

An Adaptive Sampling Algorithm with Applications to Denial-of-Service Attack Detection

Animesh Patcha and Jung-Min Park
Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061
Email: {apatcha, jungmin}@vt.edu

Abstract—There is an emerging need for the traffic processing capability of network security mechanisms, such as intrusion detection systems (IDS), to match the high throughput of today’s high-bandwidth networks. Recent research has shown that the vast majority of security solutions deployed today are inadequate for processing traffic at a sufficiently high rate to keep pace with the network’s bandwidth. To alleviate this problem, packet sampling schemes at the front end of network monitoring systems (such as an IDS) have been proposed. However, existing sampling algorithms are poorly suited for this task especially because they are unable to adapt to the trends in network traffic. Satisfying such a criterion requires a sampling algorithm to be capable of controlling its sampling rate to provide sufficient accuracy at minimal overhead. To meet this utopian goal, adaptive sampling algorithms have been proposed. In this paper, we put forth an adaptive sampling algorithm based on weighted least squares prediction. The proposed sampling algorithm is tailored to enhance the capability of network based IDS at detecting denial-of-service (DoS) attacks. Not only does the algorithm adaptively reduce the volume of data that would be analyzed by an IDS, but it also maintains the intrinsic self-similar characteristic of network traffic. The latter characteristic of the algorithm can be used by an IDS to detect DoS attacks by using the fact that a change in the self-similarity of network traffic is a known indicator of a DoS attack.

I. INTRODUCTION

The Internet today continues to grow and evolve as a global infrastructure for new services. Business needs have dictated that corporations and governments across the globe should develop sophisticated, complex information networks, incorporating technologies as diverse as distributed data storage systems, encryption and authentication mechanisms, voice and video over IP, remote and wireless access, and web services. As a result, Internet service providers and network managers in corporate networks are being motivated to gain a deeper understanding of the network behavior through monitoring and measurement of the network traffic flowing through their networks.

Network-based security systems, like intrusion detection systems (IDS), have not kept pace with the increasing usage of high-speed networking technologies such as Gigabit Ethernet. The repeated occurrences of large-scale attacks (such as distributed denial-of-service (DDoS) attacks and worms) that exploit the bandwidth and connectivity of networks made possible by such technologies are a case in point.

The single biggest reason that can be attributed to the incapability of current solutions to detect intrusions in high-speed networks is the prohibitively high cost of using tra-

ditional network monitoring schemes like host and router based monitoring solutions. These schemes typically measure network parameters of every packet that passes through a network device. This approach has the drawback that it becomes extremely difficult to monitor the behavior of a large number of sessions in high-speed networks. In other words, traditional network monitoring schemes are not *scalable* to high-speed networks.

To alleviate the aforementioned problem, sampling algorithms have been proposed. Over the years, network managers have predominantly relied on static sampling algorithms for network monitoring and management. In general, these sampling algorithms employ a strategy where the samples are taken either randomly or periodically at some fixed interval. The major advantage of using a sampling algorithm is that it reduces bandwidth and storage requirements. Traditional sampling algorithms typically use a static or fixed rule to determine when to sample the next data. Static sampling of network traffic was first proposed by Claffy et al. [1] in the early 1990’s for traffic measurement on the NSFNET backbone. In their much cited paper, Claffy et al. describe the application of event and timed-based sampling for network traffic measurement.

Static sampling algorithms like *simple random sampling* employ a random distribution function to determine when each sample should be taken. The distribution may be uniform, exponential, Poisson, etc. In random sampling, all items have some chance of selection that can be calculated. The advantage of utilizing a random sampling algorithm is that it ensures that bias is not introduced regarding which entity is included in the sampled population.

However, given the dynamic nature of network traffic, static sampling does not always ensure the accuracy of estimation, and tends to over sample at peak periods when efficiency and timeliness are most critical. More generally, static random sampling algorithms do not take into account traffic dynamics. As a result, they cannot guarantee that the sampling error in each block falls within a prescribed error tolerance level.

