Chapter 8 (Part 4): Continuous Time Markov Chain Performability Modeling
Dept. of Electrical & Computer engineering
Duke University
Email: kst@ee.duke.edu
URL: www.ee.duke.edu/~kst
Outline

- Why performability modeling?
- Erlang loss performability model
- Modeling cellular systems with failure
- Multiprocessor Performability
- Conclusion
Outline

- Why performability modeling?
- Erlang loss performability model
- Modeling cellular systems with failure
- Multiprocessor Performability
- Conclusion
Wireless “ilities” besides performance

Reliability
- for a specified operational time

Availability
- at any given instant

Survivability
- performance under failures

Performability measures of the network’s ability to perform designated functions

R.A.S.-ability concerns grow. High-R.A.S. not only a selling point for equipment vendors and service providers. But, regulatory outage report required by FCC for public switched telephone networks (PSTN) may soon apply to wireless.
Causes of Service Degradation

- Limited Resources
  - Equipment failures
  - Software failures
  - Planned outages (e.g. upgrade)
  - Human-errors in operation

- Resource full

- Resource loss

- Long waiting-time
  - Time-out
  - Service blocking

- Service Interruption
  - Loss of information
The Need of Performability Modeling

- New technologies, services & standards need new modeling methodologies

- Pure performance modeling: too optimistic!
  Outage-and-recovery behavior not considered

- Pure availability modeling: too conservative!
  Different levels of performance not considered
Measures To Be Evaluated

- **Dependability**
  - Reliability: $R(t)$, System MTTF
  - Availability: Steady-state, Transient
  - Downtime
  - Security, safety

  "Does it work, and for how long?"

- **Performance**
  - Throughput, Blocking Probability, Response Time

  "Given that it works, how well does it work?"
Measures To Be Evaluated (Contd.)

- Composite Performance and Dependability

“How much work will be done(lost) in a given interval including the effects of failure/repair/contention?”

- Need Techniques and Tools That Can Evaluate
  - Performance, Dependability and Their Combinations
Outline

- Why performability modeling?
- Erlang loss performability model
- Modeling cellular systems with failure
- Hierarchical model for APS in TDMA
- Multiprocessor Performability
- Conclusion
Erlang Loss Pure Performance Model

- Telephone switching system: \( n \) channels
- Call arrival process is assumed to be Poissonian with rate \( \lambda \)
- Call holding times exponentially distributed with rate \( \mu \)
- A new call is accepted if at least one idle channel is available, otherwise it is blocked.
Erlang Loss CTMC Model

State index is the number of channels in use

Let $\pi_j$ be the steady state probability for the Continuous Time Markov Chain

Blocking Probability: $P_b = \pi_n$

Expected number of calls in system: $E[N] = \sum_{j=0}^{n} j \pi_j$

Desired measures of the form: $E[M] = \sum_{j=0}^{n} r_j \pi_j$
Sharpe Textual Input File:

* Code for the Pure Performance Model
* note the use of loop in the specification of CTMC
* This allows size of the CTMC to be variable
* use repeated pattern of transitions for conciseness

```
bind
  lambda 49
  mu 0.35
end
markov perf(n)
  loop i,0,n-1
    $(i) $(i+1) lambda
    $(i+1) $(i) (i+1)*mu
  end
end
```
Sharpe code:

* Pb : Steady-state call blocking probability
func Pb(n) prob(perf,$(n);n)
* The value of n (number of channels) varied from 4 to 100
loop nb,4,100,10
    expr Pb(nb)
end
end
Blocking probability vs. Number of channels
Availability model

- Availability Analysis: (Telephone Switching system with \( n \) channels)
- Wish to compute
  - Steady-state system Unavailability: \( U \)
  - Steady-state system Availability: \( A \)
  - Instantaneous system Availability: \( A(t) \)
  - Downtime: downtime (in minutes per year)
- The times to channel failure and repair are exponentially distributed with mean \( \frac{1}{\gamma} \) and \( \frac{1}{\tau} \), respectively.
- \( \gamma = \frac{1}{\text{MTTF}} \): Failure rate of channel
- \( \tau = \frac{1}{\text{MTTR}} \): Repair rate of channel
Let $\pi_j$ be the steady state probability for the CTMC

