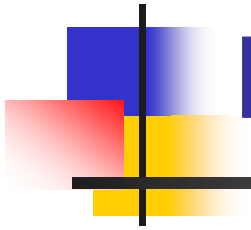
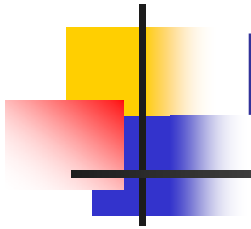


System Modeling Using Markov Chain and MATLAB

Prof. Ying-Dar Lin

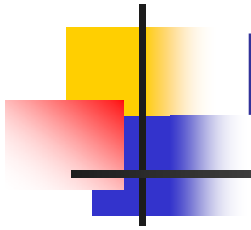
Department of Computer Science

National Chiao-Tung University



Motivation

- Systems become complicated
  - Multithreaded multiprocessor architecture
  - Communication networks
    - Bandwidth allocation
    - Scheduling
    - Routing
  - Control systems
- Too complex for simple queuing analysis
  - Multi-dimensional modeling is required



Markov Chain

- A mathematical tool to model a system
- Time-dependent variables/dimensions
  - Form a *state*
- Conditional independence of future evolution on the past
  - Past history *summarized* in current state
- An object in a discrete set of locations
  - Modeled over time
  - Next position depends *only* on current one
- Can be discrete or continuous

Discrete Time Markov Chains (DTMC)

- Assume a stochastic process

$$\{X_0, X_1, X_2 ..., X_n, ...\}$$

- Transition probability matrix *P*

$$P = \{p_{ij}, i, j \in S\} \qquad p_{ij} \geq 0, \sum_{j \in S} p_{ij} = 1$$

Let $\pi_i(n) = \Pr\{X_n = i\}, i \in S$ be the unconditional distribution at time *n*, and

$$\pi(n) = [\pi_0(n), \pi_1(n), ..., \pi_n(n), ...]$$

be the corresponding row vector

DTMC

- For n=1

$$\pi_i(1) = \sum_{j \in S} \Pr\{X_0 = j, X_1 = i\} = \sum_{j \in S} \pi_j(0) p_{ji}$$

and $\pi(1) = \pi(0)P$

Iterate the equation below

$$\pi(n+1) = \pi(n)P, n \geq 0$$

we have $\pi(n) = \pi(0)P^n$ and the *limiting state prob.* as

$$\tilde{\pi} = \lim_{n \to \infty} \pi(n) = \lim_{n \to \infty} \pi(0)P^n = \pi(0) \lim_{n \to \infty} P^n = \pi(0)\tilde{P}$$

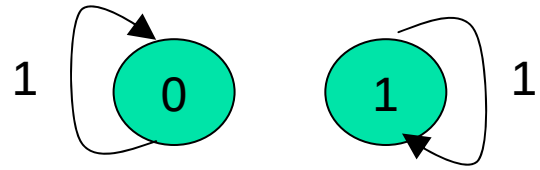
DTMC – Simple Examples

State transitions

Initial state

Transition probability matrix

System1:

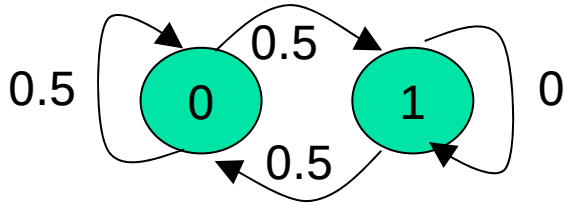


$$\begin{pmatrix} 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$[0\ 1] \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [0\ 1]$$

(The state does not involve)

System2:

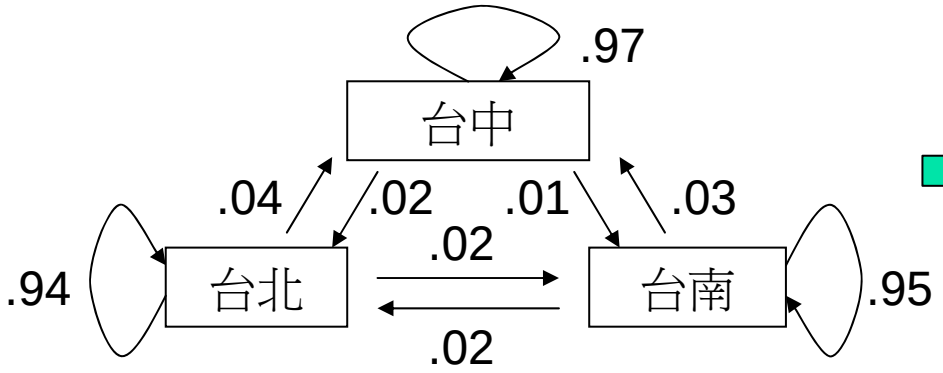


$$\begin{pmatrix} 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} .5 & .5 \\ .5 & .5 \end{pmatrix}$$

$$[0\ 1] \times \begin{bmatrix} .5 & .5 \\ .5 & .5 \end{bmatrix} = [.5\ .5]$$

(After the 1st round)

