Using Direct Sequence Spread Spectrum to Determine the Responsiveness of a TCP Aggregate to Packet Drops

by Mehdi Kalantari, Mark Shayman

TR 2003-37

Using Direct Sequence Spread Spectrum to Determine the Responsiveness of a TCP Aggregate to Packet Drops

Mehdi Kalantari and Mark Shayman
University of Maryland at College Park

Abstract— **In this paper we introduce a test through which the responsiveness of a TCP aggregate can be measured. The first introduced test is based on dropping a few packets from the aggregate and measuring the resulting rate decrease of that aggregate. This kind of test is not robust to multiple simultaneous tests performed at different routers. Extensions are done to make the test robust to multiple simultaneous tests by inspiring from the CDMA approach in the literature of multiple access channels in the communication theory. The measurements of responsiveness can be utilized for different purposes like congestion control or mitigating a Distributed Denial of Service Attack.**

I. Introduction

A key characteristic of TCP traffic is its responsiveness to packet drops. This forms the basis for congestion control. The degree to which a TCP aggregate reduces its rate in response to packet drops depends on packet size, roundtrip time and the distribution of window sizes among the constituent flows. An aggregate may also include noncooperative or malicious flows that do not participate in the TCP congestion control algorithm. Such flows are called *nonconformant*.

The goal of this paper is to introduce a technique for quantifying the responsiveness of a TCP aggregate to packet drops and for estimating the fraction of traffic that is nonconformant. Such a technique is useful for congestion control based on random early drop (RED). With quantitative information about the responsiveness of TCP traffic, when a router gets close to congestion it will know how many drops are needed to keep the rate of the traffic within the capacity of its outgoing links.

Another application is to the mitigation of distributed denial of service (DDoS) attacks. As perpetrators become more sophisticated, it can be anticipated that DDoS attacks will become increasingly stealthy with attack traffic designed to closely resemble ordinary Internet traffic. Studies show that more than 90 percent of the Internet traffic is generated by TCP traffic sources [4]. Furthermore, http traffic accounts for more than 42 percent of current traffic, while the average amount of packets exchanged per http flow is on the order of 10 packets. Consequently, a stealthy attacker might well choose to clog access links by generating large numbers of short-lived TCP flows. Such flows would be difficult to distinguish from ordinary traffic; they would also be difficult to trace back. Even if an attack source generated packets over a significant duration of time, by changing the spoofed source address or the source port number it could make the traffic appear to be composed of many small flows.

In order to mitigate DDoS attacks, it is desirable to have the capability to filter attack flows at backbone routers. However, it is not practical to filter at the granularity of individual flows. This is especially true when there are many small flows that consume a considerable percentage of bandwidth overall, but each of which lasts for a short period of time. An attack flow may have ended before it can be identified and filtered. A more practical alternative is to partition the traffic into a number of aggregates, identify those aggregates that contain a significant amount of attack traffic and filter a portion of these aggregates in order to reduce the effects of the attack.

To make aggregate-based filtering feasible, it is necessary to be able to estimate what proportion of traffic in an aggregate is likely to have been generated by attack sources. For stealthy attacks generating TCP packets, there may be no obvious characteristics that identify the attack packets as such. However, it is reasonable to assume that flows generated by attackers will differ from benign flows by failing to conform to TCP congestion control. Consequently, the problem of estimating the proportion of attack traffic in an aggregate can be reformulated as the problem of estimating the proportion of nonconformant traffic in a TCP aggregate.

In general terms, our approach is to perturb the arrival rate of the aggregate by intentionally dropping a small number of packets. By doing this periodically, the responsiveness of the aggregate under normal conditions can be determined. Then if a test indicates that the responsiveness of the aggregate has decreased, the presence of an attack can be inferred and the proportion of attack traffic can be estimated.

There are two difficulties that must be overcome in order to enable this approach to work. The first problem is that because of variations in the round trip times of flows, the perturbation of the arrival rate as a result of packet drops is spread out over time. We are able to overcome this problem by considering the integral of the rate decrease over time.

The second problem is more challenging and is concerned with distributed implementation. Each router should be able to apply perturbations and use these perturbations to determine the responsiveness of the aggregates it sees, either for the purposes of congestion control or to estimate the proportion of malicious traffic in a DDoS attack. However, the flows in an aggregate may experience perturbations at multiple routers. In a distributed implementation, in order to perform its estimation, a router should not need to be aware of the perturbations applied by other routers. Our approach to solving this second problem is inspired by the direct sequence spread spectrum (CDMA) approach to multiple access communication channels. Each router is assigned a dropping signature that specifies its packet dropping rate as a function of time. Different routers are assigned signatures that are orthogonal in a certain sense. Using simulations, we show that this approach enables each individual router to compute an accurate estimate of the proportion of nonconformant traffic that it sees without requiring any information to be shared among routers.

One final issue that we address has to do with the potential interaction between the applied perturbations and the effects of congestion-induced drops. Our approach assumes that the random process describing the window size of each TCP flow is a stationary random process. If congestion is building up, this assumption is no longer valid. Thus, for the DDoS application, the approach is intended to be applied in the initial stage of an attack before significant congestion-induced dropping occurs near the victim(s). Our simulations indicate that the method provides an accurate estimate of the proportion of nonconformant traffic as long as the bottleneck link utilization does not exceed about 80 percent of the outgoing link capacity.

The remainder of the paper is organized as follows. In Section II, we describe related work. In Section III, we derive mathematical formulas for the rate decrease of a conformant aggregate in response to dropping a small number of packets. In Section IV, we use these results to obtain an estimator for the TCP conforming component of an aggregate. In Section V, we describe how the use of CDMA-inspired orthogonal perturbing signatures enables multiple routers to perform perturbations without mutual interference. In Section VI, we give more explore the possible applications of APM and CAPM in more detail. In Section VII, the results of the simulation experiments are presented which confirm the efficacy of the proposed method.

