

# Self-Similar Markovian Sources

Adam Domański <sup>1</sup> , Joanna Domańska <sup>2</sup> , Katarzyna Filus <sup>2</sup> , Jakub Szygula <sup>1,\*</sup>  and  
Tadeusz Czachórski <sup>2</sup> 

<sup>1</sup> Faculty of Automatic Control, Electronics and Computer Science, Department of Distributed Systems and Informatic Devices, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland; Adam.Domanski@polsl.pl

<sup>2</sup> Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland; joanna@iitis.pl (J.D.); kfilus@iitis.pl (K.F.); tadek@iitis.pl (T.C.)

\* Correspondence: jakub.szygula@polsl.pl

Received: 14 April 2020; Accepted: 25 May 2020; Published: 28 May 2020



**Abstract:** Markov queueing models are a powerful tool to evaluate the performance of computer networks and have been used in telecommunication studies for over 100 years. To apply them to the evaluation of the modern Internet, we should not only adapt them to the contemporary network structures but also include a description of the complex stochastic patterns (self-similarity and long-range dependance) of transmitted flows. We examine the features of two Markov models of an almost self-similar process, keeping in mind the modeling of Internet traffic. We have found that the obtained results are comparable with those achieved using a well-known generator of self-similar traffic.

**Keywords:** long-range dependence (LRD); self-similarity; traffic sources; Hurst parameter; Markov models

## 1. Introduction

Since the early 20th century and the times of Erlang and Engset, queueing models based on Markov chains have been used in the modeling of telecommunication systems. They have accompanied and supported the design process of telephone and telegraph systems, followed by computer systems and computer networks.

Typically, a queueing model is composed of a server and incoming customers which are queued while waiting for their turn to be served. The term “customers” may refer to packets queued in routers. Knowing the pattern of packet arrivals, ordering methods and service times, the model should predict the queueing time and packet drop probability. Both input and output information have a stochastic character. The servers are linked into a queueing network, reflecting a physical or a virtual topology of a studied computer network. The service time is the time needed to send the packet content to the output link; thus, it is proportional to the size of the packet. The packets’ waiting time contributes to the total transmission time and therefore is essential for the evaluation of transmission quality. The queues in routers have limited volume, and they may temporarily become full, meaning that arriving packets are lost; the loss probability given by a queueing model is another important factor of the quality of service. Both waiting time and loss probability depend on the intensity of the input flow and its stochastic nature. Therefore, the models should reflect the character of the flows closely. In this paper, we discuss their ability to capture the self-similarity of the flows—a feature that has a strong impact on the quality of transmission.

Queueing models are either analytical, arising from the queueing theory, or based on a discrete-event simulation. Both types are extensively used, and both are essential for different needs. Simulation models are more detailed, but generally demand more computations and a sophisticated

software. Two well-known simulators are OMNeT++ [1] and ns-3 [2], both of which strictly follow the behavior of network protocols and the router's performance. Analytical models are much simpler, leading to a decrease in the number of computational operations, and as a result, these models have a shorter execution time. This is especially beneficial in transient analysis, in which the discrete event simulation of such network behavior is very time-consuming due to the large number of randomized repetitions required to achieve a reasonable level of a statistical accuracy. This is even more critical in the case of self-similar traffic where, due to the complex changes of traffic, the simulations should be longer to assure a desired accuracy level. Additionally, the analytical models efficiently compute very small probabilities, e.g., loss probabilities, while a simulation needs to be long enough to observe rare events a sufficient number of times.

The question examined in this article is the following: are analytical models of traffic, which may be a part of entire analytical queueing models, correct and featured by sufficiently strong performance to compete with simulation traffic models? It seems that the answer is positive.

In Markov models, each state of a studied system corresponds to a state of a Markov chain. They may use continuous-time (CTMC) or discrete-time (DTMC) chains. In the first case, a change of state may occur at any time; in the second case, only at defined time moments can this change take place. The transition matrix defining the changes of states in a Markov chain contains transition intensities in the case of CTMC and transition probabilities in the case of DTMC. If we solve a system of Chapman–Kolmogorov equations determining the state probabilities in a Markov chain, we also obtain the state probabilities of the system (i.e., the number of equations and the number of states are the same).

The main feature of Markov chains is that they are memoryless, implying that the sojourn time at a state is exponentially distributed in the case of a continuous-time Markov chain and geometrically distributed in the case of a discrete-time chain. Therefore, the first Markov queueing models assumed exponentially distributed interarrival times (i.e., Poisson input stream) and exponentially distributed service times. These assumptions are distinct from what we observe in real networks. However, practically any distribution may be represented by a linear combination of exponentially distributed phases, so if we also include the number of the current phase in the state definition, we may construct a queueing Markov model with any interarrival and service time distributions. Of course, the number of states to be considered and the number of equations to be solved are increasing. Moreover, only a numerical solution is possible. Consequently, the number of states, which rapidly increase with the complexity of the model and result in so-called state explosion, limit the range of utility of Markov models. Nevertheless, the constant increase of the computing power makes increasingly complex models feasible. We use a number of tools to convert a distribution to a set of exponentially distributed phases—e.g., [3]—and a number of tools to numerically solve the equations for state probabilities—e.g., [4]. However, the representation of general interarrival time distributions is not the only issue; another challenge is the autocorrelation of the input flow intensity. This should also be represented in Markov models if we aim to apply this approach to the evaluation of the Internet; i.e., to predict queueing delays and their variability, as well as the packet drop probability (caused by the congestion of the queues). Such queueing models are also useful for the proper choice of active queueing management mechanisms in routers and for the detection of possible distributed denial of service (DDoS) attacks which influence the self-similarity degree of the traffic; see [5,6].