Population Movement



Transition probability matrix

	北	中	南
北	.94	.04	.02
中	.02	.97	.01
南	.02	.03	.95

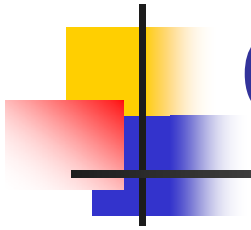
Initial state S: $\begin{bmatrix} 北 & 中 & 南 \\ .55 & .20 & .25 \end{bmatrix}$

$$S(10) = S \times P^{10} = [.38 \;\; .3733 \;\; .2464] \quad \text{(After 10 years)}$$

$$S(200) = S \times P^{200} = [.25 \;\; .5417 \;\; .2083] \quad \text{(After 200 years)}$$

$$S(201) = S \times P^{201} = [.25 \;\; .5417 \;\; .2083] \quad \text{(Steady state reached)}$$

Continuous Time Markov Chains (CTMC)

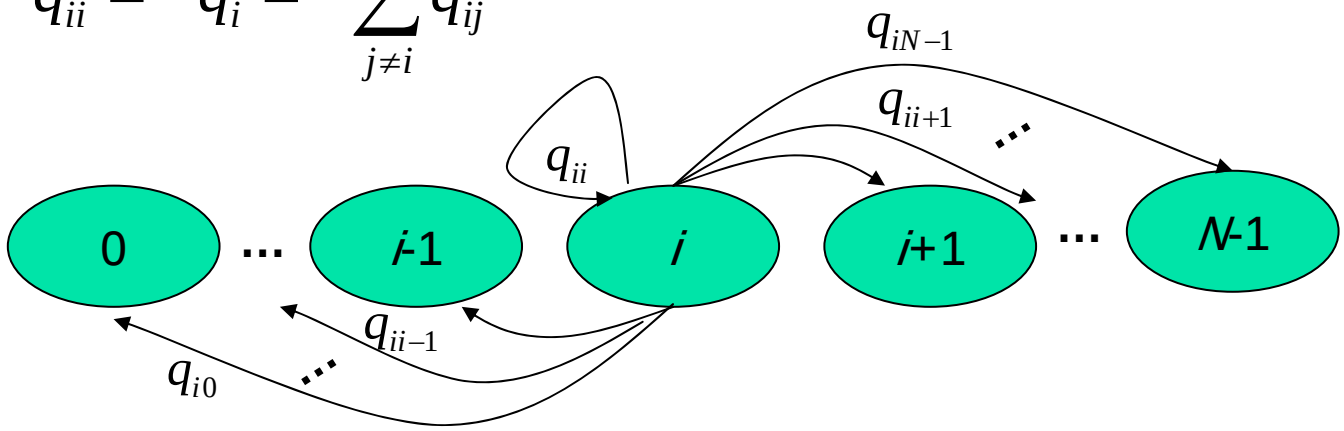


- Similar to DTMC except
  - With infinitesimally small time units
  - Transitions are activated with *rates*, not *probabilities*
- Closely model a system
  - Over time
  - Job arrival and service times are considered
- Detailed definition is not mentioned here
  - We focus on the directions of using it

Properties of CTMC

- Two properties hold for the transition rate matrix
  - Conservative:

$$q_{ii} = -q_i = -\sum_{j \neq i} q_{ij}$$



  - Stable: all q_{ij} are finite

Steady State Derivation

- Iterative multiplication: $\pi_{k+1} = \pi_k \times Q$ (transition matrix)
  - Does not converge ($\because |rate| > 1$)
- Steady-state derivation (assume NxN matrix)

$$\begin{cases} \pi \times Q = 0 & \text{----- (1)} \\ \pi \times e = 1 & \text{----- (2)} \end{cases}$$

