



ADAPTIVE FLOW SAMPLING FOR SELF-SIMILAR TRAFFIC MEASUREMENT

Altyeb Altaher Altyeb^a, Sami Mohamed Sharif^b, Iman Abuel Maaly^b and Hassan Ibrahim Saleh^a

^a Alguryat College –King Saud University –Kingdom of Saudi Arabia
altypaltaher@yahoo.com

^b Department of Electrical and Electronic Engineering-University of Khartoum-Sudan

Abstract

Previous studies of Internet traffic have shown that a very small percentage of flows consume most of the network bandwidth. It is important to take such flows into account when measuring the network traffic self-similarity. This paper proposes a traffic measurement algorithm for self similar traffic: Adaptive sampling algorithm. This algorithm uses multistage filters and adaptation mechanism to concentrate on large flows. Using analysis and simulation, this algorithm is shown to effectively reflect the bursts of self-similar traffic over multiple time scales and can improve accuracy when network flows are self-similar.

Keywords Traffic measurement ,network traffic self-similarity, traffic sampling .

1. Introduction

Recent works prove that packet network traffic shows self-similarity [1], [2], [3], [4], [5], and it has more different theoretical properties than classical queuing analysis [3]. For self-similar traffic, traffic bursts appear on a wide range of time scales. This existence of self-similarity in network traffic has led to new perspectives and raised questions regarding traffic measurement [5]. Self-similarity has great influence on network performance. As for LRD (long-range dependence) traffic, increasing buffer can smooth the bursts of traffic to certain extent. But the bursting behavior in large-time scales of network traffic requires a great deal of buffer. Therefore, increasing buffer blindly is costly and unrealistic [5], [6].

NetFlow ,first implemented in Cisco routers ,is the most widely used flow measurement solution today .NetFlow which counts periodically sampled packets, is slow, inaccurate and resource intensive [7].

The main problem with flow measurement approach is its lack of scalability. Flow measurement in [9] shows that the number of flows between end host pairs in a 1 hour period to be as high as 1.7 million and 0.8 million. The numbers of flows in real network are very large; it's unfeasible for flow measurement devices to keep up with

the increases in the number of flows. Therefore, traffic collection algorithms have to focus on the largest flow. Despite the large number of flows, a common observation found in many measurement studies is that a small percentage of flows account for a large share of the traffic [7].

In contrast to traditional sampling algorithms, MF (multistage filters) algorithm is more accurate while using much less memory [8], [10]. Based on the main idea of MF algorithm, a novel and scalable traffic collection algorithm is proposed for self-similar traffic: adaptive multistage filter algorithm, which can reflect the bursts of self-similar traffic on multiple time scales. The adaptation is related to the throughput of a flow.

The rest of the paper is organized as follows: in section 2, the definition and measurement of self-similar process are shown. The MF algorithm and the Adaptive-MF algorithm are adopted and described in section 3. Computer simulation and results are presented in section 4. Finally, concluding remarks are given in section 5.

2. Definition And Measurement Of Traffic Self-Similarity

A. Definition of Traffic Self-Similarity

Considering a wide range of stochastic process $X(t)$, stochastic variance X is self-similar with the Hurst parameter $H(0.5 \leq H \leq 1)$ in the sense of statistics, if for all $\alpha > 0$ and $t \geq 0$,

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

$\stackrel{d}{=}$ denotes equivalence in the sense of finite dimensional distributions.

The original series $X(i)$ is divided into blocks of size m and the aggregated series $X^{(m)}(k)$ is calculated as:

$$X^{(m)}(k) = \frac{1}{m} (X_{km-m+1} + \dots + X_{km}), (k = 0, 1, 2, \dots) \quad (2)$$

Let r^m denote the autocorrelation function of $X^{(m)}$ and $D[X^{(m)}]$ denote the variance.

