دانشکده مهندسی کامپیوتر

عنوان درس:

مدل‌سازی و ارزیابی سیستم‌های کامپیوتری

Modeling and Evaluation of Computer Systems (MECS)

جلسه ۱۰: مدلهای پاداش مارکوف

مدرس:

محمد عبداللهی ازگمی

(Mohammad Abdollahi Azgomi)

azgomi@iust.ac.ir

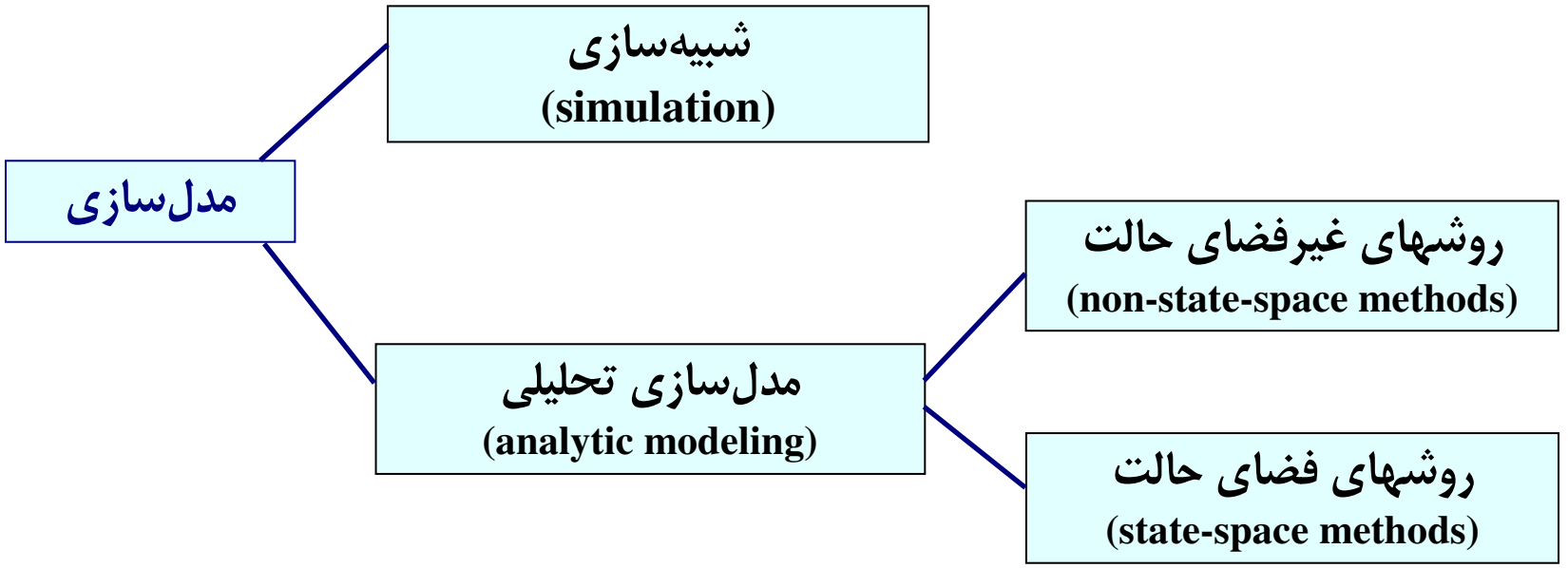
مدلهای پاداش مارکوف (Markov Reward Models)

- فهرست مطالب:
  - طبقه‌بندی مدل‌سازی (Modeling Taxonomy)
  - مدلهای پاداش مارکوف
    - تعریف صوری
    - مثال
  - ارزیابی کارایی سیستم با استفاده از مدلهای پاداش مارکوف
- مرجع:

