Matching Multi-Fractal Process Parameters Against Real Data Traffic

Patrik Carlsson[1], Markus Fiedler[2] and Arne A. Nilsson[3]

[1]Blekinge Institute of Technology, 371 79 Karlskrona, Sweden. E-mail: Patrik.Carlsson@bth.se
[2]Blekinge Institute of Technology, 371 79 Karlskrona, Sweden. E-mail: Markus.Fiedler@bth.se
[3]Blekinge Institute of Technology, 371 79 Karlskrona, Sweden. E-mail: Arne.Nilsson@bth.se

Abstract

Recent analyses of real data/internet traffic indicate that data traffic exhibits long-range dependence as well as self-similar or multi-fractal properties. By using mathematical models of Internet traffic that share these properties we can perform analytical studies of network traffic. This gives us an opportunity to analyse potential bottlenecks and estimate delays in the networks.

Processes with multi-fractal properties can be modeled by multiplying the output of Markov Modulated Rate Processes (MMRP) [1] each defined by four parameters. The MMRP are easily used in stochastic fluid flow modeling. This model is also suited for analysis of other traffic types e.g. VoIP and thus, it allows for integration of different traffic types, i.e. time-sensitive voice traffic with best-effort data traffic. Using this model we can calculate performance parameters for each individual stream that enters the system/model.

In this paper we show how to construct a multi-fractal process that is matched to measured data from MMRP sub processes.

Keywords: Multi-fractal traffic, fluid flow model, numerical analysis, performance evaluation

1 Introduction

The reason for this research is that we want to develop a mathematical model for Internet traffic, that can be used in a fluid flow analysis. In earlier papers [1] and [2] we demonstrated that the fluid flow model is a usable tool for analytical performance studies. We have used data processes that had multi-fractal properties (*MFP*), and we also expressed desires to do similar studies based on processes that have been modeled based on measured network traffic. In this paper we show how a certain class of multi-fractal processes can be modeled such that they match up well against real network traffic. The tool we now have permits us to do better capacity management instead of today's *rule-of-thumb*.

Since we are to match a mathematical model against measured data, we will start by describing the mathematical model and its basic properties. Following that we will verify this model against simulations that have been performed. The actual matching of the parameters based on real measured traffic will then be described, and finally some conclusions will be drawn.

2 Processes description

The mathematical model that is used to generate a process with multi-fractal properties is formed by multiplying the output of independent 2-state Markov modulated rate processes (called sub-processes). The process will be formed as

$$R(t) = \prod_{i=0}^{n-1} R_i(t) \qquad (1)$$

Here $R_i(t)$ is the output rate from sub-process i at time t (data-unit/time-unit). The 2-state Markov-modulated rate process is depicted in Figure 1. This is a well known and often used model. The model was originally suggested in [4]

2.1 Description of a Sub-Process

Each sub process is a simple low-high process, as described in Figure 1. The sub processes transition matrix and rate matrix are formed as

$$\mathbf{M}_i = \begin{bmatrix} -\lambda_i & \lambda_i \\ \mu_i & -\mu_i \end{bmatrix} \qquad \mathbf{R}_i = \begin{bmatrix} l_i & 0 \\ 0 & h_i \end{bmatrix} \qquad (2)$$

The first two moments of the MFP rate process is going to be used for the matching of processes to MFP. Thus it is interesting to have analytical expressions for the two first moments.

$$\mathbf{E}[R_i(t)] = \frac{l_i\mu_i + h_i\lambda_i}{\lambda_i + \mu_i} \qquad (3)$$

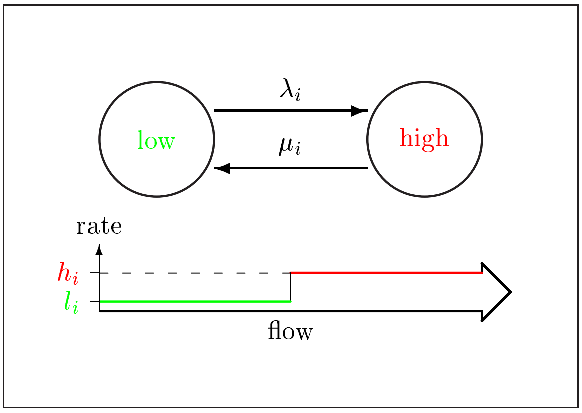

Figure 1: 2-state Markov Modulated Rate Process, the i:th a sub-process

$$\mathbf{E}\left[R_i^2(t)\right] = \frac{2\lambda_i\mu_i(l_i - h_i)^2}{t^2(\lambda_i+\mu_i)^3}\left(t - \frac{1}{\lambda_i+\mu_i} + \frac{e^{-(\lambda_i+\mu_i)t}}{\lambda_i+\mu_i}\right) + \frac{(l_i\mu_i + h_i\lambda_i)^2}{(\lambda_i+\mu_i)^2} \tag{4}$$

