Active Queue Management for Self-Similar Network Traffic

Farnaz Amin*, Kiarash Mizanain**, and Ghasem Mirjalily***

* Electrical Engineering and computer science Department, Yazd University, farnaz.amin@stu.yazduni.ac.ir

**Electrical Engineering and computer science Department, Yazd University, k.mizanian@yazduni.ac.ir

*** Electrical Engineering and computer science Department, Yazd University, mirjalily@yazduni.ac.ir

Abstract: *Recent studies have shown that network traffic is Self-Similar, and it has a great impact on network performance. Self-similar traffic can lead to large queuing delays and packet loss rates. In this paper, we devise an active queue management algorithm which takes the Self-similarity of traffic into account. Hurst is a key parameter describing self-similar processes, which is designed to determine the degree of the self-similarity. In our approach, we utilize a technique based on the wavelet method to estimate the Hurst parameter. Classification is based on real-time estimation of Hurst parameter. Also, we use ns2 to simulate the network configurations and to generate traffics with Pareto distribution. The numerical results illustrate the performance of the proposed algorithm in contrast to other recently implemented buffer management algorithms in ns2.*

Keywords: Hurst parameter, self-similarity, congestion control.

1. Introduction

Recent studies have shown that the traffic on a local network is bursty at wide range of time scales. This phenomenon has been observed in different networks from Ethernet to ATM, VBR video and WWW traffic [1]. Furthermore, Studies have shown that Self-similarity has a negative effect on the network performance leading queuing delay and packet loss [2, 3, 4]

Neglecting the self-similarity may lead to optimistic performance predictions and inappropriate allocation of network resources allocation. Understanding the concept of self-similar traffic is important for the network planners and designers. It should be noticed especially in high speed networks. Guarantying the Quality of Service (QoS) for self-similar traffic in high speed networks is an important concern. For guarantying the QoS in high speed networks, it is necessary to properly allocate network resources to the service.

The best utilization of resources in a network environment depends upon characterizing the traffic itself and then setting few parameters such as buffer size and bandwidth to maximize the performance. Optimal network resource allocation is determined by optimal buffer size and the optimal allocation of bandwidth.

One of the impacts of self-similarity on a network is that it increases the packet loss. The solution in order to curtail the impact of self-similarity is structural resource allocation; one option for structural resource allocation is the buffer size in routers.

These performance effects can be limited by estimating Hurst parameter by using wavelet analysis over the network traffic in real-time. The degree of similarity will be determined by the Hurst parameter. Therefore it is important to consider it in the network control and management. Using estimation of the Hurst parameter as a metric could be useful to achieve more stable routes and buffer management [5].

In this paper, we use the wavelet-based tool to measure the Hurst parameters of the network traffic in real-time. Since the more the traffic is self-similar, the more it needs buffer, in this paper we have proposed an algorithm for queue management which takes into account the degree of the traffic self-similarity and classify the traffic based on these Hurst parameter values.

The rest of the paper is organized as follows. In section 2 we define self-similarity and Heavy-Tailed distribution and in section 3 we describe the wavelet method to estimate the Hurst parameter. Section 4 discusses about queue management and also describes our proposed algorithm. Finally simulation results and some discussions are presented in section 6.

2. Self-Similarity and Heavy-Tails

The similarity is defined as the characteristic of traffic on scale invariance. The term self-similar was first used by Mandelbrot [6] in 1960's to identify those processes that are scalable over the time without losing its scalable properties.

A stationary process is long range dependent (LRD) if its autocorrelation function $R(k)$ is non-sum-able, therefore we can define long-range dependence only when the infinite series exist [7].

Suppose we have $X = (X_k : k = 0, 1, 2, \dots)$ as a covariance stationary stochastic process for which μ is mean, σ^2 is variance, and $R(k)$ for $k \geq 0$ is autocorrelation function. Also, suppose that the autocorrelation function of X has the following form:

$$R(\mathrm{k}) \sim \eta_1 k^{-\beta}, \text{ as } k \rightarrow \infty \tag{1}$$

where $0 < \beta < 1$. (The constants $\eta_1, \eta_2, \dots$ are finite positive integers.) For $m = 1, 2, \dots$, let $X^{(m)} = (X^{(m)}{}_k : k = 1, 2, 3, \dots)$ is the covariance stationary time series with corresponding autocorrelation function $R^{(m)}$ that is obtained by calculating the average of the original series X over the non-overlapping time periods of size m.