**Steady state unavailability:**

$$\overline{A} = \pi_0$$

**Expected number of non-failed channels:**

$$E[N] = \sum_{j=0}^{n} j \pi_j$$

**Desired measures of the form:**

$$E[M] = \sum_{j=0}^{n} r_j \pi_j$$
Sharpe code:

bind
  MTTF 1000
  MTTR 24
end
markov avail(n)
  loop i,n,1,-1
    $(i) $(i-1) i/MTTF
    $(i-1) $(i) 1/MTTR
  end
end
* Initial probability, assume that n channels are up initially
  $(n) 1
end
Sharpe code:

func U(n) prob(avail,0;n)
func A(n) 1-U(n)
func At(t,n) 1-tvalue(t;avail,0;n)
func downtime(n) 60*8760*U(n)

loop nb,4,20,1
  expr U(nb),A(nb),downtime(nb)
end
format 8
bind N 4
loop t,1,291,10
  expr At(t,N)
end
end
Downtime vs. Number of channels
Instantaneous Availability vs. Number of channels
Erlang loss composite model

- A telephone switching system: \( n \) channels
- The call arrival process is assumed to be Poissonian with rate \( \lambda \), the call holding times are exponentially distributed with rate \( \mu \)
- The times to channel failure and repair are exponentially distributed with mean \( 1/\gamma \) and \( 1/\tau \), respectively
- The composite model is then a homogeneous CTMC
The state \((i, j)\) denotes \(i\) non-failed channels and \(j\) ongoing calls in the system.

CTMC with \((n+1)(n+2)/2\) states.

Total call blocking probability:

\[
T_b = \overline{A} + \pi_{n,n} + \sum_{i=1}^{n-1} \pi_{i,i}
\]

Example of expected reward rate in steady state.
Total call blocking probability

\[ T_b = \overline{A} + \pi_{n,n} + \sum_{i=1}^{n-1} \pi_{i,i} \]

- Blocked due to unavailability
- Blocked due to buffer full
- Blocked due to buffer full in degraded levels
Problems in composite performability model

- **Largeness:** Number of states in the Markov model is rather large
  - Tolerance: Automatically generate Markov reward model starting with an SRN (stochastic reward net)
  - Avoidance: Use a two-level hierarchical model

- **Stiffness:** Transition rates in the Markov model range over many orders of magnitude
  - Tolerance: Use stiffly stable methods of num. Solution
  - Avoidance: Use a two-level hierarchical model

- Potential solution to both problems is a hierarchical performability model
The hierarchical performability model provides an approximation of the exact composite model.

Each state of pure availability model keeps track of the number of non-failed channels. Each state of the performance model represents the number of talking channels in the system.

Call blocking probability is computed from pure performance model and supplied as reward rates to the availability model states.

---

Erlang loss hierarchical model

- **Upper availability model**

- **Lower performance model**
Erlang loss model (contd.)

- The steady-state system unavailability

\[ \overline{A} = \pi_0^{(u)} = \left[ \sum_{i=0}^{n} \frac{1}{i!} \left( \frac{r}{\gamma} \right)^i \right]^{-1} \]

- The blocking probability with \( i \) non-failed channels

\[ P_b(i) = \frac{(\lambda/\mu)^i / i!}{\sum_{j=0}^{i}(\lambda/\mu)^j / j!} = \pi_i^{(l)} \]

- Total blocking probability

\[ \hat{T}_b = \sum_{i=0}^{n} r_i \pi_i^{(u)} = \overline{A} + P_b(n)\pi_n^{(u)} + \sum_{i=1}^{n-1} P_b(i)\pi_i^{(u)} \]

- Blocked due to buffer full
- Blocked due to buffer full in degraded levels
- Blocked due to unavailability
Erlang loss model (cont’d)

Compare the exact total blocking probability with approximate result

- Advantages of the hierarchical
  - Avoid largeness
  - Avoid stiffness
  - More intuitive
  - No significant loss in accuracy

Total blocking probability in the Erlang loss model
Total blocking probability

- Has three summands
  - Loss due to unavailability (pure availability model will capture this)
  - Loss when all channels are busy (pure performance model will capture this)
  - Loss with some channels busy and others down (degraded performance levels)
- Performability models captures all three types of losses
- Higher level, lower level model or both can be based on analytic/simulation/measurements
Performability Evaluation (1)

- Two steps
  - The construction of a suitable model
  - The solution of the model
- Two approaches are used
  - Combine performance and availability into a single monolithic model
  - Hierarchical model where lower level performance model supplies reward rates to the upper level availability model
Performability Evaluation (2)

- Measures of performability [Haver01]
  
  - Expected steady-state reward rate (we have only used this measure in the section)
  - Expected reward rate at given time
  - Expected accumulated reward in a given interval
  - Distribution of accumulate reward
  - Expected task completion time
  - Distribution of task completion time
Outline

- Why performability modeling?
- Markov reward models
- Erlang loss performability model
- Modeling cellular systems with failure
- Multiprocessor Performability
- Conclusion
Handoffs in wireless cellular networks

- **Handoff:** When an MS moves across a cell boundary, the channel in the old BS is released and an idle channel is required in the new BS.