π : **stationary prob. vector**

e : **column matrix with all elements being 1**

After some matrix manipulations with (1) and (2), we have

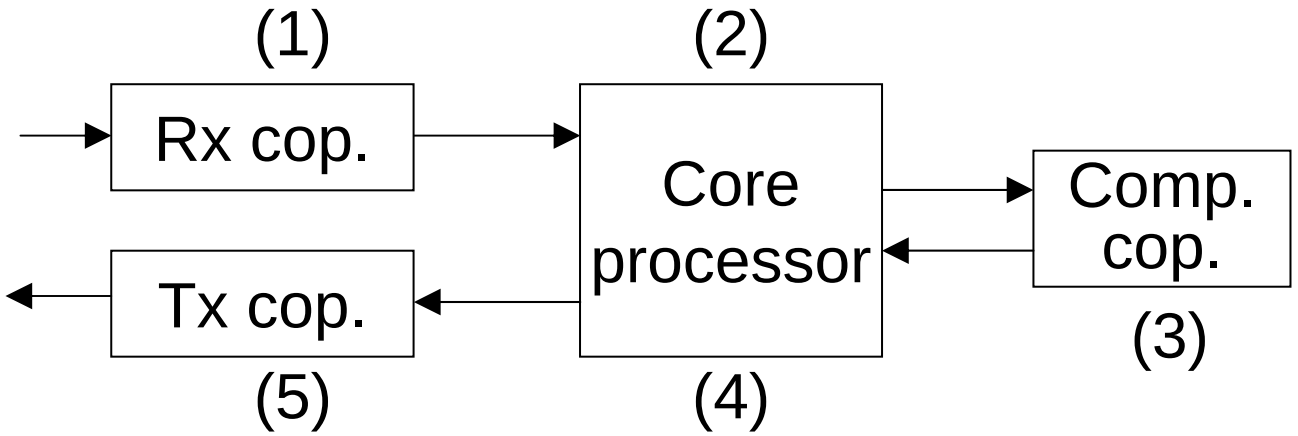
$\pi \times T = (0,0,0....,1)$, where (0,0,0,…,1) is the initial state and

$$T_{n,m} = \begin{cases} Q_{n,m}, m = 0,..., N-2 \\ 1; m = N-1 \end{cases}$$

. Therefore, we can have

$$\pi = \pi \times T \times T^{-1} = (0,0,0...,1) \times T^{-1}$$

A Sample CTMC Model



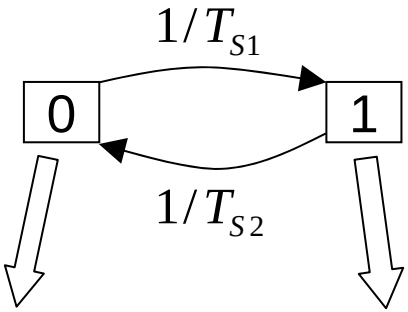
A core processor with a number of coprocessors

Task name	Processing time (μsec)
(1) Receive	27.3
(2) Core_A	31
(3) Crypto	12.6
(4) Core_B	49
(5) Transmit	27.3

Model Description

λ : Pkt arrival rate
$\lambda_{S1}, \lambda_{S2}$: Switching rate
T_{S1}, T_{S2} : Run length
μ_X : Service time of *X*
buf_X: Buffer size of *X*

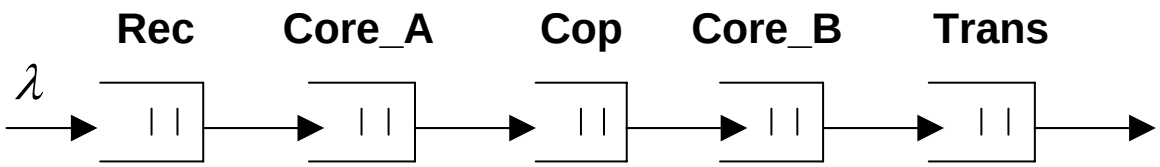
Processor Control Switch (*PCS*)



State definition:

(R, A, C, B, T, S)

buffer length



Conditions of the state transitions

Rec: *A*<buf_A => rate= μ_R;
Core_A: *S*=0, C<buf_C => rate= μ_A ;
Cop: *B*<buf_B=> rate= μ_C;
Core_B: *S*=1, *T*<buf_T => rate= μ_B ;
Trans: rate= μ_T