In recent years, the problem of the self-similarity and long-range dependence (LRD) of network traffic has become a popular topic of research [5–9]. Several articles [10,11] have shown that ignoring these phenomena in computer network analysis causes an underestimation of performance measures. Many studies, both theoretical and empirical, have shown that network traffic exhibits self-similar properties. The self-similarity of the network traffic has a large impact on a network performance; it enlarges the queue occupancy and increases the number of packets dropped in the network nodes [12]. Thus, it is necessary to take this feature into account in the process of creating realistic models of traffic sources [13]. If we want to apply the Markov approach in this case, we require the Markov models

of the input flow (of the source generating this flow) together with the Markov model of the server. Only then may we construct a Markov chain in which states are defined by the states of the source and the server simultaneously.

In statistics, the term “self-similar” indicates that a portion of the object can be considered as the rescaled image of the whole. It was first introduced by Benoit Mandelbrot in 1967 [14]. If we consider a continuous stochastic process, this means that the rescaled process and the original process have the same distribution [15]. Let us denote by  $Y(t)$  a continuous stochastic process. The condition that a process must satisfy to be self-similar can be written as follows [16]:

$$Y(t) \stackrel{d}{=} a^{-H} Y(at), \quad (1)$$

for  $t \geq 0, a \geq 0$  and  $0 < H < 1$ , where  $H$  denotes the Hurst parameter used to estimate the degree of self-similarity.

When dealing with network traffic, we usually examine the data in the form of a time series instead of a continuous process [17]. Any measurement of traffic intensity within a certain interval gives us a discrete time stochastic process, determined by the obtained samples; if the sequence  $X_1, X_2, \dots, X_k, \dots$  is self-similar, then for the aggregated sequence

$$X_k^{(m)} = \frac{1}{m} (X_{km-m+1} + \dots + X_{km}), \text{ where } m > 1 \quad (2)$$

the variance is equal to

$$\text{Var}[X^{(m)}] = \frac{\text{Var}[X]}{m^\beta}, \text{ where } 0 < \beta < 1, \quad H = 1 - \beta/2 \quad (3)$$

or

$$\log \text{Var} (X_k^{(m)}) \approx \log \text{Var} (X) - \beta \log m \quad (4)$$

Thus, the log–log plot of  $\text{Var} (X_k^{(m)})$  versus  $m$  is a straight line with the slope  $-\beta$ . The equations above are used in a well-known method of Hurst parameter estimation known as the aggregated variance method (it is described in [7,15]). We can use many different methods to estimate this value utilizing the aggregated series presented above or some more complex methods operating on a transformed domain [17].

The notions of self-similarity and long-range dependence (LRD) are often used as equivalents, although they are not [18]. A process that exhibits LRD is in fact an asymptotically second-order self-similar process. LRD means that some temporal similarities occur in the examined data. Let us denote by  $X_n$  a second-order stationary process. Its covariance function can be written as follows [17]:

$$r(k) = \text{Cov} (X_n, X_{n-k}) \stackrel{\Delta}{=} \mathbb{E} [(X_n - \mu)(X_{n-k} - \mu)] \quad (5)$$

and the autocorrelation is equal to

$$\rho(k) \stackrel{\Delta}{=} \frac{r(k)}{r(0)} \quad (6)$$

LRD occurs when the autocorrelation function of a self-similar process follows a power law, which means that the value of the autocorrelation function is slowly decreasing as  $k \rightarrow \infty$ :

$$\rho(k) \sim k^{-\beta} \text{ as } k \rightarrow \infty, \quad (7)$$

where  $\beta \in (0, 1)$ . Returning to Equation (1); for  $H \in (0, 0.5)$ , LRD does not occur, and for  $H > 0.5$ , the process is correlated, which means that LRD occurs. For the correlated processes, the value of  $H$  determines the “length of the memory”. For the value  $H = 0$ , the process is a random curve (it is memoryless) and for  $H = 1$  it is deterministic. The number of time-scales for which the plot changes

linearly defines a number of time-scales for which the autocorrelation persists. A straight-line fit can be used to determine the averaged value of the Hurst parameter over the examined time-scales. We will refer to it while displaying the results of experiments in the form of figures in Section 3.

The physical consequence of the self-similar intensity of flows in computer network transmissions is that very long periods of high traffic may happen during which queues and waiting times continuously increase, and we observe massive losses of packets.

Many methods are used to simulate self-similar internet traffic, among which fractional Gaussian noise (FGN) (introduced in [19]) and Poisson Pareto burst process (introduced in [20]) are the most popular. Nevertheless, the Markovian models can still be useful in this field [18]. In Section 2 we briefly describe two Markovian models: the special semi-Markov process (SSMP), and Markov modulated Poisson process (MMPP) and one non-Markovian Poisson Pareto burst process (PPBP). SSMP may be a part of a DTMC queueing model, MMPP may be used in a CTMC model, and PPBP is often recommended for simulations of self-similar Internet traffic; however, neither of them is a perfect self-similar process. In Section 3 we present the results achieved in the experiments. We investigate and compare their ability to mimic self-similarity over several time scales. Section 4 concludes the article.

## 2. Markov Chains and the Markov Traffic Sources

Several models have been introduced for self-similar and LRD traffic source modeling. The majority of these models have been based on approaches which are different than the Markovian approach. Some of them use the  $\alpha$ -stable distribution [21], chaotic maps [22], fractional autoregressive integrated moving average (fARIMA) [23] or fractional Levy motion [24]. The advantage of the non-Markovian models is the easiness of obtaining a good description of the network traffic (degree of self-similarity, traffic intensity). The disadvantage of these solutions is their inability to use the traditional and well-known queueing models and modeling techniques for computer network analysis.

Markov models were introduced by A.A. Markov in 1907. They have been used in efficiency analysis tasks since the 1950s. Due to the advances made in the field of numerical methods, they have gained popularity as an analytical tool. The behaviour of a physical system can be described as a set of possible states, which are the possible actions or events that happen in the system, and the transitions between these states. If the next system's state depends only on the current one (i.e., it is memoryless), the examined system can be represented using a Markov process, which is one of the stochastic processes. For a discrete number of states, a Markov process is called a Markov chain. The states of a Markov chain can be defined using the parameters describing the system. The transitions between the states are represented by the transition matrix. They are defined using transition intensities in the case of a CTMC and transition probabilities in the case of a DTMC.

