# Lectures 3: Resource Management

Ing-Ray Chen

CS 6204 Mobile Computing

Virginia Tech

Courtesy of G.G. Richard III for providing some of the slides

# Challenges of Integrated Services

- True combination of real-time and non-real-time services
- Maximize the utilization of network infrastructure
- Quality of service (QoS)
- Handoff handling
  - Forced termination of an outgoing call is more annoying than blocking of a new call

# Handoff Design Issues

- Forced termination vs. new call blocking
- Increased channel utilization in a fair manner
- Goal:
  - Minimization of forced termination of real-time handoff calls without drastically sacrificing other QoS parameters
- Need for support of multiple service classes simultaneously
- Keys for good designs:
    - Delay sensitivity: non-real-time vs. real-time
    - Preemptive model: priority reservation for handoff calls over new calls to minimize forced termination of real-time handoff calls

# Resource Reservation for Call Admission Control (Ref [7])

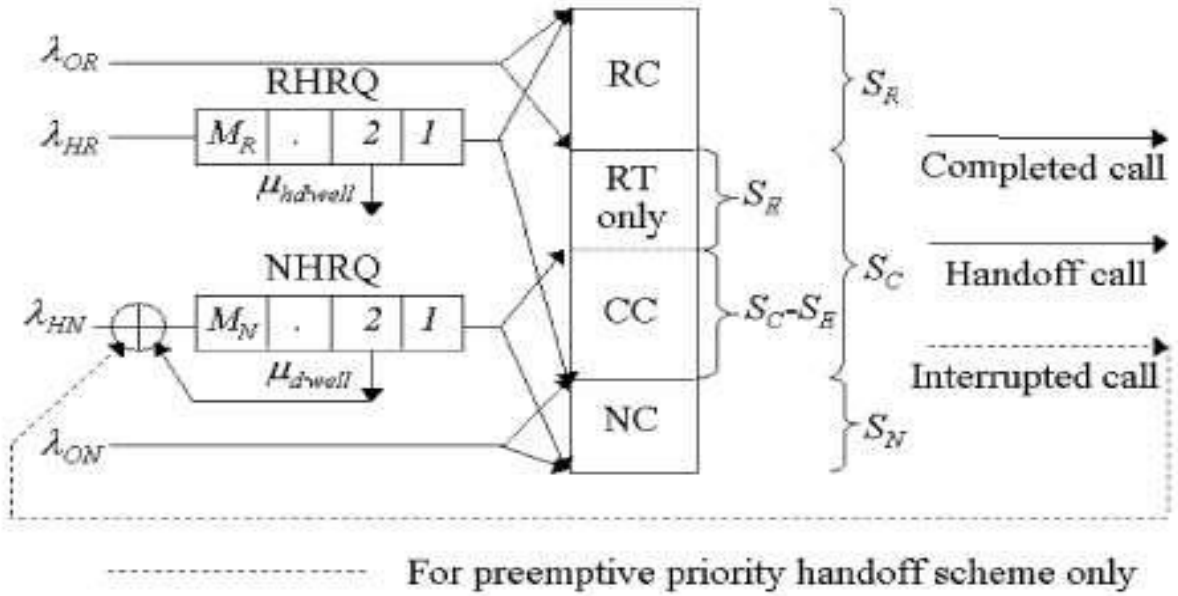



Fig. 1. System model for a reference cell.

## Partitions:

RC: # of real-time calls – maximum capacity $S_R$

RT only: # of real-time handoff calls (part of CC) – maximum capacity $S_E$

CC: # of handoff calls - maximum capacity $S_C$

NC: # of non-real-time calls – maximum capacity $S_N$

Some partition may be shared, e.g., RC may be used by real time new calls or real-time handoff calls.

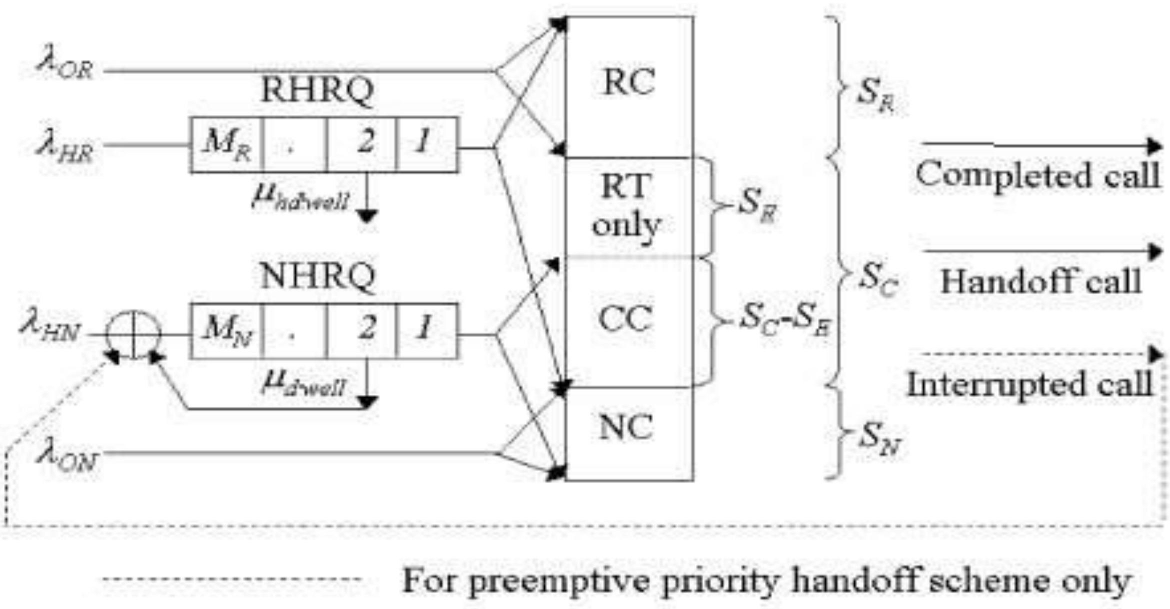



Fig. 1. System model for a reference cell.

## Traffic Sources:

- $\lambda_{OR}$: arrival rate of real-time originating new calls
- $\lambda_{HR}$: arrival rate of real-time handoff requests
- $\lambda_{ON}$: arrival rate of non-real-time originating new calls
- $\lambda_{HN}$: arrival rate of non-real-time handoff requests

## Queues (for handoff calls only)

- RHRQ: # of real-time handoff requests in queue with capacity $M_R$
- NHRQ: # of non-real-time handoff requests in queue with capacity $M_N$

# Algorithm for Originating New Calls

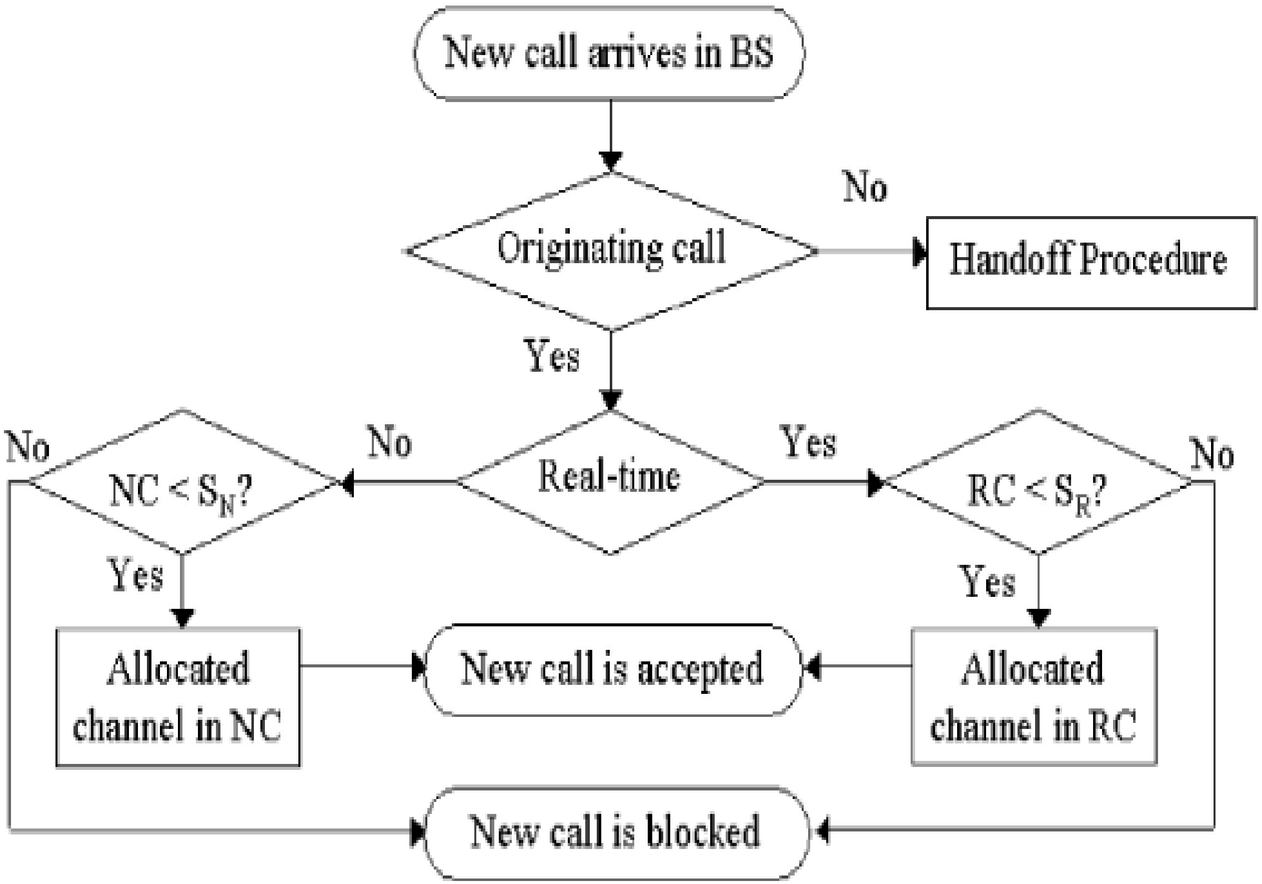



Fig. 2. Flow diagram for handling originating calls.