In the commercial world, *NetFlow* [2] is a widely deployed general purpose measurement feature of Cisco and Juniper routers. The volume of data produced by NetFlow is a problem in itself. To handle the volume and traffic diversity of high speed backbone links, NetFlow resorts to 1 in N packet sampling. The sampling rate is a configuration parameter that is set manually and is seldom adjusted. Setting it too low,

causes inaccurate measurement results; setting it too high can result in the measurement module using too much memory and processing power, especially when faced with increased volume of traffic or unusual traffic patterns.

Under dynamic traffic load conditions, simple periodic sampling may be poorly suited for network monitoring. During periods of idle activity or low network loads, a long sampling interval provides sufficient accuracy at a minimal overhead. However, bursts of high activity require shorter sampling intervals to accurately measure network status at the expense of increased sampling overhead. To address this issue, *adaptive sampling* algorithms have been proposed to dynamically adjust the sampling interval and optimize accuracy and overhead.

In this paper, we put forth an adaptive sampling algorithm that is based on *weighted least squares prediction*. The proposed sampling algorithm uses previous samples to estimate or predict a future measurement. The algorithm is used in conjunction with a set of rules which defines the sampling rate adjustments that need to be made when a prediction is inaccurate. To gauge the performance of the proposed sampling algorithm, we compared it with *simple random sampling* where the samples are taken at time intervals determined by a random distribution.

A. Motivation

The growth of the Internet and the advances in networking technologies have also brought about unwanted side effects: the proliferation of network-based attacks and cyber crime [3]. However, as pointed out above, current security mechanisms especially in the domain of attack detection have not scaled to handle the higher network throughputs.

Several approaches for either sampling or attack detection have been proposed in the research community. However, to the best of our knowledge, none of the proposed algorithms for network traffic sampling have taken an approach that is tailored to meet the needs of attack detection. From this perspective we attempt to answer one key question in this paper: Is it possible to design a low cost packet sampling algorithm that will enable accurate characterization of the IP traffic variability for the purpose of detecting DoS attacks in high throughput networks?

We believe that the proposed sampling algorithm is tailored to enhance the capability of network-based IDS at detecting a DoS attack. The proposed sampling algorithm would ideally precede the IDS and sample the incoming network traffic. The key characteristic is that it adaptively reduces the volume of data that would be analyzed by the network IDS, and also preserves the intrinsic self-similar characteristic of network traffic. We believe the latter characteristic of the proposed sampling algorithm can be used by an IDS to detect traffic intensive DoS attacks by leveraging on the fact that a significant change in the self-similarity (See Appendix A for details on self-similarity) of network traffic is a known indicator of a DoS attack [4], [5].

This paper is organized as follows. In Section II, we review the related work in the area of packet sampling. Section III presents the weighted least square predictor and the proposed adaptive weighted sampling algorithm. In Section IV, we

describe the simulation results and compare the performance of the proposed sampling algorithm with simple random sampling. In Section V we conclude the paper by summarizing the paper's contributions and suggesting possible areas for the application of the proposed sampling algorithm.

II. RELATED WORK

The biggest challenge in employing a sampling algorithm on a given network is scalability. The increasing deployment of high-speed networks, the inherently bursty nature of Internet traffic, and the storage requirements of large volume of sampled traffic have a major impact on the scalability of a sampling algorithm. In the context of packet sampling, this implies that either the selected sampling strategy should take into account the trends in network traffic or the selected sampling algorithm should sample most if not all the packets that are flowing through the network. The major impediment towards adopting the latter approach is that a higher sampling rate would imply greater memory and space requirements for the sampling device. In addition, a higher sampling rate would run the risk of not being scalable to high-speed networks.

Packet sampling has been previously proposed for a variety of objectives in the domain of computer networking. Sampling network traffic was advocated as early as 1994. As mentioned above, Claffy et al. [1] compared three different sampling strategies to reduce the load on the network parameter measurement infrastructure on the NSFNET backbone. The three algorithms studied in [1] were, systematic sampling (deterministically taking one in every N packets), stratified random sampling (taking one packet in every bucket of size N), and simple random sampling (randomly taking N packets out of the whole set). The results showed that event-driven algorithms were more accurate than time-driven ones, while the differences within each class were small. This was attributed to trends in network traffic.