II. Related Work

Many researchers have conducted studies to do identification and modeling of TCP traffic in granularity of flow under steady state conditions. In [1] the authors propose a method of testing a flow by comparing the steady state throughput of a TCP flow with the theoretical predicted value for conforming flows. If the response and model are similar, the flow is called TCP conformant. The objective of that study is to identify and penalize the non-conformant flows for congestion control purposes. The approach in [1] describes how large sustained individual flows may be tested for TCP conformance. However, a significant percent of the Internet traffic may be composed by short lived flows. Stochastic Fair Blue (SFB) is proposed in [11], and it offers per flow test for responsiveness by mapping different flows to the parallel bins. The approach is based on the fact that the bins containing a non-conforming flow are likely to be overloaded. However, if there are many non-conforming flows in a traffic aggregate, it is likely that all bins are overloaded, and the algorithm will not be able to distinguish between conforming and nonconforming flows.

There are many other works dealing with the TCP dynamics and its throughput analysis. In [10] the authors have offered the throughput model for a TCP traffic under assumption of stationary random losses. In [7] authors offer a flow based analysis of TCP dynamics in the Active Queue Management(AQM) routers by using stochastic differential equations.

III. Mathematical Formulation

To introduce the Aggregate Perturbation Method, we need to study how a TCP aggregate responds to packet drops in the network. To do so, we start with definitions of a flow and an aggregate:

Definition 1: A flow is a stream of data packets with the same source, destination and application, traversing along the same path.

Definition 2: An aggregate is a set of flows that have at least one link or node in common in their path from source to destination, and all packets belonging to these flows have a common property.

To define an aggregate in a more precise way, we have to define the common property of flows that belong to that aggregate. An example of an aggregate is all FTP flows that pass through a router.

Now we will continue by making some further assumptions about a given aggregate:

A1: The aggregate is composed of TCP flows that conform to TCP-Reno congestion control.

A2: All TCP flows in the aggregate are in the congestion avoidance phase.

A3: The random process describing the window size of each TCP flow is a stationary random process. We call this stationary random process $W(t)$.

Assumption **A1** is a key assumption since it states how a flow decreases its rate as a response to packet drops. TCP Reno has two different phases known as *slow start* and *congestion avoidance*. Slow start begins after making a connection, and upon successful transmission of every packet and receiving acknowledgement from the receiver the window size is increased by one. Congestion avoidance starts after the window size exceeds a threshold value, and in this phase the window size is increased one per round trip time, and upon experiencing a drop it is decreased to half its current value. By stating **A3**, we have assumed each TCP flow in the aggregate is in steady-state. This assumption is consistent with **A2**. Now we can state and prove the following lemma about the response of the aggregate when we drop D packets from it. For the purpose of this lemma, the packet size B and roundtrip time R are assumed constant. The extension to variable roundtrip times is described later.

Lemma 1: Assume a TCP traffic aggregate satisfying **A1-A3**. Let $\lambda(t)$ be the instantaneous arrival rate of this aggregate. If D packets are dropped from the aggregate instantaneously at time $t = 0$, and if D is small compared to the number of flows in the aggregate, then the aggregate will experience the following average instantaneous decrease in its rate:

$$E[\Delta\lambda(t_1)] = -\frac{BD}{2R}\frac{E[W^2]}{E[W]} \qquad (1)$$

where $t_1 > 0$ is the time at which the aggregate experiences its minimum rate as a result of packet drops, and $\Delta\lambda(t) = \lambda(t) - \lambda(0^-)$ for any $t > 0$.

Proof: Assume at time t the number of active flows in the aggregate is $N(t)$. Furthermore, let $p_j(t)$ denote the unconditional probability that the window size of a flow at time t is equal to j. Then we will have the following for the total instantaneous rate of the aggregate:

$$\lambda(t) = \sum_{k=1}^{N(t)} \lambda_k(t) \qquad (2)$$

where $\lambda_k(t)$ is the incoming arrival rate of traffic of the k^{th} flow belonging to the aggregate at time t. Since we have assumed $D << N(t)$, we may conclude that the probability of receiving multiple drops by the same flow is small, and so D flows receive drops. We call these flows $\{f_1, f_2, \ldots, f_D\}$. Assume for a flow f_i, the window size at time t is $W_{f_i}(t)$, the instantaneous rate is $\lambda_{f_i}(t)$ and the decrease in the instantaneous rate as a result of dropping one packet is $\Delta\lambda_{f_i}(t)$. Since the overall instantaneous decrease in the rate of the aggregate is sum of the rate decreases for the flows that experienced drops, we will have:

$$E[\Delta\lambda(t)] = \sum_{i=1}^{D} E[\Delta\lambda_{f_i}(t)] \qquad (3)$$

From the symmetry of the problem with respect to $\{f_1, f_2, \ldots, f_D\}$, we can conclude that the D random variables $\{\Delta\lambda_{f_1}(t), \Delta\lambda_{f_2}(t), \ldots, \Delta\lambda_{f_D}(t)\}$ are identically distributed and hence we will have:

$$E[\Delta\lambda(t)] = \sum_{i=1}^{D} E[\Delta\lambda_{f_i}(t)] = DE[\Delta\lambda_{f_1}(t)] \qquad (4)$$

$E[\Delta\lambda_{f_1}(t)]$ can be written as:

$$E[\Delta\lambda_{f_1}(t)] = \sum_{j=1}^{\infty} E[\Delta\lambda_{f_1}(t)|W_{f_1}(0) = j]P(W_{f_1}(0) = j) \qquad (5)$$