# Algorithm for Handoff Requests

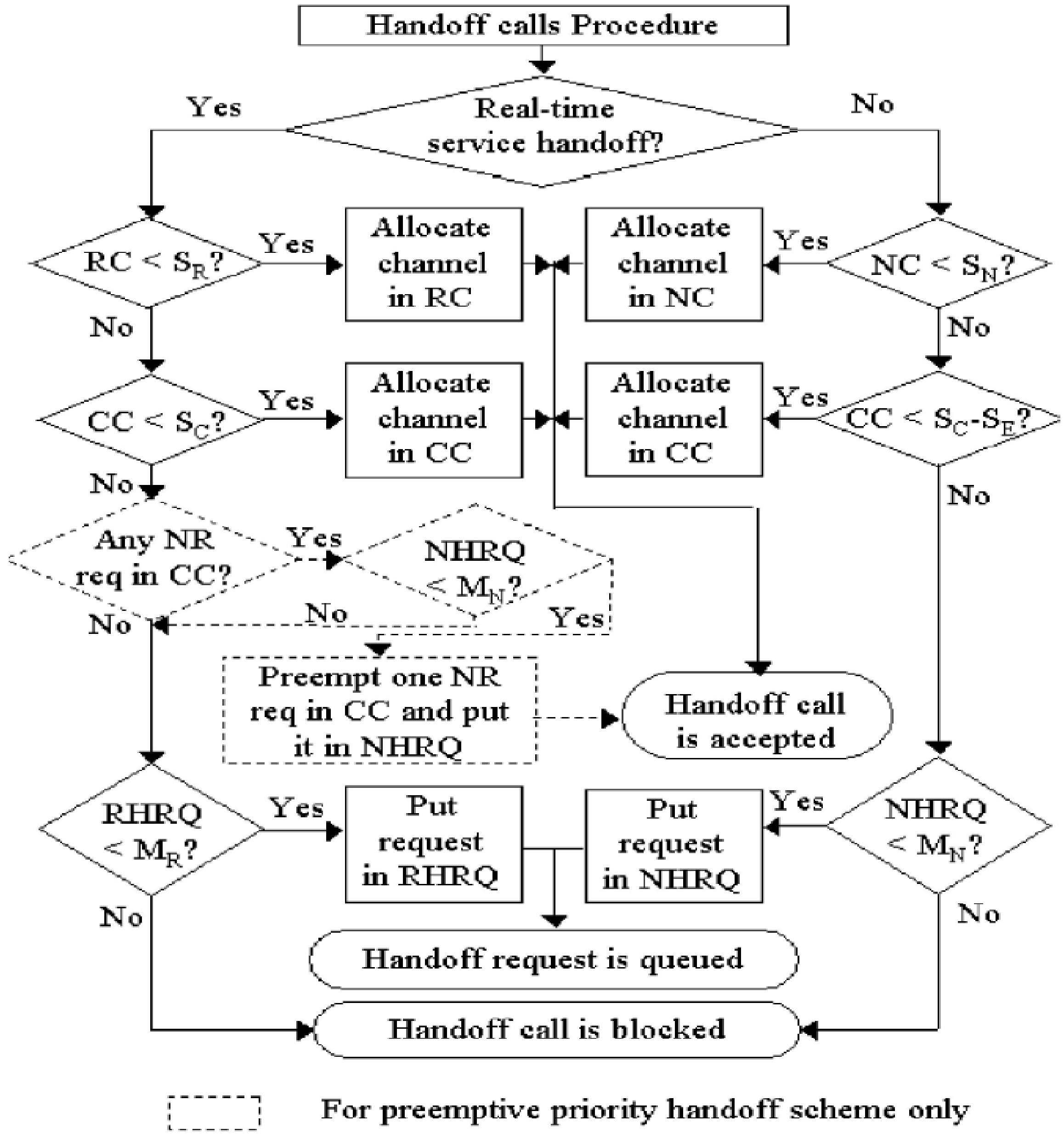



Fig. 3. Flow diagram for handling handoff request calls.

# PERFORMANCE ANALYSIS

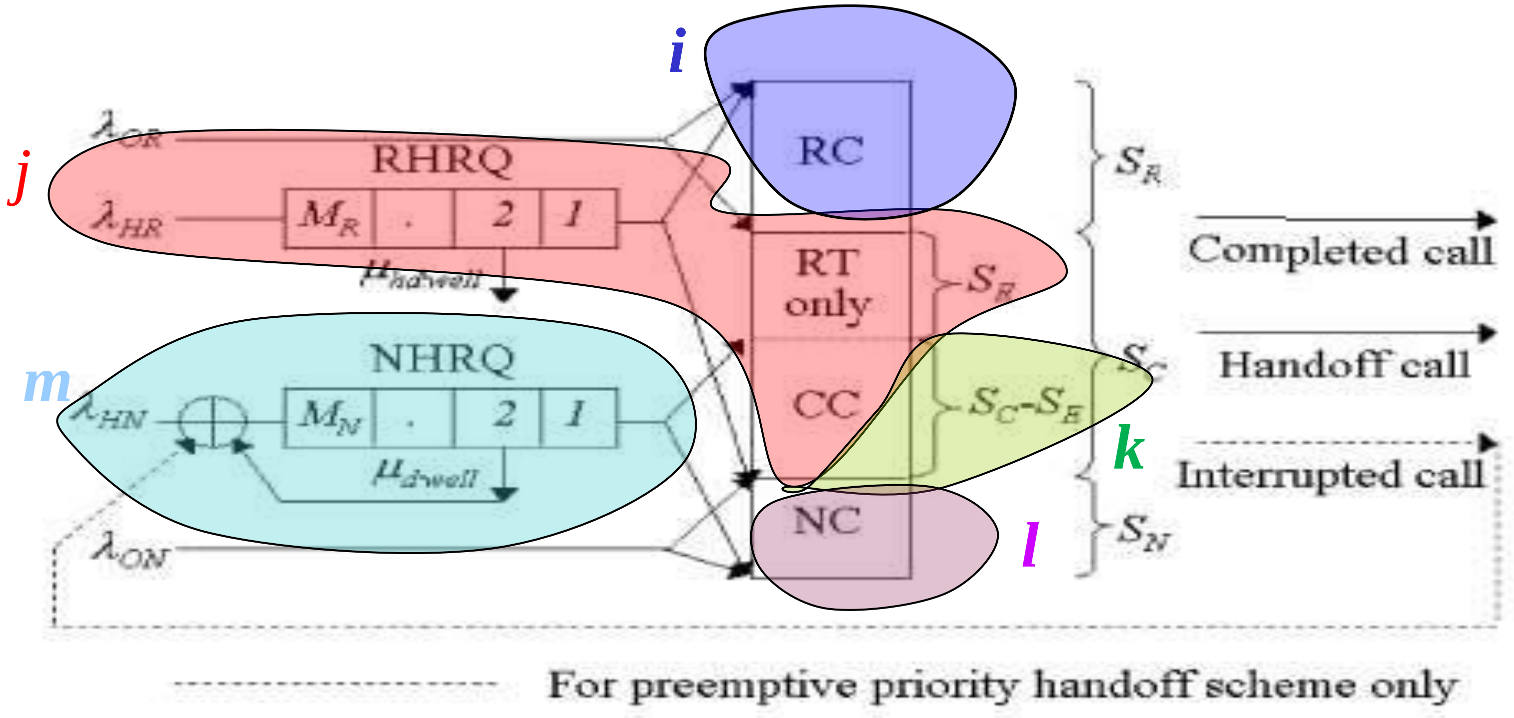



Fig. 1. System model for a reference cell.