Drobisz et al. [6] proposed a rate adaptive sampling algorithm to optimize the resource usage in routers. The authors proposed using the packet inter-arrival rates and CPU usage as the two methodologies to control resource usage and vary the sampling rate. They also showed that adaptive algorithms produced more accurate estimates than static sampling under a given resource constraint. In another paper, Cozzani et al. [7] used the simple random sampling algorithm to evaluate the ATM end-to-end delays. In the SRED scheme in [8], Ott et al. use packet sampling to estimate the number of active TCP flows in order to stabilize network buffer occupancy for TCP traffic. The advantage of this scheme is that only packet headers need to be examined.

Another approach taken by Estan and Varghese [9], involved a random sampling algorithm to identify large flows. In the algorithm, proposed in [9], the sampling probability is determined according to the inspected packet size. In another study, Cheng et al. [10] proposed a random sampling scheme based on Poisson sampling to select a sample that is representative of the whole dataset. The contend that using Poisson sampling is better as it does not require the packet arrival to conform to a particular stochastic distribution. Sampling strategies were also used in [11] for the detection of DoS attacks. Sampling

has also been proposed to infer network traffic and routing characteristics [12]. In [13], Duffield et al. focused on the issue of reducing the bandwidth needed for transmitting traffic measurements to a remote server for later analysis, and devised a size-dependent flow sampling algorithm. In another paper, Duffield et al. [14] investigated the consequences of collecting packet sampled flow statistics. They found that flows in the original stream whose length is greater than the sampling period tend to give rise to multiple flow reports when the packet inter arrival time in the sampled stream exceeds the flow timeout.

Sampling for intrusion detection entails a more thorough examination of the sampled packets. In addition, unlike some of the sampling applications mentioned above, sampling for intrusion detection and more specifically for anomaly detection requires near line-speed packet examination. This is especially because a *store-and-process* approach towards sampled packets or packet-headers for off-line analysis is not sufficient to prevent intruders. Hence, in the design of an intrusion detection algorithm, sampling costs are of paramount importance.

III. THE PROPOSED SAMPLING ALGORITHM

Traffic measurement and monitoring serves as the basis for a wide range of IP network operations and engineering tasks such as trouble shooting, accounting and usage profiling, routing weight configuration, load balancing, capacity planning, etc. Traditionally, traffic measurement and monitoring is done by capturing every packet traversing a router interface or a link. With today's high-speed (e.g., Gigabit or Terabit) links, such an approach is no longer feasible due to the excessive overheads it incurs on line-cards or routers. As a result, packet sampling has been suggested as a scalable alternative to address this problem.

Early packet sampling algorithms assumed that the rate of arrival of packets in a network would average out in the long term. However, it has been shown [15] that network traffic exhibits periodic cycles or trends. The main observation of [15] and other studies have been that not only does network traffic exhibit strong trends in the audit data but these trends also tend to be long term.

This section presents the proposed sampling algorithm. In Section III-A, we describe the *weighted least squares* predictor that is utilized for predicting the next sampling interval. This predictor has been adopted because of its capability to follow the trends in network traffic. Thereafter, in Section III-B we describe the sampling algorithm itself.

A. Weighted Least Square Predictor

Let us assume that the vector \mathbf{Z} holds the values of the N previous samples, such that Z_N is the most recent sample and Z_1 is the oldest sample. Having fixed a window size of N , when the next sampling occurs, the vector is right shifted such that Z_N replaces Z_{N-1} and Z_1 is discarded. The weighted prediction model therefore predicts the value of Z_N given Z_{N-1}, \dots, Z_1 . In general, we can express this predicted value as a function of the N past samples i.e.,

$$\hat{Z}_N = \alpha^T \tilde{\mathbf{Z}}, \quad (1)$$

where \hat{Z}_N is the new predicted value, $\tilde{\mathbf{Z}}$ is the vector of past $N - 1$ samples, and α^T is a vector of predictor coefficients distributed such that newer values have a greater impact on the predicted value \hat{Z}_N . A second vector, \mathbf{t} , records the time that each sample is taken and is shifted in the same manner as \mathbf{Z} . The objective of the weighted prediction algorithm is to find an appropriate coefficient vector, α^T , such that the following summation is minimized

$$S = \sum_{i=1}^{N-1} w_i \left(Z_i - \hat{Z}_i \right)^2, \quad (2)$$

where w_i , Z_i , and \hat{Z}_i denote the weight, the actual sampled value, and the predicted value in the i^{th} interval, respectively.

