Application of DPR-Based Splitting to Simulation of Networks with Self-Similar Traffic

Ozdemir Akin, J. Keith Townsend
The Center for Advanced Computing and Communication
Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, N.C. 27695-7914

***Abstract*— Recent studies of different networks have shown that the network traffic is statistically self-similar which makes the simulation of these networks less stable. In this paper we investigate the effect of the direct probability redistribution (DPR) based splitting technique on networks with self-similar input traffic. The DPR-based splitting has shown to increase the efficiency of simulations of a number of different network models. However, because of the long-range dependent structure of the self-similar traffic, the DPR method fails to give correct results. In order to solve this problem we suggest some modifications to the DPR method. To demonstrate the validity and efficiency of the new method we use a simple buffer simulation which has a self-similar input traffic. We used ONxOFF sources with heavy-tailed ON and OFF time distributions to generate the self-similar input traffic.**

I. Introduction

Recent measurement studies of different networks have shown that network traffic is bursty over a wide range of time scales (i.e., *long-range dependent* (LRD) or *self-similar*). These studies include Ethernet LAN traffic [1], WAN traffic [2], and VBR video traffic [3], [4]. To obtain accurate results from the simulation of these networks, self-similarity must be taken into consideration. Otherwise, using the traditional traffic models, such as Poisson, Markov-Modulated Poisson or Markov Chain models, so called *short-range dependent* (SRD) traffic models, can give misleading results [5].

One method for generating self-similar traffic is using *heavy-tailed* distributions, whose tail follows a power law. The tails of these distributions decline slowly with respect to a regular exponential distribution, causing non-negligible probabilities even for very large values of the support. The result is that larger sample sizes are required to obtain accurate results from simulation. Basically using heavy-tailed distributions has two effects on simulations, first it increases the time until which the simulation reaches the steady-state, and second it increases the variability at the steady-state. A complete discussion about the effect of heavy-tail distributions on simulations can be found in [6]. Considering these facts about the simulation with heavy-tailed distributions, generating accurate results needs more computational effort —even for high probability events— than a regular simulation of a Poisson system. So any technique that can speed-up the simulation of such networks is of value. There are a number of techniques, including *direct probability redistribution* (DPR)-based trajectory splitting, that are shown to provide speed-up for estimating network parameters [7], [8], [9], [10], [11]. In this paper we present a modification of DPR-based splitting to speed-up the simulation of systems with heavy-tailed, self-similar traffic.

This work was supported by The Center for Advanced Computing and Communication, NC State University.

The DPR-based splitting method was originally developed to speed-up the simulation of Markovian systems [7], [8]. However, methods for generating self-similar traffic incorporates statistical dependence between the states which makes it impossible to use the original subset indicator function (SIF) strategies. In order to solve this problem we developed a different subset indicator function which takes the dependence into consideration. By using this kind of SIF, DPR-based splitting can give high speed-up values for simulations of systems with self-similarity without any loss of accuracy of the estimates. In order to show this we use a simple switch which is fed by self-similar traffic. The (asymptotically) self-similar traffic is generated by superpositioning many identical ONxOFF sources, which have heavy-tailed distributions for their ON and OFF time length distributions.

In the following section we will give a brief overview of DPR-based splitting and give the mathematical definition and statistical properties of self-similarity. Also in this section the system model that is used in the simulation examples is described in detail. In Section III, the reasons the original SIF strategies fail for heavy tailed distributions is investigated, and an alternative SIF is introduced. Simulation results demonstrate the efficacy of the new strategy are given in Section IV. Conclusion remarks given in Section V.

II. Background and System Model

A. DPR-Based Splitting

Here we will briefly explain the basic principles of DPR-Based splitting —a more complete presentation can be found in [7], [8]. Let $s_1, s_2, \ldots, s_n \epsilon S$ be the state space of the system, where s_i's form an irreducible discrete-time Markov chain. Also let $V_0, V_1, \ldots (V_k \epsilon 1, 2, \ldots, n)$ denote the system evolution. In DPR the state space is partitioned into m mutually exclusive subsets, $S_1, S_2, \ldots, S_m$. The subset indicator function (SIF) which is given by,

$$\Gamma(i) = j, \; i = 1, 2, ...n \text{ and } j = 1, 2...m \tag{1}$$

is used to find the corresponding subset for each state. During the simulation this function is used for each observation to map the observation to the corresponding subset, i.e. $\Gamma(V_i) = j$ indicates that the observation V_i is in subset j.

