A Strategy For Teaching Markov Chains

Juraj Smieško

Department of InfoCom Networks
University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovakia
smiesko@kis.fri.uniza.sk

Abstract

This article presents a new strategy for teaching Markov chains to computer engineering students. The process starts with simulation methods, continues with numerical computations and culminates in analytical solution. At the beginning, we motivate students by explanation of importance of 2-state Markov chain in IP traffic modeling. Using the simulations, we perform a statistical estimate of steady-states distribution. Next, we construct a mathematical model and solve it using numerical methods. We teach the analytical solution, which in this case is the most effective one, last. The same strategy can be used to model Ethernet using a 3-state Markov chain.

Keywords

Markov chains, Markov modulated process, Bernoulli process, Matlab, teaching strategy

1. Introduction

Currently we see a problem in teaching mathematics and mathematical modeling in undergraduate courses of engineering majors. The students often come from their previous schools with bad habits and distaste for mathematics. Therefore when teaching the computer engineering students, we try various ways how to motivate them to look for their own solutions. One of the successful approaches is to first simulate the problem, which makes it attractive to the students, and then lead them to look for the analytical solution by statistical analysis of the simulation. In this article, we demonstrate how we introduce the Markov chains in the course IP Network Theory Fundamentals.

2. Motivation

The simplest stochastic model of packet flow, i.e. IP traffic, is a Bernoulli process. It is defined by two properties - the events (packets, ATM cells, bits...) in flow are independent, and the probability of an event occurring in the given time slot does not change in time, i.e. is the same for all time slots. In other words, a Bernoulli process is a sequence of independent, identically distributed random experiments with exactly two possible outcomes. We assume that there can be only one event in any given time slot. This assumption is justified by the IP network technology, where in the lowest physical layer, we cannot have 2 bits at the same time, no matter how fast the transmission is.

The independence of packets is also a strong assumption. Students usually object that the IP traffic does not have this property. Of course, if we talk about only one IP service, the independence of packets is out of question. However, if we are considering the core network flow, in which thousands of IP services are combined, the dependency of individual packets vanishes. We can confirm this statement by offering students the measured samples of core network IP traffic. Statistical analysis of these samples confirms that their behavior corresponds to the Bernoulli process.

Bernoulli process can be simulated easily (Fig. 1). A good way how to introduce the concept to the students is to explain that the process can be considered equivalent to tossing an unfair coin – with heads falling with probability p and tails with q =1-p (we can call this a p-coin).

All included code snippets and figures are from Matlab. Process increments are represented using vector a.

```
max = 50000;
p = 0.5; a(1) = 1;
for i = 2:max
    r = rand;
    if r<p
         a(i) = 1;
    else a(i) = 0;
    end
end
t = linspace(1,max,max); plot (t,a,0,2)
```

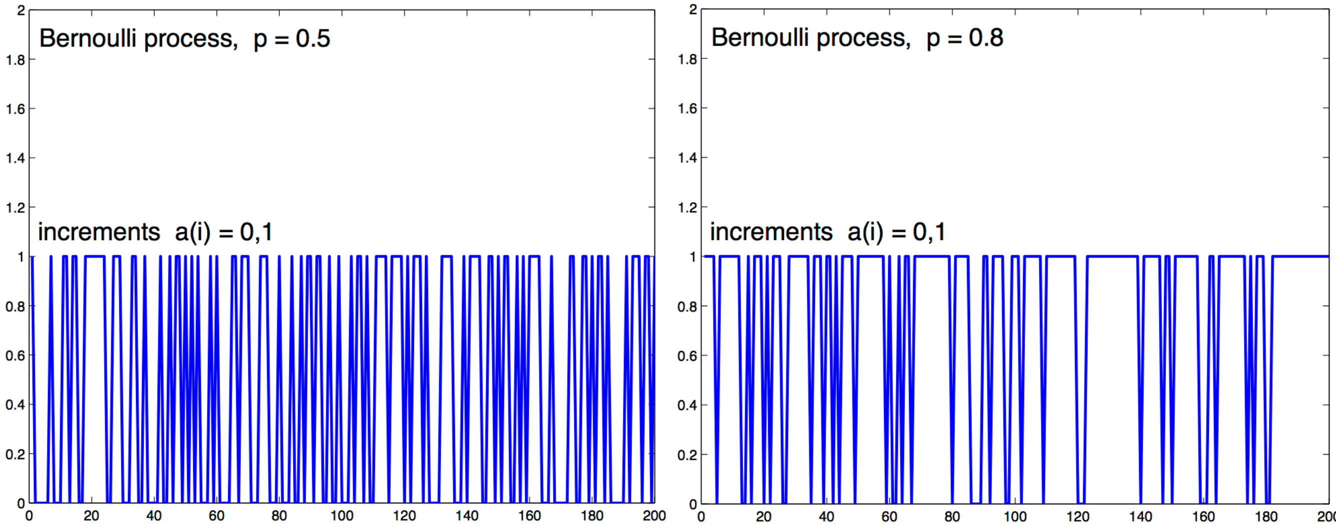


Figure 1. Bernoulli process with p = 0.5 and p = 0.8.