Using assumption **A1** and **A2** recall that we are dealing with TCP Reno compatible version of TCP. Hence, upon receiving a single drop, a TCP flow that is in congestion avoidance phase does not revert to slow start, but continues in congestion avoidance phase and halves its window size [6], [3]. Furthermore, this reduction happens by receiving a duplicate ACK of the dropped packet by the sender. So rate reduction happens at the sender at a time that is a fraction of round trip time R, but the router that did the drops observes the rate decrease for flow f_1 at time t_1 that theoretically is around the round trip time R. Since in the congestion avoidance phase the window size of a TCP flow cannot grow more than one per round trip time, we can see $W_{f_1}(t_1^-) \leq W_{f_1}(0) + 1$, so $W_{f_1}(t_1^-) \approx W_{f_1}(0)$. Then as a result of halving the window size we will have $W_{f_1}(t_1) \approx W_{f_1}(0)/2$. Hence

$$E[\Delta\lambda_{f_1}(t_1)|W_{f_1}(0) = j] = -\frac{Bj}{2R} \qquad (6)$$

It should be noted that since we have assumed the round trip times of all flows are the same, the time at which the aggregate experiences its minimum rate as a result of drops at $t = 0$ is t_1, the same as the time at which the $f_1, f_2, \ldots, f_D$ experience their minimum rate. After t_1 flows start to recover their rates.

The other term in (5) can be written in the following way:

$$P(W_{f_1}(0) = j) = \frac{jp_j(0)}{\sum_{i=1}^{\infty} ip_i(0)} \qquad (7)$$

The numerator of (7) is proportional to the number of packets generated by the flows with window size j in some time interval around $t = 0$, while the denominator is proportional to the number of packets generated by all flows in the same time interval. So given a packet drop, the probability that this packet belongs to a flow of window size j is given by (7).

Substituting (7) and (6) in (5) gives:

$$E[\Delta\lambda_{f_1}(t_1)] = -\frac{B}{2R}\frac{\sum_{j=1}^{\infty} j^2 p_j(0)}{\sum_{i=1}^{\infty} ip_i(0)} = -\frac{B}{2R}\frac{E[W^2]}{E[W]} \qquad (8)$$

The result follows immediately from (4) and (8). **QED**
In the proof, we have made use of stationary assumption **A3**, and so the expectations in the middle term of (8) are independent of time. This is true as long as the flows belonging to the aggregate are in their congestion avoidance state. It is straightforward to use similar proof to extend the above result to the case in which the flows forming the aggregate do not have the same round trip times. In this case, the rate reduction of different flows that received drops does not happen at the same time, and the rate reductions happen as smaller decreases spread over time. By using a similar derivation, it follows that in this case the expected sum of these smaller rate decreases is:

$$-\frac{BD}{2}\frac{E[W^2/R^2]}{E[W/R]} \qquad (9)$$

in which R is the random variable describing the round trip time of a packet belonging to a flow in the aggregate.

Now, we can make the following useful observations about the equation (1):

Remark 1: The amount of rate decrease as a result of dropping D packets from the aggregate is independent of the absolute instantaneous rate of aggregate $\lambda(t)$, and the number of flows contributing to the aggregate $N(t)$.

Remark 2: The rate decrease shows a linear behavior versus the number of dropped packets D. Later we will see how this linearity helps us to define a simple estimator of the portion of the aggregate which is not responsive to packet drops or congestion control. It is important to notice that this linearity is valid if $D << N(t)$

Since equation (1) only gives the instantaneous rate decrease, it is useful to study how the aggregate responds over time after drops. For this purpose, and under the same terms as Lemma 1, we can state:

Lemma 2: Assume **A1** and **A3** are true for a traffic aggregate. If we drop D packets from this aggregate at time $t = 0$, and D is small compared to the number of flows in the aggregate, then the average decrease in the rate of aggregate at time $t > 0$ is:

$$E[\Delta\lambda(t)] = -\frac{BD}{R}\Theta(F_W, t) \tag{10}$$

where F_W is the probability density function of window size $W(t)$ and $\Theta(.,.)$ is a nonnegative known function.

Proof: The proof is very similar to that of Lemma 1. Observe that similar to (6), for a flow f_1 that has received a drop at time $t = 0$ we can write:

$$E[\Delta\lambda_{f_1}(t)|W_{f_1}(0) = j] = \frac{B}{R}E[W_{f_1}(t) - W_{f_1}(0)|W_{f_1}(0) = j] \tag{11}$$

But $E[W_{f_1}(t)|W_{f_1}(0) = j]$ describes how the window size of a flow is increased after receiving a single drop at time $t = 0$ from a given initial condition $W_{f_1}(0) = j$, and it is independent of B, D and R. The statement of Lemma 2 is proved by substituting (11) and (7) in (5). It can be seen that $\Theta(F_W, t)$ can be written in the following way:

$$\Theta(F_W, t) = \frac{\sum_{j=1}^{\infty} jp_j(0)(j - E[W_{f_1}(t)|W_{f_1}(0) = j])}{\sum_{i=1}^{\infty} ip_i(0)} \tag{12}$$

QED

In this case it is hard to find a closed form for $\Theta(F_W, t)$ for an arbitrary time t. However, the comments given in **Remark 1** and **Remark 2** are still true, and those are the facts that we will use later to introduce an estimator of the TCP Conforming Component of an Aggregate.

For the case of different round trip times, we will have a similar statement:

$$E[\Delta\lambda(t)] = -BD\Theta_1(G_{WR}, t) \tag{13}$$

in which G_{WR} is the joint PDF of Window size $W(t)$ and the round trip time R, and $\Theta_1(.,.)$ is a nonnegative function. For the analytical results in the remainder of the paper, we will limit our attention to the case in which the round trip times are constant, but the results can be logically extended for the case with different round trip times. The simulation experiments include variable round trip times.