The DPR technique redistributes the steady state probability by altering the transition probability matrix of the system. This is done in such a way that state i is visited relatively $\mu_{\Gamma(i)}\theta$ more than a regular Monte Carlo (MC) simulation where $\mu_{\Gamma(i)}$ is the oversampling factor and θ is a normalization constant. For convenience let $\mu = \{\mu_1, \ldots, \mu_m\}$. The oversampling factors are the main parameters that control the redistribution of the steady state probabilities. We assume that $\mu_1 = 1$ and $\mu_1 \leq \mu_2 \ldots \leq \mu_m$ which can be achieved by reordering the subset indexes and normalizing μ by μ_1. Iterative subset exploration (ISE) is used to find the oversampling factors. This method sets the μ values such that each subset has asymptotically the same hit probability, the details of this method can be found in [7], [8]. However, this method has not been proven to give the optimal μ values.

The DPR method can be implemented by using trajectory splitting, the details of the method and the algorithm can be found in [7], [8]. Transition from subset i to j, $(i < j)$, causes Y new sub-trajectories where Y is a random variable with an expected value $E[Y] = \mu_j/\mu_i - 1$. Whenever a splitting occurs each new sub-trajectories is given a ticket value, which has the following distribution,

$$Pr(\Omega = l | S_i \to S_j) = \begin{cases} \frac{\mu_l - \mu_{l-1}}{\mu_j - \mu_i} & \text{if } i < l \leq j \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The DPR method starts with a single trajectory (main trajectory) which has the minimum possible ticket value. A sub-trajectory is terminated when it attempts to enter a subset with an index less than its ticket value, as the main trajectory has the minimum ticket value it can not be terminated in any transition.

In DPR technique each state, $s_i, i = 1, 2 \ldots, n$, is visited $\mu_{\Gamma(i)}$ times more than a regular MC simulation. Therefore, when an event of interest occurs while the system is in subset j the event counter should be incremented by $1/\mu_j$ instead of unity in order to have unbiased estimates. Also the DPR based splitting does not require the event of interest to be in a single subset, which means that we still have unbiased estimates even if the event occurs in many different subsets.

The speed-up factor, G, is the ratio of the computational effort required for the DPR-based splitting and MC simulation to have a fixed estimator variance. In this paper we will use a slightly different method to find G than the previous methods. This new G is given by,

$$\hat{G} = \frac{\hat{\sigma}^2_{\text{MC}} N_{\text{MC}}}{(1 + \hat{H})\hat{\sigma}^2_{\text{DPR}} N_{\text{DPR}}} \tag{3}$$

,where $\hat{\sigma}^2_{\text{X}}$ is the estimated variance of the estimator using method X (MC or DPR), and N_{x} is the length of the simulation. $\hat{H}$ is the estimated splitting overhead. From this it is seen that we also need a MC simulation to estimate the speed-up, but it is important to note that this is only needed for speed-up estimation, in general the DPR technique does not require this. However, for this equation it is assumed that the variance of the MC simulation decreases proportional to $1/N_{\text{MC}}$. In this paper the parameter we are interested in (buffer length) satisfies this requirement so we can use (3).

B. *Self-Similarity*

A basic definition for self-similarity in discrete time domain can be give as follows, let $X = \{X_n, n = 1, 2, ...\}$ be a stationary time series and also let $X^{(m)} = \{X_n^{(m)}, n = 1, 2, ...\}$ be another (m-aggregated) time-series where $X_n^{(m)} = \frac{1}{m}\sum_{i=nm-(m-1)}^{nm} X_i$. The time-series X is called exactly self-similar with parameter $\beta(0 < \beta < 1)$ if, $\text{Var}[X^{(m)}] = \text{Var}[X]/m^\beta$ and $R_{X^{(m)}}(k) = R_X(k)$, where $\text{Var}[X]$, and $R_X(\cdot)$ are the variance and autocorrelation of X respectively.