The coefficient vector is given by:

$$\alpha^T = \left(\tilde{\mathbf{Z}}^T \mathbf{W} \tilde{\mathbf{Z}} \right)^{-1} \tilde{\mathbf{Z}}^T \mathbf{W} \mathbf{w}, \quad (3)$$

where $\mathbf{W} = \mathbf{w}^T \mathbf{w}$ is a $(N - 1) \times (N - 1)$ diagonal weight matrix and \mathbf{w} is a $N \times 1$ weight vector with weight coefficient's w_i that are determined according to two criteria:

- 1) The "freshness" of the past $N - 1$ samples. A more recent sample has a greater weight.
- 2) The similarity between the predicted value at the beginning of the time interval and the actual value. The similarity between the two values is measured by the distance between them. The smaller the Euclidean distance is, the more similar they are to each other.

Based on the above two criteria, we define a weight coefficient as

$$w_i = \frac{1}{(t_N - t_i)} \left(\frac{1}{|Z_i - \hat{Z}_i|^2 + \eta} \right), 1 \leq i \leq N - 1, \quad (4)$$

where η is a quantity introduced to avoid division by zero.

B. Adaptive Weighted Sampling

Adaptive sampling algorithms dynamically adjust the sampling rate based on the observed sampled data. A key element in adaptive sampling is the prediction of future behavior based on the observed samples. The weighted sampling algorithm described in this section utilizes the weighted least squares predictor (see section III-A) to select the next sampling interval. Inaccurate predictions by the weighted least squares predictor indicates a change in the network traffic behavior and requires a change in the sampling rate.

The proposed adaptive sampling algorithm consists of the following steps (see Fig. 1):

- 1) Fix the first N sampling intervals equal to τ . (In our simulations we used $\tau = 60$ sec. and $N = 10$)
- 2) Apply the weighted least squares predictor to predict the anticipated value, \hat{Z}_N , of the network parameter.
- 3) Calculate the network parameter value at the end of the sampling time period.
- 4) Compare the predicted value with the actual value.

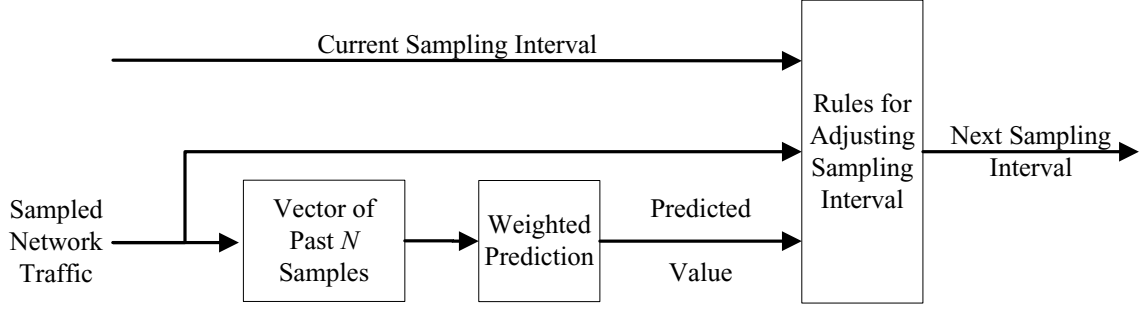


Fig. 1: Block diagram of the adaptive sampling algorithm

- 5) Adjust sampling rate according to the predefined rule set if the predicted value differs from the actual value