As a next step, we explain to the students that another important type of network process is the so-called bursty process (Fig. 2), which significantly influences, for example, the buffer activity. Bursty process has an On, i.e burst, period, and Off, i.e. silent, period. We show the students that they cannot simulate this process just by changing the parameter p, as in the previous figures. This way we lead them to the idea that in order to generate a bursty process, they need two unfair coins, α-coin and β-coin. The α-coin will generate 1 in case of Heads and switches to the β-coin in case of Tails. The β-coin works similarly, but in case of Heads, generates 0. The probabilities of switching between the coins can be denoted by α and β and are called transition probabilities between coins.

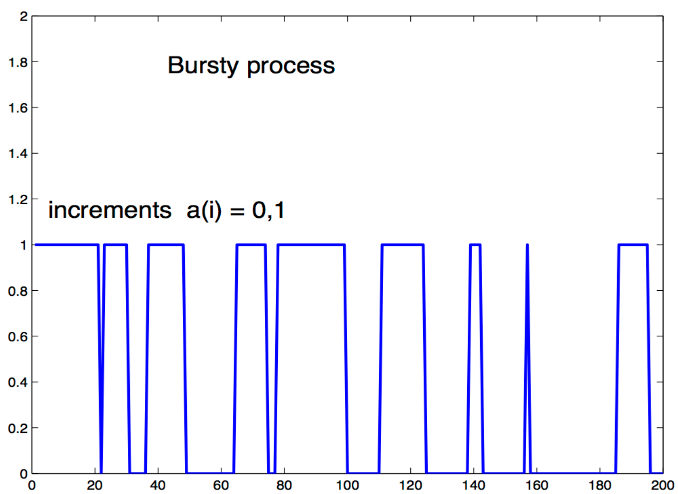
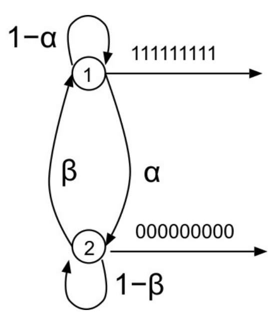


Figure 2. Bursty process.

3. Simulation of the On/Off process



So far, we have described, how we lead the students to the idea that in order to simulate an On/Off process, we will need to switch between two unfair coins using certain predefined rules. We can state that one toss models one time slot – this means we work with a random process in discrete time. If we get Heads on the α-coin, 1 is generated, if we get Tails, we switch coins. Similarly, if we get Heads on β-coin, 0 is generated, if we get Tails, we switch coins. The simulation process can be illustrated using the following 0/1 sequence:

$a(i)$	1	1	0	0	0	1	0	1	1	1	0	0	0	1	1
Pr(.)	1	1-α	α	1-β	1-β	β	α	β	1-α	1-α	β	1-β	1-β	β	1-α

We point out to the students that the probability of getting 1 is not always the same – it can be either

β or $1-\alpha$. The same is true for occurrences of 0, with probabilities α or $1-\beta$. This is the main difference from the Bernoulli process, where 1 is always generated with the same probability p and 0 with the same probability $q = 1 - p$.

The values 0 and 1 represent the traffic increments in the time slot and we record them in the variable $a(.)$. The default initial condition is $a(1) = 1$. Since we consider the coin tosses to be independent and we do not change coins in time, the values $a(i) = 0/1$ are generated using the same probability rules, that means we are working with a random process with i. i. d. (independent, identically distributed) increments.

For simulations, we can use the previous code for both coins, we just need to introduce a new variable that will indicate which coin is currently in use, i.e. the current state of the system.

```
max = 50000;
alpha = 0.1; beta = 0.3; s=1; a(1)=1;
for j=2:max
    if s==1
        r=rand;
        if r<alpha
             s=2; a(j)=0;
        else a(j)=1;
        end
    else r = rand;
        if r<beta
            s=1; a(j)=1;
        else a(j)=0;
        end
    end
end
x = linspace(1,max,max); plot (t,a,0,2)
```

The students can experiment with α, β parameter settings (Fig. 3) so that they get either a bursty process or process without bursty periods and consider whether this model can be used to simulate the Bernoulli process.