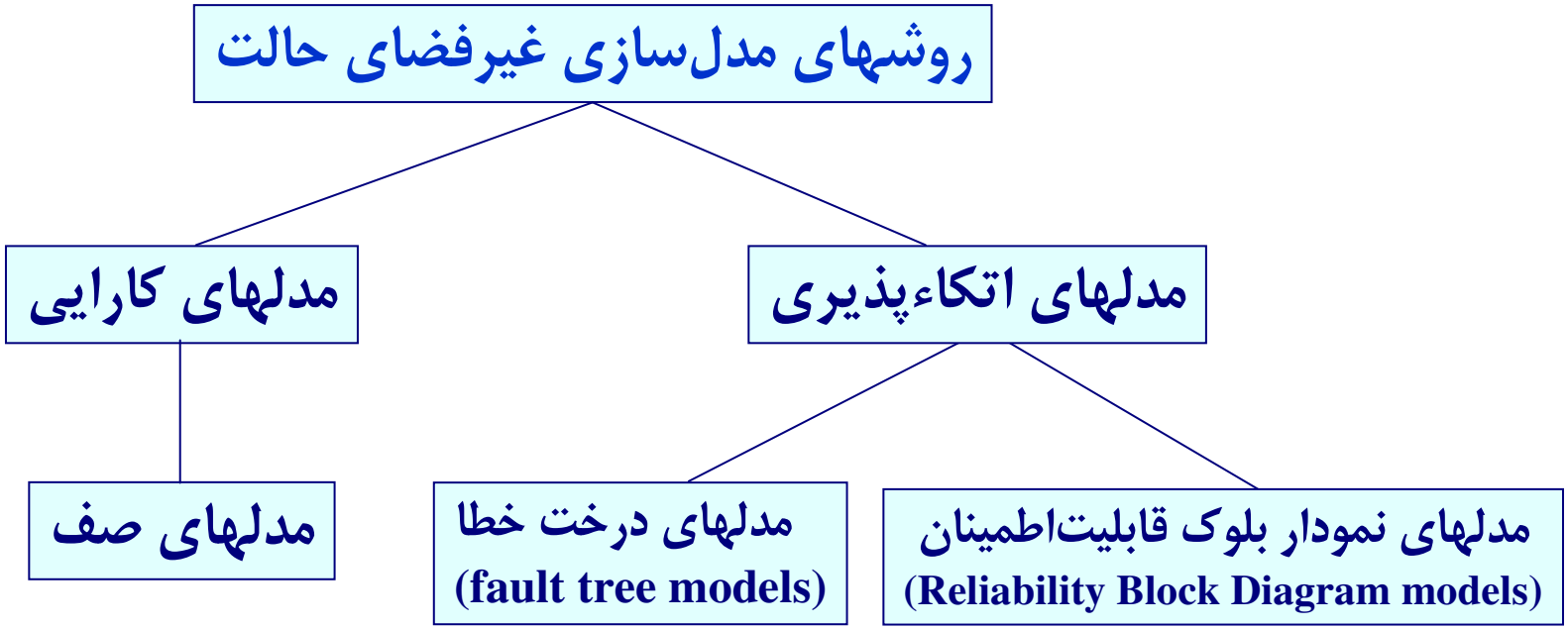
- G. Bloch et al., *Queuing Networks and Markov Chains*, 2nd Ed., John Wiley and Sons (2006)

طبقه‌بندی مدل‌سازی

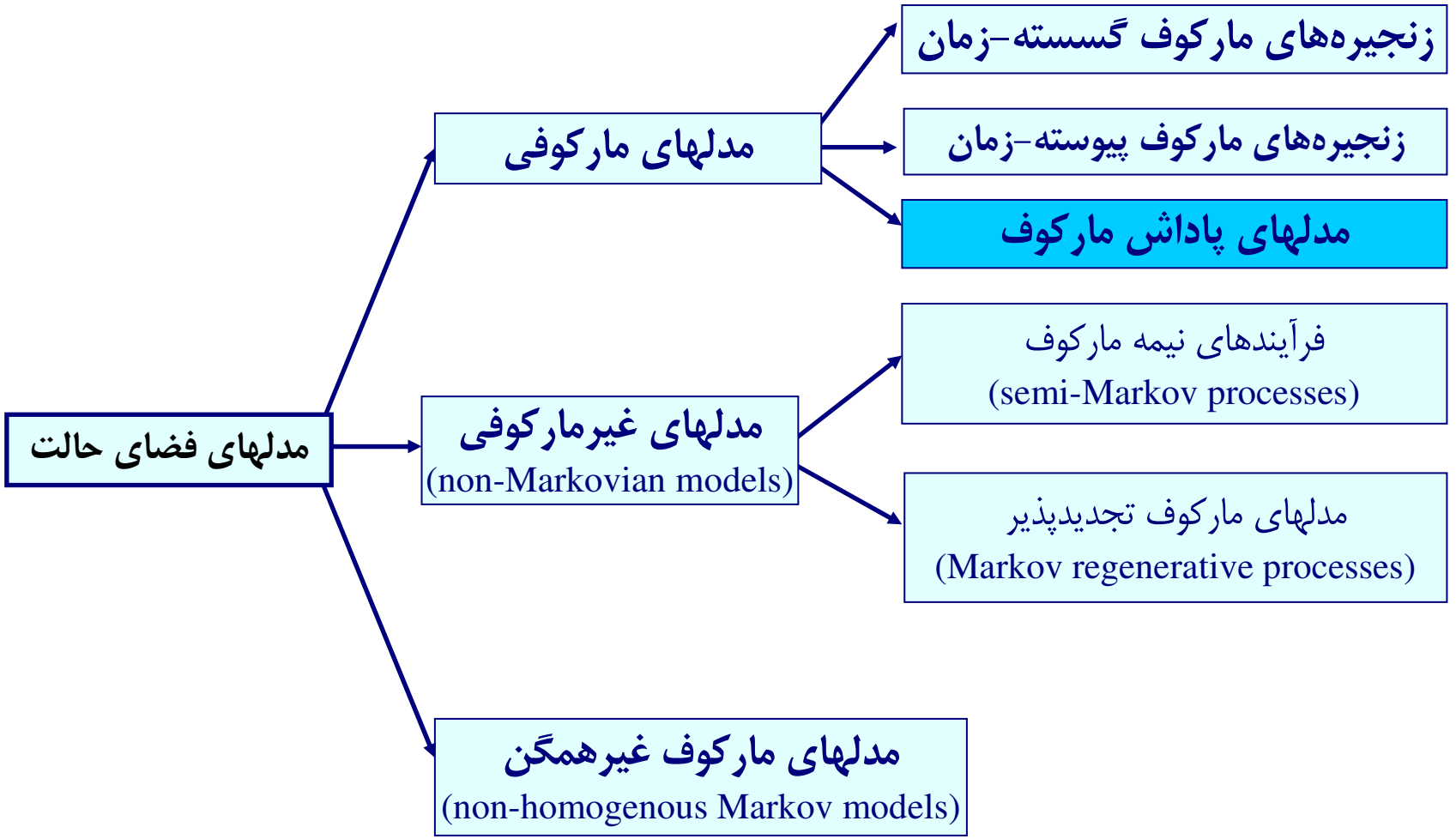
- روشهای مدل‌سازی را می‌توان به صورت زیر طبقه‌بندی نمود:



طبقه‌بندی روشهای مدل‌سازی غیرفضای حالت



طبقه‌بندی روشهای مدل‌سازی فضای حالت



مدلهای پاداش مارکوف

- مدلهای پاداش مارکوف (MRMs) یک بسط CTMC است که میزان سودمندی آنها را افزایش می‌دهد.
- این مدلها برای محاسبه معیارهای کارآیی و اتکاءپذیری مورد استفاده قرار می‌گیرند.

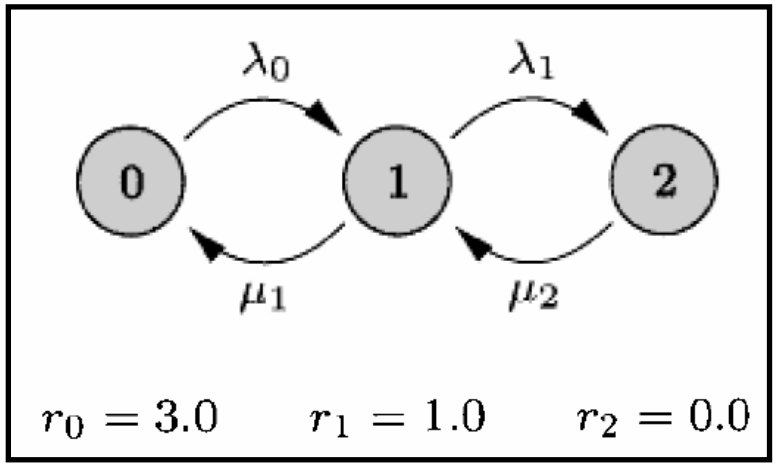
تعریف صوری

- یک مدل پاداش مارکوف شامل یک CTMC $X=\{X(t), t\geq 0\}$ با یک فضای حالت S و یک تابع پاداش r (reward function) است که : $r: S\rightarrow\Re$
- اغلب برای هر حالت $i\in S$، $r(i)$ پاداش بدست آمده در واحد زمان برای آن حالت را نشان می‌دهد.
- امکان انتساب پاداشها هم به حالتها و هم به گذرها وجود دارد.
  - *اما در ادامه فقط درمورد انتساب پاداش به حالتها صحبت می‌شود.*
- نرخهای پاداش (reward rates) مبتنی بر نیازمندیهای سیستم، نظیر کارایی، قابلیت دسترسی، قابلیت اطمینان و غیره تعریف می‌شوند.

مثال ۱

- یک CTMC سه حالته را در نظر بگیرید:

$$Q = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 \\ \mu_1 & -(\mu_1 + \lambda_1) & \lambda_1 \\ 0 & \mu_2 & -\mu_2 \end{pmatrix}$$



- بردار احتمال اولیه (initial probability vector) برابر است با: $\pi(0) = (1, 0, 0)$
- بردار نرخ پاداش (reward rate vector) برابر خواهد بود با: $r = (3,1,0)$

تعریف صوری (ادامه)

- نرخ پاداش سیستم در زمان t را با $Z(t) = r_{X(t)}$ نشان می‌دهیم.
- پاداش تجمعی (accumulated reward) در بازه [0, t] به صورت زیر نشان داده می‌شود:

$$Y(t) = \int_0^t Z(\tau)\,\mathrm{d}\,\tau = \int_0^t r_{X(\tau)}\,\mathrm{d}\,\tau$$

- امید ریاضی پاداش تجمعی (expected accumulated reward) به صورت زیر خواهد بود:

$$E[Y(t)] = \sum_{i \in S} r_i L_i(t)$$

- آنگاه $L_i(t)$ امید ریاضی کل زمانی (expected total time) است که CTMC در طی بازه [0, t] در حالت i صرف می‌کند را نشان می‌دهد، که در آن داریم:

$$\mathbf{L}(t) = \int_0^t \boldsymbol{\pi}(u)\,\mathrm{d}\,u$$

تعریف صوری (ادامه)