The predicted output \hat{Z}_N which has been derived from the previous N samples, is then compared with the actual value of the sample, Z_N . A set of rules is applied to adjust the current sampling interval, $\Delta T_{Curr} = t_N - t_{N-1}$, to a new value, ΔT_{New} , which is used to schedule the sampling query. The rules used to adjust the sampling interval compare the rate of change in the predicted sample value, $\hat{Z}_N - Z_{N-1}$, to the actual rate of change, $Z_N - Z_{N-1}$. The ratio, R , between the two rates is defined as:

$$R = \left| \frac{\hat{Z}_N - Z_{N-1}}{Z_N - Z_{N-1}} \right|. \quad (5)$$

Based on the value of R , which ranges from R_{MIN} to R_{MAX} ¹, we define the next sampling interval ΔT_{New} as shown in Equation (6). The variables β_1 and β_2 , in Equation 6, are tunable parameters. When determining the values for β_1 and β_2 , one needs to consider the rate of change of the network parameter under consideration. As in [16], we used the values $\beta_1 = 2$ and $\beta_2 = 2$ in our simulations.

$$\Delta T_{New} = \begin{cases} (1 + R) \times \Delta T_{Curr} & \text{if } R > R_{MAX} \\ \beta_1 \times \Delta T_{Curr} & \text{if } R_{MIN} < R < R_{MAX} \\ R \times \Delta T_{Curr} & \text{if } R < R_{MIN} \\ \beta_2 \times \Delta T_{Curr} & \text{if } R \text{ is Undefined} \end{cases} \quad (6)$$

The value of R is equal to 1 when the predicted behavior is the same as the observed behavior. If the value of R is greater than R_{MAX} , it implies that the measured value is changing more slowly than the predicted value and this means that the sampling interval needs to be increased. On the other hand, if R is less than R_{min} , it implies that the measured value of the network parameter is changing faster than the predicted value. This indicates more network activity than predicted, so the sampling interval should be decreased to yield more accurate values for future predictions of the network

¹Based on the results obtained from simulations performed by us, we selected a value of $R_{MIN} = 0.82$ and $R_{MAX} = 1.21$. These values were selected because they provided good performance over a wide range of traffic types.

parameter. The value of R may be undefined. This case arises when both the numerator and denominator of Equation (5) are zero. This condition is generally indicative of an idle network or a network in steady state. In such a scenario, the sampling interval is increased by a factor of $\beta_2 (> 1)$.

IV. SIMULATION RESULTS

Simulations were conducted to evaluate the performance of the proposed adaptive sampling algorithm. We evaluated the proposed sampling algorithm using data from the Widely Integrated Distributed Environment (WIDE) project [17]. The WIDE backbone network consists of links of various speeds, from 2Mbps CBR (Constant Bit Rate) ATM up to 10 Gbps Ethernet. The WIDE dataset we analyzed consisted of a 24-hour trace that was collected on September 22, 2005.

When comparing the performance of the proposed adaptive sampling algorithm with the simple random sampling algorithm, a useful criterion to use is the mean square error (MSE) of the estimate or its square root, the root mean squared error, measured from the population that is being estimated. Formally we can define the mean square error of an estimator X of an unobservable parameter θ as $MSE(X) = E[(X - \theta)^2]$. The root mean square error is the square root of the mean square error and the root mean square error is minimized when $\theta = E(X)$ and the minimum value is the standard deviation of X .

In Fig. 2, we compare the proposed adaptive sampling scheme with the simple random sampling algorithm using the standard deviation of packet delay as the comparison criterion. Packet delay is an important criterion for detecting DoS attacks, especially attacks that focus on degrading the quality of service in IP networks [18]. The results show that over different block sizes, the proposed adaptive scheme has a lower standard deviation when compared with the simple random sampling algorithm. Since standard deviation is directly proportional to the root mean square error criterion, this implies that the proposed algorithm predicts the packet mean delay better than the simple random sampling algorithm while reducing the volume of traffic.