A more detailed derivation can be found in [3]. It is obvious that the equation for the first moment is not affected by any changes in t. The second-moment however

$$\lim_{t\to 0} \mathbf{E}\left[R_i^2(t)\right] = \frac{\lambda_i\mu_i(l_i-h_i)^2}{(\lambda_i+\mu_i)^2} + \mathbf{E}\left[R_i^1(t)\right]^2 \tag{5}$$

$$\lim_{t\to \infty} \mathbf{E}\left[R_i^2(t)\right] = \frac{(l_i\mu_i+h_i\lambda_i)^2}{(\lambda_i+\mu_i)^2} = \mathbf{E}\left[R_i^1(t)\right]^2 \tag{6}$$

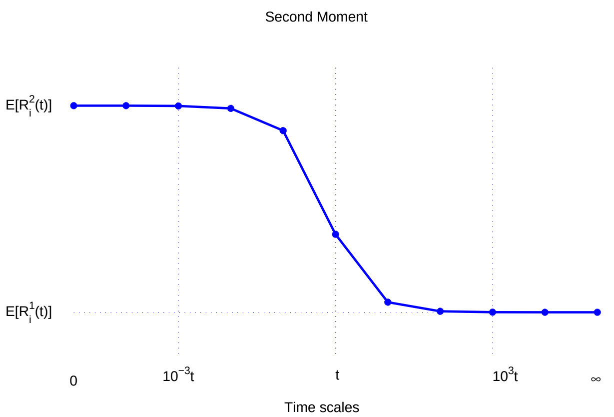

Figure 2: Typical shape of the second moment of a subprocess. $\lambda_i = \mu_i$ and $\mathbf{E}[R_i(t)] = 1$

2.2 Process

As mentioned before the process is formed by multiplying the output from the sub-processes. The transition matrix is formed using Kronecker addition:

$$\mathbf{M}_{\mathrm{D}} = \mathbf{M}_0 \oplus \mathbf{M}_1 \oplus \cdots \oplus \mathbf{M}_{n-1} \tag{7}$$

and the rate matrix as

$$\mathbf{R}_{\mathrm{D}} = \mathbf{R}_0 \odot \mathbf{R}_1 \odot \cdots \odot \mathbf{R}_{n-1}. \tag{8}$$

Where

$$\mathbf{A} \oplus \mathbf{B} = \mathbf{A} \otimes \mathbf{I}_B + \mathbf{I}_A \otimes \mathbf{B} \tag{9}$$

$$\mathbf{A} \odot \mathbf{B} = \mathbf{A} \otimes \mathbf{I}_B \cdot \mathbf{I}_A \otimes \mathbf{B} \tag{10}$$

and $\mathbf{I}_x$ is a diagonal matrix with ones on it and the same size as matrix x, where $\otimes$ is defined as

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & \dots & a_{1,m} \\ \dots & \dots & \dots \\ a_{n,1} & \dots & a_{n,m} \end{bmatrix}_{n\times m}$$

$$\mathbf{B} = \begin{bmatrix} b_{1,1} & \dots & b_{1,q} \\ \dots & \dots & \dots \\ b_{n,1} & \dots & b_{p,q} \end{bmatrix}_{p\times q}$$

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & \dots & a_{1,m}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{n,1}\mathbf{B} & \dots & a_{n,m}\mathbf{B} \end{bmatrix}_{np\times mq} \tag{11}$$

The moments for the process are easily obtained simply by multiplication of the sub-process moments

$$\mathbf{E}[R(t)] = \prod_{i=0}^{n-1} \mathbf{E}\left[R_i^1(t)\right] \tag{12}$$

$$\mathbf{E}\left[R^2(t)\right] = \prod_{i=0}^{n-1} \mathbf{E}\left[R_i^2(t)\right] \tag{13}$$

3 Simulations

To compare the analytical formulas of the first and second moments, some simulations have been run. In Figure 3 a set of simulations are shown where we compare results from simulation with result from the analytical method. In Table 1 the process parameters are presented. In Figure 3 the first and the second moment are shown. Each row corresponds to a given set of parameters, starting with 1. The columns corresponds to the moments, column one is the first moment and column two the second moment.

Looking at the first row the match is perfect for both moments. However this process corresponds to a non varying process, with $l_i = h_i = 2$ hence no variations. Looking at row two and three, a slight difference is detected. It can probably be accredited to a too small sample space for the simulation. But the match is fairly good at least at higher time scales for the second moment.