- اجازه دهید π_i احتمال حالت پایدار (steady-state probability) حالت i باشد.
- امید ریاضی نرخ پاداش حالت پایدار (expected steady-state reward rate) برابر خواهد بود با:

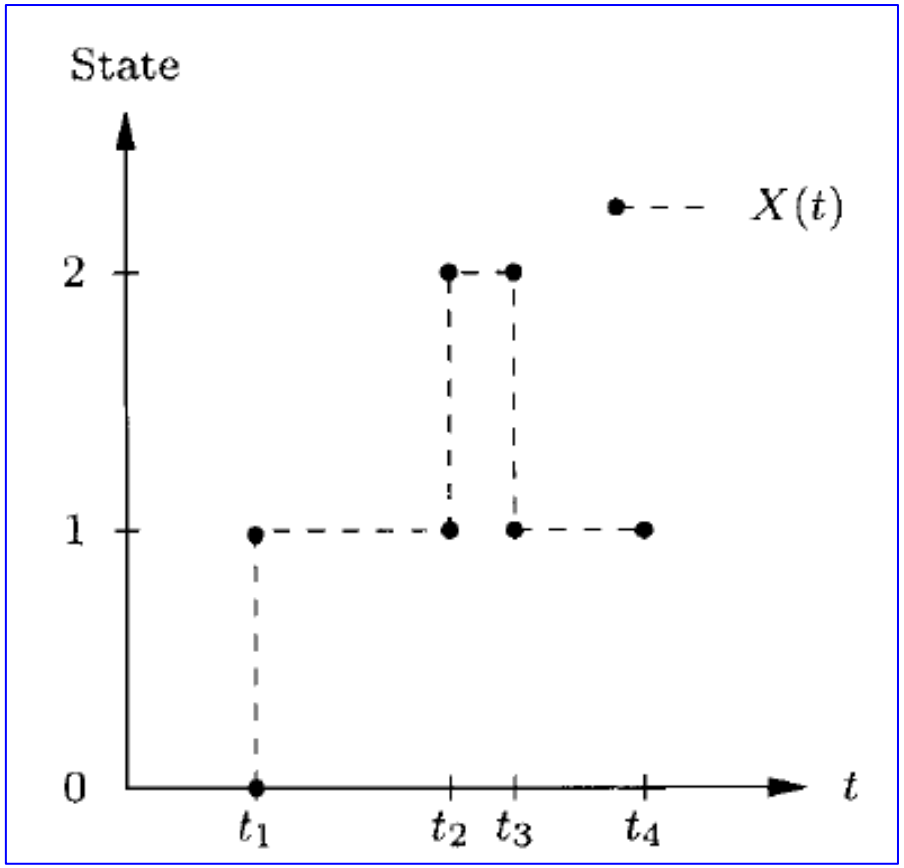
$$\lim_{t\to\infty}\left(\frac{1}{t}E[Y(t)]\right) = E[Z(\infty)] = E[Z] = \sum_{i\in S} r_i\pi_i$$

- امید ریاضی نرخ پاداش لحظه‌ای (expected instantaneous reward rate) برابر خواهد بود با:

$$E[Z(t)] = \sum_{i\in S} r_i\pi_i(t)$$

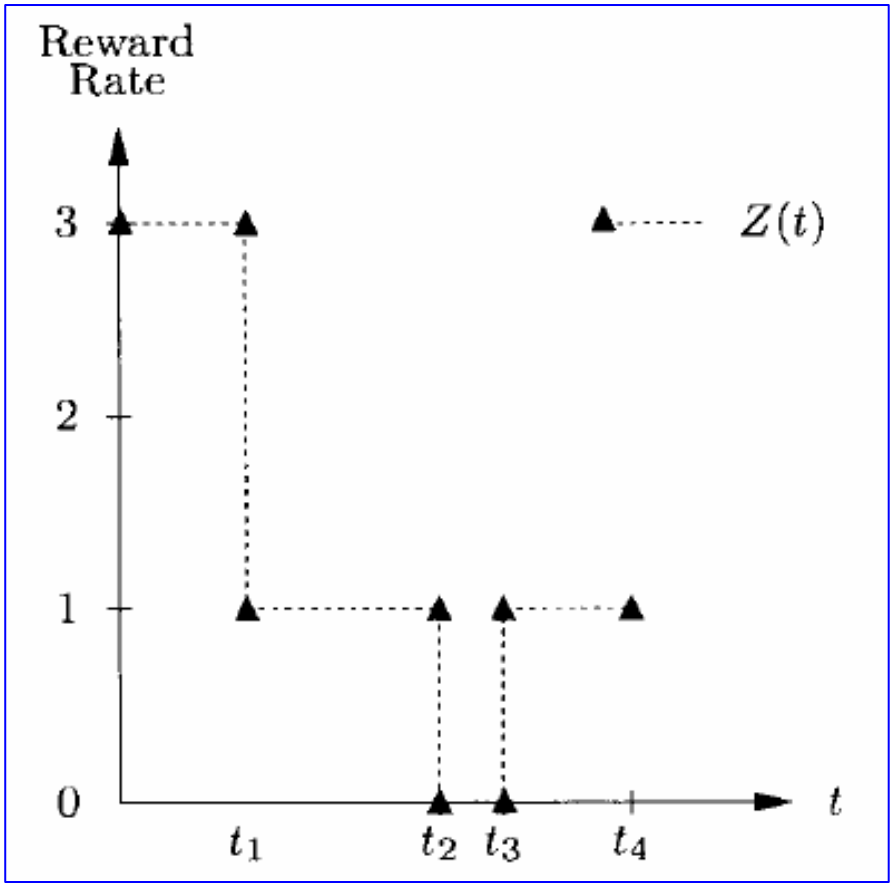
ادامه مثال ۱

- در مورد MRM مثال ۱، مسیر نمونه (sample path) رفتار فرآیند X(t) را از زمان 0 الی t مشاهده کرده‌ایم که در نمودار زیر حالتهایی را که در آن قرار داشته نمایش داده شده است:



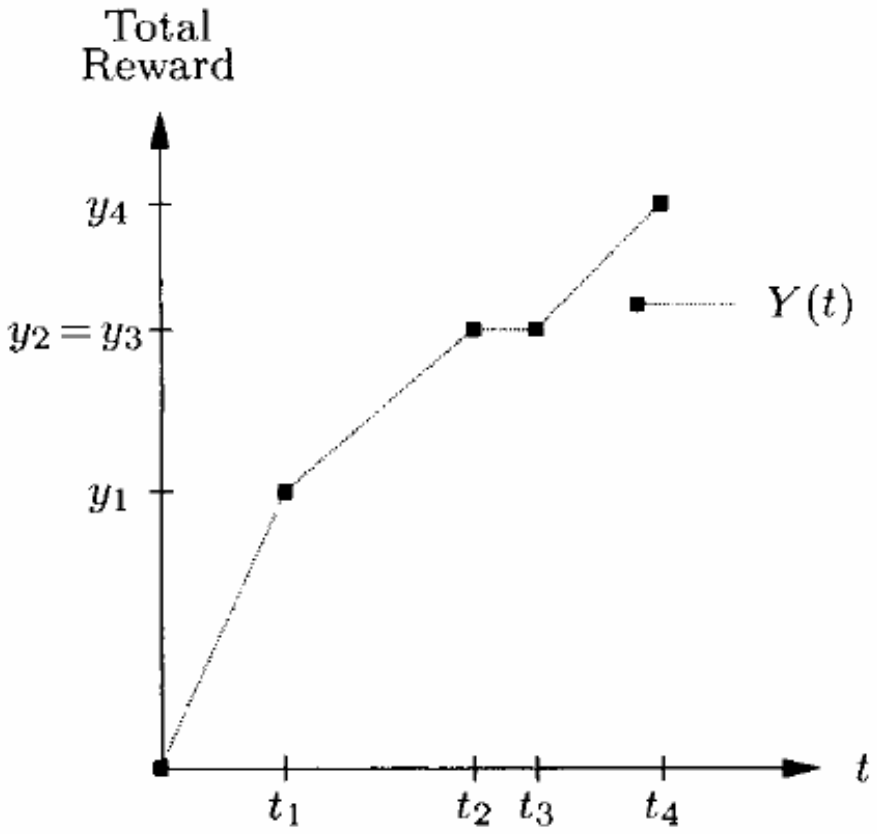
ادامه مثال ۱

- نمودار زیر نرخ پاداش سیستم در زمان t یا Z(t) را نشان می‌دهد:



ادامه مثال ۱

- حال با توجه به نمودارهای قبلی، پاداش تجمعی یا Y(t) را بدست می‌آوریم که در نمودار زیر ارائه شده است:



$$y_1 = r_0 t_1 = 3t_1$$
$$y_2 = y_1 + r_1(t_2 - t_1) = y_1 + t_2 - t_1$$
$$y_3 = y_2 + r_2(t_3 - t_2) = y_2$$
$$y_4 = y_3 + r_1(t_4 - t_3) = y_3 + t_4 - t_3$$

Finish here for DSD!