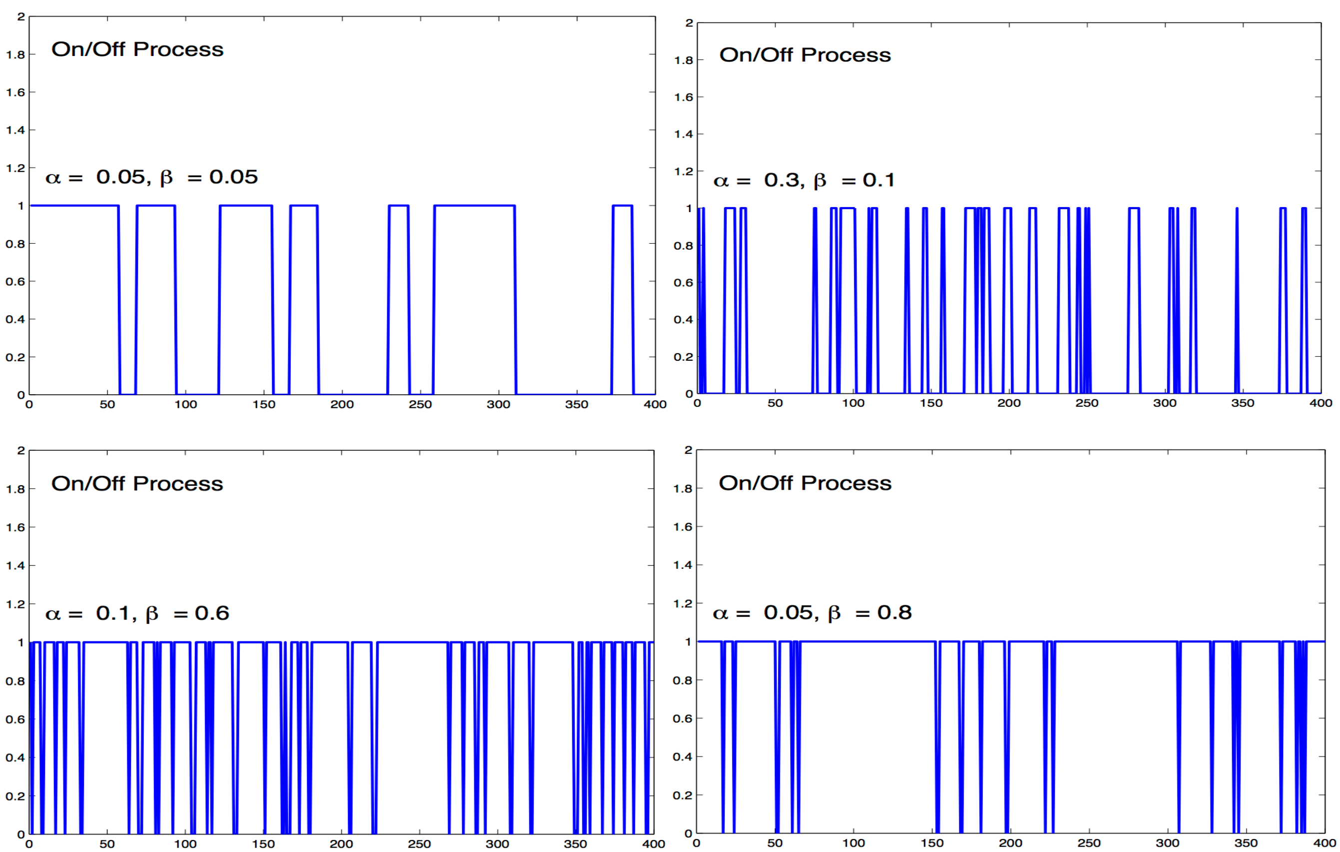


Figure 3. On/Off process with different parameters.

For better visual comparison of traffic, we introduce the so-called cumulative process $A(t)=\sum_{i=1}^{t} a(i)$ (Fig. 4). In Matlab code, we change the graphical visualization at the end of program:

```
A=cumsum(a);  t=linspace(1,max,max);  plot(t,A);  hold on
```