Samples are given by

$$x_i^m = \frac{1}{m}(x_{im-m+1}) + \cdots + x_{im}, i \geq 1 \tag{2}$$

The process x is considered as a second-order self-similar with a self-similarity Hurst parameter

$$H = 1 - \beta/2 \tag{3}$$

if the corresponding $X^{(m)}$ processes have the same correlation functions as X. In another words, $R^m(k) = R(k)$ for $m = 1, 2, \dots$ and $i = 0, 1, 2, \dots$.

The Hurst parameter represents the degree of self-similarity in the observed traffic. When the value of the Hurst parameter is between 0.5 and 1, then the traffic is said self-similar and closer values of H to 1 indicate a higher degree of self-similarity.

2.1. Heavy-tailed Distribution

Another property of self-similarity is called long-range dependence (LRD) which satisfies the autocorrelation function in the most common definition of self-similarity [5]. Long-range dependence of the process is characterized by a slowly decaying autocorrelation function and the non-sum-able autocorrelation function. By increasing the k, the autocorrelation function decays hyperbolically. This is opposed with the property of short-range dependence (SRD), where the autocorrelation function decay is exponential.

There is a common relation between short-range and long-range dependences with the value of the Hurst parameter of the self-similar process as follows [5]:

$$0 < H < 0.5 \rightarrow \text{SRD}$$
$$0.5 < H < 1 \rightarrow \text{LRD}$$

Network traffic is a stochastic process and is described as such. Self-similarity and long-range dependence (LRD) properties are described by heavy-tailed distributions. The heavy-tailed distributions are depicted as hyperbolic, while this is not true for light-tailed distributions (i.e. exponential distribution) where distributions decay exponentially.

In this paper we use the distributions with the property of being *heavy-tailed*. We say a distribution is heavy-tailed if

$$P[X \geq x] \sim x^{-\alpha}, \text{ As} x \rightarrow \infty, \ 0 < \alpha < 2 \tag{3}$$

This means that, regardless of the behavior of the distribution for small values of the random variable, if the asymptotic shape of the distribution is hyperbolic, it is considered as heavy-tailed. The simplest heavy-tailed distribution is the *Pareto* distribution. It is hyperbolic over its entire range. The probability mass function of *Pareto* distribution is

$$p(x) = \alpha k^\alpha x^{-\alpha-1}\alpha, \ k > 0 \ \ x \geq k \tag{4}$$

Also its cumulative distribution function is as follows:

$$F(x) = P[X \leq x] = 1 - \left({}^{k}\!/_{x}\right)^\alpha \tag{5}$$

In the above equation, the parameter k represents the smallest possible value of the random variable. Pareto distributions have some properties that qualitatively differ from distributions more commonly encountered such as the exponential distribution, normal distribution, or Poisson distribution.

3. Self-similar Traffic Parameters Estimation Using Wavelet

In this paper, the first phase of classification is real-time estimation of the Hurst parameter. We use a technique based on the wavelet method to estimate the Hurst parameter. In the review of different estimation methods, the Abry-Veitch (AV) wavelet-based was adopted because it has low space and time complexity and also good statistical properties [8].

The wavelet method used for estimation of the Hurst parameter offers a semi parametric estimator which can be more efficient than other estimation methods in terms of statistical properties and particularly space complexity. In the wavelet method, the differences in aggregated series will be analyzed if Y^j is aggregated series. Then,

$$Y_{2k}^{j+1} = \left(Y_{2k}^j - Y_{2k-1}^j\right) 1/\sqrt{2}, k = 1,2, \dots, n/2j \tag{6}$$

where $j = 1,2, \dots$. Note that the expectation of Y is zero. In frequency domain, the variance is equivalent to the signal energy in a frequency band depending on j (E_j). In our approach, we plot E_j versus 2^j on log-log scale. Linearity must be checked for all scales *j*. Then the Hurst parameter will be calculated by calculating the slope of the line.

4. Queue Management

In general there are two major approaches used in routers for buffer management in order to congestion control and bandwidth allocation: packet scheduling and queue management

The first method which is called link scheduling as well uses several logical or physical queues and allocates output bandwidth to each class of traffic where a class can be a flow or a stream group of similar flows. This is the basic idea of the Fair Queuing (FQ) and the Class-Based Queuing (CBQ) algorithms [9]. Stochastic Fairness Queuing (SFQ) is a simple implementation of the fair queuing algorithms family. It is less accurate than others, but it also requires less calculation while being almost perfectly fair. Adopting this mechanism requires a change in network management and billing practices. Also, the complexity of the algorithms and state requirements of scheduling makes it difficult to use [10].