مطالعه موردی

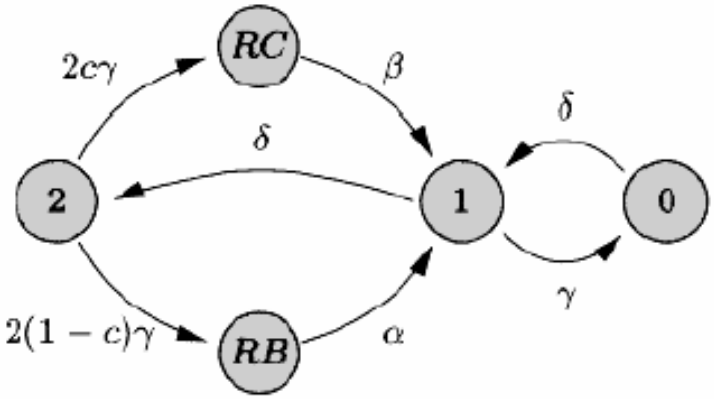
- Consider a multiprocessor system with *n* processor elements processing a given workload.
- Each processor is subject to failures with a *mean time to failure* (MTTF), $1/\gamma$.
- In case of a failure, recovery can successfully be performed with probability *c*.
- Probability *c* is called the *coverage factor* and is usually close to 1.
- Typically, recovery takes a brief period of time with mean $1/\beta$.
- Sometimes, however, the system does not successfully recover from a processor failure and suffers from a more severe impact.
- In this case, we assume the system needs to be rebooted with longer average duration of $1/\alpha$.
- Unsuccessful recovery is most commonly caused by error propagation when the effect of a failure is not sufficiently shielded from the rest of the system.

مطالعه موردی

- **Failed processors need to be repaired, with the *mean time to repair* (MTTR) of 1/δ.**
- **Only one processor can be repaired at a time and repair of one processor does not affect the proper working of the remaining processors.**
- **If no processor is running correctly, the whole system is out of service until first repair is completed.**
- **Neither reboot nor recovery is performed when the last processor fails. If all times are assumed to be independent, exponentially distributed random variables then a CTMC can be used to model the scenario…**

مطالعه موردی

- **In the following figure an example is given for the case of $n = 2$ processors and state space S = (2, *RC*, *RB*, 1, O}.**



Parameter	Meaning
$1/\gamma$	mean time to failure
$1/\delta$	mean time to repair
$1/\alpha$	mean time to reboot
$1/\beta$	mean time to recover
c	coverage probability

State	Meaning
$i \in \{0, 1, 2\}$	i working processors
RC	recovery
RB	reboot

مطالعه موردی

- **Since CTMC in of the model is ergodic, the unique steady-state probability vector will be:**

$$\pi_r = (\pi_2, \pi_{RC}, \pi_{RB}, \pi_1, \pi_0)$$

- **From Fig. 2.6 the infinitesimal generator matrix can be derived:**

$$\mathbf{Q} = \begin{pmatrix} -2\gamma & 2c\gamma & 2(1-c)\gamma & 0 & 0 \\ 0 & -\beta & 0 & \beta & 0 \\ 0 & 0 & -\alpha & \alpha & 0 \\ \delta & 0 & 0 & -(\gamma+\delta) & \gamma \\ 0 & 0 & 0 & \delta & -\delta \end{pmatrix}$$

مطالعه موردی

- **Clearly, the model is already an abstract representation of the system.**
- **Transient and steady-state probability vectors could be computed, but it is not yet clear what kind of measures should be calculated because we have said nothing about the application context and the corresponding system requirements.**
- **We take this model as a high-level description, which may need further elaboration so that a computational model can be generated from it.**
- **We consider four classes of system requirements for the example:**
  - ***System availability***
  - ***System reliability***
  - ***System performance***
  - ***Task completion***

System Availability

- *System availability* is the probability of an adequate level of sorvice, or, in other words, the long-term fraction of time of actually delivered service.
- Usually, short outages can be accepted, but interruptions of longer duration or accumulated outages exceeding a certain threshold may not be tolerable.
- Accordingly, the model must be evaluated with respect to requirements from the application context.
- First, tolerance thresholds must be specified as to what extent total outages can be accepted.
- Second, the states in the model must be partitioned into two sets:
  - one set comprising the states where the system is considered "up," i.e., the service being actually delivered, and
  - the complementary set comprising the states where the system is classified as "down."
- In our example, natural candidates for down states are in the set: {0, *RC, RB}*.

System Availability

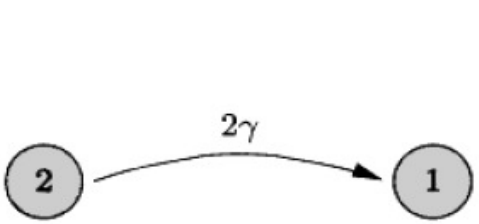
- But not all of them are necessarily classified as down states.
- Since reconfiguration generally takes a very short time, applications may well not be susceptible to such short interruptions.
- As a consequence, the less significant state *RC* could even be eliminated in the generation of the computational model.
- Finally, the measures to obtain from the computational model must be decided.
- For example, the transient probability of the system being in an up state at a certain time *t* conditioned on some initial state, the proportion of time being in one of the up states in a given finite time horizon *[0, T)* conditioned on an initial state, or the long-term proportion of time being in up states are some of the desired measures.