4 Matching

Our developed matching method assumes that the first moment has been normalized in such a way that

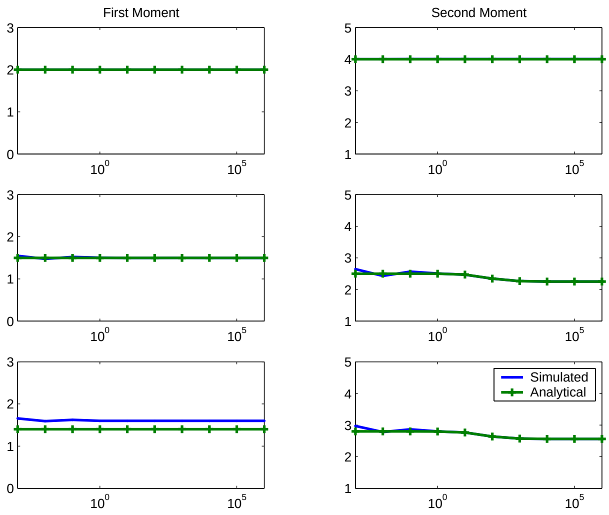

Figure 3: A comparison between moments that where calculated from simulations and moments based on the analytical formulas in Equation 3 and 4

ID	τ	λ	μ	l_i	h_i	Mean
1	100	0.02	0.02	2	2	2
2	100	0.02	0.02	1	2	1.5
3	83.333	0.03	0.02	1	2	1.4

Table 1: Parameters used in simulation

the mean of the first moments at different time scales will be 1.

When matching real-world data there are usually some trade-offs to be made. We have chosen to over-estimate the moments. The main reason for this is that it is better to work against a worst case scenario. Another consideration is the number of processes that are used to match the process, since in an analysis step to follow [1] there is a upper-bound on the processing power. That is if we use an infinite number of processes we would probably get a good match, but it would not be usable in future work.

When matching we are trying to match the first two moments. There are a number of methods that could be used to do this. In this paper we have chosen to use a simple and heuristic method. The method works roughly like this:

1. Initialize

$\vec{D}^2$	The second moment
$\vec{D}^1$	The first moment
$\vec{T}$	The time scales

where D_i^x denotes the i:th element in the vector corresponding to the x:th moment.

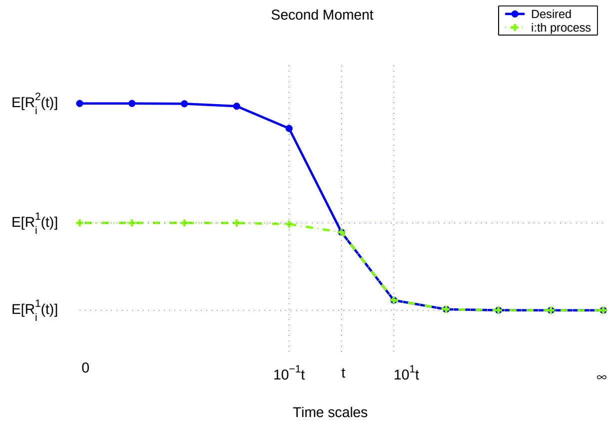

Figure 4: How sub-process i is identified from the desired signal

2. For each time scale in $\vec{T}$, try to find a sub-process that has properties that match it. As if the process were *constant* at time scales smaller than t. Cf. Figure 4.

3. Loop through $\vec{T}$ starting at the largest time scale and going down to the smallest, where i denotes the position of the current time scale in $\vec{T}$.

   (a) Set $t = T_i$, τ is estimated to be $10t$

   $$M^2 = D_i^2 \quad M^1 = D_i^1 \tag{14}$$

   These will be used as the desired value of the sub-process as $t \to 0$, since the process has been normalized in such a manner that the mean will be one and remembering that the limit value of a sub-process in equation 5. This value can be too large for one single sub-process. The solution is the usage of sub-sub-processes, processes that are identical and together achieve the desired properties on a given time scale.

   (b) Estimate how many sub-sub-processes that are needed to achieve this limit.

   $$k = \left\lceil \frac{\log M^2}{\log 2} \right\rceil \tag{15}$$

   (c) If $k < 0$ then this time scale is O.K. goto 3.