A weaker type of self-similarity is asymptotic self-similarity. A process X is said to be asymptotically self-similar with parameter $\beta(0 < \beta < 1)$ if for k large enough $\text{Var}[X^{(m)}] = \text{Var}[X]m^\beta$ and $R_{X^{(m)}}(k) \to R_X(k)$ as $m \to \infty$.

The degree of self-similarity is indicated by the Hurst parameter (H, where $0.5 < H \leq 1$), where $H = 1 - \beta/2$. The process is not self-similar when H is equal to $1/2$, and self-similarity increases with increasing H.

For packet-switch network traffic, self-similarity translates to burstiness across all time scales (up to infinity). Of course for a realistic system it is not possible to have an infinite range but in our simulations the self-similarity continues for aggregation levels up to the simulation length. So we will call the generated traffic in our simulations as self-similar.

C. *Generating Self-Similar Traffic with ON×OFF Sources*

In order to generate self-similar traffic we used multiple ONxOFF sources where ON and OFF periods have

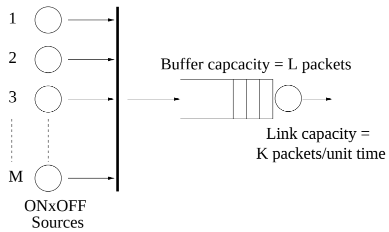

Fig. 1. The system model used throughout the paper. The lengths of the ON and OFF periods have heavy-tailed distributions.

heavy tailed distributions, a proof for this can be found in [12]. In our system the ON and OFF periods strictly alternate, and during one time unit each ON source sends one packet, while OFF sources are idle.

The incoming traffic can be approximated by a Poisson-Zeta process [5], where the number of bursts generated at each time unit has a Poisson distribution. Also the length of each burst has a Zeta (Zipf or discrete Pareto) distribution, with probability mass function (PMF) given by $p(k) = Ck^{-(1+\rho)}$ where $C = 1/\sum_{k=1}^{\infty} k^{-(1+\rho)}$. It has been shown that the superposition of many of these independent ONxOFF sources generates an asymptotically self-similar process [13]. In the Poisson-Zeta model there is no restriction for the length of the OFF period distribution, but in our system we use the Zeta distribution for length of the OFF periods.

This traffic is sent to a switch with a single input buffer that serves K packets at each time unit. The maximum buffer length is L. Also for convenience the processing time of all packets is assumed to be constant and equal to $1/L$ time units. A block diagram of the overall system is shown in Fig. 1.

III. Using DPR-Based Splitting with Self-Similar Traffic

In this section we first look at the reason why the original SIF strategy for DPR does not work with the system model shown in Fig. 1. We then introduce a new SIF as a solution. Here the network parameter of interest is the probability mass function of the buffer length.

In DPR-based splitting in order to get high speed-up values it is important to choose a SIF, that will increase the number of occurrences of the events of interest (in this case, large buffer lengths). In previous studies with Markovian traffic accurate estimates of the buffer length distribution were achieved when the SIF was also chosen to be the buffer length itself [7], [8]. However, when this SIF strategy is used for the system shown in Fig. 1 the DPR method gives inaccurate results, as shown in Fig. 2. This was not an unexpected result as it is known that the DPR method does not

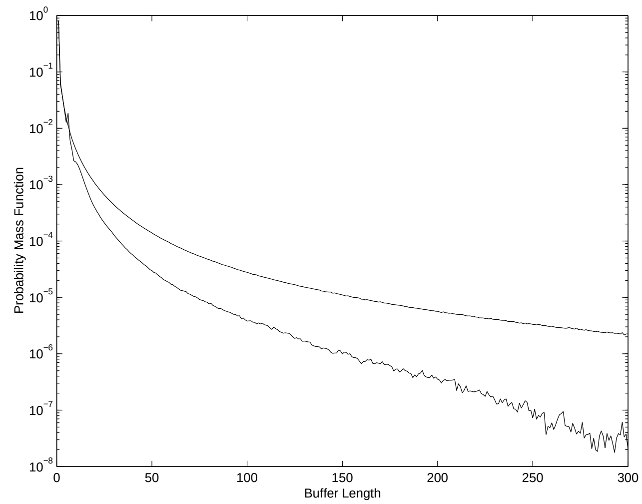

Fig. 2. In this figure the curve at the top represents the correct PMF while the bottom curve is the result by using the buffer length as the SIF.

work well for systems with memory. A more serious issue is that the error is manifest as a bias in the estimated PMF but one important point to note is that DPR does not give any kind of warning and also it gives reasonable results. The main reason why this SIF does not work is because of the termination policy of the sub-trajectories. The sub-trajectories are killed when they attempt to enter a subset value which is smaller than its ticket value regardless of the source states as discussed in Section II. This process unintentionally truncates the ON states of the sources, which eventually changes the distribution of the ON periods. As a result the probability mass function of the buffer length changes.