***i*** **is the number of real-time calls in RC**

***j*** **is the sum of the number of real-time handoff calls in both CC and RHRQ**

***k*** **is the number of non-real-time handoff calls in CC**

***l*** **is the number of non-real-time calls in NC**

***m*** **is the number of non-real-time handoff calls waiting in NHRQ**

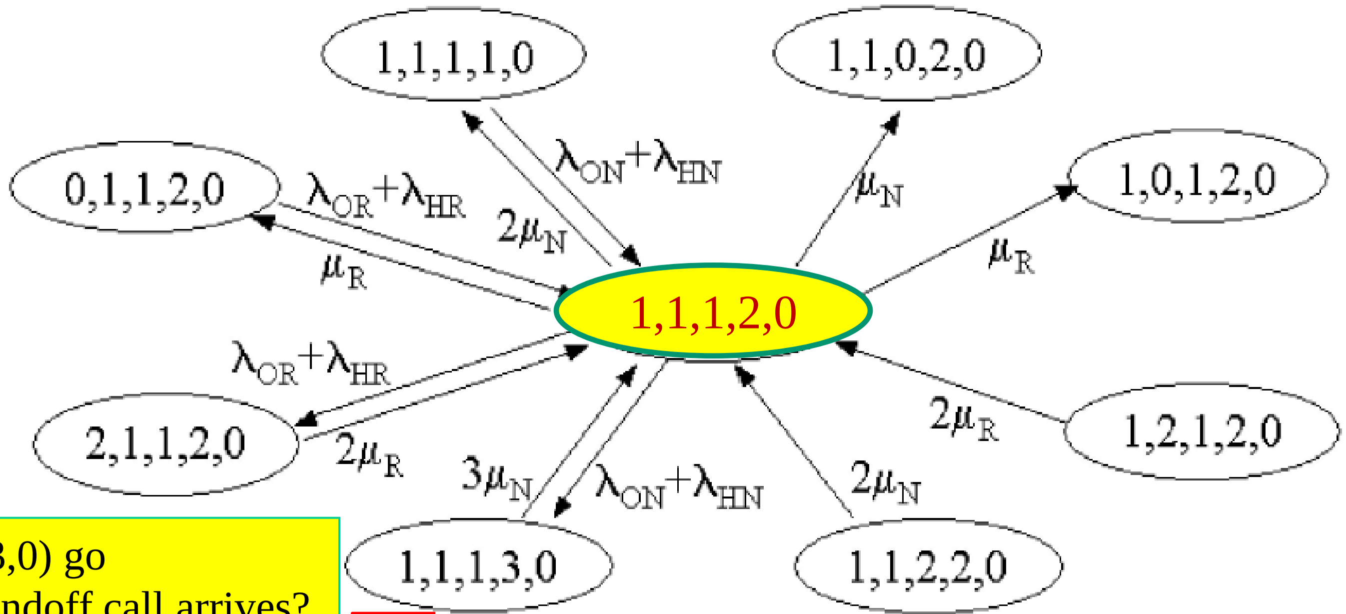
# A Partial Markov Model for (i=1, j=1, k=1, l=2, m=0), i.e., (RC=1, RT+RHRQ=1, CC-RT=1, NC=2, NHRQ=0)

1 real-time call in RC

1 real-time handoff call in CC

1 non-real-time handoff call in CC

2 non-real-time calls in NC




Which state will (1,1,1,3,0) go when a non-real-time handoff call arrives?

Fig. 4. Stable state transition diagram for State (i = 1, j = 1, k = 1, 1 = 2, $m = 0$).

$$(\lambda_{OR} + \lambda_{HR} + \lambda_{ON} + \lambda_{HN} + 2\mu_R + 3\mu_N)P_{(1,1,1,2,0)} = (\lambda_{OR} + \lambda_{HR})P_{(0,1,1,2,0)} + (\lambda_{ON} + \lambda_{HN})P_{(1,1,1,1,0)} + 3\mu_N P_{(1,1,1,3,0)} + 2\mu_N P_{(1,1,2,2,0)} + 2\mu_R P_{(1,2,1,2,0)} + 2\mu_R P_{(2,1,1,2,0)}. \tag{17}$$

(1,1,2,3,0)

Given: $S = S_R + S_C + S_N = 12$
$S_R = 6$; $S_C = S_N = 3$; $S_E = 1$
$M_R = 5$; $M_N = 50$

# QoS Metric Calculation

- Based on the state diagram, the steady state probability that the system is in a state can be calculated, from which QoS metrics may be calculated. For example: the blocking probability of non-real-time handoff calls ($B_{HN}$) can be calculated by conditioning on CC-RT is full ($j+k \geq S_C - S_E$), NC is full ($l = S_N$) and NHRQ is full ($m = M_N$):

$$B_{HN} = \sum_{i=0}^{S_R} \sum_{k=0}^{S_C-S_E} \sum_{\cancel{j=S_C-k+1}}^{S_C+M_R-k} P(i, j, k, S_N, M_N). \qquad (37)$$

$$\boxed{j = S_C - k}$$

*Recall:*

$i$ is the number of real-time calls in RC

$j$ is the sum of the number of real-time service handoff calls in both CC and RHRQ

$k$ is the number of non-real-time handoff calls in CC

$l$ is the number of non-real-time calls in NC

$m$ is the number of non-real-time handoff calls waiting in NHRQ

# Admission Control for Revenue Optimization with QoS Guarantees (Ref [8])

- Call admission control (CAC) algorithms that make acceptance/rejection decisions based on:
  - Satisfying QoS requirements, and
  - Optimizing system “revenue” or “reward”
- Integrating pricing with CAC
- Assume “charge-by-time” pricing

# System Model- From A Cell's Perspective

Multiple Service classes:

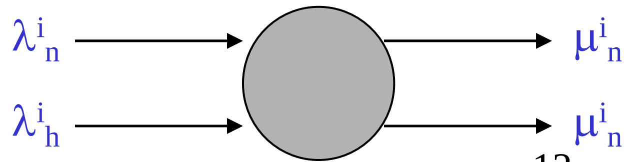
$\lambda^i_n$ – Arrival rate of new calls of service class i

$\mu^i_n$ – Departure rate of new calls of service class i

$\lambda^i_h$ – Arrival rate of handoff calls of service class i

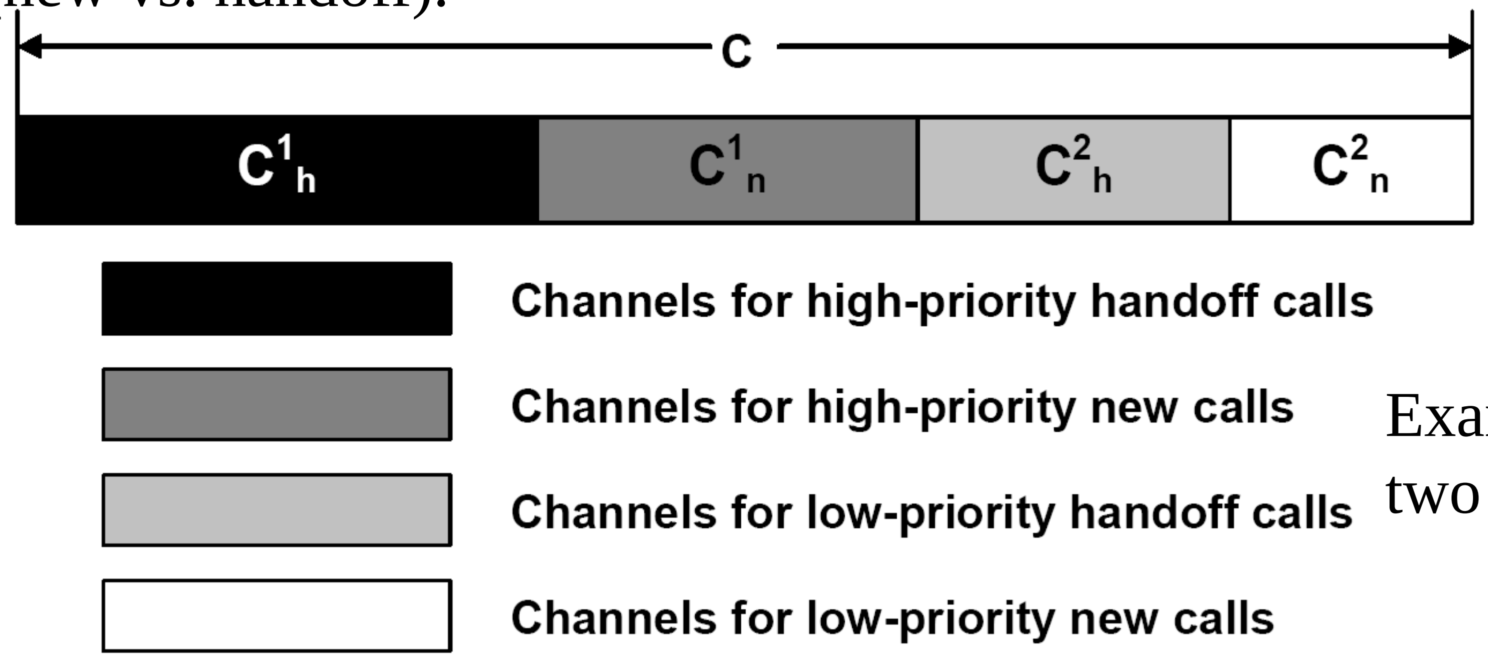
$\mu^i_h$ – Departure rate of handoff calls of service class i

- A cell has C channels
- Service call of class i requires $k^i$ channels
- Price rate is $v^i$ for class i



# Partitioning Admission Control

Partitioning CAC divides the total number of channels in a cell into several fixed partitions, with each partition specifically reserved to serve a particular service class (real-time vs. non-real-time) and call type (new vs. handoff).