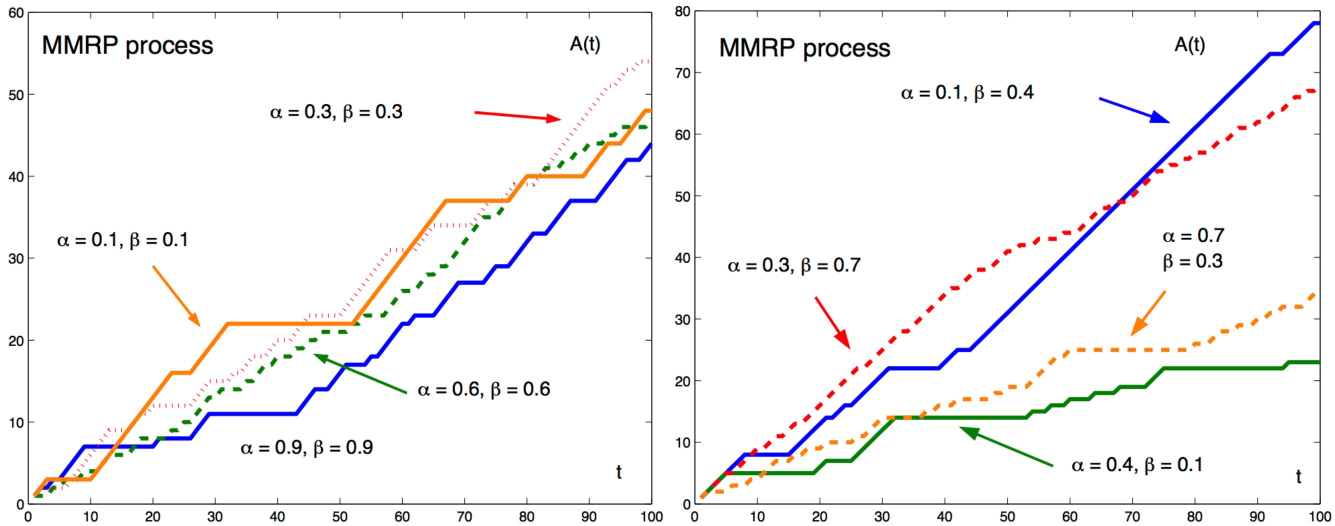


Figure 4. Cumulative process.

At the beginning of the simulation, we set the system into the On state, using the α-coin, and value 1 was generated. We denote this state by s_1. How can we determine the probability that the system will be in the state s_1 again at the time t = 8 using the simulation?

We introduce the term state probabilities. We denote the probability that the system is in state s_1 at the time t by $p_1(t)$. At the beginning of the simulation, the system was set to state 1, that means that for the probability of the initial state s_1 we have $p_1(0) = 1$ and for the state s_2 (Off) $p_2(0) = 0$. We demonstrate the state probabilities using a short sequence 10110100:
$p_1(1) = 1$, $p_1(2) = 1/2$, $p_1(3) = 2/3$, $p_1(4) = 3/4$, $p_1(5) = 3/5$, $p_1(6) = 4/6$, $p_1(7) = 4/7$, $p_1(8) = 4/8$

The probability of the state s_1 changes depending on the number of 1s generated into the given time slot. The information about the number of generated 1s is stored in the cumulative process $A(t)$. For the probability of the state s_2, we have $p_2(t) = 1 - p_1(t)$. Again, we modify the final part of our Matlab code:

```
t=linspace(1,max,max);  p1=A./t;  p2=1-p1;  plot(t,p1,t,p2);
```