IV. Estimating TCP Conforming Component of an Aggregate (APM)

We recall that a TCP flow is said to be conformant if it responds to packet drops in the manner prescribed by the congestion control algorithm of TCP-Reno. The results of the preceding section predict the response of an aggregate consisting entirely of conformant TCP flows to a perturbation consisting of a small number of packet drops. In this section, we describe how it can be used to estimate the proportion of conformant traffic in an arbitrary aggregate of TCP flows. This estimator works based on either a single pass or multiple pass test on the aggregate.

Equation (1) suggests the basis of an algorithm for testing an aggregate: At time $t = 0$ some randomly selected packets are dropped from the aggregate. Based on the comparison of the observed decrease of the aggregate rate to the expected decrease, the non-conforming component of the traffic can then be estimated. One important problem that should be solved is the values of $E[W^2]$ and $E[W]$ which are not known in advance by the algorithm. But the algorithm can simply do some test and observations during the normal conditions when all traffic is assumed conformant and estimate the ratio $E[W^2]/E[W]$.

Another practical difficulty of using equation (1) arises when we need to measure the absolute value of rate decrease as a result of packet drops. Equation (1) assumes that the router observes all responses simultaneously, but there may be some mismatch in the time that different flows respond to packet drops. This problem becomes more obvious when the round trip times of different flows are not the same. In this case the responses of each of the individual flows that have received drops may be spread out in time, and (10) gives the sum of these smaller rate decreases which happen at different times.

To overcome this problem we define another metric to measure the degradation of the aggregate as a result of packet drops. This metric is defined as:

$$\eta(D) = \int_0^{t_r} (\lambda(0^-) - \lambda(t))\, \mathrm{d}t \tag{14}$$

t_r is a nonnegative finite time, and it can be chosen to be the minimum time for the recovery of all flows that received drops, and $\lambda(0^-)$ is the instantaneous rate at the moment before dropping the first packet. To get better results, $\lambda(0^-)$ may be replaced by a short-term average of the rate of aggregate in a time interval earlier than $t = 0$. $\eta(D)$ is simply a measure of how many more packets could have been sent by the aggregate if we had not dropped D packets. Based on the statement of Lemma 2 we will have:

$$E[\eta(D)] = \frac{BD}{R}\theta(F_W) \tag{15}$$

in which

$$\theta(F_W) = \int_0^{t_r} \Theta(F_W, t)\,\mathrm{dt} \tag{16}$$

Practically, it is hard to find closed form expressions for $\theta(F_W)$ defined in (16). But as a matter of fact, for our purpose it is not necessary to have such closed form expressions. In our algorithm, both $E[W^2]/E[W]$ or $\theta(F_W)$ should be estimated during the normal conditions of the network. Fortunately, $E[\eta(D)]$ is still independent of the number of flows in the aggregate $N(t)$ and the absolute rate of traffic $\lambda(t)$. In addition, $\eta(D)$ is a linear function of D. These properties of $\eta(D)$ help us to define a simple and scalable tool to estimate how much an aggregate is responsive to packet drops. It is important to notice that t_r is a parameter of the estimator, and it should not necessarily be the minimum time for the recovery of all flows that received drops. A smaller or larger value of t_r still preserves linear dependence of $E[\eta(D)]$ on D. However, a shorter value means discarding some useful information and a longer value means adding more noise and variance to the observation of $\eta(D)$.

Now we can continue toward introducing the estimator of the non-responsive component of an aggregate. Assume the TCP-conforming component of the traffic aggregate sends its traffic with unknown rate $\lambda_T(t)$. Likewise, the non-TCP-conforming component sends its traffic with rate $\lambda_N(t)$. Obviously we have $\lambda_T(t) + \lambda_N(t) = \lambda(t)$, and $\lambda(t)$ is completely observable. For the purpose of estimating $\lambda_T(t)$ and $\lambda_N(t)$, at time $t = 0$, we drop D packets from the aggregate randomly. Then let the conformant component receives $D_T(t)$ drops. The expected value of $D_T(t)$ is:

$$E[D_T(t)] = \frac{\lambda_T(t)}{\lambda_N(t) + \lambda_T(t)}D = \rho D \tag{17}$$

in which $\rho = \frac{\lambda_T(t)}{\lambda_N(t)+\lambda_T(t)}$ is the fraction of aggregate that is TCP conforming. Since the nonconforming part does not respond to packet drops, the rate decrease of the entire aggregate is equal to the rate decrease of the TCP-conforming component of that aggregate, which by using equation (15) can be written as:

$$E[\eta(D)] = \rho\frac{BD}{R}\theta(F_W) \tag{18}$$

We use equation (18) to construct an estimator for the ratio of the aggregate traffic that is conformant. Let $\bar{\rho}$ denote the estimate of this ratio. Note, in (18), $\eta(D)$ is fully observable and can be measured by using (14) after dropping D packets. Using (18), we will define the following estimator for the value of ρ:

$$\bar{\rho} = \frac{R}{DB}\frac{1}{\theta(F_W)}\eta(D) \tag{19}$$

It should be noted that in (19), the value of $\frac{R}{B\theta(F_W)}$ is constant and independent of D. So it can be estimated during the normal conditions of the network where the aggregate is fully conformant to TCP congestion control when $\rho = 1$. This quantity can be used later to estimate ρ. In order to decrease the variance of the estimator, multiple tests may be performed to achieve a more accurate estimate of ρ. We call this estimation method Aggregate Perturbation Method or APM.