Over the years, many different software tools have been developed to deal with the Markov models; e.g., OPEDo [25], PARSECS [26], WinPEPSY-QNS [27], PRISM [28], GTAexpress [29] or our own Olymp2 [4]. Olymp2 uses the Java language to define the network nodes and the interactions between them. Due to the potentially very large sizes of the models' transition matrices, their generation is parallelized, and they can be compressed in real-time using a dedicated compression method based on a finite-state automata. Markov based models are currently applied to analyze and evaluate self-similar network traffic [30–34]. We are currently able to generate and solve Markov chains with 150 million states. The solution uses one of the projection methods based on the Arnoldi process using Krylov subspaces to determine the approximate eigensolution. It is achieved using the orthogonal projection of a large matrix onto a small Krylov subspace—see [35]—then, the computations using the uniformization method and Padé approximations [36] are made much easier. This approach is supplemented by a direct numerical solution of large systems of ordinary differential equations (ODE) using uniformization—i.e., discretization—which replaces a continuous Markov chain with a discrete-time Markov chain and a Poisson process. We are also increasing the size of tractable Markov chains by several orders through the use of graphical processing units and a careful design of computational algorithms for parallel computing and the optimization of memory usage. We make

these remarks to underline that we are well prepared to integrate Markov self-similar traffic models with known queue and server models and obtain quantitative results which are important in the engineering use of these models.

Below, we briefly describe two Markovian models—the special semi-Markov process (SSMP) and Markov modulated Poisson process (MMPP)—and a non-Markovian model—Poisson Pareto burst process (PPBP). An SSMP may be a part of a DTMC model, an MMPP may be used in a CTMC model, and a PPBP is frequently used in network traffic simulations.

### 2.1. Special Semi-Markov Process (SSMP)

The SSMP process was first introduced as a new type of a stochastic process by Lévy and Smith [37]. These processes are a generalization of the discrete and continuous Markov processes with a countable number of states. Here, we follow a proposition of a discrete time process which has only two values, 0 and 1, established at the beginning of each time-slot and determined by a discrete time Markov chain [38] (modulator) with  $n$  states. If it currently is in state 1, the value of the process is 1, and it is 0 for any other state of the modulator. State 1 means an arrival in this time slot and state 0 means that there is no arrival. The transition matrix  $\mathbf{A} = [p_{ij}]$ , where  $p_{ij}$  is the probability of going from state  $i$  to state  $j$  of the modulator, should be chosen to match the process to the arrival pattern observed in real traffic. To simplify this task, in [38], the authors suggest representing  $p_{ij}$  by only three parameters,  $a$ ,  $q$  (auxiliary parameters that can be determined based on the observed data) and  $n$  (the number of states in a Markov chain), in the following way:

$$\mathbf{A} = \begin{bmatrix} 1 - 1/a - 1/a^2 - \dots - 1/a^{n-1} & 1/a & 1/a^2 & \dots & 1/a^{n-1} \\ ineq/a & 1 - q/a & 0 & \dots & 0 \\ ine(q/a)^2 & 0 & 1 - (q/a)^2 & \dots & 0 \\ ine \dots & \dots & \dots & \dots & \dots \\ ine(q/a)^{n-1} & 0 & 0 & \dots & 1 - (q/a)^{n-1} \end{bmatrix}$$

The expected value of such a flow is

$$E[X] = \frac{1 - (1/q)}{1 - (1/q)^n} \quad (8)$$

and from this, we can determine the value of  $q$ . With the increase in the number of states in the Markov chain, we should also observe an increase in the number of time-scales over which the process exhibits a self-similar behavior as  $\lim_{n \rightarrow \infty} E[X] = 1 - (1/q)$ ,  $1/q < 1$ . The parameter  $a$  is determined by the Hurst parameter value as the slope of the plot  $\log_{10} \frac{\text{var}(X^{(m)})}{\sigma_2^2}$  versus  $\log_{10}(m)$ . We find this value using an iterative algorithm; e.g., the Newton–Raphson equation.

This model may be used together with any discrete time Markov server model; see [39].

### 2.2. Markov Modulated Poisson Process (MMPP)

The Markov modulated Poisson process is a Poisson process in which the intensity of arrivals is defined by the state of a modulator. In the simplest case, the modulator has two states, and if there is no traffic in one of these states, which means that  $\lambda = 0$ , we can refer to an ON–OFF process. One of the properties of the Markov modulated Poisson process is that a superposition of MMPPs is also an MMPP. A sum of  $d$  ON–OFF processes, as proposed in [40], may be used to represent a self-similar process. The obtained MMPP is a process with  $2^d$  states. This model consists of  $d$  ON–OFF processes, and each of them may be parametrized by two square matrices:

$$\mathbf{D}_0^i = \begin{bmatrix} -(c_{1i} + \lambda_{1i}) & c_{1i} \\ c_{2i} & -(c_{2i} + \lambda_{2i}) \end{bmatrix}, \quad \mathbf{D}_1^i = \begin{bmatrix} \lambda_{1i} & 0 \\ 0 & \lambda_{2i} \end{bmatrix}, \quad i = 1, \dots, d.$$

The element  $c_{1i}$  is the transition rate (if multiplied by  $dt$ , it becomes the transition probability) from the ON state to OFF of the  $i$ -th source, and  $c_{2i}$  is the rate from the OFF state to ON.  $\lambda_{1i}$  is the traffic rate when the  $i$ -th source is ON. In the OFF state, a source is inactive. The sum of  $\mathbf{D}_0^i$  and  $\mathbf{D}_1^i$  is an

irreducible infinitesimal generator  $\mathbf{Q}^i$  with the stationary probability vector  $\pi_i$ , which can be defined as follows:

$$\pi_i = \left( \frac{c_{2i}}{c_{1i} + c_{2i}}, \frac{c_{1i}}{c_{1i} + c_{2i}} \right)$$

To determine the parameter matrices  $\mathbf{D}_0$  and  $\mathbf{D}_1$  of an MMPP obtained as a superposition of  $d$  MMPPs with two states, it is possible to use the Kronecker sum of those  $d$  two-state MMPPs [41]:

$$(\mathbf{D}_0, \mathbf{D}_1) = \left( \oplus_{i=1}^d \mathbf{D}_0^i, \oplus_{i=1}^d \mathbf{D}_1^i \right)$$

To fit the examined model to the specific data, two approaches can be used: moment-based approach and likelihood-based approach [42]. It has been presented in [40] that it is possible for the fitting procedure for a superposition of two-state MMPPs to be based not only on moments, but also on a self-similar measure—the Hurst parameter.