In the second set of experiments, we verified whether the traffic data sampled by the proposed sampling scheme has the self similar property. For this verification, we used two

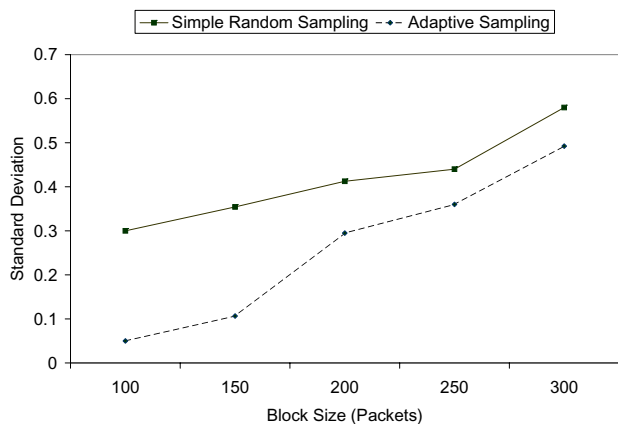


Fig. 2: Standard deviation of packet delay.

different parameters: the mean of the packet count and the Hurst parameter. The peak-to-mean ratio (PMR) can be used as an indicator of traffic burstiness. PMR is calculated by comparing the peak value of the measure entity with the average value from the population. However, this statistic is heavily dependent on the size of the intervals, and therefore may or may not represent the actual traffic characteristic. A more accurate indicator of traffic burstiness is given by the Hurst parameter (See Appendix A for details).

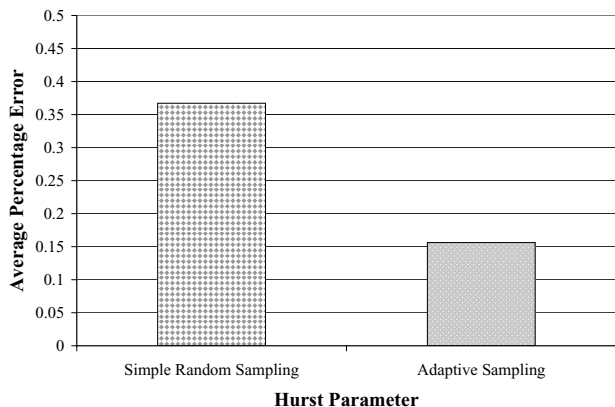


Fig. 3: Average percentage error for the Hurst parameter.

Fig. 3 and Fig. 4 show the average sampling error for the Hurst parameter and the sample mean, respectively. As one can see from Fig. 3, the random sampling algorithm resulted in higher average percent error for the Hurst parameter when compared to adaptive sampling. This could be the result of missing data spread out over a number of sampling intervals. In Fig. 4, the average percentage error for the mean statistic was marginally higher for our sampling algorithm when compared with the simple random sampling algorithm, albeit the difference was insignificant. One possible reason for this marginal difference is the inherent adaptive nature of our sampling algorithm—i.e., the proposed sampling algorithm is more likely to miss short bursts of high network activity in periods that typically have low network traffic. The simple

random sampling scheme would be less likely to have the same problem.

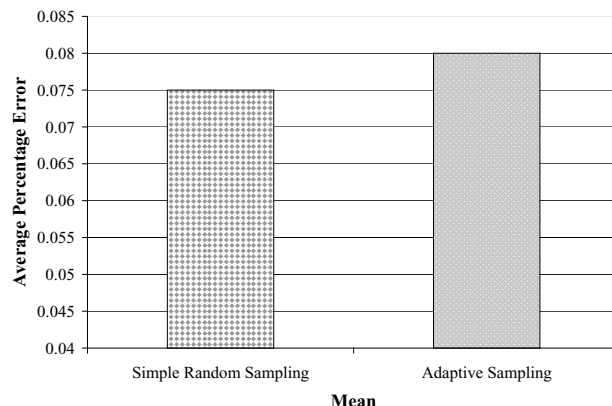


Fig. 4: Average percentage error for the mean statistic.

V. CONCLUSION

In this paper, we have presented an adaptive sampling algorithm which uses weighted least squares prediction to dynamically alter the sampling rate based on the accuracy of the predictions. Our results have shown that compared to simple random sampling, the proposed adaptive sampling algorithm performs well on random, bursty data. Our simulation results show that the proposed sampling scheme is effective in reducing the volume of sampled data while retaining the intrinsic characteristics of the network traffic.