From the above example we see that even if the system is a Markovian system, DPR may fail because the algorithm may change some of the *predefined distributions* of the system. By *predefined distribution* we mean the distributions that are a part of the system description. Thus, care must be exercised using DPR-based splitting when a system variable has a dependence structure given by,

$$P(X_t|X_{t-1}) = P(X_t|X_{t-2}) = \ldots = P(X_t|X_{t-i}),\ i > 1$$

where X is a random variable with a predefined distribution. Of course for small i DPR may work well, but when i gets large it may alter the predefined distribution. This dependence must be considered when choosing the SIF.

To circumvent this problem we use a new SIF that keeps the sub-trajectories alive until all the ON sources (at the splitting point) can change to the OFF state at least once. To achieve this, we need to define a new variable $\Omega_k (k = 1, 2, ...M)$, where M is the number of sources. The variable Ω_k represents the length of the ON period when source k goes from OFF to ON. This value does not change until the source transitions back to the OFF state, and becomes zero during the OFF period. Then the new SIF is given by,

$$\Gamma(V_t) = \lfloor \frac{\sum_{i=1}^{M} \Omega_i}{h} \rfloor + 1, \ t = 1, 2, ... \quad (4)$$

where $\lfloor \cdot \rfloor$ function takes the integer part of the real number. As Ω_k can take quite large values we use a normalization constant, h, which prevents the short ON periods from changing the subset. This constant also it decreases the number of subsets used by an order of h. This function keeps the sub-trajectories alive until the sources, even with long burst lengths, transition to the OFF state.

However, as seen from (4) the new SIF is related to the parameter of interest in a very complicated manner. So we can expect that the statistical properties of these two variables can be quite different, as the DPR method alters the characteristics of the simulation significantly.

IV. Numerical Results

In this section we present simulation results to demonstrate the efficacy of the new SIF introduced above. In these simulations the PMF of the buffer length of the system shown in Fig. 1 is observed. As it is very difficult to obtain theoretical results for the system parameter we are interested in (buffer length), we use MC simulations to compare our results. To be able to do this we use parameter values that would give high probabilities (minimum probability values on the order of 10^{-7}) so that running the MC simulations would not take a prohibitive amount of time. As shown in an example below, it can be expected that DPR-based splitting may give even higher speed-up for lower probability estimates.

To find the estimates we ran a independent simulations each having 10^b (not including the warm-up period) observations, which gives a total length of $a \times 10^b$. In the simulations we use a very long (10^6 time units) warm-up period to eliminate all start-up effects.

In the simulations we used a maximum buffer length of $L = 800$ packets, and a link capacity (the number of packets that can be served in a unit time) of $K = 4$ packets/unit time. The number of ONxOFF sources were set to $M = 150$ and the lengths of the ON and OFF periods have Zipf distribution with parameters $\rho_{on} = 1.6, \rho_{off} = 1.1$. Using these values gives self-similar traffic with a Hurst parameter, H, of 0.7. We used $a = 20$, and used three different b values, 6, 7 and 8, both for MC and DPR simulations. We assumed the longest MC simulation result as the correct PMF and compared the DPR results with this one. For all