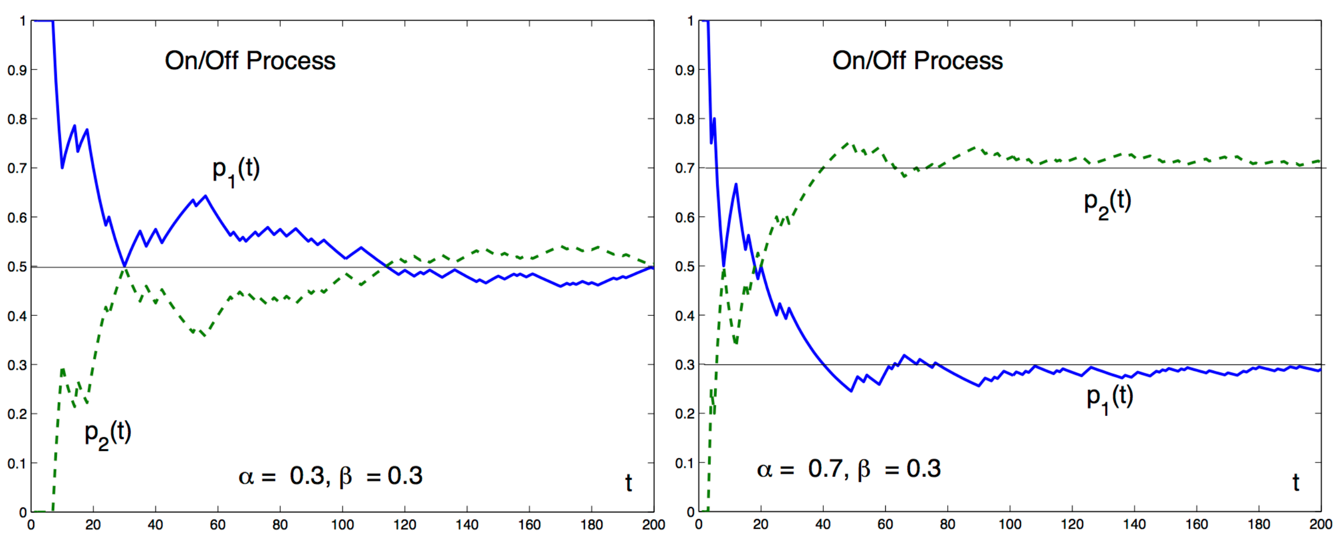


Figure 5. State distributions of On/Off process in time.

After experimenting with various parameter values and different simulation lengths, the students know that the probability values converge in time. This is the proper time to state the Markov theorem either in proper mathematical terms or in a simplified version:

In a non-periodical transitive chain (all states are mutually accessible and there exists at least one loop), the state probabilities converge in time to the time-independent steady-state distribution.

4. Numerical solution

From simulations, we move on to the numerical solution. The first question is, how to represent our system in the computer. We need to know the system states and how to switch between them. Let us denote the transition probability between states s_i a s_j in one time slot by $p_{i,j}$. The rules for switching between states can be described using the transition matrix $P = \begin{pmatrix} p_{1,1} & p_{1,2} \\ p_{2,1} & p_{2,2} \end{pmatrix} = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}$. At the beginning, time $t = 0$, we are in the on state and therefore the initial distribution is the vector $\mathbf{p}(0) = (p_1(0), p_2(0)) = (1, 0)$.

We will compute the probability that the system will still be in the state s_1 after the time slot, taking into account all possibilities, i.e. using the complete probability theorem. For the ease of understanding, we take specific values of parameters, e.g. $\alpha = 0.1$ and $\beta = 0.7$:

$$p_1(1) = p_1(0)p_{1,1} + p_2(0)p_{2,1} = 1\cdot 0.9 + 0\cdot 0.7 = 0.9$$

Similarly, we compute the probability that the system stopped transmission in the next time step:

$$p_2(1) = p_1(0)p_{1,2} + p_2(0)p_{2,2} = 1\cdot 0.1 + 0\cdot 0.3 = 0.1$$

Both of these calculations can be done using matrices:

$$\mathbf{p}(1) = \mathbf{p}(0) \cdot \mathbf{P} = (1, 0) \begin{pmatrix} 0.9 & 0.1 \\ 0.7 & 0.3 \end{pmatrix} = (0.9, 0.1)$$

Similarly, we calculate the state of the system after 2 ms, i.e. the chain distribution at $t = 2$:

$$\mathbf{p}(2) = \mathbf{p}(1) \cdot \mathbf{P} = (0.9, 0.1) \begin{pmatrix} 0.9 & 0.1 \\ 0.7 & 0.3 \end{pmatrix} = (0.88, 0.12)$$

Using iterations in Matlab, we can calculate the probability distributions in the subsequent time slots:

$$\mathbf{p}(3) = (0.876, 0.124),\ \mathbf{p}(4) = (0.8752, 0.1248),\ \mathbf{p}(5) = (0.87504, 0.12496).$$

We see that the probability values converge in time and this corresponds to the Markov theorem. Now, let us return to the recurrent formula for probability:

$$\mathbf{p}(n) = \mathbf{p}(n-1)\cdot\mathbf{P} = \mathbf{p}(n-2)\cdot\mathbf{P}^2 = \mathbf{p}(n-3)\cdot\mathbf{P}^3 = \ldots = \mathbf{p}(0)\cdot\mathbf{P}^n$$

In Matlab, we obtain the n^{th} power of matrix **P** using a simple command **P**^n:

$$\mathbf{P}^2 = \begin{pmatrix} 0.88 & 0.12 \\ 0.84 & 0.16 \end{pmatrix},\ \mathbf{P}^3 = \begin{pmatrix} 0.876 & 0.124 \\ 0.868 & 0.132 \end{pmatrix},\ \mathbf{P}^4 = \begin{pmatrix} 0.8752 & 0.1248 \\ 0.8736 & 0.1264 \end{pmatrix},$$

$$\mathbf{P}^5 = \begin{pmatrix} 0.87504 & 0.12496 \\ 0.87472 & 0.12528 \end{pmatrix},\ \mathbf{P}^6 = \begin{pmatrix} 0.875008 & 0.124992 \\ 0.874944 & 0.125056 \end{pmatrix}$$

Looking at the transitional matrices, we see that the probabilities of states at time t, $p_j(t)$, and probabilities of transitions at time t, $p_{k,j}(t)$, converge to the same limits for any states k. This again follows from the Markov theorem.