System Reliability

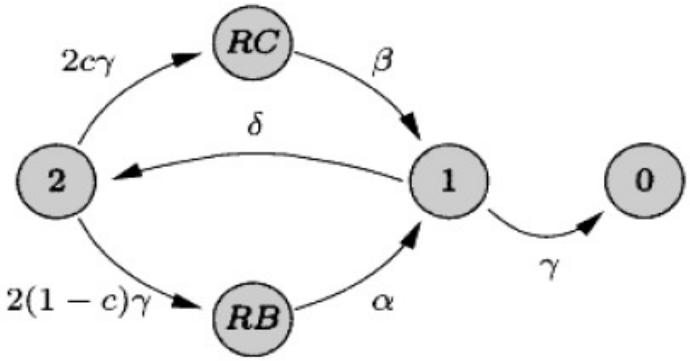
- ***System reliability*** **is the probability of uninterrupted service exceeding a certain length of time.**
- **By definition, no interruption of service at all can be tolerated.**
- **But note that it still needs to be exactly specified as to what kind of event is to be considered as an interruption.**
- **In the most restrictive application context, reconfiguration (*RC*) might not be accept.able.**
- **In contrast, nothing prevents the assumption to be made that even a rehoot (*RB*) can be tolerated and only the failure of the last component leads to the single down state (0).**
- **As a third alternative, reconfiguration may be tolerated but not reboot and not the failure of the last component.**
- **The three possible scenarios are captured in the following figure…**

Model Variants with Absorbing States Capturing Reliability Requirements.

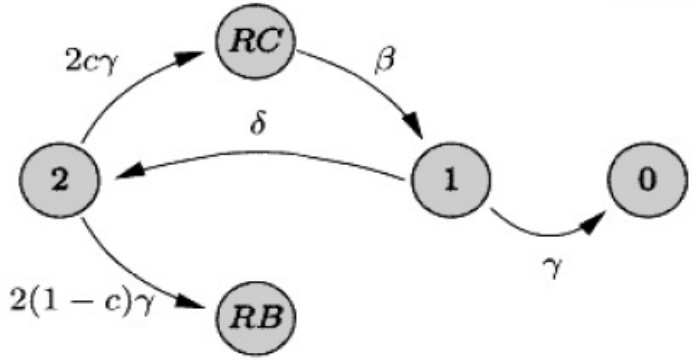
- **The model structures have also been adapted by introducing absorbing down states t,hat reflect thr, fact that down states are considered as rcpresenting catastrophic events.**



Variant 1



Variant 2



Variant 3

System Performance

- *System performance* takes the capacity of different configurations into account.
- Typical nieasures to be calculated are thc utilization of the rebources or system throughput.
- Other measures of interest relate to the fwquency with which certain incidents occur.
- With respect to the model in Fig. 2.6, the individual states need to be characterized by their contribution to a successful task completion.
- The higher the degree of parallelism, the higher the expected accomplishment will be.
- But it is a nontrivial task to firid the right performance indices attributable to each particular state in the computational model.

System Performance

- The easiest way would be to assign a capacity proportional to the number of working processors.
- But because each such state represents a whole configuration, where each system resource and the imposed workload can have an impact on the overall performance, more accurate characterization is needed.
- One way would be to execute separate modeling studies for every possible configuration and to derive some more detailed measures such as statedependent effective throughputs or response time percentiles.
- Another way is to expand the states and to replace some of them with a more detailed representation of tht: actual configuration and the workload.
- This approach will lead to a model of enormous size.

Task Completion

- *Task completion* is reflected in the probability that a user will receive service at the required quality, or in other words, in the proportion of users being satisfied by the received service and its provided quality.
- Many different kinds of measures could be defined in this category.
- It could be the proportion of tasks being correctly processed or the probability that computation-time thresholds are not exceeded.
- With advanced applications such as continuous media (e.g., audio and video streams), measures are investigated relating the degree of user's satisfaction to the quality of the delivered service.
- Usually, such application-oriented measures are composed of different constituents such as timeliness, delay-variation, loss, and throughput measures.

محاسبه معیارها با MRM

- در ادامه در کتاب با تعریف پاداشهای متناسب با محاسبه معیارهای فوق آنها را محاسبه نموده است...

ارزیابی کارایی سیستم با مدلهای پاداش مارکوف

- استفاده از نرخهای پاداش محدود به مدلهای قابلیت دسترسی، قابلیت اطمینان و انجام‌پذیری نبوده و می‌تواند در مدلهای کارایی خالص که عاری از خطا (fault-free) بوده و برای محاسبه معیارهایی نظیر توان عملیاتی، زمان پاسخ، بهره‌وری، و احتمال اتلاف کار کل، نیز مورد استفاده قرار گیرد.
- در ادامه می‌بینیم که چگونه می‌توانیم برای ارزیابی کارایی از این مدلها استفاده کنیم.

ارزیابی کارایی سیستم با مدلهای پاداش مارکوف (ادامه)

- مقادیر l_i برای مشخص کردن درصد اتلاف کارهای (percentage loss of tasks) وارده به سیستم در حالت $i \in U$ استفاده می‌شوند.