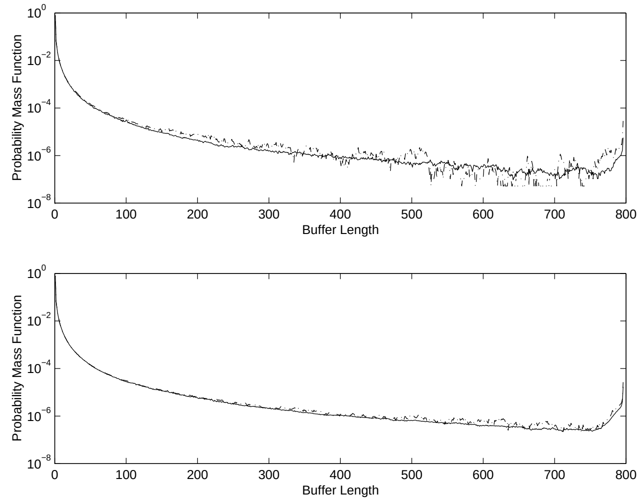

Fig. 3. Probability mass function of the buffer length is shown in these figures. The solid curves are for the DPR method and the dashed ones are for the MC simulations. The total length of the simulations are 20×10^6 and 20×10^7.

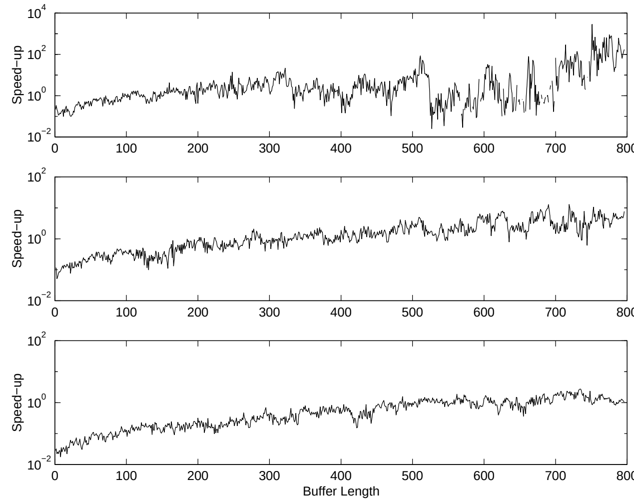

Fig. 4. In this figure the estimated speed-up values are shown for three different simulation lengths. The plots are for 10^6, 10^7 and 10^8 from top to bottom.

simulation run lengths we see that the DPR method gives the same PMF as the MC simulations thus validating the DPR method with the new SIF strategy. Comparisons between the DPR method and MC simulation are shown in Fig. 3. As expected for small sample sizes MC results are noisy. Estimated speed-up values are presented in Fig. 4. From the figure we can see that the speed-up values decrease with increasing simulation length. This is mainly because the observed parameter is different than the subset indicator function, so even if the variance of the subsets decreases proportional to $1/b$ the variance of the buffer lengths decreases in a different manner. The variance versus buffer length plot

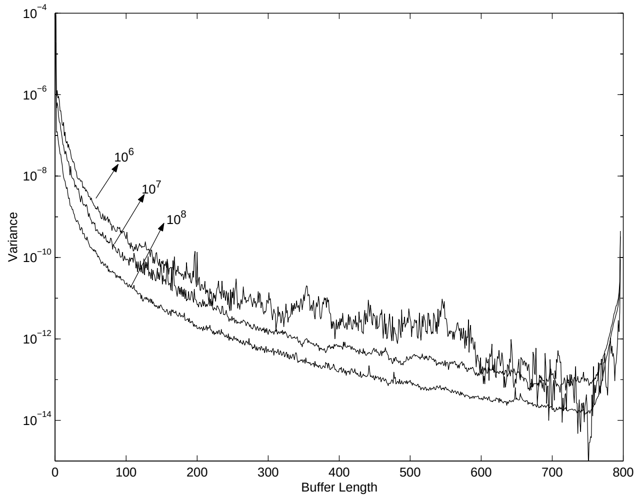

Fig. 5. Estimate of the variance of the buffer length as a function of queue length. It is seen that even the simulation length increases ten fold, the variance does not decrease in the same order.