The second method, called active queue management, uses advanced packet queuing disciplines in contrast to traditional FIFO drop-tail queuing on an outbound queue of a router to actively handle (or avoid) congestion[11].

One of the oldest and best-known active queue management mechanisms is Random Early Detection (RED), which prevents congestion by monitoring the output buffer to detect impending congestion, and randomly select and notify the sender of network congestion so that they can reduce their transmission rate [12]. RED has a critical problem and it is that non-TCP flows that are unresponsive than TCP can take more share of the output bandwidth [10, 13].

An active queue management scheme, called class-based threshold (CBT), has been proposed in [10], which supports the UDP flows by adding simple class-based static bandwidth reservation mechanism to RED.

CBT implements explicit resource reservation capabilities of CBQ in a single queue, managed by RED without packet scheduling. However, as in the case of CBQ, static resource reservation mechanism of CBT could result in poor performance for speed-changing traffic and certainly unfair since it changes the best effort nature of the Internet.

To eliminate the limitations of explicit reservation of CBT resources and to maintain its good features simultaneously, [14] proposed Dynamic-CBT (D-CBT). D-CBT fairly allocates the bandwidth of a congested link to the traffic classes by dynamically assigning the UDP thresholds such that the summation of the fair share of flows in each class is assigned to the class on any given time era if the mix of flows changes**.**

4.1. Proposed Alghorithm

In this paper, we have suggested a queue management algorithm for self-similar traffic which is supposed to have advantages of fair queuing and its implementation is simple. In this algorithm, the degree of self-similarity is taken into account for allocation of buffer to the flows.

We apply the Abry-Veitch (AV) wavelet-based tool to estimate the Hurst parameters to determine the degree of self-similarity of the generated traffic.

When a packet is delivered to router, it should be classified first. Three distinct queues with different sizes are considered for the flows; first queue is used for those flows whose self-similarity is higher than 0.7 and their Hurst parameter is between 0.7 And 1. Second queue is used for those flows whose degree of self-similarity is between 0.5 and 0.7. Finally, third queue is used for those flows whose degree of self-similarity is less than 0.5. Sending packets from theses queues is round robin. For active queue management, we have implemented RED on each queue.

By receiving a new packet at a FIFO output queue, two tasks must be done like RED algorithm. First, it calculates the average length of the queue, called *avg*. Then *avg* will be compared to two thresholds: minimum threshold and maximum threshold. If this average value is less than minimum threshold, it assumes that there is no congestion or it is at the minimum level. In such circumstances, the received packet will be placed in the queue. But if the average value is greater than the maximum threshold, it concludes that the congestion is at a serious level. So the incoming packet will be discarded. A region for the value of *avg* between the above two thresholds is called critical region. In this region, a probability *p* will be computed by using the current value of *avg*. Then, every packet will be discarded with the probability *p* (i.e. it will be queued with the probability *1 – p*) when the queue is in the critical region. These threshold values have an important effect on the isolation and allocation in the algorithm.

According to [10], we dynamically assign the maximum threshold value to each class. In this way, we consider the number of existing packet in a class in order to allocate the bandwidth to it fairly. The following equation holds between the threshold value and the bandwidth:

$$Th_i = \left(\frac{B_i \times L}{P_i}\right) \times H_i \qquad (7)$$

in which P_i is mean packet size, L is packet processing delay, H_i is Mid-range of Hurst parameter for each class and B_i is bandwidth of class *i*, where:

$$B_i = sharebandwidth_i \times B \qquad (8)$$

and:

$$sharebandwidth_i = \frac{num\ of\ packets\ in\ class_i}{total\ num\ of\ packets} \qquad (9)$$

We use the mean Hurst in equation (7) because we need more buffer for higher values of Hurst. Other parameters in our algorithm are set exactly the same as RED.

5. Simulation Results

To evaluate the performance of our algorithm, we use Network Simulator NS2 to simulate a series of scenarios. The symmetrical network topology has been adopted for the simulation. As shown in Fig. 1, 2 and 3, there are *n* connections that are established to the bottleneck link. We choose the following values for *n*: 5, 10 and 15.

The more the number of connections leads to more congestion in the bottleneck. The bottleneck link's bandwidth is 1.7Mbps, and all of the *n* connections' output are 10Mbps (Table1).

On the other hand, integration of individual ON-OFF sources allows explaining the observed similarity in wide-area traffic, because it has the self-similar behavior [15]. In this case the traffic source is transmitting packets at constant rate in the ON period and is idle in the OFF period. The ON-OFF periods are taken from a heavy tailed distribution such as Pareto distribution, which has infinite variance and finite mean. When a large number of these sources are aggregated, it concludes the traffic has self-similar characteristics.