Example: two classes

Constraints: $C^1_h, C^1_n, C^2_h, C^2_n \leq C$ & $C^1_h + C^1_n + C^2_h + C^2_n = C$

# Partitioning-based CAC

- Channels in a partition cannot be shared.
- If a class 1 new call arrives and all channels allocated to serve class 1 new calls are used up, this class 1 new call is rejected. This applies to all service classes (i=1, 2, etc.) and call types (new vs. handoff).
- Each partition thus can be modeled as a $M/M/n^i/n^i$ queue with $n^i$ being the number of call slots in the partition (determined by $k^i$), $\lambda^i$ being the arrival rate and $\mu^i$ being the service rate. The subscript of "n" or "h" is dropped from the notation above.

# Input Parameters to a Cell: Two Classes

The following parameters are used by a cell's CAC algorithm:

**Arrival Rates**

$\lambda^1_h$
$\lambda^1_n$
$\lambda^2_h$
$\lambda^2_n$

**Departure Rates**

$\mu^1_h$
$\mu^1_n$
$\mu^2_h$
$\mu^2_n$

**Price Rates**

$v^1$
$v^2$

**Number of Channels required per call**

$k^1$
$k^2$

**QoS: Maximum Blocking Probability Thresholds**

$B^1_h t$
$B^1_n t$
$B^2_h t$
$B^2_n t$

# QoS Metric Calculation

- Blocking probability is the probability a call is rejected.
- QoS constraints: Blocking probabilities of new and handoff calls for both classes 1 and 2 must be satisfied.
- We would like to partition the channels such that the following QoS constraints are satisfied:

$$B^1_h < B^1_h t$$
$$B^1_n < B^1_n t$$
$$B^2_h < B^2_h t$$
$$B^2_n < B^2_n t$$

The blocking probability (new or handoff calls of class i) is equal to the probability of the partition allocated to service class i new or handoff calls being full, which can be calculated by the probability that all $n^i$ slots are full in the $M/M/n^i/n^i$ model.

# Revenue Calculation

The revenue that a successfully terminated or handed-off call brings to the cell is calculated by the product of the call's price rate parameter $v^i$ with the duration of the call in the cell.

The revenue rate earned by the partitioning-based CAC may be calculated as follows:

$$PR(C, \lambda^1_h, \lambda^1_n, \lambda^2_h, \lambda^2_n) = PR^1_h + PR^1_n + PR^2_h + PR^2_n$$

where $PR^1_h$, $PR^1_n$, $PR^2_h$, and $PR^2_n$ stand for the revenues generated per unit time due to high-priority handoff calls, high-priority new calls, low-priority handoff calls, and low-priority new calls, respectively.

For example: $PR^1_h$ is calculated by $(1- B^1_h)\ \lambda^1_h\ v^1/\mu^1_h$

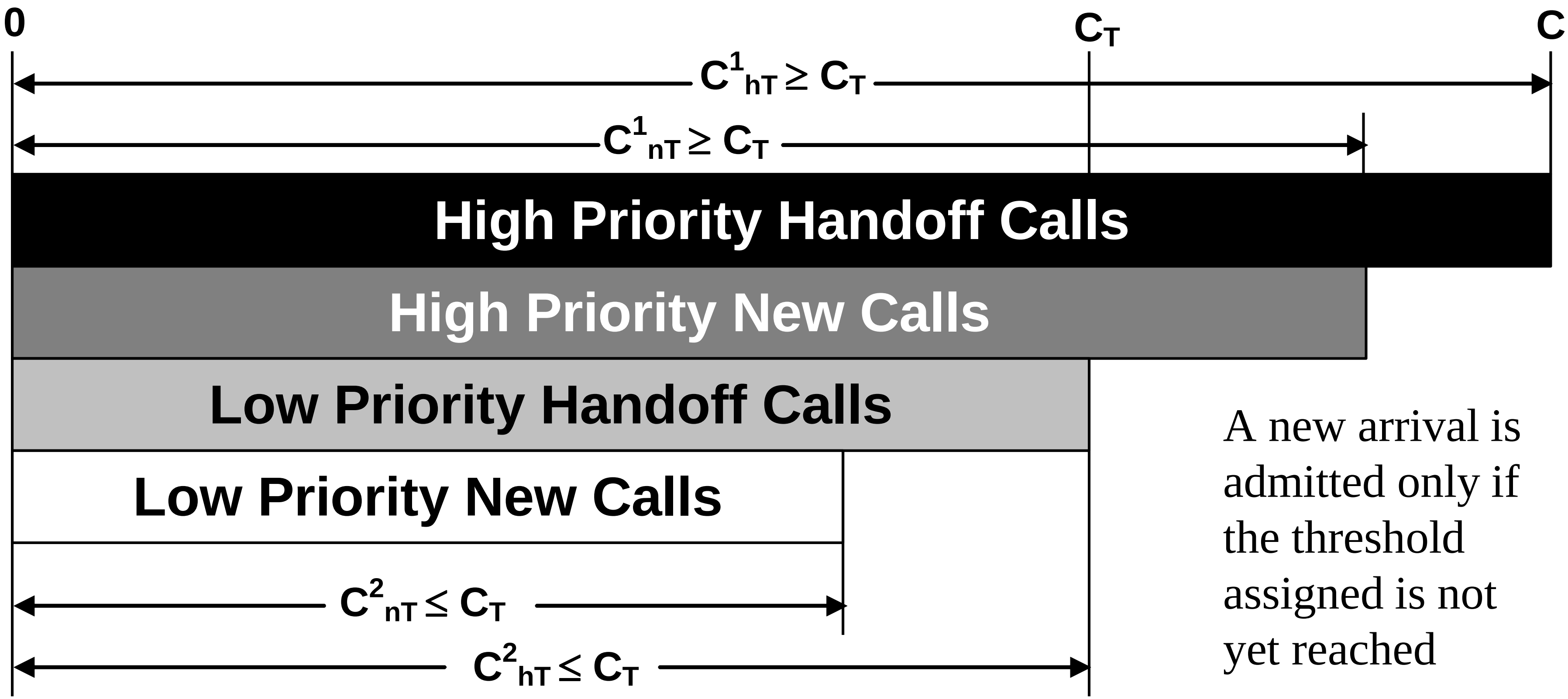
# Revenue Optimization Problem Under Partitioning-based CAC

- Identify the best partition sizes ($C^1_h$, $C^1_n$, $C^2_h$, $C^2_n$) that will maximize the cell's revenue PR(C, $\lambda^1_h$, $\lambda^1_n$, $\lambda^2_h$, $\lambda^2_n$) subject to the imposed QoS constraints:

$$B^1_h < B^1_h t$$
$$B^1_n < B^1_n t$$
$$B^2_h < B^2_h t$$
$$B^2_n < B^2_n t$$

# Threshold-Based CAC

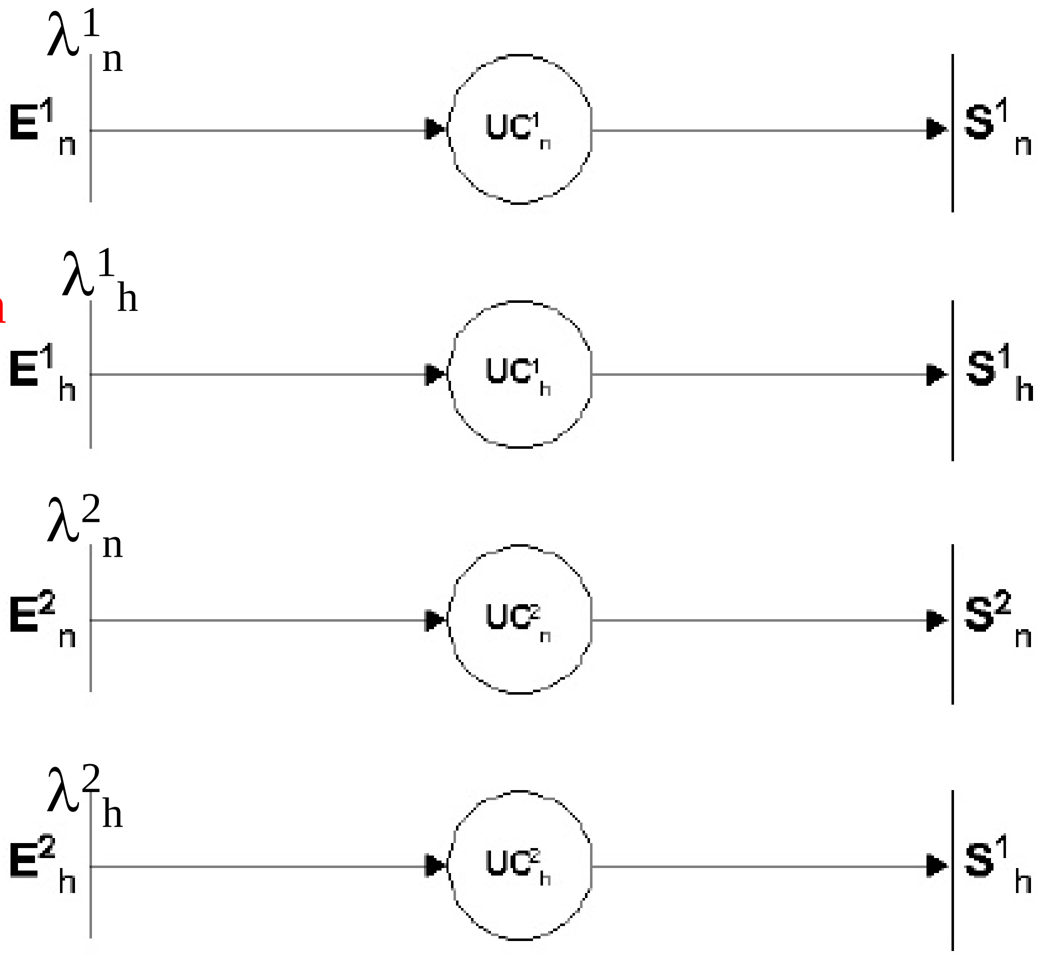
Example: two classes (class 1 is high priority)