نرخهای پاداش برای توان عملیاتی

State i	Reward Rate $r(i)$
2	$1 - l_2$
RC	0
RB	0
1	$1 - l_1$
0	0

نرخهای پاداش برای محاسبه احتمال اتلاف کل

State i	Reward Rate $r(i)$
2	l_2
RC	1
RB	1
1	l_1
0	1

ارزیابی کارایی سیستم با مدلهای پاداش مارکوف (ادامه)

- امید ریاضی احتمال اتلاف کل (TLP) در حالت پایدار و حالت گذرا ، TLP(t) به صورت زیر خواهند بود:

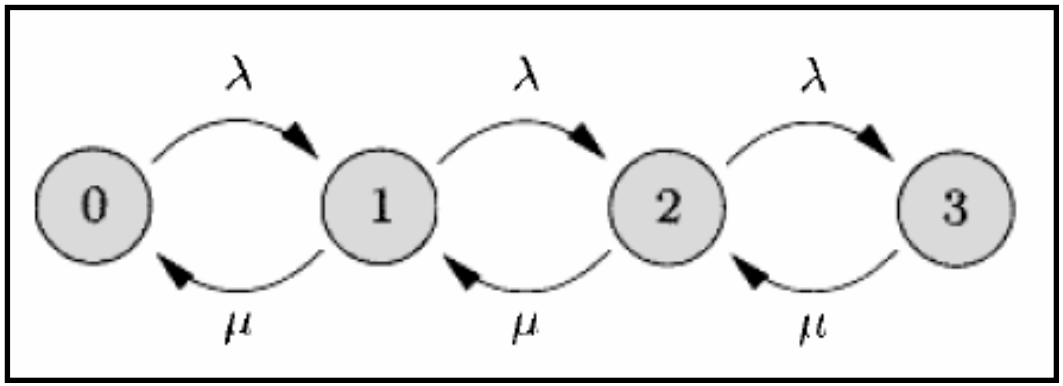
$$TLP(t) = E[Z(t)] = \sum_{i \in U} l_i \pi_i(t) + \sum_{i \in D} 1 \pi_i(t)$$
$$= l_2 \pi_2(t) + l_1 \pi_1(t) + \pi_{RC}(t) + \pi_{RB}(t) + \pi_0(t)$$
$$TLP = E[Z] = \sum_{i \in U} l_i \pi_i + \sum_{i \in D} 1 \pi_i$$
$$= l_2 \pi_2 + l_1 \pi_1 + \pi_{RC} + \pi_{RB} + \pi_0 .$$

ارزیابی معیارهای کارایی

- **توان عملیاتی:** می‌توان با انتساب نرخهای گذر حالت متناظر با خروجیها از صف (یعنی کامل شدن سرویس) به عنوان پاداش این معیار را محاسبه نمود.
- **میانگین زمان پاسخ:** می‌توان با انتساب تعداد مشتریان موجود در یک حالت به عنوان پاداش این معیار را محاسبه نمود.
- **بهره‌وری:** مبتنی بر ساختار پاداش دودویی است:
  - اگر یک منبع خاص در یک حالت مورد نظر مشغول است، نرخ پاداش ۱ خواهد بود.
  - در غیر این صورت نرخ پاداش ۰ خواهد بود، که نشان‌دهنده بیکاری منبع است.

ارزیابی معیارهای کارایی (ادامه)

- فرض کنید که مشتریان با نرخ λ به یک سیستم وارد شوند، نرخ سرویس μ بوده و یک سرویس‌دهنده داریم...



نرخهای پاداش بهره‌وری

State i	Reward Rate r_i
3	1
2	1
1	1
0	0

نرخ‌های پاداش میانگین تعداد مشتریان

State i	Reward Rate r_i
3	3
2	2
1	1
0	0

نرخ‌های پاداش توان عملیاتی

State i	Reward Rate r_i
3	μ
2	μ
1	μ
0	0

ارزیابی معیارهای کارایی (ادامه)

- توان عملیاتی:

$$\lambda^{(3)} = E[Z] = \sum_i r_i \pi_i = \mu\left(\pi_3 + \pi_2 + \pi_1\right)$$

- میانگین تعداد مشتریان:

$$\overline{K^{(3)}} = E[Z] = \sum_i r_i \pi_i = 3\pi_3 + 2\pi_2 + 1\pi_1$$

- میانگین زمان پاسخ:
  - با استفاده از قانون لیتل:

$$\overline{T^{(3)}} = \frac{1}{\lambda}\overline{K^{(3)}} = \frac{1}{\lambda}\left(3\pi_3 + 2\pi_2 + 1\pi_1\right)$$

- بهره‌وری:

$$\rho^{(3)} = E[Z] = \sum_i r_i \pi_i = \pi_3 + \pi_2 + \pi_1$$

مدلهای مبتنی بر پاداش

- مدلهای سطح بالایی وجود دارند که تولید مدلهای پاداش مارکوف را می‌توانیم به‌طور خودکار با استفاده از ابزارهای مربوط به آنها انجام دهیم.
- نمونه‌هایی از این مدلهای سطح بالا عبارتند از:
  - شبکه‌های پاداش تصادفی (SRNs: stochastic reward nets)
  - مدلهای پاداش مبتنی بر شبکه‌های فعالیت تصادفی

    (stochastic activity network (SAN) based reward models)