Transition Matrix Generation

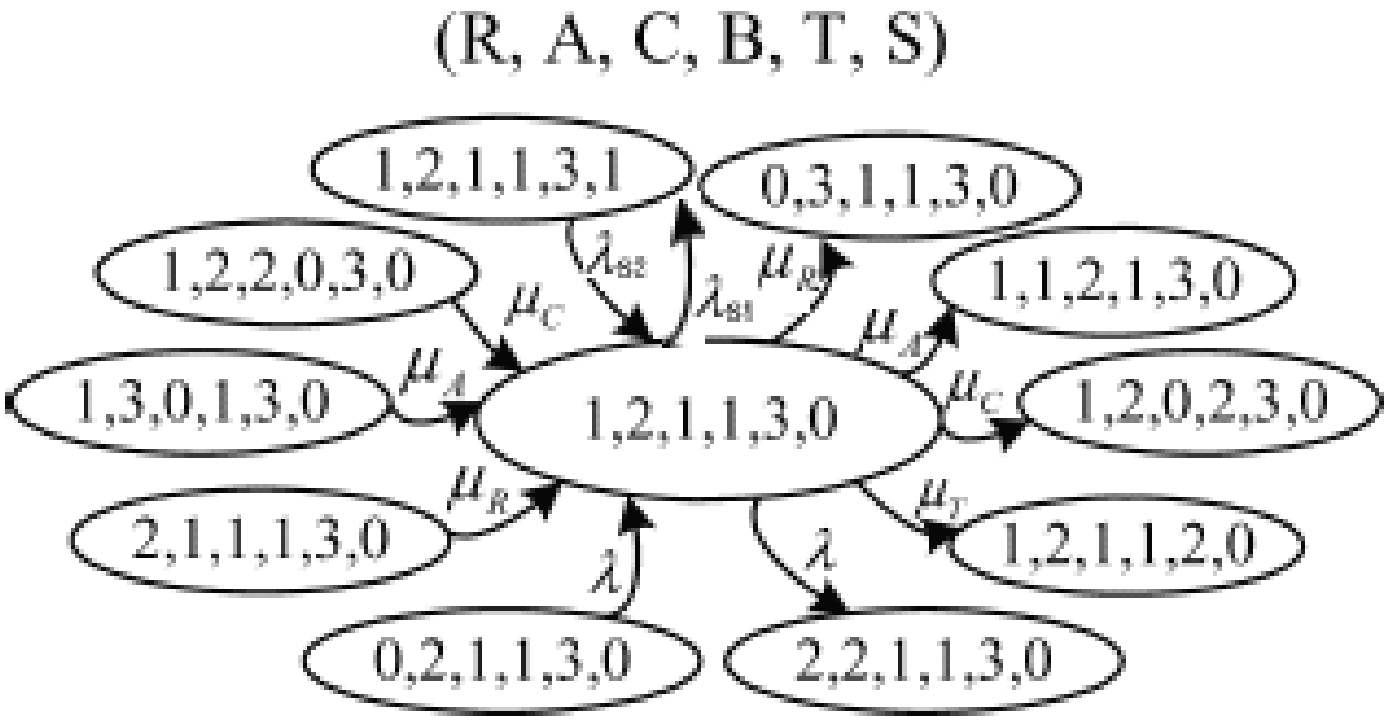
PCS: (0=>1) rate=

$$\begin{cases} 1/T_{S1} & (\text{if } A>0,\ B>0) \\ 0 & (\text{if } A=0,\ B>0) \\ 0 & (\text{if } A \geqq 0,\ B=0) \end{cases}$$

PCS: (1=>0) rate=

$$\begin{cases} 1/T_{S2} & (\text{if } A>0,\ B>0) \\ 0 & (\text{if } A>0,\ B=0) \\ 0 & (\text{if } B \geqq 0,\ A=0) \end{cases}$$