The model has been implemented in NS2. To generate traffic, we used Pareto distribution in NS2 and the α parameter has been given to each source randomly between 1 and 2.

Duration of the simulation process is 80 seconds. The simulation parameters have been described in Table 1.

We compared the throughput of our algorithm with Drop-tail, SFQ and RED that are the currently implemented buffer management algorithms in NS2.

Figures 4-6 show the Packet delivery Ratio (PDR) of the simulation results of Drop-Tail, RED, SFQ and our proposed algorithm.

Obviously, network performance will be decreased after the increasing of the number of nodes. According to these figures, in all these queues, we witness such situation.

The performance of our proposed algorithm in all 3 networks is better than SFQ and Drop-Tail and DRR queues (TABLE 2). Because classification has been used for queuing, this algorithm offers more fairness (balance) among the flows in comparison to drop-tail and RED and also prevents the impact of other flows in the case of congestion.

TABLE 1: Simulation Parameters

Botteleneck link bandwidth	1.7 [Mbits/s]
Delay of the Botteleneck link	20 [ms]
Packet Size	1,000 [bytes]
Buffer Size	100 [packets]
Rate	2 [Mbits/s]

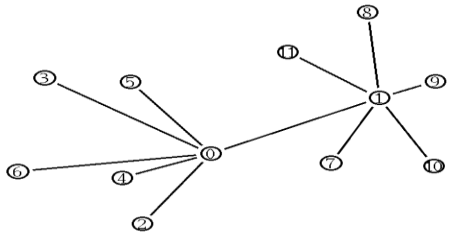

Fig. 1: Simulation model for 5 node network

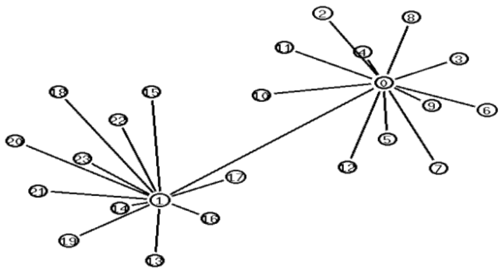

Fig. 2: Simulation model for 10 node network

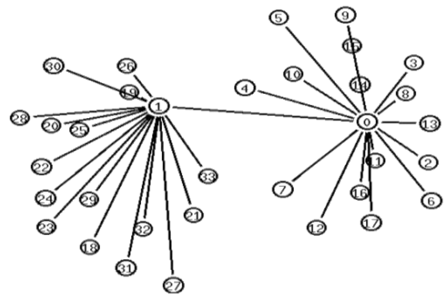

Fig. 3: Simulation model for 15 node network

TABLE I: Packet Delivery Ratio

No. of nodes	Drop-Tail	RED	SFQ	Proposed algorithm
5	33.27	37.65	32.85	39.74
10	15.89	17.39	15.85	17.43
15	10.75	11.49	10.74	11.5