Assume the autocorrelation function of $X^{(m)}$ has form $r(k) \sim k^{-\beta} L_1(k)$, when $k \rightarrow \infty$ with $0 < \beta < 1$ where L_1 satisfies

$$\lim_{t \rightarrow \infty} L_1(tx)/L_1(t) = 1 \quad \text{for all } x > 0.$$

Definition 1: X is exactly second-order self-similar with autocorrelation parameter $H = 1 - \beta/2$, if for All m

$$D[X^{(m)}] \approx D[X]/m^\beta \quad (3)$$

$$r^{(m)} = r(k) \quad (4)$$

Definition 2: X is asymptotically second-order self-similar with autocorrelation parameter $H = 1 - \beta/2$,

When $m \rightarrow \infty, k = 1, 2, 3, \dots$

$$D[X^{(m)}] \sim D[X]/m^\beta \quad (5)$$

$$r^{(m)} \sim r(k) \quad (6)$$

B. Traffic Self-Similarity Measurement

Rescaled Adjusted Range Statistic (also called the R/S statistic) is used to measure the degree of Traffic self-similarity. The Hurst parameter is an important measure of traffic self-similarity. If $H = 1/2$ then a data set has independent data. If $H > 1/2$ then the data set has self-similarity.

Historically, self-similar processes are marked because these processes provide an elegant interpretation of the empirical phenomenon. Given a series of observations $X_1, X_2, X_3, \dots, X_n$ with Sample mean:

$$\bar{x}(n) = \frac{1}{n} \sum_{i=1}^n x_i \quad (7)$$

sample variance

$$s^2(n) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}_n)^2 \quad (8)$$

and
$$l_i(n) = \sum_{k=1}^i x_k - \bar{x}_n \quad (9)$$

the rescaled adjusted range (the R/S statistics) is defined as

$$\frac{R(n)}{S(n)} = \frac{\max_{1 \leq r \leq n} l_i(n) - \min_{1 \leq r \leq n} l_i(n)}{S(n)} \quad (10)$$

The Hurst parameter H is given by the equation $R(n)/S(n) \sim cn^H$

$$(11)$$

Where c is a finite positive constant and Hurst parameter is in the range of $(0.5, 1)$.

3. Adaptive Multistage Filters

A. Multistage filters algorithm

The main idea of multistage filters algorithm is utilizing multiple stages in identifying large flows in network [8], [10]. MF algorithm uses only a few memory references per packet, making it suitable for use in high speed routers. MF algorithm produces more accurate estimates than NetFlow, but they do processing and access memory for each packet [8]. A flow ID is firstly defined as below:

Definition 1: A flow ID is a formed as (S, R) , with S denoting source node and R denoting destination node. A typical MF algorithm presented in Fig.1.

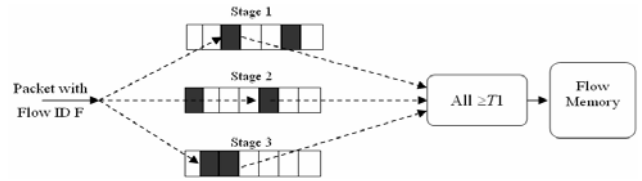


Figure 1. A typical MF algorithm

MF algorithm identifies flows by multistage. A stage is a table of buckets that are indexed by a hash function computed on a packet flow ID. When a packet comes in, a hash on its flow ID is computed and the size of packet is added to the corresponding bucket. Since each stage uses independent hash function to compute the index, the same flow ID will have great probability to stay in distinct buckets.

MF algorithm uses filter intervals in identifying large flows. A filter interval is a continued time period. After each filter interval, the MF algorithm will select candidate large flows. During the consequent measurement period, this algorithm will focus only on the selected flows. After the filter interval, MF algorithm scans all the stages.

A flow F can be accepted only if its sizes in bucket of each stage all exceed threshold T . Since the number of buckets in a stage that can be afforded is significantly smaller than the number of flows in network [9], many flows will map to the same buckets. Multistage filters algorithm can reduce the following problems brought out by traditional sampling algorithms: first, small flows can map to buckets that hold large flows and get added to flow memory; second, several small flows can hash to the same bucket and add up to a number larger than the threshold.

If M is the available memory, the error-ratio of MF algorithm is proportional to $1/M$, contrast to the error of an algorithm based on classical sampling is proportional to $1/\sqrt{M}$, thus providing much less accuracy for the same amount of memory [8].

B. Adaptive -MF algorithm

According to [1], [2], if the lived time of a flow is longer, it will have greater probability to live during the last time in self-similar conditions. [7] and [9] show that 9% of the flows between AS pairs account for 90% of the byte traffic between all AS pairs. What's more, the largest percentage account for occupying the bandwidth is the flows whose throughput is much higher. Based on multistage, adaptive-MF algorithm takes throughput of flows into account. Size of a flow may seem small in a special interval, but for the particularity of self-similar traffic, if its throughput is comparatively larger, it will have greater probability to live during the last measurement time. Hence, the amount sizes of this flow will possibly be much larger during the whole measurement time.