V. Orthogonal Perturbing Signatures (CAPM)

One of the problems of distributed implementation of APM is the potential of simultaneous perturbations. The measurements of a perturbing router on an aggregate can be falsified by the simultaneous perturbations of a downstream or upstream router. In this section we introduce CAPM to overcome this problem. In CAPM every perturbing router uses a unique perturbing pattern. We will show that under proper assignment of the perturbing patterns and proper definition of aggregate degradation measure for each perturbing router, the test and measurment of each router will be robust to the interference caused by the other simultaneous perturbing routers.

Assume at time $t_1 < t$ we drop D_1 packets and at time $t_2 < t$ we drop D_2 packets from the aggregate. We are interested in the resulting average rate decrease $E[\Delta\lambda(t)]$ at time t. If $D = D_1 + D_2$ is small enough compared to the number of flows in the aggregate, then we expect that the likelihood of receiving multiple drops by the same flow is small enough, so the two perturbations independently degrade the rate of aggregate, and the decrease in the rate of aggregate at time t is the superposition of the two degradations. So based on the result of Lemma 2, we will have the following for the total degradation of aggregate at time t:

$$E[\Delta\lambda(t)] = -\frac{BD}{R}(\Theta(F_W, t - t_1) + \Theta(F_W, t - t_2)). \tag{20}$$

In (20) the first term on the right hand side is due to D_1 drops at time t_1 and the second term due to D_2 drops at time t_2. Obviously, the above argument is valid if we have more than two perturbations.

Now we can make the drops more spread out on a given interval $[0, T]$, and distribute them in a uniform or nonuniform way over that interval. For this case, let nonnegative function $r(t)$ be the number of drops per unit time at time $t \in [0, T]$. In other words, the packet drops are the output of a Poisson Process with variable rate $r(t)$. This means for a small enough interval of length Δt belonging to $[0, T]$ the probability of a single packet drop is approximately $r(t)\Delta t$. We call $r(t)$ a *drop signature* or a *drop pattern.*

Under the same terms of Lemma 1, we can state the following lemma:

Lemma 3: Assume we have a traffic aggregate for which assumptions **A1** and **A3** are valid. We drop the packets of this aggregate with rate $r(t)$ for $t > 0$, and assume that $r(t)$ is small enough such that a single flow is unlikely to experience another drop before recovery from a previous one. Then

$$E[\Delta\lambda(t)] = -\frac{B}{R}\int_0^t r(\tau)\Theta(F_W, t-\tau)\,\mathrm{d}\tau. \tag{21}$$

Proof: Denote $D_r(I)$ to be the random variable describing the number of packet drops in interval $I \subseteq [0, T]$. Now fix t, and consider $k+1$ time values $t_0 = 0 < t_1 < t_2 < \ldots < t_k = t$. Define $I_j = [t_j, t_{j+1}]$ for $0 \leq j < k$. If we assume the length of I_j is small enough, then we can think of $D_r(I_j)$ packet drops in that interval as $D_r(I_j)$ instantaneous drops at the beginning of interval t_j. Then drops in I_j make the following contribution to the aggregate rate decrease at time t:

$$E[\Delta\lambda(t)] = E[E[\Delta\lambda(t)]|D_r(I_j)] = -\frac{BE[D_r(I_j)]}{R}\Theta(F_W, t-t_j). \tag{22}$$

Note that in the middle term of (22) we have used iterated expectations, where the outer expectation is with respect to $D_r(I_j)$, and inner expectation is with respect to $\Delta\lambda(t)$. Now recall that we have assumed that length of I_j is small enough, so we can assume $r(t)$ has almost a constant value over this interval (to be more precise, we should assume $r(t)$ is continuous on $[0, T]$, except for a finite set of points). Then $[D_r(I_j)] \approx (t_{j+1} - t_j)r(t_j)$. Hence:

$$E[\Delta\lambda(t)] \approx -\frac{B(t_{j+1} - t_j)r(t_j)}{R}\Theta(F_W, t - t_j). \tag{23}$$

Since the value of $E[\Delta\lambda(t)]$ given in (23) is the contribution to the rate decrease of aggregate by drops in interval I_j, from the superposition property stated in (20), the total average rate decrease of aggregate can be written in the following way:

$$E[\Delta\lambda(t)] \approx -\sum_{j=0}^{k-1}\frac{B(t_{j+1} - t_j)r(t_j)}{R}\Theta(F_W, t - t_j). \tag{24}$$

Now let $k \to \infty$ in a way such that $\max_j(t_{j+1} - t_j) \to 0$. Then the summation in (24) tends to the following convolution integral in the limit:

$$E[\Delta\lambda(t)] = -\frac{B}{R}\int_0^t r(\tau)\Theta(F_W, t-\tau)\,\mathrm{d}\tau \tag{25}$$

which is the claimed result. **QED**

Remark 3: The system with input $r(t)$ and output $E[\Delta\lambda(t)]$ is a linear time invariant system with the impulse response $-\frac{B}{R}\Theta(F_W, t)$.

From the linearity proved in Lemma 3 we can expect that the effect of multiple simultaneous perturbations made by different drop signatures additively show up in the rate of aggregate in average sense. For instance, if we perturb the aggregate with rate $r_1(t)$ in one experiment that results in rate decrease of $\Delta\lambda_1(t)$, and in another experiment we use $r_2(t)$ that causes rate decrease of $\Delta\lambda_2(t)$, then if we perturb the traffic with rate $r(t) = r_1(t) + r_2(t)$ in a third experiment, we will expect the average rate decrease to be $E[\Delta\lambda(t)] = E[\Delta\lambda_1(t)] + E[\Delta\lambda_2(t)]$. Later in this section we will see how we can take advantage of this property to permit multiple perturbing routers.