The input parameters that are required in the fitting method can be described as follows [32]:

- $\lambda^*$ —the mean rate of the process to be modeled,
- $n$ —number of time scales,
- $d$ —number of active MMPPs,
- $H = 1 - \frac{\beta}{2}$ —the Hurst parameter,
- $\rho$ —lag 1 correlation.

The MMPP model may be integrated with any CTMC queueing model. We have successfully used this approach, for example, to model an IP router in which a system of four MMPPs reflected data published by CAIDA and which came from the link Equinix Chicago, collected during one hour on 18 February 2016, with 22,644,654 packets belonging to 1,174,515 IPv4 flows. The service time distribution at the server corresponded to the histogram of IP packet sizes and was represented by a hyper-Erlang distribution, with over 1400 phases in total [43].

### 2.3. Poisson Pareto Burst Process (PPBP)

The PPBP is a process in which the overlapping of multiple bursts can be observed. Burst arrivals follow the Poisson process with intensity  $\lambda_p$ , and the burst duration is defined by the Pareto (thus, heavy-tailed) distribution. The arrivals of customers during a burst are given by a constant bit rate, denoted as  $r$  [20]. PPBP was implemented, for example, in the broadly used simulation tool NS-3 [44].

Formally, the Poisson Pareto burst process is a continuous time stochastic process, for which  $t \in [0, T] \rightarrow B_t$ , where  $B_t$  is the number of active bursts in the data traffic at the time  $t$ . The active burst is a Poisson process of rate  $\lambda$ . The time of activity of each burst is independent and given by the Pareto distribution (Pareto( $\alpha, \beta$ )). The function of Pareto( $\alpha, \beta$ ) cumulative distribution is given by

$$F(x) = \begin{cases} 1 - \left(\frac{\beta}{x}\right)^\alpha & \text{for } x \geq \beta \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

For  $\alpha \in (1, 2)$ , the random variable representing the duration of a single burst  $D$  has infinite variance and a finite expectation, defined as follows:

$$E(D) = \frac{\alpha\beta}{\alpha - 1} \quad (10)$$



At time  $t = 0$ , the number of active bursts is a Poisson random variable of parameter  $\Lambda = \lambda E(D)$ . The periods of activity of these initial bursts are independent and identically distributed random variables. The cumulative distribution of these bursts can be defined by

$$G(x) = \begin{cases} 1 - \frac{1}{\alpha} \left(\frac{\beta}{x}\right)^\alpha - \alpha & \text{for } x \geq \beta \\ 1 - \frac{\alpha-1}{\alpha} \left(1 - \frac{x}{\beta}\right) - \frac{1}{\alpha} & \text{for } 0 < x < \beta \end{cases} \quad (11)$$

where  $\alpha, \beta$  are the parameters of the Pareto distribution.

The probability distribution with a cumulative distribution function defined as above is called the distribution of Pareto forward recurrence times. The mean and variance of this function is infinite [45]. The process exhibits LRD when its autocorrelation persists over all times scales. The PPBP process exhibits long range dependence (LRD) as long as the Pareto distributed burst durations have infinite variance.

The generation of the Pareto burst process requires the following parameters to be set:

- The traffic time duration  $t$ ,
- The intensity of the Poisson burst  $\lambda$ ,
- The mean intensity of bursts  $E(D)$ ,
- The Hurst parameter  $H \in (0.5, 1)$ .

The translation into  $\text{Pareto}(\alpha, \beta)$  distribution can be achieved as follows: from  $H$ , we get  $\alpha = 3 - 2H$ , and from  $E(D)$ , we obtain the parameter  $\beta$ .

### 3. Experimental Results

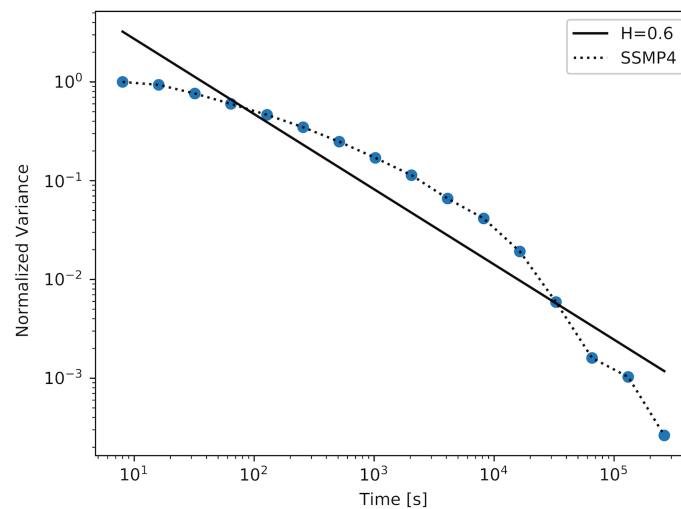
The traffic models presented in Section 2 were programmed using Python and used in its discrete event simulation module SimPy [46]. We have used the aggregated variance method, briefly described in Section 1, to determine the value of the Hurst parameter for the generated samples. We have plotted the values of the normalized variance versus the size of the time slots used in the aggregation process in the log-log scale (see Equation (4)). The relationship between the slope  $-\beta$  of the plot and the Hurst parameter  $H$  is presented in Equation (3). Note that for a steeper slope of the plot, the corresponding Hurst parameter takes smaller values. With the increase in the value of the slope, the value of the Hurst parameter decreases. The slope of the plot can slightly vary within a few first time-scales. This means that the value of the Hurst parameter within this range is non-identical for different time-scales. On the other hand, the visible decrease of the slope after the initial time-scales means that a process is no longer able to maintain self-similar properties. Such processes are self-similar only in a finite number of time-scales. We have fitted a straight-line for each of the plots using the least squares technique. At this point, it is necessary to mark that the values of the vertical axis differ between the figures. As a result, it is possible to observe that the slope of the straight-line fits differs between the figures. For every plot, we have computed the average and the maximum distance between a straight-line fit and the actual plot. The results of these computations have been presented in Table 1.