A new arrival is admitted only if the threshold assigned is not yet reached

Constraints: $C^1_{hT} \geq C_T$, $C^1_{nT} \geq C_T$, $C^2_{hT} \leq C_T$, $C^2_{nT} \leq C_T$

# Threshold-Based Admission Control Performance Model




modeling admission of class 1 handoff calls with rate $\lambda^1_h$

# Threshold-Based CAC Performance Model

**$UC^i_n$**– holding the number of class i new calls admitted

**$UCi_h$**– holding the number of class i handoff calls admitted

**$E^i_h$**– models admission of class i handoff calls with rate $\lambda^i_h$

**$E^i_n$**– models admission of class i new calls with rate $\lambda^i_n$

**$S^i_h$**– models service of class i handoff calls with a service rate of $M(UC^i_h)$ multiplied with $\mu^i_h$ where $M(UC^i_h)$ stands for the number of tokens in place $UC^i_h$

**$S^i_n$**– models service of class i new calls with a service rate of $M(UC^i_n)$ multiplied with $\mu^i_n$ where $M(UC^i_n)$ stands for the number of tokens in place $UC^i_n$

# Threshold-Based CAC

- A new service request arrival is admitted only if the threshold assigned is not yet reached.
  - → Assign an enabling predicate to guard $E^i_n$, $E^i_h$ with thresholds $C^i_{nT}$ and $C^i_{hT}$

# Threshold-Based CAC: Use Enabling Predicate for Admission Control

- Enabling predicate of $E^1_n$

  $[M(UC^1_n) + M(UC^1_h)]\ k^1 + \mathbf{k^1} + [M(UC^2_n) + M(UC^2_h)]\ k^2 \le C^1_{nT}$

- Enabling predicate of $E^1_h$ is

  $[M(UC^1_n) + M(UC^1_h)]\ k^1 + \mathbf{k^1} + [M(UC^2_n) + M(UC^2_h)]\ k^2 \le C^1_{hT}$

- Enabling predicate of $E^2_n$ is

  $[M(UC^1_n) + M(UC^1_h)]\ k^1 + \mathbf{k^2} + [M(UC^2_n) + M(UC^2_h)]\ k^2 \le C^2_{nT}$

- Enabling predicate of $E^2_h$ is

  $[M(UC^1_n) + M(UC^1_h)]\ k^1 + \mathbf{k^2} + [M(UC^2_n) + M(UC^2_h)]\ k^2 \le C^2_{hT}$

# QoS Metric Calculation

- Blocking probabilities

$$B_n^1 = \frac{\left(\lambda_n^1 - rate(E_n^1)\right)}{\lambda_n^1} \qquad B_h^1 = \frac{\left(\lambda_h^1 - rate(E_h^1)\right)}{\lambda_h^1}$$

$$B_n^2 = \frac{\left(\lambda_n^2 - rate(E_n^2)\right)}{\lambda_n^2} \qquad B_h^2 = \frac{\left(\lambda_h^2 - rate(E_h^2)\right)}{\lambda_h^2}$$

Note: rate($E^1_n$) is the admission rate of class 1 new calls and is calculated by the expected value of a random variable X which has a value of $\lambda^1_n$ if $E^1_n$ is enabled and a value of 0 otherwise.

# Revenue Calculation

The revenue generated per unit time from the threshold-based CAC algorithm to the cell is defined by:

$$TR(C, \lambda^1_h, \lambda^1_n, \lambda^2_h, \lambda^2_n) = TR^1_h + TR^1_n + TR^2_h + TR^2_n$$

Where $TR^1_h$, $TR^1_n$, $TR^2_h$, and $TR^2_n$ stand for the revenues generated per unit time due to high-priority handoff calls, high-priority new calls, low-priority handoff calls, and low-priority new calls, respectively, given by:

$$TR^i_h = (1 - B^i_h)\, \lambda^i_h\, v^i / \mu^i_h$$

$$TR^i_n = (1 - B^i_n)\, \lambda^i_n\, v^i / \mu^i_n.$$

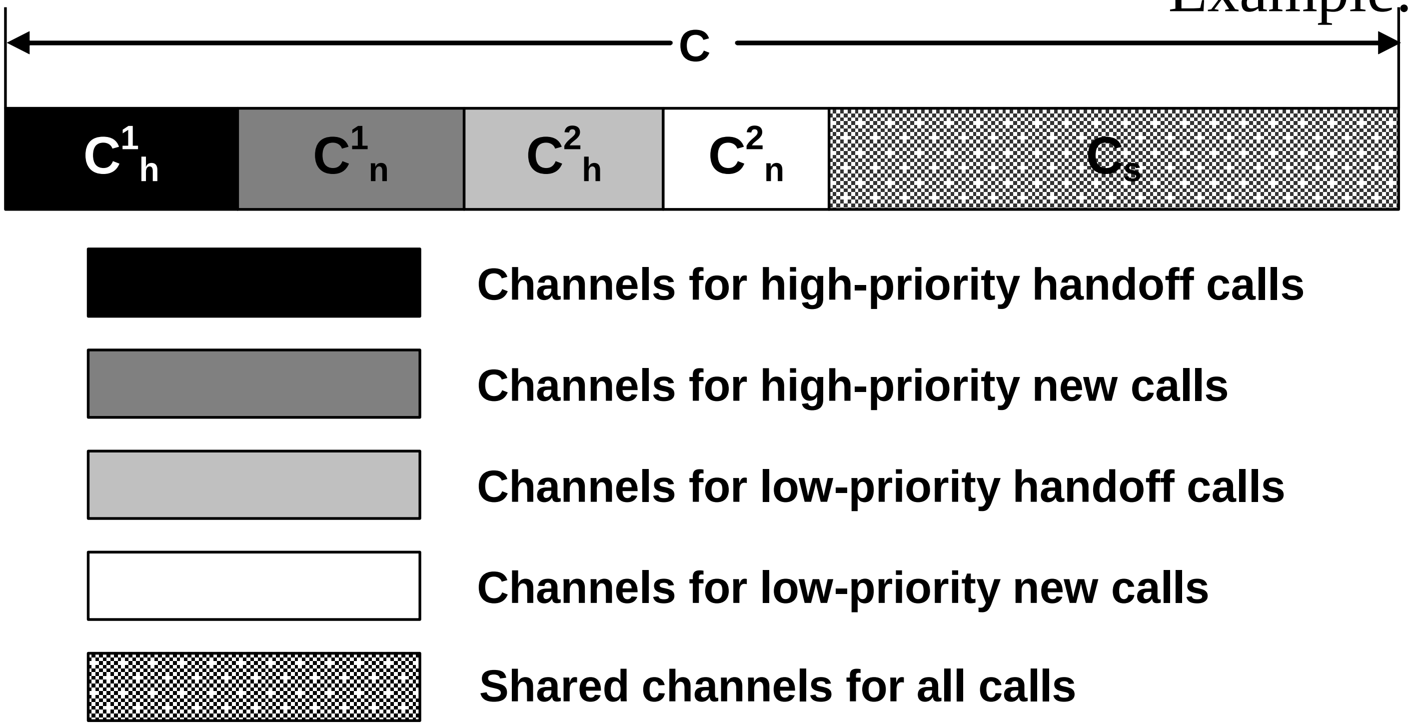
# Revenue Optimization Problem Under Threshold-based CAC

- Identify the best threshold set ($C^1_{hT}$, $C^1_{nT}$, $C^2_{hT}$, $C^2_{nT}$) that will maximize the cell's revenue subject to the imposed QoS constraints.