In order to find the limits of the probability states, we need to compute a high power of the matrix. In our case, it is sufficient to take $n = 20$, $\mathbf{P}^{20} = \begin{pmatrix} 0.875 & 0.125 \\ 0.875 & 0.125 \end{pmatrix}$.

Using simple numerical methods, we have obtained the steady-state probability distribution of Markov chain, which we denote by $\boldsymbol{\pi} = (\pi_1, \pi_2) = (0.875, 0.125)$. Note that both lines of $\mathbf{P}^{20}$ matrix are the same, which means that steady-state distribution $\boldsymbol{\pi}$ is independent of initial distribution $\mathbf{p}(0)$, which is again a consequence of the Markov theorem.

5. Analytical solution

We are at the point when the students can simulate a 2-state Markov chain and also numerically find its steady-state distribution. This is the time to introduce all the used ideas in proper mathematical terms.

A random process in discrete time $X(t)$ is a sequence of random variables $\{X_i\}_i$, which at time $i = t_i$ have values s_i, $X(t_i) = X_i = s_i$. We will call the random process in discrete time a random chain. We denote the probability that a chain at time t is in the state s_k by:

$$p_k(t) = \Pr(X(t) = s_k)$$

The vector $\mathbf{p}(t) = (p_0(t), p_1(t), p_2(t), \ldots)$ is called the state distribution of chain at the time t and the vector $\mathbf{p}(0) = (p_0(0), p_1(0), p_2(0), \ldots)$ the initial distribution at the time $t = 0$.

We call the stochastic process $X(t)$ a **Markov chain**, if it has a countable set of states $S = \{s_0, s_1, \ldots, s_n\}$ and Markov property holds for the conditional probabilities:

$$\Pr(X(t_n) = s_{m_n} \mid X(t_{n-1}) = s_{m_{n-1}}, X(t_{n-2}) = s_{m_{n-2}}, \ldots, X(t_0) = s_{m_0}) = \Pr(X(t_n) = s_{m_n} \mid X(t_{n-1}) = s_{m_{n-1}})$$

Markov property means that there is no memory, i.e. the state in the next time slot only depends on current time slot and not on the previous ones.

We call the Markov chain **homogenous** if the transition probability between states does not depend on time t:

$$\Pr(X(t_1) = s_j | X(t_1 - 1) = s_i) = \Pr(X(t_2) = s_j | X(t_2 - 1) = s_i) = p_{i,j}$$

In homogenous Markov chain, the probability $p_{i,j}$ denotes the probability of transition from state s_i to state s_j in one time step regardless the current time slot:

$$p_{i,j} = \Pr(X(t) = s_j | X(t - 1) = s_i)$$

Markov chain is uniquely determined by its set of states and transition probabilities between these states.

We call the Markov chain **transitive** if all states are mutually achievable: $\forall\, s_i, s_j,\ \exists\, t_n, t_m;\ p_{i,j}(t_n) > 0,\ p_{j,i}(t_m) > 0$

When modeling the On/Off process, we considered independent coin tosses without change in coins in time. Therefore, the On/Off process can be modeled as a 2-state homogenous transitive Markov chain.

Markov theorem

Let Markov chain $X(t)$ be transitive and have a loop in at least one state, $p_{i,i}(1) > 0$. Then the state probabilities and transit probabilities will eventually stop being time dependent:

$$\forall j : \lim_{n\to\infty} p_j(n) = \pi_j, \forall i, j : \lim_{n\to\infty} p_{i,j}(n) = \pi_j$$

The Markov theorem says that the Markov chain with certain properties stabilizes in time, i.e. the probability distribution stops being time dependent. We call the probability vector $\boldsymbol{\pi} = (\pi_0, \pi_1, \pi_2, \ldots.)$ steady-state probabilities of Markov chain. It can be obtained by taking the recurrent relations to the limit:

$$\mathbf{p}(n) = \mathbf{p}(n-1)\cdot\mathbf{P} \Rightarrow \boldsymbol{\pi} = \boldsymbol{\pi}\cdot\mathbf{P}$$

The system of equations is linearly independent, so we can replace any one equation by the normalization condition

$$\sum_{\forall j} \pi_j = 1$$

In order to compute the invariant distribution, we can use the following operations:

$$\boldsymbol{\pi} = \boldsymbol{\pi}\cdot\mathbf{P} \ \Rightarrow\ \boldsymbol{\pi}[\mathbf{E}-\mathbf{P}] = 0 \Rightarrow [\mathbf{E}-\mathbf{P}]^{\mathrm{T}}\, \boldsymbol{\pi}^{\mathrm{T}} = 0$$