Example State Transitions

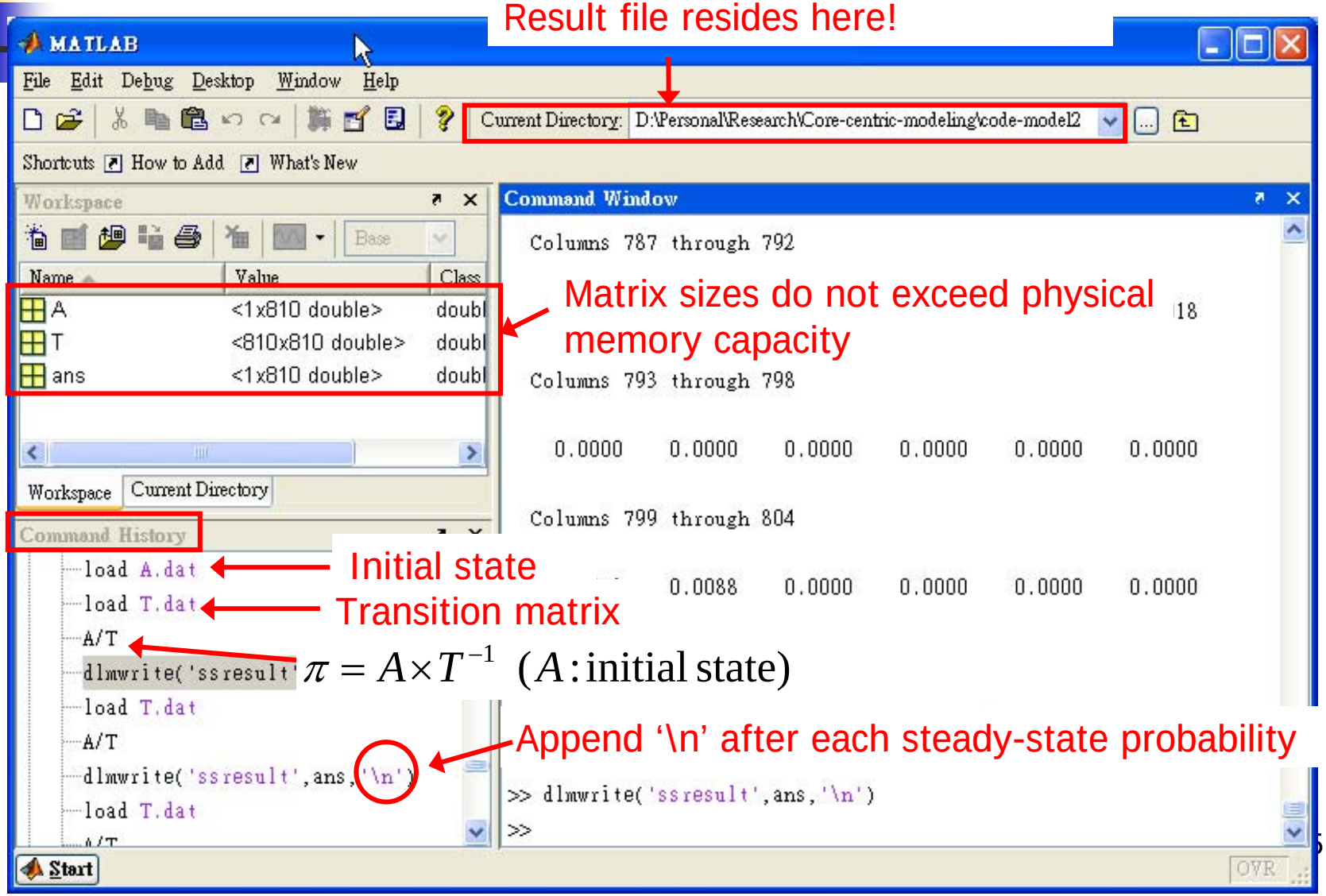


Yi-Neng Lin, Ying-Dar Lin, Yuan-Cheng Lai, "Modeling and Analysis of Core-centric Network Processors," ACM Transactions on Embedded Computing Systems, Vol. 7, No. 4, Article 41, July 2008.