# Hybrid CAC

**The hybrid CAC algorithm divides the channels into fixed partitions the same way partitioning-based CAC does. In addition, a "shared" partition is reserved to allow calls of all service classes to compete for usage based on threshold-based CAC**

Example: two classes




Constraints: $n^1_h k^1 + n^1_n k^1 + n^2_h k^2 + n^2_n k^2 \leq C - C_s$ & $C^1_h + C^1_n + C^2_h + C^2_n + C_s = C$

# Hybrid Admission Control

- The shared partition is available for use by a service class only if the partition reserved for that service class and service type (new vs. handoff) is used up.
- QoS constraints the same as before:

$$\boxed{\begin{array}{l} B^1_h < B^1_h t \\ B^1_n < B^1_n t \\ B^2_h < B^2_h t \\ B^2_n < B^2_n t \end{array}}$$

# Hybrid CAC Performance Model

- Hybrid CAC encompasses both partitioning and threshold CAC algorithms as special cases
  - Partitioning CAC: there is no shared partition, so $C_s=0$
  - Threshold-based CAC: no fixed partitions, so $C^1_h$, $C^1_n$, $C^2_h$, $C^2_n$ all equal to 0
- Hybrid CAC performance model has two sub-models:
  - Partitioning: $C^1_h$, $C^1_n$, $C^2_h$, $C^2_n$
  - Threshold-based: $C=C_s$

# Hybrid CAC Performance Model

- What is the arrival rate to the shared partition?
  - Arrival rate is the spill over rate from each fixed partition (modeled as an M/M/n/n queue)

$$\lambda_{hs}^{1} = \lambda_{h}^{1} \boxed{\frac{\frac{1}{n_h^1!}\left(\frac{\lambda_h^1}{\mu_h^1}\right)^{n_h^1}}{1+\sum_{j=1}^{n_h^1}\frac{1}{j!}\left(\frac{\lambda_h^1}{\mu_h^1}\right)^{j}}}$$

Erlang's B formula for rejection probability

Arrival rates into shared partition

$\lambda^{1}_{hs}$ = class 1 handoff calls

$\lambda^{1}_{ns}$ = class 1 new calls

$\lambda^{2}_{hs}$ = class 2 handoff calls

$\lambda^{2}_{ns}$ = class 2 new calls

Expressions for $\lambda^{1}_{ns}$, $\lambda^{2}_{hs}$, and $\lambda^{2}_{ns}$ are similar

# Hybrid CAC Revenue Generation

- Revenue generated per unit time from hybrid CAC is the sum of revenues earned from the fixed partitions plus that earned from the shared partition:

$$HR(C, \lambda^1_h, \lambda^1_n, \lambda^2_h, \lambda^2_n) =$$

$$PR(C\text{-}C_s, \lambda^1_h, \lambda^1_n, \lambda^2_h, \lambda^2_n) + TR(C_s, \lambda^1_{hs}, \lambda^1_{ns}, \lambda^2_{hs}, \lambda^2_{ns})$$

**Optimization Problem for hybrid CAC:** Identify the best partition and the best threshold set within the shared partition ($C^1_h$, $C^1_n$, $C^2_h$, $C^2_n$, $C_s$) to maximize the revenue subject to the imposed QoS constraints

# CAC Comparison, with varying $\lambda^1_h$

| $\lambda^1_h$ | **Partitioning CAC** | | **Hybrid CAC** | | **Threshold-based CAC** | |
|---|---|---|---|---|---|---|
| | $(C^1_h, C^1_n, C^2_h, C^2_n)$ | **Revenue/Time** | $(C^1_h, C^1_n, C^2_h, C^2_n, C_s)$ | **Revenue/Time** | $(C^1_{hT}, C^1_{nT}, C^2_{hT}, C^2_{nT})$ | **Revenue/Time** |
| 1 | (16,56,4,4) | 577.391 | (8, 72,0,0,36) | 580.000 | (80,80,80,80) | 579.95 |
| 1.5 | (20,52,4,4) | 615.486 | (12,36,0,0,32) | 620.000 | (80,80,80,80) | 619.88 |
| 2 | (20,52,4,4) | 652.304 | (12,32,0,0,36) | 659.997 | (80,80,80,80) | 659.75 |
| 2.5 | (28,44,4,4) | 686.660 | (16,32,0,0,32) | 699.986 | (80,80,80,80) | 699.485 |
| 3 | (32,40,4,4) | 717.032 | (16,32,0,0,32) | 739.949 | (80,80,80,80) | 739.023 |
| 3.5 | (32,40,4,4) | 754.215 | (16,28,0,0,36) | 779.842 | (80,80,76,76) | 778.258 |
| 4 | None | None | (16,28,0,0,36) | 819.565 | (80,80,76,76) | 817.058 |
| 4.5 | None | None | (20,24,0,0,36) | 858.998 | (80,80,76,76) | 855.266 |
| 5 | None | None | (20,24,0,0,36) | 897.974 | (80,80,76,76) | 892.708 |
| 5.5 | None | None | (20,24,0,0,36) | 936.137 | (80,80,76,76) | 929.203 |
| 6 | None | None | (20,20,0,0,40) | 973.303 | (80,80,76,76) | 964.569 |
| 6.5 | None | None | (20,20,0,0,40) | 1009.098 | (80,76,75,72) | 992.917 |
| 7 | None | None | (24,20,0,0,36) | 1043.262 | None | None |
| 7.5 | None | None | (24,20,0,0,36) | 1075.786 | None | None |

C=80, $\mu^1_h = 1.0$, $\lambda^1_n = 6.0$, $\mu^1_n = 1.0$, $\lambda^2_h = 1.0$, $\mu^2_h = 1.0$, $\lambda^2_n = 1.0$, $\mu^2_n = 1.0$, $v^1 = 80$, $v^2 = 10$, $k^1 = 4$, $k^2 = 1$, **$B^1_h t = 0.02$, $B^2_h t = 0.04$, $B^1_n t = 0.05$, $B^2_n t = 0.1$.**

# CAC Comparison, varying QoS Constraints

| $(B^1_h t, B^2_h t)$ | Partitioning CAC | | Hybrid CAC | | Threshold-based CAC | |
|---|---|---|---|---|---|---|
| | $(C^1_h, C^1_n, C^2_h, C^2_n)$ | Revenue/ Time | $(C^1_h, C^1_n, C^2_h, C^2_n, C_s)$ | Revenue/ Time | $(C^1_{hT}, C^1_{nT}, C^2_{hT}, C^2_{nT})$ | Revenue/ Time |
| **Low Class 1 Call Arrival Rates ($\lambda^1_h = 1.0$, $\lambda^1_n = 1.0$)** | | | | | | |
| $(0.02,0.04) \times 2^{-1}$ | (20,20,20,20) | 359.135 | (16,12,5,5,42) | 360.000 | (80,80,80,80) | 359.999 |
| $(0.02,0.04) \times 2^{-4}$ | (24,20,20,20) | 358.345 | (16,12,5,5,42) | 360.000 | (80,80,80,80) | 359.999 |
| $(0.02,0.04) \times 2^{-6}$ | (28,16,22,14) | 353.041 | (16,12,5,5,42) | 360.000 | (80,80,80,80) | 359.999 |
| $(0.02,0.04) \times 2^{-7}$ | (28,16,23,13) | 350.311 | (16,12,5,5,42) | 360.000 | (80,80,80,80) | 359.999 |
| … | None | None | … | … | … | … |
| $(0.02,0.04) \times 2^{-21}$ | None | None | (16,12,5,5,42) | 360.000 | (80,76,76,61) | 359.991 |
| $(0.02,0.04) \times 2^{-22}$ | None | None | (16,12,5,5,42) | 360.000 | (80,76,76,54) | 359.904 |
| $(0.02,0.04) \times 2^{-23}$ | None | None | (16,12,5,5,42) | 360.000 | (80,76,76,48) | 359.409 |
| $(0.02,0.04) \times 2^{-24}$ | None | None | (16,12,5,5,42) | 360.000 | (80,76,76,42) | 357.231 |
| $(0.02,0.04) \times 2^{-25}$ | None | None | (16,12,5,5,42) | 360.000 | None | None |
| **High Class 1 Call Arrival Rates ($\lambda^1_h = 3.5$, $\lambda^1_n = 4.5$)** | | | | | | |
| $(0.02,0.04) \times 2^{0}$ | None | None | (12,16,2,2,48) | 834.544 | (80,80,76,76) | 830.610 |
| $(0.02,0.04) \times 2^{-1}$ | None | None | (12,16,2,2,48) | 834.544 | (80,80,76,76) | 830.610 |
| $(0.02,0.04) \times 2^{-2}$ | None | None | (20,8,1,1,50) | 830.078 | (80,76,76,76) | 826.298 |

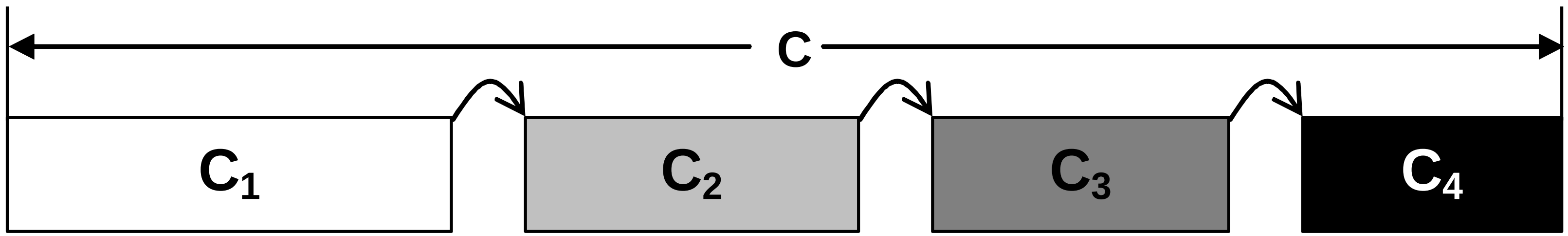
$C=80$, $\mu^1_h = 1.0$, $\mu^1_n = 1.0$, $\lambda^2_h = 10.0$, $\mu^2_h = 1.0$, $\lambda^2_n = 10.0$, $\mu^2_n = 1.0$, $v^1 = 80$, $v^2 = 10$, $k^1 = 4$, $k^2 = 1$, $B^1_n t=0.05$, $B^2_n t=0.1$.