**Table 1.** The comparison of distances between calculated values, and the fitted values provided by the method of least squares. SSMP: special semi-Markov process; MMPP: modulated Markov Poisson process; PPBP: Poisson Pareto burst process.

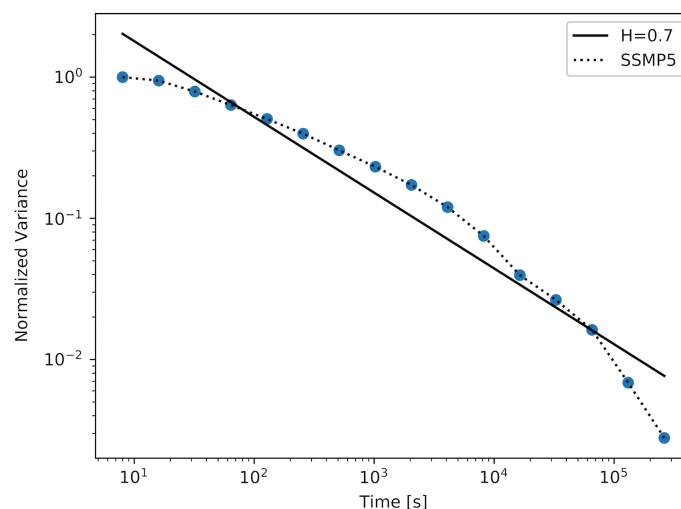
Method	Max Distance	Avg Distance
SSMP4	1.0722	0.5264
SSMP5	0.8912	0.3974
SSMP6	0.7088	0.2664
MMPP5	0.3999	0.1087
MMPP22	0.3257	0.0991
PPBP	0.1790	0.0542

For all three models, we wanted to produce traffic with at least  $H = 0.8$ . It was not possible for every Markov model to achieve a desired value of the Hurst parameter. To achieve such samples, it was necessary to repeat the experiment with modified parameters of the examined model. In the case of the SSMP model, we changed the number  $n$  of modulator states, and for MMPP model, we changed the number  $d$  of used active ON-OFF sources.

Figures 1–3 show the results achieved using the SSMP model with a different number of states in a modulator ( $n$ ). For the model with  $n = 4$ , which is the simplest model of its kind, the achieved approximated value of the Hurst parameter was 0.6. The plot for this experiment is shown in Figure 1. After increasing the number of states  $n$ , it was possible to achieve a sample with a higher value of the Hurst parameter, which reached the value 0.7. The corresponding plot is shown in Figure 2. To achieve a sample with the desired value of the Hurst parameter, which was 0.8, it was necessary to use an SSMP model with  $n = 6$ . As we can observe in Table 1, the maximum and average distance between the actual plot and the straight-line fit are decreasing with the increase in the number of states in the modulator. Returning to the three figures described above, we can observe that the number of time-scales for which the considered plots can have persistent linearity (and as a result, a stable value of the Hurst parameter) increases with the number of states in the modulator.

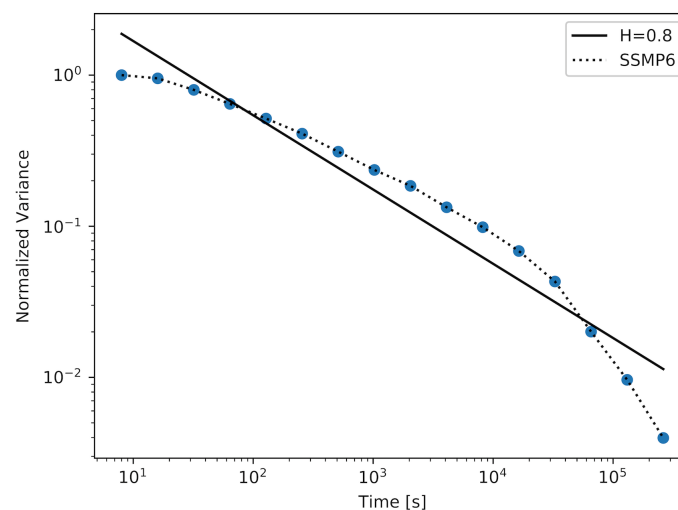


**Figure 1.** The aggregate variance of the sample in several time scales with the SSMP model,  $n = 4$ .



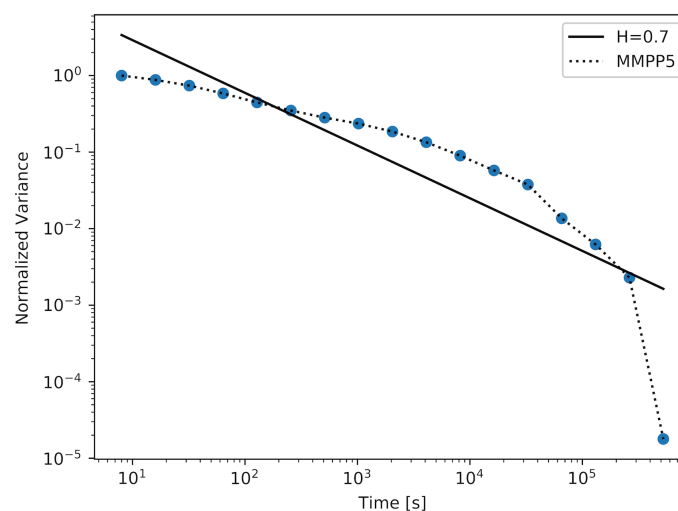
**Figure 2.** The aggregate variance of the sample in several time scales with the SSMP model,  $n = 5$ .