- **Hard handoff:** the old radio link is broken before the new radio link is established (AMPS, GSM, DECT, D-AMPS, and PHS).
Wireless Cellular System
Traffic in a cell

New Calls

Handoff Calls
From neighboring cells

Common Channel Pool

Call completion

Handoff out
To neighboring cells

A Cell
Performance Measures: Loss formulas or probabilities

- When a new call (NC) is attempted in an cell covered by a base station (BS), the NC is connected if an idle channel is available in the cell. Otherwise, the call is **blocked**.

- If an idle channel exists in the target cell, the handoff call (HC) continues nearly transparently to the user. Otherwise, the HC is **dropped**.

- **Loss Formulas**
  - New call blocking probability, $P_b$: Percentage of new calls rejected.
  - Handoff call dropping probability, $P_d$: Percentage of calls forcefully terminated while crossing cells.
Guard Channel Scheme

Handoff dropping less desirable than new call blocking!

Handoff call has Higher Priority: Guard Channel Scheme

GCS: $g$ channels are reserved for handoff calls.

$g \uparrow \downarrow$ trade-off between $P_b \uparrow$ & $P_d \uparrow \downarrow$
Assumptions

- Poisson arrival stream of new calls $\lambda_1$
- Poisson stream of handoff arrivals $\lambda_2$
- Limited number of channels: $n$
- Exponentially distributed completion time of ongoing calls $\mu_1$
- Exponentially distributed cell departure time of ongoing calls $\mu_2$
Modeling for cellular network with hard handoff

Markov chain model of wireless hard handoff

\( n(t) \): the number of busy channels at time \( t \)

\[ \Lambda(j) = \begin{cases} \lambda_1 + \lambda_2, & \text{if } i < n-g \\ \lambda_2, & \text{if } n-g \leq i < n \end{cases} \]

\[ \lambda = \lambda_1 + \lambda_2, \quad \mu = \mu_1 + \mu_2. \]

\[ A = \lambda / \mu, \quad A_1 = \lambda_2 / (\mu_1 + \mu_2). \]

\[ \pi_i = \lim_{t \to \infty} \text{Prob}(C(t) = i), \quad n \in \Omega = \{0, 1, 2, \ldots, n\} \]
Loss formulas for wireless network with hard handoff

Dropping probability for handoff:

\[ P_d(n, g) = \pi_n = \frac{\frac{A^{n-g}}{n!} A_1^g}{\sum_{i=0}^{n-g-1} \frac{A^i}{i!} + \sum_{i=n-g}^{n} \frac{A^{n-g}}{i!} A_1^{i-(n-g)}}. \]

Blocking probability of new calls:

\[ P_b(n, g) = \sum_{i=n-g}^{n} \pi_i = \frac{\sum_{i=n-g}^{n} \frac{A^{n-g}}{i!} A_1^{i-(n-g)}}{\sum_{i=0}^{n-g-1} \frac{A^i}{i!} + \sum_{i=n-g}^{n} \frac{A^{n-g}}{i!} A_1^{i-(n-g)}}. \]

\( \square \) Notation: if we set \( g=0 \), the above expressions reduces to the classical Erlang-B loss formula

\[ A = \lambda / \mu, \ A_1 = \lambda_2 / (\mu_1 + \mu_2). \]
Computational aspects

- Overflow and underflow might occur if $n$ is large
- Numerically stable methods of computation are required
  - Recursive computation of dropping probability for wireless networks
  - Recursive computation of the blocking probability
  - For loss formula calculator, see webpage: 
    http://www.ee.duke.edu/~kst/wireless.html
Optimization problems

- **Optimal Number of Guard Channels**

  **O1:** Given $n$, $A$, and $A_1$, $P_{d_0}$ determine the optimal integer value of $g$ so as to minimize $P_b(g)$ such that $P_d(g) \leq P_{d_0}$