We believe that the proposed adaptive sampling scheme can be used for a variety of applications in the domain of network monitoring and network security. The variations in the self similarity and long range dependence of network traffic are known indicators of a denial-of-service attack [5]. Therefore, an anomaly detection scheme could successfully use the proposed sampling algorithm to sample and reduce the volume of inspected traffic while still being able to detect minor variations in the self-similarity and long range dependence of network traffic.

REFERENCES

- [1] K. C. Claffy, G. C. Polyzos, and H.-W. Braun, "Application of sampling methodologies to network traffic characterization," in *SIGCOMM '93: Proceedings of the Conference on Communications architectures, protocols and applications*, (New York, NY, USA), pp. 194–203, ACM Press, 1993.
- [2] C. NetFlow, "CISCO NetFlow." http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html.
- [3] E. Millard, "Internet attacks increase in number, severity." http://www.toptechnews.com/news/Internet-Attacks-Increase-in-Severity/story.xhtml?story_id=0020007B77EI, 2005.
- [4] M. Li, W. Jia, and W. Zhao, "Decision analysis of network based intrusion detection systems for denial-of-service attacks," in *Proceedings of the IEEE Conferences on Info-tech and Info-net*, vol. 5, Dept. of Computer Sci., City Univ. of Hong Kong, China, IEEE, October 2001.
- [5] P. Owezarski, "On the impact of DoS attacks on internet traffic characteristics and QoS," in *ICCCN '05: Proceedings of the 14th International Conference on Computer Communications and Networks*, pp. 269–274, LAAS-CNRS, Toulouse, France, IEEE, October 2005.

- [6] J. Drobisz and K. J. Christensen, "Adaptive sampling methods to determine network traffic statistics including the hurst parameter," in *IEEE LCN '98: Proceedings of the IEEE Annual Conference on Local Computer Networks*, pp. 238–247, IEEE, 1998.
- [7] I. Cozzani and S. Giordano, "A measurement based qos evaluation through traffic sampling," in *SICON '98: Proceedings of the 6th IEEE Singapore International Conference on Networks (SICON)*, (Singapore), IEEE, June 30–July 3 1998.
- [8] T. J. Ott, T. Lakshman, and L. Wong, "Sred: Stabilized red," in *INFOCOM '99: Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, (New York, NY), pp. 1346–1355, Bellcore, USA, IEEE, March 1999.
- [9] C. Estan and G. Varghese, "New directions in traffic measurement and accounting," in *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 323–336, ACM Press, 2002.
- [10] G. Cheng and J. Gong, "Traffic behavior analysis with poisson sampling on high-speed network," in *ICII '01: Proceedings of the International Conferences on Info-tech and Info-net, 2001*, vol. 5, (Beijing, China), pp. 158–163, Computer Science Dept., Southeast Univ., Nanjing, China, IEEE, 29 Oct.–1 Nov 2001.
- [11] Y. Huang and J. M. Pullen, "Countering denial-of-service attacks using congestion triggered packet sampling and filtering," in *ICCCN '01: Proceedings of the Tenth International Conference on Computer Communications and Networks*, (Scottsdale, AZ), pp. 490–494, Dept. of Comput. Sci., George Mason Univ., Fairfax, VA, USA, IEEE, October 2001.
- [12] N. Duffield, C. Lund, and M. Thorup, "Properties and prediction of flow statistics from sampled packet streams," in *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, (New York, NY, USA), pp. 159–171, ACM Press, 2002.
- [13] N. Duffield, A. Greenberg, and M. Grossglauser, "A framework for passive packet measurement," Internet Draft draftduffield-framework-papame-01, IETF, February.
- [14] N. Duffield, C. Lund, and M. Thorup, "Charging from sampled network usage," in *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pp. 245–256, November 2001.
- [15] D. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot, "Long-term forecasting of internet backbone traffic: Observations and initial models," in *INFOCOM '03: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, (Burlingame, CA, USA), pp. 1178–1188, Spring ATL., IEEE Press, 30 March–3 April 2003 2003.
- [16] E. A. Hernandez, M. C. Chidester, and A. D. George, "Adaptive sampling for network management," in *Journal of Network and Systems Management*, vol. 9, pp. 409–434, HCS Research Laboratory, University of Florida, December 2001.
- [17] WIDE Project, "The widely integrated distributed environment project." <http://tracer.csl.sony.co.jp/maw1/>.
- [18] E. Fulp, Z. Fu, D. S. Reeves, S. F. Wu, and X. Zhang, "Preventing denial of service attacks on quality of service," in *DISCEX '01: Proceedings of the DARPA Information Survivability Conference and Exposition II*, vol. 2, pp. 159–172, IEEE Press, June 2001.
- [19] W. E. Leland, M. S. Taq, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic," in *SIGCOMM '93: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications* (D. P. Sidhu, ed.), (San Francisco, California), pp. 183–193, 1993.
- [20] M. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," in *SIGMETRICS'96: Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems*, (Philadelphia, Pennsylvania), p. 160, May 1996. Also, in Performance evaluation review, May 1996, 24(1):160-169.
- [21] Y. Xiang, Y. Lin, W. L. Lei, and S. J. Huang, "Detecting DDOS attack based on network self-similarity," in *Proceedings of IEE Communications*, vol. 151, pp. 292–295, June 2004.