For the case of using drop signature $r(t)$, we define the following measure of degradation of aggregate:

$$\eta_h(r) = \int_0^T h(t)\Delta\lambda(t)\,\mathrm{dt} \tag{26}$$

in which $h(t)$ is a weighting function that states at what time instants the results are more important to us, and at what time instants we are less interested in the rate decrease of aggregate. Based on the result of Lemma 3 and the linearity of the $\eta_h(r)$ in $\Delta\lambda(t)$, it can be observed that $E[\eta_h(r)]$ is linearly dependent on the drop signature $r(t)$.

Now we try to use an approach similar to Direct Sequence Spread Spectrum CDMA in multiple access communication to solve interfering problems of multiple simultaneous perturbing routers. In this approach, each router perturbs the traffic according to its unique CDMA code. In this case, the drop signature of the i^{th} perturbing router can be written as:

$$r_i(t) = A_i\sum_{j=1}^{N} c_j p_{T_c}(t - (j-1)T_c) = A_i s_i(t) \tag{27}$$

in which, A_i is a known perturbation amplitude of the i^{th} router, N is a positive integer called the spreading factor, $T_c = T/N$, $(c_1, c_2, \ldots, c_N)$ is a binary sequence assigned to the particular router known as the code of the user. In (27), $s_i(t)$ denotes the *normalized drop signature,* and $p_{T_c}(t)$ is a real-valued function known as the chip waveform and it satisfies the following property:

$$\int_{-\infty}^{\infty} p_{T_c}(t)p_{T_c}(t - nT_c)\,\mathrm{dt} = 0,\ \text{n} = 1, 2, \ldots. \tag{28}$$

The estimation of the i^{th} router about the conformance of the aggregate is made based on the *Matched Filter* output. The matched filter output is the value of $\eta_h(r_i)$ evaluated at $h(t) = s_i(t)$:

$$y_i = \int_0^T s_i(t)\Delta\lambda(t)\,\mathrm{dt}. \tag{29}$$

Since in our problem $r_i(t)$ is a drop rate, it should be nonnegative, and hence $p_{T_c}(t)$ should be nonnegative. For this purpose we suggest the popular simple rectangular chip waveform:

$$p_{T_c}(t) = \begin{cases} 1 & if\ 0 < t < T_c \\ 0 & otherwise. \end{cases} \tag{30}$$

Usually, in the CDMA systems assignment of the codes is very important. Users with a potential of high interference (e.g., neighbor routers in our problem) are assigned to codes that cause their drop signatures to be orthogonal (or close to orthogonal)

$$\int_0^T s_i(t)s_j(t)\,\mathrm{dt} = 0, \text{ for i} \neq \text{j}. \tag{31}$$

Unfortunately, the statement of (31) cannot be satisfied with the current definition of drop signatures defined in (27). That is because both $s_i(t)$ and $s_j(t)$ are nonnegative rate functions, and hence the integral defined in (31) can never be zero. We can solve this problem with a minor change of the orthogonality requirement and the structure of the matched filter. First, we replace the orthogonality condition with a similar condition in which the normalized drop signatures are orthogonal after removing their DC components:

$$\int_0^T s_i^a(t)s_j^a(t)\,\mathrm{dt} = 0, \text{ for i} \neq \text{j} \tag{32}$$

in which $x^a(t)$ is $x(t)$ after eliminating its DC component over $[0, T]$:

$$x^a(t) = x(t) - \frac{1}{T}\int_0^T x(t)\,\mathrm{dt} \tag{33}$$

Furthermore, we change the matched filter output for the i^{th} router in the following way:

$$y_i = \eta_{s_i^a}(r) = \int_0^T s_i^a(t)\Delta\lambda(t)\,\mathrm{dt} \tag{34}$$

y_i is the value of η_h in (26) evaluated for $h(t) = s_i^a(t)$. One important note about notation $\eta_h(r)$ in (34) and (26) is that in both equations r is the total perturbing function, since the rate decrease $(\lambda(0^-) - \lambda(t))$ is affected by this total drop rate. Hence $r(t) = \sum_k r_k(t)$, where k is an index that covers the set of all perturbations that the aggregate experiences. Now we can state the following lemma:

Lemma 4: Assume assumptions **A1** and **A3** are valid for a traffic aggregate, and assume that the overall drop rate $r(t) = \sum_k r_k(t)$ is small enough such that a single flow is unlikely to experience another drop before recovery from a previous one. Furthermore the rate of change of each perturbation $r_k(t)$ is small enough compared to the response time of the aggregate. Then under orthogonality assumption of (32) we have:

$$E[y_i] = E[\eta_{s_i^a}(r)] = E[\eta_{s_i^a}(r_i)] \tag{35}$$

Proof: Denote $\Delta\lambda^x(t)$ to be the rate change of aggregate when the aggregate is perturbed with drop rate $x(t)$. By definition we will have:

$$E[\eta_{s_i^a}(r)] = \int_0^T s_i^a(t)E[\Delta\lambda^r(t)]\,\mathrm{dt}. \tag{36}$$

From the result of Lemma 3 and linearity of $E[\Delta\lambda^r(t)]$ in $r(t)$ we can conclude

$$E[\Delta\lambda^r(t)] = E[\Delta\lambda^{r_i}(t)] + \sum_{j\neq i} E[\Delta\lambda^{r_j}(t)]. \tag{37}$$

Substituting (37) in (36) yields:

$$E[\eta_{s_i^a}(r)] = E[\eta_{s_i^a}(r_i)] + \sum_{j\neq i} E[\eta_{s_i^a}(r_j)]. \tag{38}$$