- **Optimal Number of Channels**

  **O2:** Given $A_1$, $P_{b_0}$, $P_{d_0}$ determine the optimal integer values of $n$ and $g$ so as to minimize $n$ such that $\begin{cases} P_b(n, g) \leq P_{b_0} \\ P_d(n, g) \leq P_{d_0}. \end{cases}$
Fixed-Point Iteration

- Handoff arrival rate will be a function of new call arrival rate and call completion rates
- Handoff arrival rate will have to be computed from handoff-out throughput
- Assuming that all cells are statistically identical, handoff out throughput from a cell equals the handoff arrival rate o the cell
Loss Formulas—Fixed Point Iteration

- A fixed point iteration scheme is applied to determine the Handoff Call arrival rates:

The arrival rate of HCs = the actual throughput of handed out calls from the cell

\[ x = \lambda_2 = \frac{\mu_2 \lambda_1 (1 - P_b(x))}{\mu - \mu_2 (1 - P_d(x))}. \]

- We have theoretically proven: the given fixed point iteration is exists and is unique

- A solution by successive substitution converges fairly rapidly in practice

- A good initial value is suggested in the paper
Modeling cellular systems with failure and repair (1)

- The object under study is a typical cellular wireless system
  - The service area is divided into multiple cells
  - There are $n$ channels in the channel pool of a BS
  - Hard handoff. $g$ channels are reserved exclusively for handoff calls
  - Let $\lambda_1$ be the rate of Poisson arrival stream of new calls and $\lambda_2$ be the rate of Poisson stream of handoff arrivals
  - Let $\mu_1$ be the rate that an ongoing call completes service and $\mu_2$ be the rate that the mobile engaged in the call departs the cell
  - The times to channel failure and repair are exponentially distributed with mean $1/\gamma$ and $1/\tau$, respectively.
Modeling cellular systems with failure and repair (2)

- Upper availability model (same as that in Erlang loss model)

![Diagram of an availability model with states n, n-1, n-2, ..., 1, 0, and transitions labeled with failure rates and repair times.]

- The steady state unavailability

\[
\bar{A} = \pi_0^{(ω)} = \left[ \sum_{i=0}^{n} \frac{1}{i!} \left( \tau / \gamma \right)^i \right]^{-1}
\]
Modeling cellular systems with failure and repair (3)

- Lower performance model

\[ \Lambda(j) = \begin{cases} 
\lambda_1 + \lambda_2, & \text{if } j < i - g \\
\lambda_2, & \text{if } i - g \leq j < i 
\end{cases} \]

\[ M(j) = j(\mu_1 + \mu_2), \quad j = 1, 2, \ldots, i \]

Let \( \lambda = \lambda_1 + \lambda_2, \mu = \mu_1 + \mu_2, \) and \( A = \frac{\lambda}{\mu}, \quad A_1 = \frac{\lambda_2}{\mu_1 + \mu_2} \)
Modeling cellular systems with failure and repair (4)

- Solve the Markov Chain, we get pure performance indices
  - The dropping probability $P_d^{(l)}(i)$
    $$P_d^{(l)}(i) = \pi_{i}^{(l)} = \frac{A_i^{i-g} A_1^{i}}{i!} \frac{A_i^{i-g} A_1^{i}}{i!} \sum_{j=0}^{i-g-1} \frac{A_j}{j!} + \sum_{j=i-g}^{i} \frac{A_j}{j!} A_1^{j-(i-g)}$$
  - The blocking probability $P_b^{(l)}(i)$
    $$P_b^{(l)}(i) = \sum_{j=i-g}^{i} \pi_j^{(l)} = \sum_{j=i-g}^{i} \frac{A_i^{i-g} A_1^{j-(i-g)}}{j!} \frac{A_i^{i-g} A_1^{j-(i-g)}}{j!} \sum_{j=0}^{i-g-1} \frac{A_j}{j!} + \sum_{j=i-g}^{i} \frac{A_j}{j!} A_1^{j-(i-g)}$$
Modeling cellular systems with failure and repair (5)

Loss probability (now call blocking or handoff call dropping) is computed from pure performance model and supplied as reward rates to the availability model states

- **Total dropping probability**

  \[ T_d = \overline{A} + \pi_n^{(u)} P_d^{(l)}(n) + \sum_{i=1}^{n-1} (\pi_i^{(u)} P_d^{(l)}(i)) \]