   (d) Determine the limit for a sub-sub-process

   $$O^2 = \sqrt[k]{M^2} \quad O^1 = \sqrt[k]{M^1} \tag{16}$$

   (e) Based on the symmetry from $\lambda = \mu = 2/\tau$ an expression for l and h can be obtained in the following manner:

$$\Delta L = 2\sqrt{O^2 - (O^1)^2} \quad (17)$$
$$l = M_i^1 - \frac{\Delta L}{2} \quad (18)$$
$$h = M_i^1 + \frac{\Delta L}{2} \quad (19)$$

(f) The parameters that identify sub-process i are known (the output rates in the two states, the transition rates and the number of sub-sub-processes that form the sub-process). Create two vectors $\vec{M}_i^2$ and $\vec{M}_i^1$ that span all time scales defined by $\vec{T}$ from these parameters. These vectors specify the moments for sub-process i. Store the parameters needed to create these vectors.

(g) Remove the influence on the desired process that this sub-process has by simply dividing it out, element-by-element. j is the size of the $\vec{T}$.

$$\vec{D}^1 = \left[\frac{D_1^1}{M_1^1}, \ \ldots, \ \frac{D_j^1}{M_j^1}\right] \quad (20)$$
$$\vec{D}^2 = \left[\frac{D_1^2}{M_1^2}, \ \ldots, \ \frac{D_j^2}{M_j^2}\right] \quad (21)$$

4. Now the matched process can be created by combining the sub-processes using element-by-element multiplication using Equation 3 and 4 from the stored parameters. Recreate the moment vectors for each sub-process call them $\vec{P}_i^x$ where i indicate that this is sub-process i and x indicate the x:th moment (one or two). The moments will then be

$$\vec{M}^1 = \begin{bmatrix} P_{1,1}^1 \times P_{2,1}^1 \ldots P_{j,1}^1, \\ \ldots, \\ P_{1,n-1}^1 \times P_{2,n-1}^1 \ldots P_{j,n-1}^1 \end{bmatrix} \quad (22)$$
$$\vec{M}^2 = \begin{bmatrix} P_{1,1}^2 \times P_{2,1}^2 \ldots P_{j,1}^2, \\ \ldots, \\ P_{1,n-1}^2 \times P_{2,n-1}^2 \ldots P_{j,n-1}^2 \end{bmatrix} \quad (23)$$

5. Repeat step 3 as many times as desired (could be based on maximal error or maximal number of processes).

Now it is possible to list the parameters that are needed to match the desired input. When comparing a matched process with a measured process we define the error as

$$\overrightarrow{Error} = \left[\left|\frac{D_1^2 - M_1^2}{D_1^2}\right|, \ \ldots, \ \left|\frac{D_j^2 - M_j^2}{D_j^2}\right|\right] \quad (24)$$

4.1 Example 1: Simulated data

In Figure 5 an example of the matching is shown. Here the desired signal is a simulated one, it was created by five sub-processes listed in Table 2. The first moment is a perfect match, the second moment is over estimated. Looking at the error the over estimation is not that large.

ID	λ	μ	l_i	h_i
1	512	512	0.5	1.5
2	128	128	0.5	1.5
3	32	32	0.5	1.5
4	8	8	0.5	1.5
5	2	2	0.5	1.5

Table 2: Parameters used to create the desired signal that is matched in Example 1.

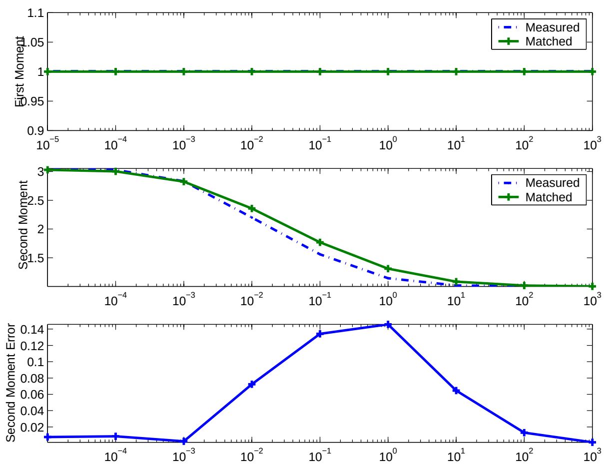

Figure 5: Comparison between a simulated process and a created process. Example 1.

4.2 Example 2: Uplink on small workgroup switch

The data used here came from a mid-size workgroup switch(Cisco 2948G). There were about 36 hosts connected to it, distributed into two categories: department computers and 16 computers accessible to students. The measurements were performed during the summer of 2001, from the 7:th to the 30:th of July, which resulted in more than 2×10^6 samples. The measurement focused on the interface that corresponded to the uplink of the switch. The data was collected using a simple SNMP (Simple Network Management Protocol) polling tool (developed during 2001-2002). The tool polled the device roughly once every second, with the odd delay (probably due to processing times in the device and intermediate devices). This jitter is of no greater importance since the values are recalculated and stored as rates, thus if the measure took too long time the time sample would account for it.