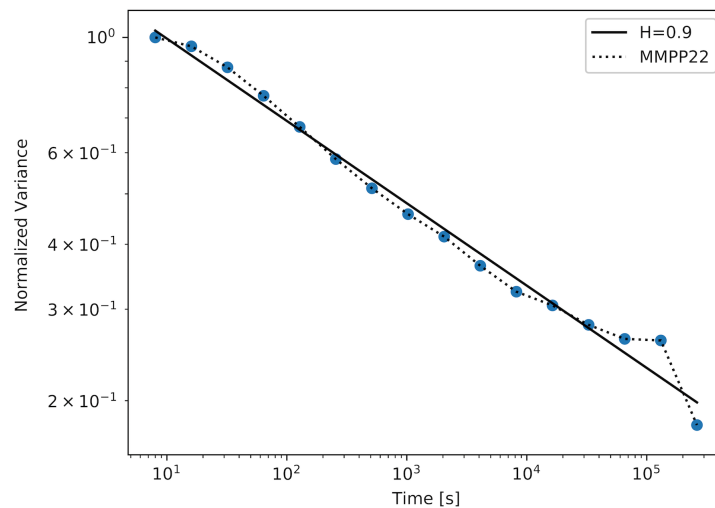
**Figure 3.** The aggregate variance of the sample in several time scales with the SSMP model,  $n = 6$ .

Figures 4 and 5 show the results achieved using the MMPP model with two different numbers of active ON–OFF sources ( $d$ ). For the MMPP with  $d = 5$ , we were able to achieve a sample with  $H = 0.7$ . The corresponding plot is shown in Figure 4. It was necessary to increase the number of active MMPP sources to achieve samples with higher values of the Hurst parameter. In Figure 5, we show the plot for the sample with  $H = 0.9$ , which is a very high value of the Hurst parameter. We can observe in Table 1 that, for a more complex MMPP model, the values of the average and the maximum distance between the straight-line fit and the actual plot decrease, which means that the approximation reflects the actual plot more closely.

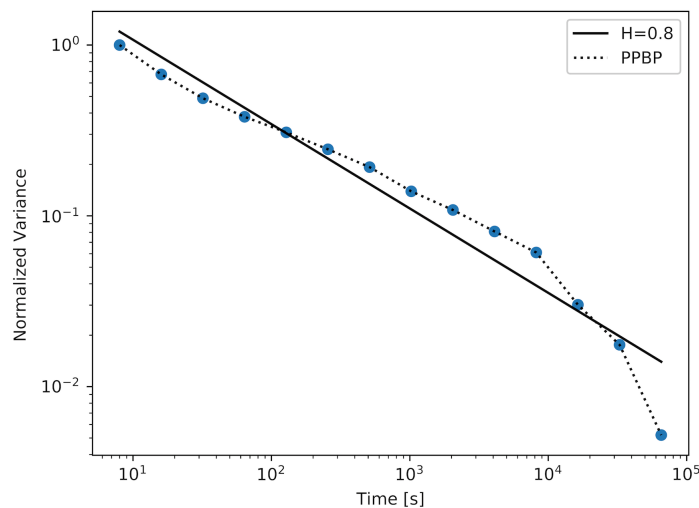


**Figure 4.** The aggregate variance of the sample in several time scales with the MMPP model,  $d = 5$ .

In Figure 6, we present a plot for a non-Markovian model—PPBP. This is a model which is frequently used in network traffic analysis. The PPBP process consists of a simultaneous generation of packets by a larger number of Poisson sources. This number was changed during the experiment. Using this model, we obtained a sample with  $H = 0.8$ . We can observe that the generated sample displays self-similarity over most of the measured time scales. The maximum and average distances between the actual plot and the straight-line fit for this model (presented in Table 1) are the smallest.



**Figure 5.** The aggregate variance of the sample in several time scales with the MMPP model,  $d = 22$ .



**Figure 6.** The aggregate variance of the sample in several time scales with the PPBP model.

The sample with the highest value of the Hurst parameter was achieved using a model with 22 active MMPPs. It is important to note that by generating this sample, the Markovian model has outperformed the PPBP model in terms of achieving the sample with the highest value of the Hurst parameter. We can also compare the plots achieved for these models (Figures 5 and 6). The plot created for the MMPP model and its straight-line fit almost perfectly overlap for the majority of time scales, which means that the generated sample maintains a constant value of the Hurst parameter over these time scales. For the PPBP model, this overlapping is also visible.

All processes are not perfect self similar-processes but may imitate their features over the range of several scales, and this is sufficient for the needs of network traffic representation. In Table 2, one can find a comparison of the time required to generate 1 million packets for the different examined models. With the increase in the complexity of the Markovian models (parameters  $d$  and  $n$ ), we can observe an increase in the generation time. However, the time required to generate a sample for all the Markovian models is one order of magnitude shorter than for the non-Markovian model.

**Table 2.** Comparison of time needed to generate 1 million packets using the different methods.

Method	Avg Time [ms]
SSMP4	12.1257
SSMP5	12.8312
SSMP6	13.5641
MMPP5	18.1601
MMPP22	32.6147
PPBP	197.1943

Note that the results of all three models were obtained here using the simulation. PPBPs can only be used in simulation models, unlike SSMPs and MMPPs, which may also be integrated with Markov queueing models. The usage of the Markov chains becomes intractable because of the number of states, which is rapidly increasing with the complexity of the modeled object. The introduction of self-similar sources increases the number of states, but we can cope with this issue by using the numerical methods described in Section 2 for the larger models. Increasing the efficiency of the hardware also alleviates the problem.

#### 4. Summary

The accuracy in creating Internet traffic models is crucial in the design and evaluation of secure and high-performance networks. It is necessary to understand the nature of Internet traffic to successfully optimize its performance and efficiency in the face of growing demands for fast information transfer. This can result in better job scheduling and resource management, ensuring the quality of transmission and preventing system malfunctions.