  - Buffer full
  - Unavailability

- **Total blocking probability**

  \[ T_b = \overline{A} + \sum_{i=1}^{g} \pi_i^{(u)} + \pi_n^{(u)} P_b^{(l)}(n) + \sum_{i=g+1}^{n-1} (\pi_i^{(u)} P_b^{(l)}(i)) \]

  - Degraded buffer full
Assumptions:

- $\lambda_1$: The arrival rate of new calls
- $\lambda_2$: The arrival rate of hand-off calls into the cell
- $\mu_1$: Service completion rate of on going calls (new or hand-off)
- $\mu_2$: Service rate of hand-off outgoing calls from the cell
- $n$: Total number of channels
- $g$: Number of guard channels
Pure Performance (traffic) model:

Markov Model:
- State index indicates the number of channels in use
- Steady-state call blocking probability ($P_b$)
- Steady-state call dropping probability ($P_d$)

Markov Chain Diagram:
- State 0 to 1 with transition rate $\lambda_1 + \lambda_2$ and rate $(\mu_1 + \mu_2)$
- State $1$ to $n-g-1$ with transition rate $\lambda_1 + \lambda_2$ and rate $(C-g)(\mu_1 + \mu_2)$
- State $n-g$ to $n-g+1$ with transition rate $\lambda_2$ and rate $(C+1-g)(\mu_1 + \mu_2)$
- State $n-1$ to $n$ with transition rate $\lambda_2$ and rate $C(\mu_1 + \mu_2)$

Call blocking probability:
- $P_d = \pi_C$
- $P_b = \pi_{C-g} + ... + \pi_C$
Sharpe code for Markov Model

* Code for the Pure Performance Model
bind
  lambda1 49
  lambda2 21
  * mu + mu2 = 1
  mu1 0.35
  mu2 0.65
  g 3
end
Sharpe code (contd.)

markov perf(n)
  loop i,0,n-g-1
    $(i) $(i+1) lambda1+lambda2
    $(i+1) $(i) (i+1)*(mu1+mu2)
  end
  loop i,n-g,n-1
    $(i) $(i+1) lambda2
    $(i+1) $(i) (i+1)*(mu1+mu2)
  end
end
end
* **Pd**: Steady-state call dropping probability
  
  ```
  func Pd(n) prob(perf,$(n);n)
  ```

* **Pb**: Steady-state call blocking probability
  
  ```
  func Pb(n) sum(i,n-g,n,prob(perf,$(i);n))
  ```

* The value of n (number of channels) taken from g+1 (4) to 100

  ```
  loop nb,g+1,100,10
    expr Pd(nb), Pb(nb)
  end
  ```

end
Blocking probability vs. Number of channels
Dropping probability vs. Number of channels
Hierarchical Approximation (Performability model)

- Top level is the Availability Model
- Bottom level is a sequence of Performance Models
\( \lambda_1 + \lambda_2 \) 

\( (\mu_1 + \mu_2) \) 

\( \lambda_1 + \lambda_2 \) 

\( (C - g)(\mu_1 + \mu_2) \) 

\( \lambda_2 \) 

\( (C + 1 - g)(\mu_1 + \mu_2) \) 

\( C(\mu_1 + \mu_2) \)
Sharpe code: total call dropping probability

* function to use to define the reward rates for the measure
* the total call dropping probability
* Reward function used for \( k > g \)
  func RewDbig(k) prob(big,\$(k);k)

* Reward function used for \( k \leq g \)
  func RewDsmall(k) prob(small,\$(k);k)

* \( k \) is the number of non-failed channels
Sharpe code : total call block probability

* function to use to define the reward rates for measure
* the total call blocking probability
* The function RewB1 will be called only when $k > g$

\[
\text{func RewBbig}(k) \ \backslash \\
\sum(j, k-g, k, \text{prob}(\text{big}, $(j); k))
\]

Name of the state \quad Number of channels
Sharpe code: initialize

bind
  lambda1 49
  lambda2 21
  * mu + mu2 = 1
  mu1 0.35
  mu2 0.65
  g 3
  MTTF 1000
  MTTR 24
end
Sharpe code: model setup