APPENDIX

A. Self Similarity and the Hurst Parameter

Self-similarity, a term borrowed from fractal theory, implies that an object (in our case network traffic) appears the same regardless of the scale at which it is viewed. In a seminal paper published in 1994, Leland et al. [19] showed that the

traffic captured from corporate networks as well as the Internet exhibits *self-similar* behavior. Prior to the publication of [19], network traffic was assumed to be Poisson in nature. However, modeling network traffic using the Poisson distribution implied that it would have a characteristic burst length which would tend to be smoothed by averaging over a long enough time scale. This was in contrast to the measured values, which indicated that there was a significant *burstiness* in network traffic over a wide range of time intervals.

The self-similar nature of network traffic can be explained by assuming that network workloads are described by a power-law distribution; e.g., file sizes, web object sizes, transfer times, and even users think times have heavy-tailed distributions which decay according to a power-law distribution. A possible explanation for the self-similar nature of Internet traffic was given in [20], where the authors suggest that many *ON/OFF* sources with heavy-tailed *ON* and/or *OFF* periods resulting in core network traffic to be self-similar. The main properties of self-similar processes include *slowly decaying variance* and *long-range dependence*. An important parameter of a self-similar process is the *Hurst parameter*, H , that can be estimated from the variance of a statistical process. Self-similarity is implied if $0.5 < H < 1$.

The Hurst parameter is defined as follows: For a given set of observations X_1, X_2, \dots, X_n with sample mean, M_n defined as $(1/n) \sum_j X_j$, adjusted range $R(n)$ and sample variance S^2 , the rescaled adjusted range or the R/S statistic is given by

$$\frac{R(n)}{S(n)} = \frac{1}{S(n)} \cdot A, \quad (7)$$

where

$$A = \left(\text{Max} \sum_{j=1}^k (X_j - M_n) - \text{Min} \sum_{j=1}^k (X_j - M_n) \right)$$

Hurst discovered that many naturally occurring time series are well represented by the relation

$$E \left[\frac{R(n)}{S(n)} \right] \sim cn^H, \text{ as } n \rightarrow \infty \quad (8)$$

with the Hurst parameter H normally around 0.73, and a finite positive constant, c , independent of n . On the other hand, if the X_k 's are Gaussian pure noise or *short range dependent*, then $H = 0.5$ in equation (8).

Li, et al. [4], demonstrated mathematically that a significant change in the Hurst parameter can be used to detect a DoS attack, but their algorithm requires an accurate baseline model of the normal (non-attack) traffic. In another paper, Xiang et al. [21] contend that DDOS attacks can be detected by adopting a modified version of the rescaled range statistic.