The Adaptive-MF algorithm is defined as bellow:

Definition 2: Let $t_i (1 \leq i \leq M)$ denote the time interval, where M is the total number of time intervals in a trace. For all i , the length of t_i is the same time interval τ . For each t_i , we define variable $x(t_i)$ as the throughput in t_i . It is calculated with formula $x(t_i) = b(t_i)/\tau$, where

b(t_i) is the total number of bits sent during the time interval t_i. It is found that there are throughput variables x(t_i), whose values are extremely large. These extremely large throughput values may cause severe packet loss or long queuing delay at routers. That is, they are strongly related to network performance. Therefore, it is meaningful to characterize them.

Definition 3 :

The throughput variable $\hat{x}(k)$ is called an extremely large throughput variable if it satisfies

$$\hat{x}(k) \geq \bar{x}(t_i) + 2\sigma_x \quad (10)$$

Where $\bar{x}(t_i)$ stands for the mean of $x(t_i)$, σ_x is the standard deviation of $x(t_i)$.

We set a mean value plus two standard deviation for the threshold value of large throughput variables. Thus $\hat{x}(k)$ can be viewed as threshold and Adaptive-MF algorithm is degenerated to MF algorithm.

The detailed description about Adaptive-MF algorithm is presented as below:

- (1) Sizes of all buckets in each stage are initialized to 0 at the start of a filter interval;
- (2) When a packet comes on, mark this packet by flow ID;
- (3) Calculate packet's position in each bucket by independent hash functions;
- (4) Add the size of this packet to each bucket;
- (5) Update the lived time of each bucket;
- (6) Is filter interval terminating? No, repeat step 2-5;
- (7) Scan all stage indices by flow ID, check if exist a packet, and its throughput larger than $\hat{x}(k)$?

If it's true, add the flow to flow memory.

During the following measurement time, Adaptive-MF algorithm will only focus on these selected flows.

- (8) Collect traffic information on a continued time;
- (9) Restarting; a new filter interval.

4. Simulations Results

Simulations are conducted to investigate the conservation of the self-similarity characteristic while using the Adaptive multistage filters method. Simulations investigate the accuracy of the prediction on sampled traffic represented by traffic throughput rates over different sampling time intervals. We study real Ethernet traffic trace that is part of a data set collected at Bellcore in August of 1989. The Ethernet traffic trace correspond to one "normal" hour's worth of traffic, collected every 10 milliseconds, thus resulting in a length of 360,000. The data set measures the number of packets per unit time. This data set has been widely used and was first analyzed in [1].

In this study, the Rescaled Adjusted Range Statistic (also called the R/S statistic) is used to measure the degree of Traffic self-similarity. First, we set the sampling interval size $\tau = 10, 50, 100, 200, 300, 400, 500, 600, 700$. Using the Rescaled Adjusted Range Statistic method, we calculated the Hurst parameter of these processes to

verify the conservation of the self-similarity characteristic.

Fig.2 shows the obtained values of Hurst parameter calculated for different sampling interval size τ considering packets traffic traces. It is clearly shown that the self-similarity property is preserved for different sampling interval size τ . Moreover, the Hurst parameter remains fairly the same as the original sampled traffic trace. The second set of experiments investigates the LMMSE-based prediction accuracy performed on the sampled packets. Fig. 3 shows the obtained prediction error for different sampling interval size τ considering packets traffic trace. The prediction error remains fairly steady for different sampling interval size τ .