We then compute the steady-state distribution of the On/Off process:

$$\mathrm{P} = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}, [\mathrm{E} - \mathrm{P}]^T \pi^T = \begin{pmatrix} \alpha & -\beta \\ -\alpha & \beta \end{pmatrix} \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow \alpha\pi_1 - \beta\pi_2 = 0 \Rightarrow \pi_2 = \frac{\alpha}{\beta}\pi_1 \Rightarrow \pi_1 + \frac{\alpha}{\beta}\pi_1 = 1 \quad \Rightarrow \pi_1 = \frac{\beta}{\alpha+\beta}, \pi_2 = \frac{\alpha}{\alpha+\beta}$$

For our example we get:

$$\alpha = 0.1, \beta = 0.7, \pi_1 = 0.875, \pi_2 = 0.125$$

In the previous section, we have obtained an estimate of these values using numerical methods.

6. Probability flow conservation law

We can find the recurrent relations that govern the probability states using a fast and efficient method (Fig. 6), which is also suitable for engineering. It follows from the **probability flow conservation law**.

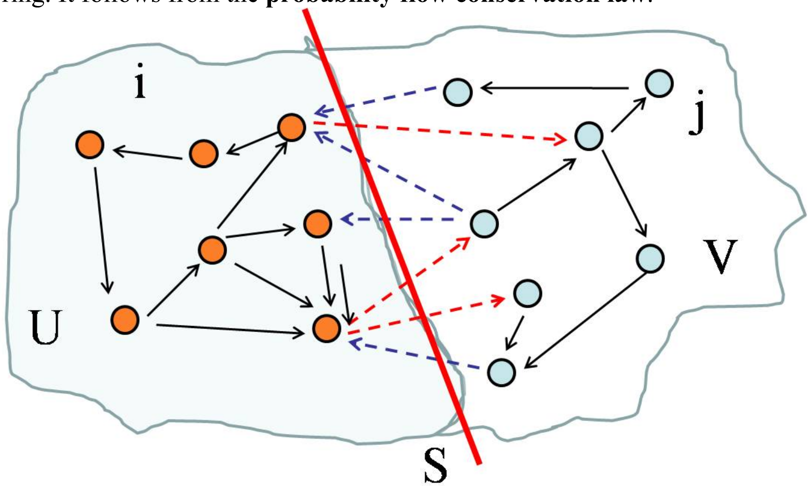


Figure 6. Conservation of probability flow.

Suppose the Markov chain set of states S has a partition U a V. Then we have (proof in [1]):

$$U \cap V = \varnothing, U \cup V = S, \sum_{i \in U} \sum_{j \in V} \pi_i p_{i,j} = \sum_{j \in V} \sum_{i \in U} \pi_j p_{i,j}$$

Let us return to the 2-state Markov model of the On/Off source of IP traffic. According to this law, we can make a cut (Fig. 7) between the states s_1 and s_2 and write the equations for steady-state distribution using the probabilities of flow through the cut.

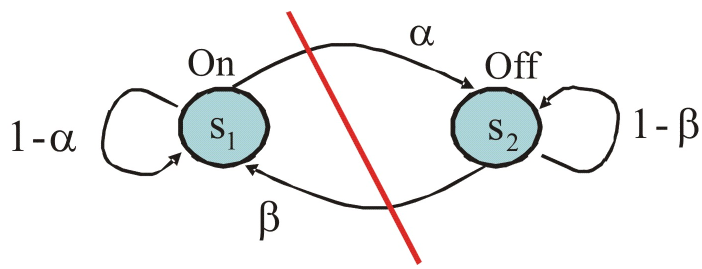


Figure 7. The cut between states s_1 and s_2.

$$\pi_1 \alpha = \pi_2 \beta \Rightarrow \pi_2 = \frac{\alpha}{\beta}\pi_1 \Rightarrow \pi_1 + \frac{\alpha}{\beta}\pi_1 = 1 \quad \pi_1 \alpha = \pi_2 \beta \Rightarrow \pi_2 = \frac{\alpha}{\beta}\pi_1 \Rightarrow \pi_1 + \frac{\alpha}{\beta}\pi_1 = 1 \quad \Rightarrow \pi_1 = \frac{\beta}{\alpha + \beta}, \pi_2 = \frac{\alpha}{\alpha + \beta}$$

Using the cut and normalization condition, we have derived the equations for π_1 and π_2.