To complete the proof, it suffices to prove $E[\eta_{s_i^a}(r_j)] = 0$ for $j \neq i$. We have $r_j(t) = A_j s_j(t)$. Now we use the assumption that $r_j(t)$ changes slower than the aggregate response time. Hence $r_j(t)$ can be approximated by using a piecewise constant function. For an interval on which $r_j(t)$ is constant, the traffic aggregate responds and settles down to a value. In the next interval $r_j(t)$ jumps to a new value, and so $E[\Delta\lambda^{r_j}(t)]$ responds accordingly, and after experiencing a small transient time settles down to a new steady state value. According to the linearity result of Lemma 3 the steady state value of $E[\Delta\lambda^{r_j}(t)]$ on each interval is proportional to the constant value of $r_j(t)$ on that interval. This means that $E[\Delta\lambda^{r_j}(t)]$ tracks the piecewise constant shape of $r_j(t)$. So by ignoring the short transients of $E[\Delta\lambda^{r_j}(t)]$ at the beginning of each interval we will have:

$$E[\Delta\lambda^{r_j}(t)] \approx C_j r_j(t) = C_j A_j(s_j^d + s_j^a(t)) \tag{39}$$

in which s_j^d is the DC component of $s_j(t)$ over interval $[0, T]$. Recall

$$E[\eta_{s_i^a}(r_j)] = \int_0^T s_i^a(t)E[\Delta\lambda^{r_j}(t)]\,\mathrm{dt}. \tag{40}$$

Substituting (39) in (40) and using orthogonality assumption of (32) yields: $E[\eta_{s_i^a}(r_j)] = 0$. **QED**

One useful observation about (34) is:

$$\int_0^T s_i^a(t)\lambda(0^-)\,\mathrm{dt} = 0 \tag{41}$$

And so we have the following simple equation for the output of the matched filter for the i^{th} router:

$$y_i = -\int_0^T s_i^a(t)\lambda(t)\,\mathrm{dt} \tag{42}$$

Now we continue toward defining an estimator of nonconformant portion of the TCP aggregate. From (25),(26) and (35), we have the following expression for the average output of the matched filter of the i^{th} perturbing router:

$$E[y_i] = \eta_{s_i^a(r_i)} =$$

$$-\frac{A_i B}{R}\int_0^T s_i^a(t)\int_0^t s_i(\tau)\Theta(F_W, t-\tau)\,\mathrm{d}\tau\,\mathrm{dt} \qquad (43)$$

Equation (44) gives the basis for the estimator of non-conformant portion of the TCP aggregate. Denote:

$$K_i = -\frac{B}{R}\int_0^T s_i^a(t)\int_0^t s_i(\tau)\Theta(F_W, t-\tau)\,\mathrm{d}\tau\,\mathrm{dt}. \qquad (44)$$

Notice that K_i is a coefficient that describes how much the aggregate is responsive to packet drops. We call this quantity the *response coefficient* of the aggregate. For a fully conformant aggregate we have:

$$E[y_i] = A_i K_i \qquad (45)$$

With an argument similar to that in Section (IV), we can derive the following formula for the case the aggregate is not fully conformant to congestion control:

$$E[y_i] = \rho A_i K_i \qquad (46)$$

in which ρ is the ratio of the bandwidth of conformant portion to the bandwidth of the aggregate. Then we will come up with the following simple estimator of ρ

$$\bar{\rho} = \frac{y_i}{A_i K_i}. \qquad (47)$$

Note that y_i is fully observable, and it can easily be measured by using (42). The amplitude of perturbing function A_i is known to the router that does perturbation. Finding the K_i is the only problem of the estimator. This coefficient can be estimated during the times that we know the aggregate is fully conformant (i.e., $\rho = 1$). Or it can be estimated by a long term average of y_i/A_i. Once K_i is estimated, we use (47) to estimate ρ for times it is unknown (for instance during a DDoS attack or a flash crowd time).

There are some key issues about how to choose the value of T_c. As stated before, T_c should be long enough such that the rate decrease of aggregate as a result of packet drops in one chip duration can show up, and the aggregate rate settles down. On the other hand, too large T_c does not improve the performance of estimator, and it only causes longer test and more packet drops, which causes the test to be more expensive. The value of N may be chosen according to the maximum number of simultaneous perturbing users and the maximum tolerable interference in the system. In contrast to the cellular systems, security of the codes is not a major concern in our problem, however, there is no limitation on defining long signature sequences, and adding security features to the assigned codes.

We have called this method CAPM that stands for CDMA based APM.

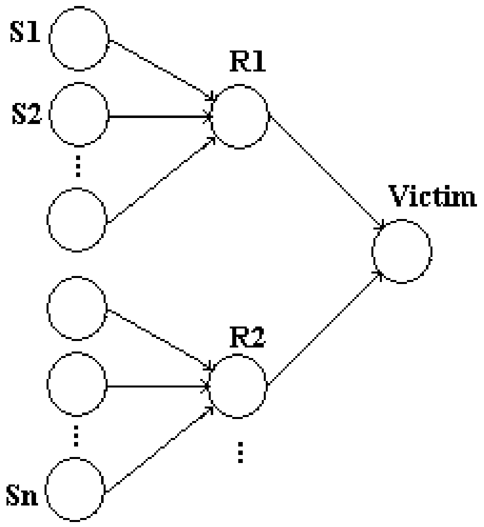

Fig. 1. The typical topology of network from the perspective of a DDoS victim

VI. Applications of APM and CAPM

A.DDoS Defense

Based on the results of Section (IV) and Section (V), we can define a method to minimize the effects of a DDoS attack that employs TCP packets. The basic idea of such a defense policy is rooted in the fact that the traffic sent by the DDoS sources over the Internet is not TCP-conforming.