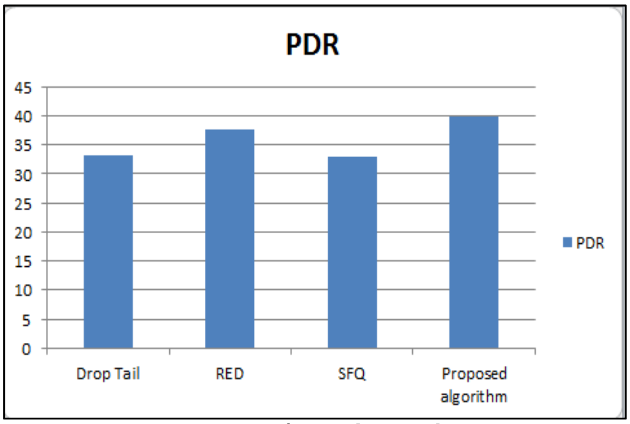

Fig. 4: PDR for 5 node network

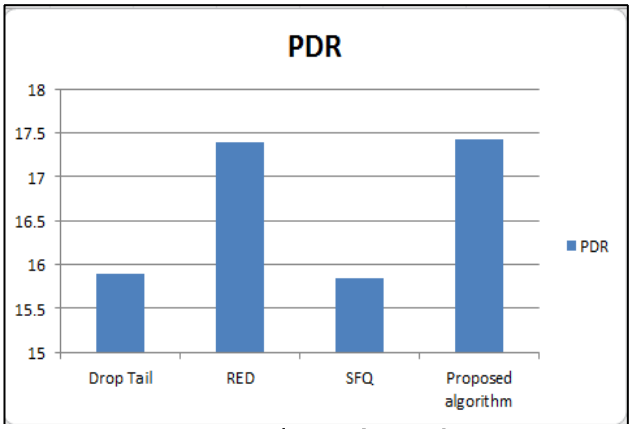

Fig. 5: PDR for 10 node network

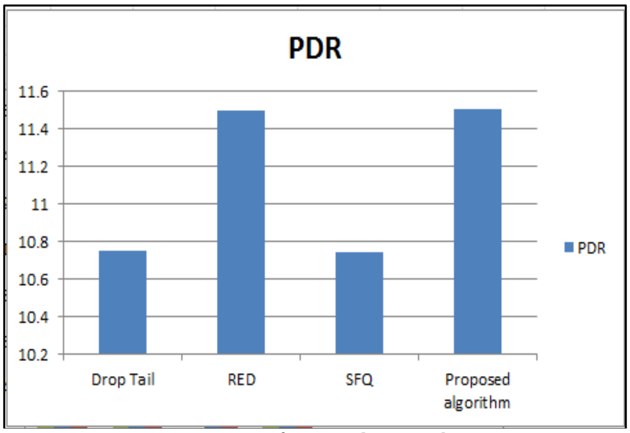

Fig. 6: PDR for 15 node network

6. Discussion

In this paper, we have proposed a new algorithm which does not require any modification to all end system TCP/IP stacks but it can be solely implemented in routers. Our scheme helps to reduce dropped packets better than SFQ and Drop-tail and RED algorithms with 100 queue size. Also it is offers more fairness between flows by using classification.

The simulation results with uniform RTT distribution were presented for Queue Managements that have been currently implemented in NS2 (Drop tail, SFQ and RED), as well as our algorithm. In our approach, in the case of congestion event, only those packets in queue which have caused congestion are dropped and their influence on other queues is avoided.

Finally, it's noticeable that the proposed algorithm is just a simple and novel idea. We aim to improve it in near future and make it more efficient.

Acknowledgements

This research was sponsored by the Telecommunication Research Center of Iran.

References

[1] M. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes, *Proc.* 1996 *ACM SIGMETRICS.*

[2] Wide Area Traffic: The Failure of Poisson Modeling Vern Paxson and Sally Floyd 1995.

[3] W. Stallings, "High-speed Networks: TCP/IP and ATM Design Principles," Prentice-Hall, 1998.

[4] K. Park, G. Kim, and M. Crovella, "On the Effect of Traffic Self-similarity on Network Performance," ***in*** *Proc. of IEEE SplE'97,* 1997.

[5] S. A. Kalim and L. Sacks, "An Investigation Using Wavelet Analysis to Detect A Change In The Characteristics Of Self-Similar Traffic," *Department of Electronic Engineering, University College London*, 1998.

[6] B.B Mandelbrot and J.W. Van Ness, "Fractional Gaussian motions, fractional noises and applications SIAM Rev, vol.10, pp. 422-437,1968.

[7] V. Paxson, "Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 5, pp. 5–18, 1997.

[8] P. Abry and D. Veitch, "Wavelet analysis of long-range-dependent traffic," *IEEE Trans. Inform. Theory*, vol. 44, no. 1, Jan. 1998.

[9] Floyd, S. and Jacobson, V., "Link-sharing and Resource Management Models for Packet Networks", IEEE/ACM Transactions on Networking, Vol. 3 No. 4,August 1995.

[10] Parris, M., Jeffay, K. and Smith, F. D., "Lightweight Active Router-Queue Management for Multimedia Networking", *Multimedia Computing and Networking*, SPIE Proceedings Series, Vol. 3020, San Jose, CA, January 1999.

[11] Floyd, S., "TCP and Explicit Congestion Notification", *Computer Communication Review*, October 1994.

[12] Floyd, S. and Jacobson, V., "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, August 1993.

[13] Lin, D. and Morris R., "Dynamics of Random Early Detection", In *Proceedings of SIGCOMM '97*, Cannes, France, September 1997.

[14] J. Chung and M. Claypool, "Dynamic-CBT–Better Performing Active Queue Management for Multimedia Networking," in *International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, 2000.

[15] M. S. Taqqu, W. Willinger, and R. Sherman, "Proof of a fundamental result in self-similar traffic modeling," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 5–23, 1997.