7. Markov model of Ethernet CSMA/CD

There are several computers connected to one bus. When a user wants to transmit, her computer sends a signal to the bus to check the bus availability. It is possible that another user will try to check availability at the same time or during the checking period. In this case, we have a signal collision and both computers stop their transmissions. They generate a random time period after which they repeat the process. If one computer finds a free bus, it will start transmission. During one transmission, other users cannot start their own transmissions.

Thus, the system can be in one of 3 states, I-idle, T-transmit and C- collision. In order to properly model the described behavior, only these transitions are possible:

$$\mathrm{I} \rightarrow \mathrm{I}, \mathrm{I} \rightarrow \mathrm{T}, \mathrm{I} \rightarrow \mathrm{C}, \qquad \mathrm{T} \rightarrow \mathrm{I}, \mathrm{T} \rightarrow \mathrm{T}, \quad \mathrm{C} \rightarrow \mathrm{I}$$

We describe the system by a homogenous 3-state Markov chain with, for example, the following transition matrix:

$P = \begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.4 & 0.6 & 0 \\ 1 & 0 & 0 \end{pmatrix}$. The task is to compute the steady-state distribution. If we wanted to simulate this problem, we would use three coins. The one representing the Collision state would be degenerate and at each Tails on this coin, the system would switch to the Idle state. Numerical solution to the precision 10^{-4} can be easily obtained using Matlab, where is enough to compute the 19th power of the transition matrix: $P^{19} = \begin{pmatrix} 0.3922 & 0.4902 & 0.1176 \\ 0.3922 & 0.4902 & 0.1176 \\ 0.3922 & 0.4902 & 0.1176 \end{pmatrix}$.

We see that the system transmits 49% of the time. The remaining 51% it is either Idle or in Collision.

In order to get the analytical solution, we can use two suitable cuts and the normalization condition. Using the probability flow conservation law we can easily derive three equations for probability states and thus solve for the general solution of this problem.

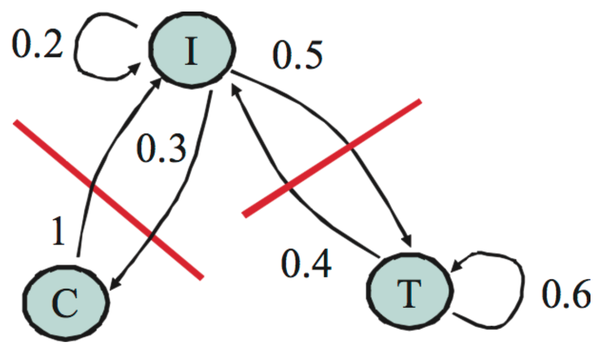


Figure 8. Two cuts for the Ethernet model.

$$0.3\pi_1 = \pi_3, 0.5\pi_1 = 0.4\pi_2 \Rightarrow \pi_2 = \frac{5}{4}\pi_1, \pi_3 = \frac{3}{10}\pi_1$$

$$\pi_1 + \pi_2 + \pi_3 = 1 \Rightarrow \pi_1 + \frac{5}{4}\pi_1 + \frac{3}{10}\pi_1 = 1 \Rightarrow \frac{51}{20}\pi_1 = 1$$

$$\Rightarrow \pi_1 = \frac{20}{51} = 0.3922, \pi_2 = \frac{25}{51} = 0.4902, \pi_3 = \frac{6}{51} = 0.1176$$

Obviously, the advantage of analytical approach is that this way we can obtain the general solution of the problem.

8. Conclusion

In this article, we have shown a teaching strategy, which appears more suitable for our students than the traditional one. The goal was not to give mathematically precise definition related to Markov chains – these can be found in [1] and [2], but rather share what our pedagogical experience confirms. Namely, that didactically, it is very appropriate to first let the students simulate the problem and look at stochastic models and the specific solutions and only later introduce the more rigorous theoretical concepts. We are trying to apply this principle also in other applied mathematics areas that we teach, e.g. [3] and [4].

References
[1] E. Gelenbere, G. Pujolle: Introduction to Queuing Networks, John Wiley & Sons Ltd., Paris, 1987
[2] J. Smieško: Základy teórie hromadnej obsluhy (Queueing Theory), textbook, 190 p., MC Energy Žilina, ISBN 80-968115-6-8, 1999
[3] K. Bachratá, I. Cimrák, O. Šuch, M. Klimo: Analýza procesov lineárnymi metódami (Process Analysis Using Linear Methods), textbook, 228 p., University of Žilina, ISBN 978-80-554-0556-8, 2012
[4] H. Bachratý, M. Grendár, K. Bachratá: Ako sa počíta pravdepodobnosť? (How To Compute Probability?), textbook, 1. Edition, 326 p., University of Žilina, Matej Bel University, 326 s., ISBN 978-80-554- 0226-0, 2010