# Other Ideas of reward optimization CAC

- Spillover CAC (Ref [9])
- Elastic Threshold CAC – two thresholds instead of just one (Ref [10])

# Spillover CAC (Ref [9])

Example: two classes



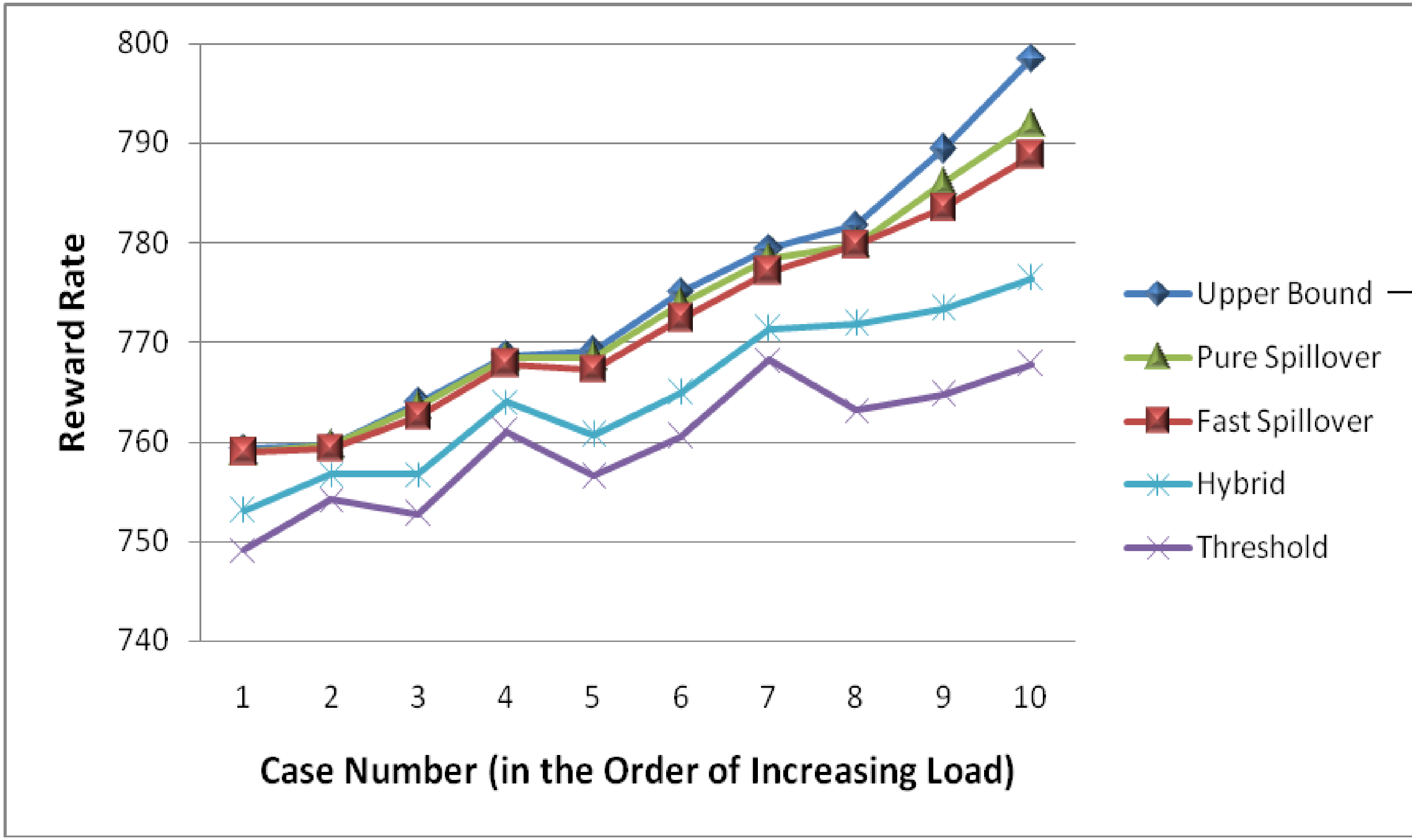

Channels shared by all calls

Channels shared by high-priority calls and low priority handoff calls

Channels shared by high-priority calls

Channels for only high-priority handoff calls

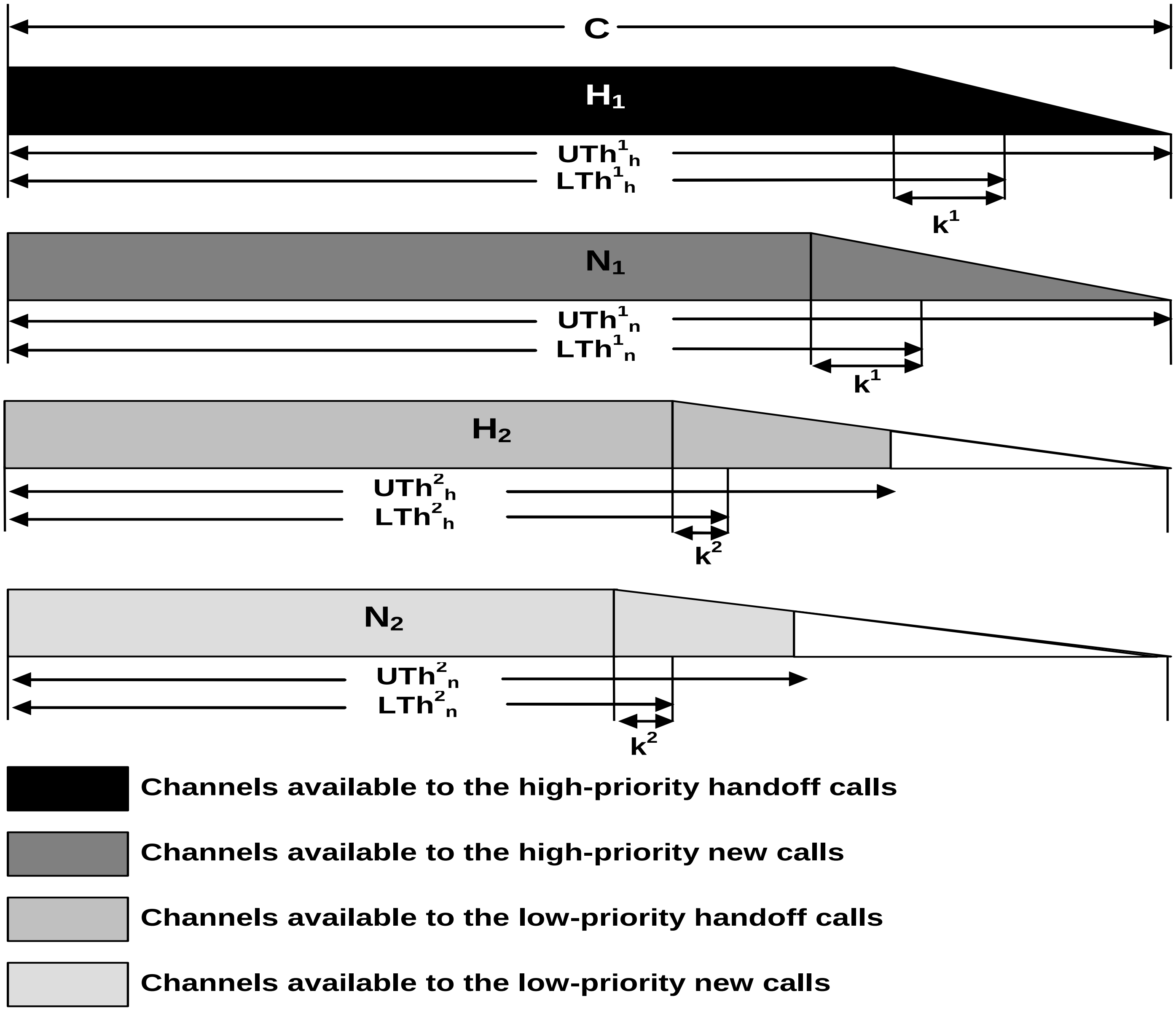
# Spillover CAC Performance




Based on infinite channels

# Elastic Threshold CAC (Ref [10])

Example: two classes



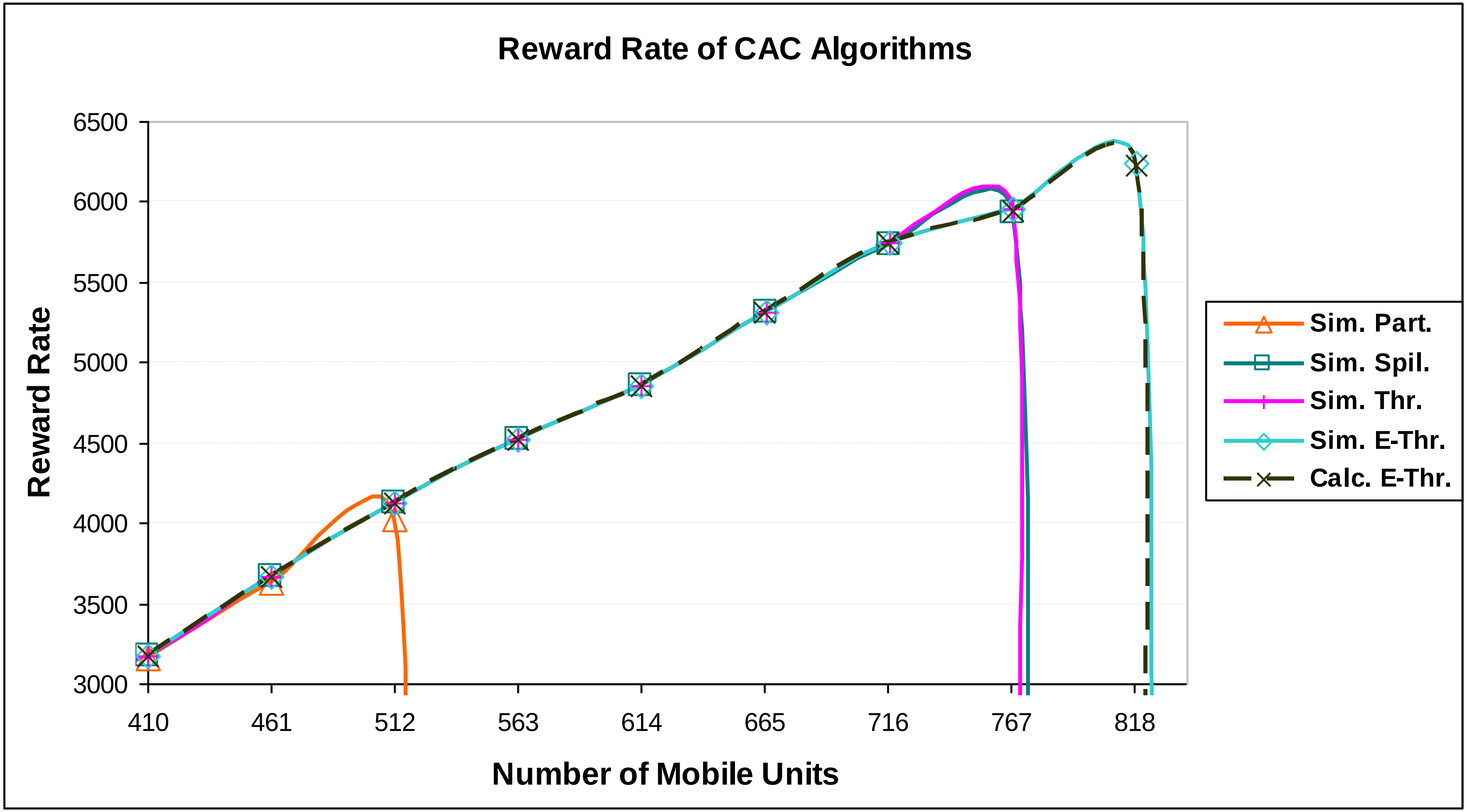
# Elastic Threshold CAC Acceptance Probability

$$P_n^i = \begin{cases} 1 & n + k^i \le LTh_n^i \\ 1 - \left[ \left( n + k^i - LTh_n^i \right) \Big/ \left( C - LTh_n^i \right) \right] & LTh_n^i < n + k^i \le HTh_n^i \\ 0 & HTh_n^i < n + k^i \end{cases}$$

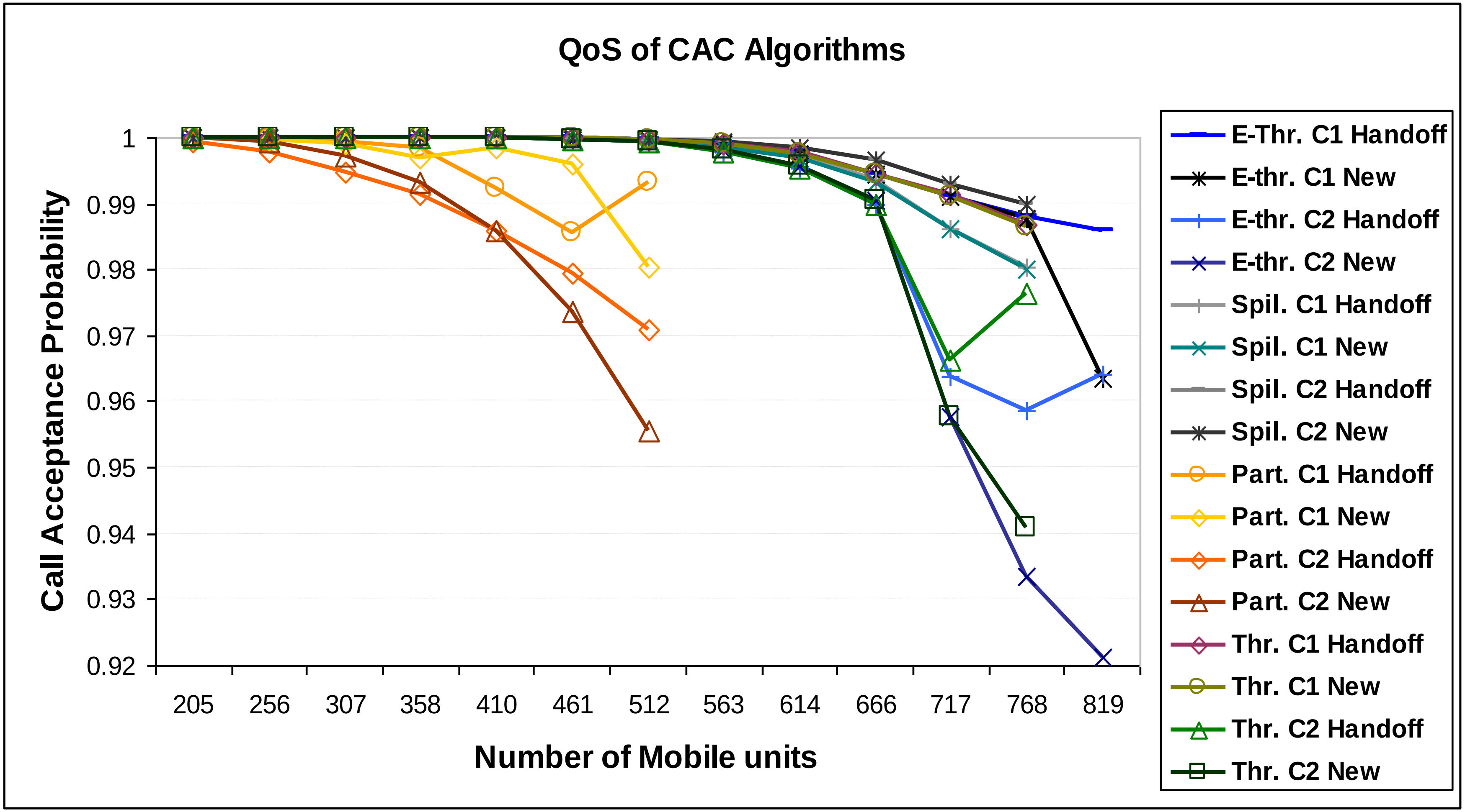
$$P_h^i = \begin{cases} 1 & n + k^i \le LTh_h^i \\ 1 - \left[ \left( n + k^i - LTh_h^i \right) \Big/ \left( C - LTh_h^i \right) \right] & LTh_h^i < n + k^i \le HTh_h^i \\ 0 & HTh_h^i < n + k^i \end{cases}$$

$n$: number of channels occupied at time of admission
$C$: total number of channels in the cell

# Comparison of CAC Algorithms in Revenue (Reward) Generated



# Comparison of CAC Algorithms in QoS




Performance Comparison: Elastic > Spillover ~ Threshold > Partitioning

# CAC Summary

- Threshold-based CAC and spillover CAC have comparable performance. Both leverage multiplexing power through channel sharing to improve acceptance ratio. Spillover CAC is able to handle higher traffic by reserving more channels to the partitions allocated to high-priority service classes. On the other hand, threshold-based CAC is able to limit the bandwidth used by low-priority service classes by setting low thresholds.