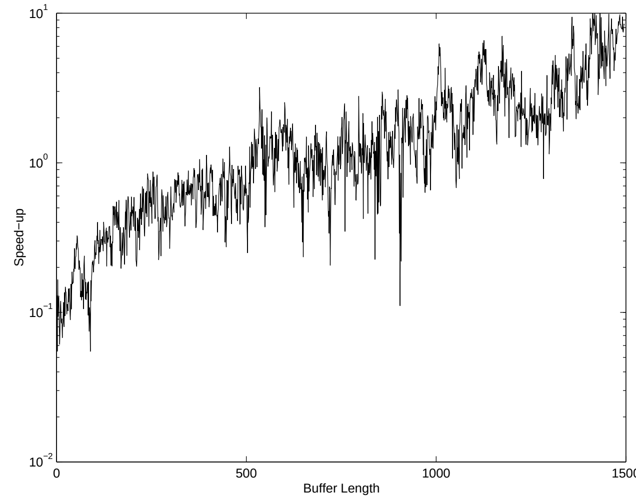

Fig. 7. Estimated speed-up values for $L = 1500$.

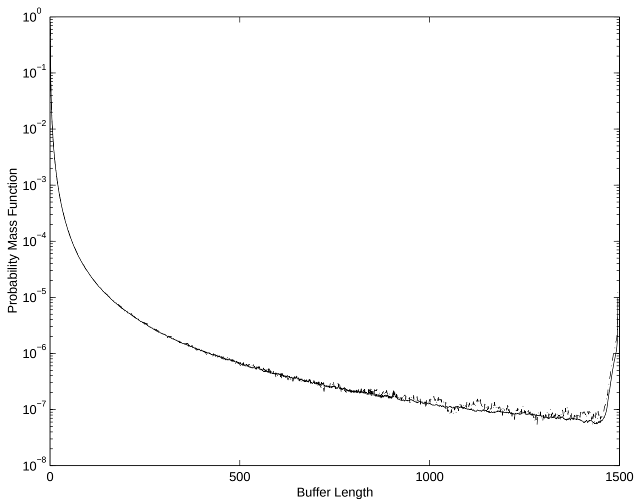

Fig. 6. PMF for buffer length when buffer size is, $L = 1500$. The dashed curve is for MC and the solid curve is for DPR simulation.

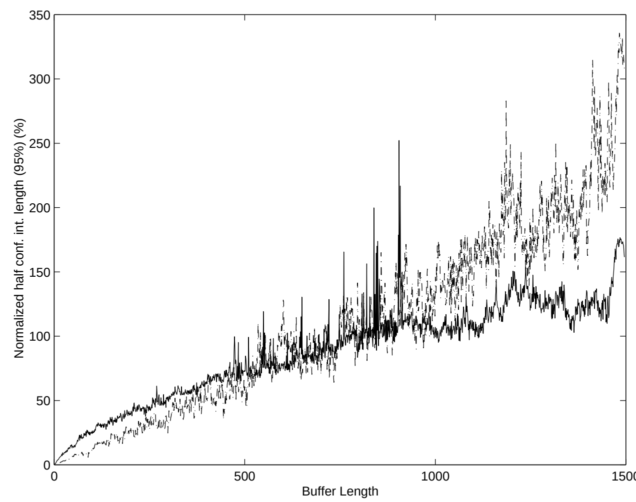

Fig. 8. Estimated half size confidence intervals (for 95%) normalized to the mean estimates for $L = 1500$. The solid line is for DPR and the dashed line is for MC method.

is shown in Fig. 5. As seen from the figure the decrease of the variance with the increasing simulation length is less than the expected value, i.e., when the simulation length is increased by an order of magnitude, there is only a slight decrease in the variance. As mentioned before this happens due to the fact that the DPR technique alters the evolution of the system. A first look at Fig. 4 may give the impression that the DPR method looses its efficacy when the simulation length is on the order of 10^8. However, it is important to note that the PMF (Fig. 3) has a minimum value greater than 10^{-7}, which makes the simulation time long enough to get a good result for even a MC simulation. We can see that even when we drop the probabilities slightly, the speed-up will again give high values. To demonstrate this we change the buffer size, L, to 1500. The new PMF and speed-up values for 20×10^8 observations can be seen in Fig. 6 and Fig. 7.

In all simulations when we compare the probability mass functions that we get using MC and DPR methods, it is seen that the MC method gives much noisier results. However, even if the DPR method gives better results the speed-up curves do not fully reflect this. In order to observe the efficiency of the DPR method in Fig. 8 (for $L = 1500$) and ,Fig. 9 (for $L = 800$) the confidence intervals for the estimates are plotted. As it is seen from the figure, the DPR method gives much better estimates than the MC simulation with increasing buffer length.