* model called when k>g
markov big(k)
  loop i,0,k-g-1
    $(i) $(i+1) lambda1+lambda2
    $(i+1) $(i) (i+1)*(mu1+mu2)
  end
  loop i,k-g,k-1
    $(i) $(i+1) lambda2
    $(i+1) $(i) (i+1)*(mu1+mu2)
  end
end
end
Sharpe code : model setup

* model called when k <= g
markov small(k)
  loop i,0,k
    $(i) $(i+1) lambda2
    $(i+1) $(i) (i+1)*(mu1+mu2)
  end
end
end
Sharpe code: total call dropping probability

markov hier
loop i,n,1,-1
    $(i) $(i-1) i/MTTF
    $(i-1) $(i) 1/MTTR
end
reward
    loop i,0,g
        $(i) RewDsmall(i)
    end
loop i,g+1,n
    $(i) RewDbig(i)
end
Sharpe code: total call dropping probability (contd.)

* Initial probability
$(n) 1$
end

var Td exrss(hier)
loop nb,4,60,10
  bind n nb
  expr Td
end
Sharpe code: total call block probability

markov hier1()
loop i,n,1,-1
   $(i) $(i-1) i/MTTF
   $(i-1) $(i) 1/MTTR
end

reward
   loop i,g+1,n
      $(i) RewBbig(i)
   end
* Since the number of non-failed channels is less or equal g
* then all new calls are blocked
loop i,0,g
   $(i) 1
end
end
Sharpe code: total call block probability (contd.)

* Initial probability
$(n) 1$
end

var Tb exrss(hier1)
loop nb1,4,60,10
  bind n nb1
  expr Tb
end
Hierarchical Performability model:

- Outputs:
  - Total call blocking probability for the approximate model: $Tb_a$
  - Total call dropping probability for the approximate model: $Td_a$
Total blocking probability: Exact vs. Approximate model
Total dropping probability: Exact vs. Approximate model
Outline

- Why performability modeling?
- Markov reward models
- Erlang loss performability model
- Modeling cellular systems with failure
- Multiprocessor Performability
- Conclusion
A Performability Example

- Consider a Multiprocessor System
- How Many Processors Should It Have?
  - Vary the # procs; each with the same capacity
- Objectives
  - System Availability
  - System Performance
  - Composite Measures of Performance & Availability
Availability Benefits Of Multiprocessor

- Higher Availability/ Reliability/ MTTF
- Lower Blocking Probability (Due to system down)
  - Note: These are potential benefits
- A simple reliability block diagram or fault tree can be used for computing reliability/availability/MTTF
- In order to capture realistic behavior, we use CTMC
System Descriptions And Parameters

- $n$ Processors, at least 1 needed for System to be UP
- Each Processor Fails at Rate $\gamma$
- Each Processor is Repaired at Rate $\tau$
- Coverage Probability $c$
System Descriptions And Parameters (Contd.)

- Average Reconfiguration Delay After a Covered Failure $1/\delta$
- Ave. Reboot Delay After an Uncovered Failure $1/\beta$
- Model System Availability Using a Markov Chain
Multiprocessor Availability Model

\[
\begin{align*}
\text{n} & \quad D_n \\
\text{n-1} & \quad D_{n-1} \\
\text{n-2} & \quad B_{n-1} \\
\text{1} & \quad 0
\end{align*}
\]

\[
\begin{align*}
\gamma & \quad \tau \\
\gamma & \quad \tau
\end{align*}
\]

\[
\begin{align*}
n \gamma c & \quad (n-1) \gamma c \\
n \gamma (1-c) & \quad B_n \\
(n-1) \gamma (1-c) & \quad B_{n-1}
\end{align*}
\]

\[
\begin{align*}
\delta & \quad \beta \\
\delta & \quad \beta
\end{align*}
\]
Compute the steady state probability \( \pi_j \) for each state \( j \)

System unavailability =

\[
1 - \sum_{j=1}^{n} \pi_j = \pi_0 + \sum_{j=2}^{n} \pi_{D_j} + \sum_{j=2}^{n} \pi_{B_j}
\]
Downtime Calculation vs. no. of Processors with *Mean Delay* as parameter.
Downtime Calculation vs. no. of Processors with \( c \) as parameter.
LESSONS

- To Realize Availability Benefits of Multiprocessing
  - Coverage Must be Near-Perfect
  - Reconfiguration Delay Must be Very Small
  OR
  - Most of the Other Processors Must Be Able to Carry Out Useful Work While 1 Fault is Being Handled.