- When we increase the number of mobile users (high traffic), only elastic threshold-based CAC is able to satisfy the high QoS requirement of class 1 handoff calls.
- We attribute the superiority of elastic threshold-based CAC to its elastic threshold functionality capable of leveraging the low threshold to regulate traffic (rejecting just a fraction of traffic) and the high threshold to reject traffic generated by service calls.

# References

Chapter 3, F. Adelstein, S.K.S. Gupta, G.G. Richard III and L. Schwiebert, *Fundamentals of Mobile and Pervasive Computing*, McGraw Hill, 2005, ISBN: 0-07-141237-9.

Other References:

7. J.W. Wang, Q.A. Zeng, and D. P. Agrawal, "Performance Analysis of a Preemptive and Priority Reservation Handoff Scheme for Integrated Service-Based Wireless Mobile Networks, " *IEEE Transactions on Mobile Computing*, Vol. 2, No. 1, 2003, pp. 65-75.
8. I.R. Chen, O. Yilmaz and I.L. Yen, "Admission control algorithms for revenue optimization with QoS guarantees in mobile wireless networks," *Wireless Personal Communications*, Vol. 38, No. 3, Aug. 2006, pp. 357-376.
9. O. Yilmaz, I.R. Chen, G. Kulczycki, and W. Frakes, "Performance Analysis of Spillover-Partitioning Call Admission Control in Mobile Wireless Networks," Wireless Personal Communications, Vol. 53, No. 1, 2010, pp. 111-131.
10. O. Yilmaz and I.R. Chen, "Elastic Threshold-Based Admission Control for QoS Satisfaction with Reward Optimization for Servicing Multiple Priority Classes in Wireless Networks," Information Processing Letters, Vol. 109, No. 15, July 2009, pp. 868-875.