The network traffic exhibits self-similarity, which has a significant impact on network performance. The most well-known self-similar traffic models are not based on the Markov approach. Nevertheless, Markov models are traditional tools in the analytical evaluation of computer networks, and they should be adapted to the case of self-similar traffic.

A few Markov models can be used to create self-similar traffic sources. We have described two of them: the SSMP model using discrete-time Markov chains and the MMPP model using continuous-time Markov chains. Both of them enable us to obtain traffic that is self-similar in a finite number of time-scales. It has been shown that it is necessary to increase the number of states in a corresponding Markov chain (i.e., increase the size  $n$  of the modulator in the case of SSMP or the number  $d$  of sources in an MMPP) to produce traffic with higher values of the Hurst parameter. In this way, we also increase the number of time-scales for which the examined traffic may sustain a relatively stable value of the Hurst parameter and consequently self-similar behavior. These Markov models may also be used in simulations. We also compare these models with a non-Markovian PPBP model, which is widely used in simulations. A more complex MMPP model (with  $d = 22$ ) can successfully “compete” with a PPBP, not only in terms of the quality of provided self-similar process but also in terms of the efficiency of its generation: the generation time of a defined number of packets is several times shorter in case of an MMPP.

The great advantage of the Markov models is that they may be used in the modeling of transient states, which is important in the evaluation of computer networks as transmitted flows are continually changing due to the variability of transmissions and due to the performance of the flow control algorithms. The evaluation of transient states by simulation is extremely costly as it requires multiple repetitions. The biggest problem of the Markov approach is the large number of Markov chain states and consequently the high computational complexity. In the past, it was impossible to implement the more complex Markov models which are necessary to mimic real traffic data accurately. Due to the advances made in computing technology, complexity is no longer a problem, and Markov models can be successfully used in optimization tasks.

**Author Contributions:** Investigation, A.D. and J.D.; Methodology, A.D. and T.C.; Software, A.D.; Validation, J.D.; Writing—original draft preparation, K.F., J.S. and T.C.; Writing—review and editing, A.D., J.D., K.F., J.S. and T.C.; Visualization, J.D.; Project administration, J.S.; Formal analysis, J.D. and K.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. OMNET++ Community Site. Available online: <http://www.omnetpp.org/> (accessed on 4 May 2020).
2. ns-3 Network Simulator. Available online: <https://www.nsnam.org/> (accessed on 4 May 2020).
3. Asmussen, S.; Nerman, O.; Olsson, M. Fitting Phase-Type Distributions via the EM Algorithm. *Scand. J. Stat.* **1990**, *23*, 419–441.
4. OLYMP2. Available online: <https://sites.google.com/site/zmioesk/software/olymp-2> (accessed on 10 April 2020).
5. Kaur, G.; Saxena, V.; Gupta, J. Detection of TCP Targeted High Bandwidth Attacks Using Self-Similarity. *J. King Saud Univ. Comput. Inf. Sci.* **2017**, *32*, 35–49.
6. Deka, R.; Bhattacharyya, D.K. Self-similarity based DDoS attack detection using Hurst parameter. *Secur. Commun. Networks* **2016**, *9*, 4468–4481. [[CrossRef](#)]
7. Park, C.; Hernandez-Campos, F.; Le, L.; Marron, J.; Park, J.; Pipiras, V.; Smith, F.; Smith, R.; Trovero, M.; Zhu, Z. Long range dependence analysis of Internet traffic. *J. Appl. Stat.* **2011**, *38*, 1407–1433. [[CrossRef](#)]
8. Pramanik, S.; Datta, R.; Chatterjee, P. Self-Similarity of Data Traffic in a Delay Tolerant Network. In *IEEE Wireless Days*; IEEE: Piscataway, NJ, USA, 2017; pp. 39–42.
9. Xu, Y.; Li, Q.; Meng, S. Self-similarity Analysis and Application of Network Traffic. *Mob. Comput. Appl. Serv. Springer Int. Publ.* **2019**, 112–125. [[CrossRef](#)]
10. Kim, Y.; Min, P. On the prediction of average queueing delay with self-similar traffic. In Proceedings of the GLOBECOM '03. IEEE Global Telecommunications Conference (IEEE Cat. No.03CH37489), San Francisco, CA, USA, 1–5 December 2003; Volume 5, pp. 2987–2991.
11. Gorrasi, A.; Restino, R. Experimental comparison of some scheduling disciplines fed by self-similar traffic. *Proc. IEEE Int. Conf. Comm.* **2003**, *1*, 163–167.
12. Stallings, W. *High-Speed Networks: TCP/IP and ATM Design Principles*; Prentice-Hall: New York, NY, USA, 1998.
13. Muscariello, L.; Mellia, M.; Meo, M.; Ajmone Marsan, M.; Lo Cigno, R. Markov Models of Internet Traffic and a New Hierarchical MMPP Model. *Comput. Commun.* **2005**, *28*, 1835–1851.
14. Mandelbrot, B. How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension. *Science* **1967**, *156*, 636–638. [[CrossRef](#)]
15. Beran, J. *Statistics for Long-Memory Processes*; CRC Press: Boca Raton, FL, USA, 1994; Volume 61.
16. Suchacka, G.; Domański, A. Investigating Long-Range Dependence in E-Commerce Web Traffic. In Proceedings of the International Conference on Computer Networks, Brunow, Poland, 14–17 June 2016; Springer: Cham, Switzerland, 2016; pp. 42–51. [[CrossRef](#)]
17. Czachórski, T.; Domańska, J.; Pagano, M. On stochastic models of internet traffic. In Proceedings of the International Conference on Information Technologies and Mathematical Modelling, Anzhero-Sudzhensk, Russia, 18–22 November 2015; Springer: Cham, Switzerland, 2015; pp. 289–303.
18. Domańska, J.; Domański, A.; Czachórski, T. Estimating the Intensity of Long-Range Dependence in Real and Synthetic Traffic Traces. *Commun. Comput. Inf. Sci.* **2015**, *522*, 11–22.
19. Mandelbrot, B.; Ness, J. Fractional Brownian Motions, Fractional Noises and Applications. *SIAM Rev.* **1968**, *10*, 422–437. [[CrossRef](#)]
20. Addie, R.G.; Neame, T.D.; Zukerman, M. Performance evaluation of a queue fed by a Poisson Pareto burst process. *Comput. Netw.* **2002**, *40*, 377–397.
21. Gallardo, J.R.; Makrakis, D.; Orozco-Barbosa, L. Use of  $\alpha$ -stable Self-similar Stochastic Processes for Modeling Traffic in Broadband Networks. *Perform. Eval.* **2000**, *40*, 71–98. [[CrossRef](#)]
22. Erramilli, A.; Singh, R.; Pruthi, P. An application of deterministic chaotic maps to model packet traffic. *Queueing Syst.* **1995**, *20*, 171–206. [[CrossRef](#)]