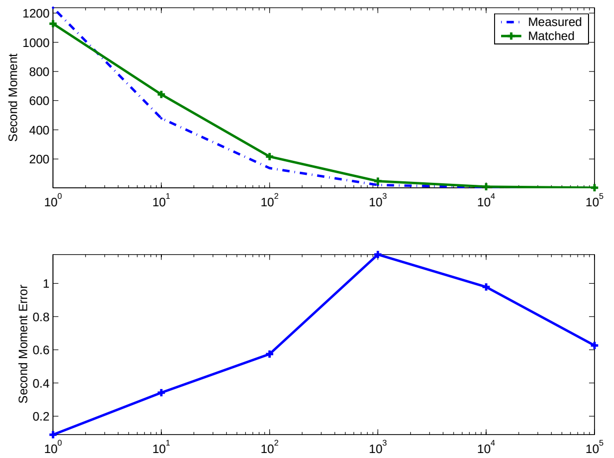

Figure 6: Comparison between a measured process and a created process. Example 2.

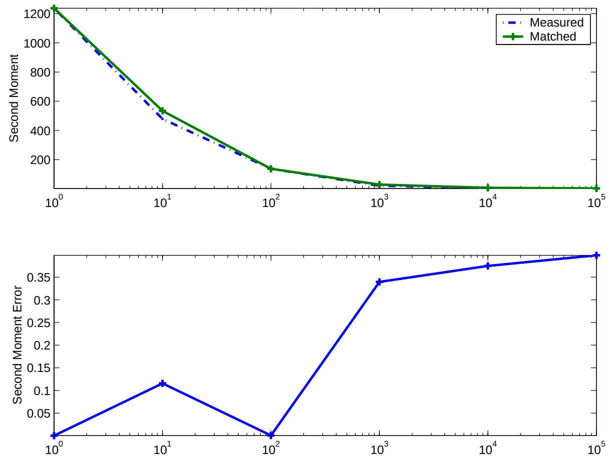

Figure 7: The resulting match after running the method 5 times, modifying the method

Looking at Figure 6 the measured process first moment behaves as expected, but the second moment seems to lack of a flat, or semi-flat section. One possible reason for this behavior could that the traffic load was low (measured during the summer), and most of the traffic that were present came from computers talking to computers in a (almost)deterministic manner, e.g. WINS, NFS, etc. This type of source generates traffic at discrete intervals, and the packet sizes are almost always the same.

Turning the focus toward the matched process a under-estimation is noticed in the first point. A solution to this problem, and perhaps a refinement of the method, was to reapply the method to the whole set of data that remained after the first cycle. By doing this a even better match was achieved in this point, but still under-estimated (the other points remained unchanged since the method found that the points already were over-estimated, $k < 0$). One way to over-estimate all-points would be to add a process with a time scale larger than the largest time scale and using it to shift the entire process upward. This would however increase the error at all time scales.

To see if it was possible to get even better matches the method was modified by setting $k = 1$ in Equation 15 and run it several times. The result was a dramatically improved match, the cost was an significant increase of processes.

5 Conclusions

In this paper we have presented a crude method that allows us to match process parameters against real network traffic. It is possible to get a better match using more processes, this, however, will cost in later analysis in processing time. We have found that it is possible to get a fairly good match using only a few processes. We need a measurement tool that allows us to measure at smaller time scales than one second. SNMP measurement seems not to be able to support this, in fact several measurements have showed that the MIB (Management Information Base) is not updated in a correct way, but rather in 10 seconds intervals (this behavior was most notably noticed in a well-known operating system).

References

[1] P. Carlsson and M. Fiedler. Multifractal products of stochastic processes: Fluid flow analysis. Proceedings of 15th Nordic Teletraffic Seminar (NTS-15), Lund, Aug. 22-24, 2000.

[2] P. Carlsson, M. Fiedler and A. Nilsson. Voice and multifractal data in the Internet. Proceedings of 26th IEEE Conference on Local Computer Networks (LCN2001), Tampa, Nov. 15-17, 2001.

[3] P. Carlsson, M. Fiedler and A. Nilsson. Moment analysis of sub-processes of multi-fractal sub-processes, Internal Report, 2002.

[4] P. Mannersalo, I. Norros, and R. Riedi. Multifractal products of stochastic processes: A preview.
COST-257 Technical Document 257TD(99)31. `http://nero.informatik.uni-wuerzburg.de/cost/Final/TDs/257td9931.pdf`