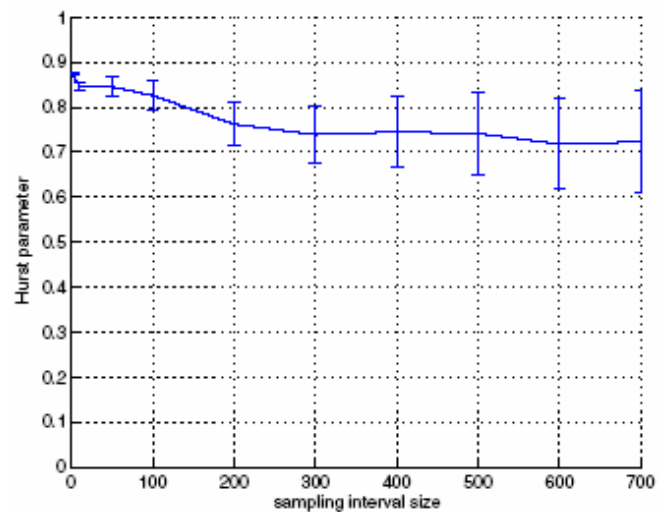


Figure 2. Hurst parameter for packets trace

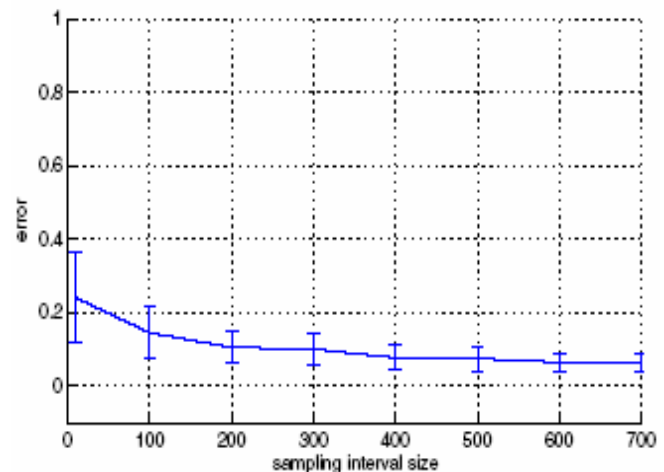


Figure 3. Prediction error for packets trace

5. Conclusion

This paper presents a traffic collection algorithm for self-similar traffic, which considers the problem of directly identifying the large flows without keeping track of potentially millions of small flows. The Adaptive-MF algorithm keeps the advantages of small memory usage of MF algorithm. Further more, Adaptive-MF algorithm reflects the bursts of self-similar traffic over multiple time scales by computing throughput threshold. This improves the accuracy of traffic collection algorithm. What's more, this algorithm also remains quite stable in

large time scales. There is an important practical implication of our study, that the measurement problems mentioned by this paper also are similar to the measurement problems faced by other areas, such as data mining, internet searching.

Acknowledgment

We would like to thank Dr Satam Alshamri and Dr Moawia Alfaki for their support and encouragement.

References

- [1] W. Leland, M. S. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic (extended version)", IEEE/ACM Trans. Networking, vol. 2, no. 1, pp. 1-15, 1994.
- [2] Zhang. Z. L., Ribeiro. V. J., Moon. S., et al. "Small-Time Scaling Behaviors of Internet Backbone Traffic: An Empirical Study", Proceeding of IEEE INFOCOM'03, San Francisco, 2003.
<http://citeseer.ist.psu.edu/zhang03smalltime.htm>
- [3] Paxson V., Royd S. "Wide-area traffk: The failure of Poisson modeling", IEEE/ACM Transactions on Networking, Vol3, pp. 226-244,1995.
- [4] Stephen. D., Ravi. K., Kevin. S., er al. "Self-similarity in the Web", ACM Transactions on Internet Technology, Vol2, No. 3, pp. 205-223,2002.
- [5] Likhanov. N., Tsybakov. B., Georans. N. D., "Analysis of an ATM buffer with self-similar ("fractal") input traffk", Proceeding of IEEE INFOCOM'95, Boston, pp. 385-992, 1995.
- [6] Norros. I. "A storage model with self-similar input", Queuing Systems, Vol 16, pp. 387-396, 1994.
- [7] Feldmann. A., Greenberg. A., Lund. C., et al. "Deriving traffic demands for operational IP networks: Methodology and experience", Proceeding of ACM SIGCOMM'00, Stockholm, pp. 257-270,2000.
- [8] Cristian. E. and George. V. "New,. Directions in Traffic Measurement and Accounting: Focusing on the Elephants, Ignoring the Mice", ACM Transactions on Computer Systems, Vol 21, No. 3, pp. 270-313, 2002.
- [9] Fang. W. and Peterson. L. "Inter-as traffic patterns and their implications", Proceeding of IEEE GLOBECOM'99, Rio de Janeiro, pp. 1859-1868, December 1999
- [10] Feng. W. C., Kandlur. D. D., Saha. D., et al. "Stochastic fair blue: A queue management algorithm for ,enforcing fairness" Proceeding of IEEE INFOCOM'01, Anchorage Alaska, pp. * 1229-1520, April 2001.