23. Harmantzis, F.C.; Hatzinakos, D. Heavy network traffic modeling and simulation using stable FARIMA processes. In Proceedings of the 19th International Teletraffic Congress, Beijing, China, 29 August–2 September 2005; pp. 300–303.
24. Laskin, N.; Lambadatis, I.; Harmantzis, F.; Devetsikiotis, M. Fractional Levy Motion and its Application to network traffic modeling. *Comput. Netw.* **2002**, *40*, 363–375. [\[CrossRef\]](#)
25. Buchholz, P.; Müller, D.; Kemper, P.; Thümmler, A. OPEDo: A tool framework for modeling and optimization of stochastic models. In Proceedings of the 1st International Conference on Performance Evaluation Methodologies and Tools, Pisa, Italy, 11–13 October 2006; Volume 180, p. 61.
26. Bell, A.; Haverkort, B. Distributed disk-based algorithms for model checking very large Markov chains. *Form. Methods Syst. Des.* **2006**, *29*, 177–196.
27. Bazan, P.; Bolch, G.; German, R. WinPEPSY-QNS - Performance evaluation and prediction system for queueing networks. In Proceedings of the 13th GI/ITG Conference—Measuring, Modelling and Evaluation of Computer and Communication Systems, Nürnberg, Germany, 27–29 March 2006; pp. 1–4.
28. PRISM Homepage. Available online: [www.prismmodelchecker.org](http://www.prismmodelchecker.org) (accessed on 4 May 2020).
29. Czekster, R.; Fernandes, P.; Webber, T. GTAexpress: A Software Package to Handle Kronecker Descriptors. In Proceedings of the QEST—6th International Conference on the Quantitative Evaluation of Systems, Budapest, Hungary, 13–16 September 2009; pp. 281–282.
30. Chen, Z.; Wen, J.; Geng, Y. Predicting future traffic using Hidden Markov Models. In Proceedings of the IEEE 24th International Conference on Network Protocols (ICNP), Singapore, 8–11 November 2016; pp. 1–6.
31. Domańska, J.; Domański, A.; Czachórski, T. *Internet Traffic Source Based on Hidden Markov Model*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6869, pp. 395–404.
32. Domańska, J.; Domański, A.; Czachórski, T. Modeling packet traffic with the use of superpositions of two-state MMPPs. In *Communications in Computer and Information Science*; Springer: Cham, Switzerland, 2014; p. 431.
33. Gudimalla, R.; Perati, M. Loss behavior of Internet router with priority based self-similar synchronous traffic-multi server queueing system with Markovian input. *Opsearch* **2017**, *54*, 283–305.
34. Nogueira, A.; Salvador, P.; Valadas, R.; Pacheco, A. Markovian Modelling of Internet Traffic. *Lect. Notes Comput. Sci.* **2011**, *5233*, 98–124.
35. Stewart, W.J. *An Introduction to the Numerical Solution of Markov Chains*; Princeton University Press: Princeton, NJ, USA, 1994.
36. Saad, Y. *Projection Methods for Solving Large Sparse Eigenvalue Problems*; Number 973 in Lecture Notes in Math; Springer: Berlin, Germany, 1982.
37. Smith, W.L. Regenerative stochastic processes. *Proc. R. Soc. Ser. A* **1995**, *232*, 6–31.
38. Robert, S. Basic estimation of Markovian Pseudo Long-Range Dependent Processes. In Proceedings of the IEEE 14th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Pisa, Italy, 12 June 2009.
39. Bruneel, H.; Kim, B.G. *Discrete-Time Models for Communications Systems, Including ATM*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1993.
40. Andersen, A.T.; Nielsen, B.F. A Markovian Approach for Modeling Packet Traffic with Long-range Dependence. *IEEE J. Sel. A. Commun.* **2006**, *16*, 719–732. [\[CrossRef\]](#)
41. Fischer, W.; Meier-Hellstern, K. The Markov-modulated Poisson Process (MMPP) Cookbook. *Perform. Eval.* **1993**, *18*, 149–171. [\[CrossRef\]](#)
42. Okamura, H.; Kamahara, Y.; Dohi, T. Estimating Markov-modulated Compound Poisson Processes. In Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools, ValueTools '07, Nantes, France, 23–25 October 2007; pp. 28:1–28:8.
43. Domańska, J.; Domański, A.; Czachórski, T.; Klamka, J. A study of IP router queues with the use of Markov models. In Proceedings of the 24th International Conference on Computer Networks, CN 2017, Łódź, Poland, 20–23 June 2017.
44. Ammar, D.; Begin, T.; Guerin-Lassous, I. A new tool for generating realistic Internet traffic in NS-3. In Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques, Barcelona, Spain, 21–25 March 2011; pp. 81–83.

45. Zukerman, M.; Neame, T.D.; Addie, R.G. *Performance Evaluation of a Queue Fed by a Poisson Pareto Burst Process*; Elsevier: Amsterdam, The Netherlands, 2002; Volume 40, pp. 377–397.
46. Discrete Event Simulation for Python. Available online: <https://simpy.readthedocs.io/en/latest> (accessed on 4 May 2020).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).