Short Intro. to MATLAB

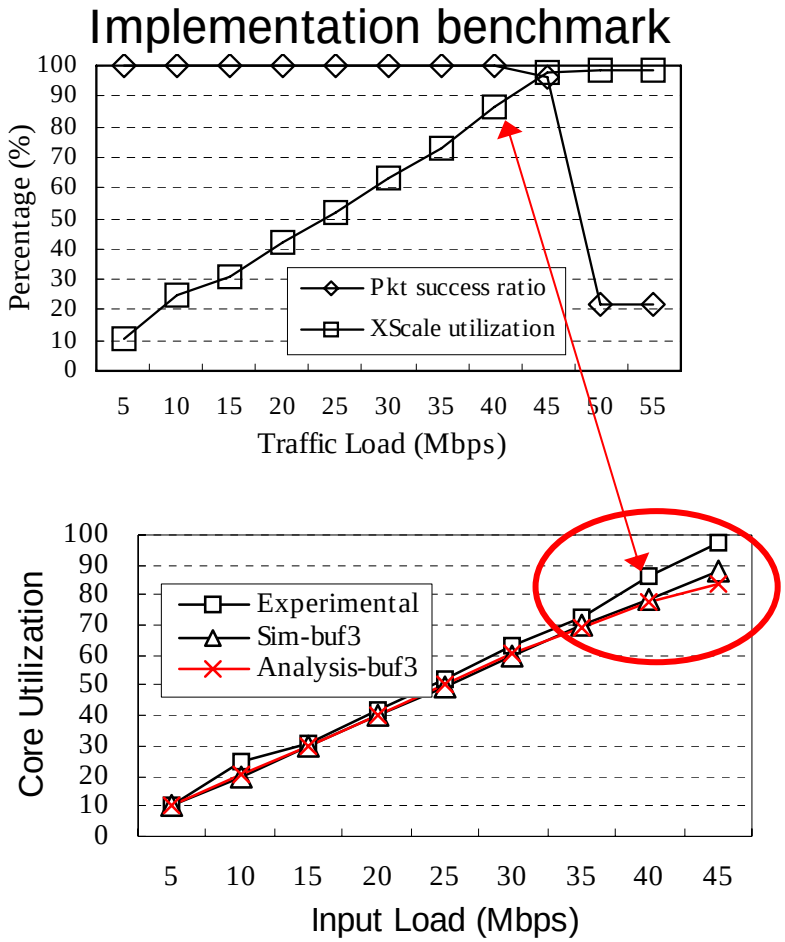
- Matrices manipulations
  - Calculate steady-state probabilities
- Commands used
  - “load”: load matrices
  - “/”: Ex: $\pi = A \times T^{-1} = A / T$ $(A : \text{initial state})$
  - “dlmwrite”: write results to a file
    - Ex: dlmwrite('ssresult',ans)
      (write results to the “ssreult” file)

User Interface

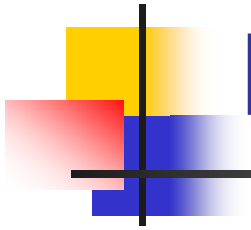


Results Collection

- Steady-state prob.
  - Recorded in “ssresult”
- Core utilization
  - Prob. that Core_A and Core_B are not 0
- Utilizations of other components?

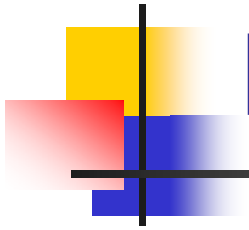


Requirements of the Mini-Project



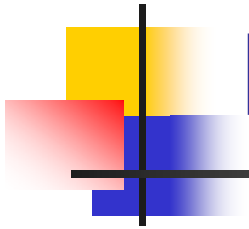
- Install Matlab7 (if you don't have one)
- Feed the input matrices
  - A: initial state matrix
  - T: transition rate matrix
  - Both of them will be provided
- Write the statistic collector
- Questions
  - Utilization of each component
    - Core (Core_A and Core_B), Rec, Cop, and Trans
  - Which one is the bottleneck?
  - What is the queue length for each component?

Requirement of the Term-Project (1/2)

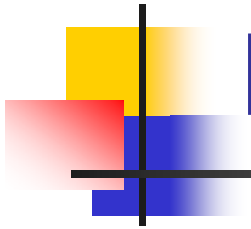


- Generate the transition rate matrix
  - According to the “Steady-State Derivation”
  - Any language is fine (Perl is recommended)
- Adopt different input loads
  - By adjusting the arrival rate (λ) (packet size = 256 bytes)
  - Step of 5% (5%, 10%, 15%, …, 40%)
  - Other parameters
    - PCS run length configured as 200 sec
    - Buffer size (buf_X) =2
    - Task processing time refer to the slides

Requirement of the Term-Project (2/2)



- Based on the mini-project
  - Feed the A and T into the MATLAB
  - Obtaining statistics of the core
    - Use the statistics collector written previously
- Task: comparison against the implementation
  - How close is your model to the implementation?
  - Why is the difference?
  - What else do you observe?



References

- *MATLAB forum*, located at http://www.mathworks.com/matlabcentral/. (the software shall be available in the Departmental Computer Center)
- *Help pages* in the MATLAB. Examples are provided for commands.
- Markov chain in Wikipedia, available at http://en.wikipedia.org/wiki/Markov_chain, and http://en.wikipedia.org/wiki/Continuous-time_Markov_process
- Any textbooks for the Markov chain.