- Must Consider Different Levels of (Degradable) Performance
Performance Benefits Of Multiprocessing

- Higher Throughput
- Lower Blocking Probability
- Lower Response Time
- Lower Prob. of Missing Deadlines

Note: These are potential benefits
Use a Finite Buffer Queuing Model To Determine Prob. that the Task is rejected due to buffer full

- Task Arrival Rate $\lambda$
- Task Service Rate $\mu$
- Number in the system $b$
- Throughput = $T_b(i)$ with $i$ Processors
- Buffer Full Prob. = $q_b(i)$ with $i$ Processors
Performance model

Queuing model

\[ M/M/i/b \]
Performance Model (Contd.)

- Compute the steady state probability \( \pi_j \) for each state \( j \)
- Throughput with \( i \) Processors

\[
T_b(i) = \sum_{j \in \Omega} r_j \pi_j, \quad r_j = \begin{cases} 
# \text{ (serving)} \mu, & \text{# (serving)} + \# \text{ (proc)} = i \\
0, & \text{otherwise}
\end{cases}
\]

- Buffer Full Prob. = \( q_b(i) \) with \( i \) Processors

\[
q_b(i) = \sum_{j \in \Omega} r_j \pi_j, \quad r_j = \begin{cases} 
1, & \# \text{ (buffer)} = b - i \\
0, & \text{otherwise}
\end{cases}
\]
Combining Performance And Availability

- Attach a Reward rate $r_i$, to State $i$ of the Failure/Repair Markov Model
- We have a Markov Reward Model
- Compute the Expected Reward Rate in the Steady-State: Weighted sum of state probabilities
Combining Performance And Availability (Contd.)

- Capacity-Oriented Availability is computed by:
  - \( r_i = \frac{\text{Number of Up Processors in State } i}{n} \)

- Throughput-Oriented Availability is computed by:
  - \( r_i = 0, \) if state \( i \) is a down state
  - \( = \frac{T_b(i)}{T_b(n)}, \) otherwise
Capacity and Throughput-Oriented Availability

- **Reward assignment:**
  - \( r_i = 0, \) if \( i \) is a DOWN state
  - \( r_i = i/n, \) if \( i \) is UP (Capacity)
  - \( r_i = T_b(i)/T_b(n), \) if \( i \) is UP (Throughput)

- **Obtained measures:**

\[
\sum_{i=1}^{n} r_i \pi_i + \sum_{i=2}^{n} r_{D_i} \pi_{D_i} + \sum_{i=2}^{n} r_{B_i} \pi_{B_i} + r_0 \pi_0
\]
Total Blocking Probability (Contd.)

- $r_i = 1$ if $i$ is a down state

- $r_i = q_b(i)$ if $i$ is an up state

$$TBP = \sum_{i=1}^{n-1} q_b(i) \pi_i + \sum_{i=2}^{n} \pi_{B_i} + \sum_{i=2}^{n} \pi_{D_i} + \pi_0$$
TOTAL BLOCKING PROBABILITY
Conclusions: Performability Example

- Optimal number of processors increases
  - With The Task Arrival Rate
  - With Smaller Buffer Space
For A General System Model

- Obtain Prob. of Rejection due to System being down numerically: SHARPE, SPNP
- Obtain Prob. of Rejection due to System being full (in each configuration):
  - Analytic Queuing Network Solver: SHARPE
  - Discrete-event Simulation: Bones
  - Stochastic Petri Net Model: SHARPE, SPNP
  - Measurements
- Compute Total Blocking Probability as before
Outline

- Why performability modeling?
- Markov Reward models
- Erlang loss performability model
- Modeling cellular systems with failure
- Hierarchical model for APS in TDMA
- Multiprocessor Performability
- SHARPE input files
- Conclusion
Conclusions

- Performability: an integrated way to evaluate a real-world system
- Two approaches
  - Composite models
  - Hierarchical models
- CTMC and MRM models for performability study of a variety of wireless systems and multiprocessor system
- A tool for solution to models: SHARPE
References


• H.-R. Sun, Y. Cao, K. S. Trivedi and J. J. Han, Availability and performance evaluation for automatic protection switching in TDMA wireless system, PRDC'99, pp15--22, Dec., 1999

• http://www.ee.duke.edu/~kst/wireless.html


• K. Trivedi, O. Ibe, A. Sathaye, R. Howe, Should I Add a Processor?, 23rd Annual Hawaii Conference on System Sciences, 1990.