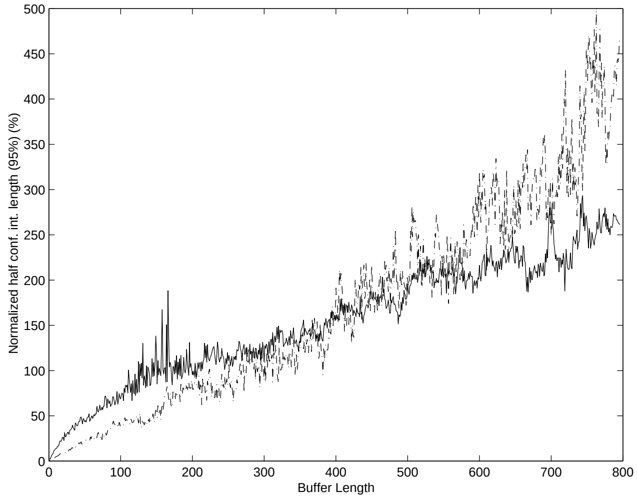

Fig. 9. Estimated half size confidence intervals (for 95%) normalized to the mean estimates for $L = 800$. The solid line is for DPR and the dashed line is for MC method each having a simulation length 20×10^7.

V. Conclusion

In this paper we investigated the shortcomings of DPR-based splitting on systems with self-similar traffic and introduced a new subset indicator function in order to apply the DPR-based splitting to these kind of systems. We demonstrated the effect of this new SIF with a simple buffer model that is fed by self-similar traffic. In order to see the efficiency of the new method we used both the traditional speed-up calculations and also the confidence intervals. The simulation results showed that the DPR method gives much better results than a regular MC simulation, especially with low probabilities.

References

[1] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of Ethernet traffic. In Deepinder P. Sidhu, editor, *ACM SIGCOMM*, pages 183–193, San Francisco, California, 1993.

[2] Vern Paxson and Sally Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[3] Mark W. Garrett and Walter Willinger. Analysis, modeling and generation of self-similar VBR video traffic. In *SIGCOMM*, pages 269–280, 1994.

[4] Taqqu M.S. Beran J., Sherman R. and Willinger W. Long-range Dependence in Variable-Bit Rate Video Traffic. *IEEE Transactions on Communications*, 43(2/3/4), February/March/April 1995.

[5] Yanhe Fan and Nicolas Georganas. Performance analysis of ATM switches with self-similar input traffic. *International Journal of Computer Systems Science and Engineering*, 12(2):95–102, 1997.

[6] Mark Crovella and Lester Lipsky. Long-lasting transient conditions in simulations with heavy-tailed workloads. In *Winter Simulation Conference*, pages 1005–1012, 1997.

[7] Z. Haraszti and J. K. Townsend. The Theory of Direct Probability Redistribution and its Application to Rare Event Simulation. *ACM Transactions on Modeling and Computer Simulation*, 9(2), April 2000.

[8] Z. Haraszti and J. K. Townsend. The Theory of Direct Probability Redistribution and its Application to Rare Event Simulation. In *Proc. IEEE Int. Conf. Commun., ICC '98*, pages 1443–1450, Atlanta, June 1998.

[9] P. Heidelberger. Fast Simulation of Rare Events in Queueing and Reliability Models. *ACM Trans. on Modeling and Comp. Sim.*, 5(1):43–85, January 1995.

[10] O. Akin and J. K. Townsend. Efficient Simulation of TCP/IP Networks Characterized by Non-Rare Events Using DPR-Based Splitting. In *Proceedings of the IEEE Global Telecom. Conf., GLOBECOM '01*, Nov. 2001.

[11] O. Akin and J. K. Townsend. Efficient Simulation of Delay in TCP/IP Networks Using DPR-Based Splitting. In *Proc. IEEE Int. Conf. Commun., ICC '02*, 2002.

[12] Walter Willinger, Murad S. Taqqu, Robert Sherman, and Daniel V. Wilson. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, 1997.

[13] N. Likhanov, B. Tsybakov, and N. Georganas. Analysis of an atm buffer with self-similar fractal input traffic. In *Proceedings of IEEE INFOCOM '95, Boston, MA.*, pages 985–992, 1995.