Assume we have a victim, and suspect traffic being forwarded to it. From the perspective of the victim, the network can be viewed as a tree as seen in figure 1. The traffic is generated at sources denoted S_i. The traffic traverses intermediate routers at which APM or CAPM is implemented. The main, and unique, property of the defense policy based on APM or CAPM is that the intermediate routers do not use any downstream feedback, statistical or otherwise, in order to determine the special characteristics or *attack signature* of the malicious traffic. This allows us to have a proactive defense to protect the victim before it is significantly impacted by the attack.

Our DDoS defense is based on dividing the traffic at the intermediate nodes into aggregates. Then the intermediate routers need only examine the aggregates rather than the many individual flows. Each aggregate is tested by dropping a few packets from it and observing the response of that aggregate to the packet drops. Based on these observations, the routers can estimate the ratio of attack traffic to the total traffic. To get more accurate results, each intermediate router may do multiple tests on each aggregate. The final step toward fully protecting the victim is to install a filter to process the traffic being forwarded. This filter is adaptively designed to have the best match to the signature or pattern of the attack traffic–i.e., it passes the packets belonging to clean aggregates with a high probability, and passes the packets belonging to the polluted aggregates with a lower probability. It should also be reiterated that the APM or CAPM are intended to be applied proactively to detect and mitigate a DDoS attack before significant increase in congestion-induced packet dropping occurs near the victim.

Pushback introduced in [8] is one of the solutions offered to mitigate DDoS attacks. In Pushback the signature of the attack traffic is identified and advertised upstream for the filtering purpose. The detection of the attack signature in the conventional pushback is based on the congestion control mechanism and observing the pattern of dropped packets or other packets at the congested routers. So the signature of attack is unknown before congestion happens as a result of DDoS. However, congestion means an advanced phase of DDoS. APM or CAPM can contribute to pushback by detecting the signature of attack in an earlier phase of a DDoS attack and before congestion happens. As stated earlier, the signature based on APM or CAPM is based on which aggregates are more likely to be polluted by the attack traffic. Therefore, by using APM or CAPM pushback can act more effectively by not depending on congestion and starting in an earlier phase of the DDoS attack.

B. Congestion Control

Congestion control is the other application of APM and CAPM. Random Early Drop [2] is one of the popular approaches to proactively prevent congestion in a router. However, tuning up the parameters of an RED congestion controller is a challenging problem. Starting drops very early causes underutilization of the resources, and doing drops very late causes this tool cannot work perfectly to prevent congestion. By utilizing either APM or CAPM a router collects information about how responsive different aggregates are -i.e., K_i coefficients defined in the previous section. By knowing these coefficients, a router can determine how much it should drop from the aggregate to reduce its bandwidth to a certain value.

Flash crowds are among the phenomena that can cause long term congestions in the Internet. In a flash crowd, a huge number of data packets are flooded toward a destination (e.g., popular site). The flash crowd may cause very heavy congestion in the links close to the destination or other places of the network like border gateways. In practice it is very likely that most of the flows of a flash crowd belong to a few aggregates. The response coefficient of aggregates that carry more traffic belonging to the flash crowd experiences a more decrease, so by filtering the packets belonging to these aggregates more aggressively and close to their sources, the other aggregates can be saved from congestion.

Another strong point of APM and CAPM is fairness. Assume a traffic composed of many aggregates is intended to be forwarded through the same outgoing link at a router, and the link is close to congestion. So it is desired to keep the traffic bandwidth within the outgoing link capacity. If the router applies equal drop probability governed by RED for all aggregates of the traffic, the aggregates with higher response coefficients will back off more aggressively compared to the aggregates with smaller response coefficients. A certain degree of fairness among aggregates can be achieved by taking into account the response coefficient of each aggregate. If p_l is the assigned drop probability of an aggregate with response coefficient K_l, then the simplest heuristic to achieve fairness is to distribute the desired drop probability of traffic among its aggregates such that $K_l p_l$ is constant for all aggregates. So all aggregates of the traffic make the same average contribution to the rate decrease.

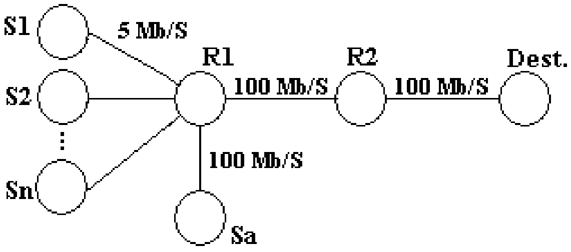

Fig. 2. The network topology used for simulation

VII. Simulations and Results

We have used the popular network simulator *ns2* to perform our experiments [9]. As explained previously, our focus is on TCP aggregates. For simulation we have used a network with fixed topology similar to that in figure 2. The nodes $S_1, S_2, \ldots, S_n$ are n sources of conformant TCP traffic, and n is 50 in our simulations. The propagation delay of the link between each source and router *R1* in figure 2 is different from a source to another source, and it has been chosen such that the round trip time of packets is uniformly distributed between 50 and 100 milliseconds under low congestion conditions. The flows at the sources are generated according to a birth-death process. Each source starts a TCP flow, and that flow ends after a random time uniformly distributed between 0 and 0.15 seconds. That source starts a new flow after waiting another random time uniformly distributed between 0 and 0.3 seconds. As shown in figure 2, there is another source *Sa*, which is connected to router R1, and it generates non-conformant traffic and sends that toward the destination. So both intermediate routers R1 and R2 receive an aggregate that is a mixture of conformant and non-conformant flows. The packet size is constant equal to 1 Kbyte for all flows. In this topology, the link between R1 and R2, and also the link between R2 and destination are bottleneck links. The capacity of these bottleneck links is 100 Mbps, that translates to 12500 packets per second.

In the first experiment, we show how a conformant aggregate responds to packet drops and show how the shape of a drop function shows up in the aggregate rate. In this experiment, there is no non-conformant component in the

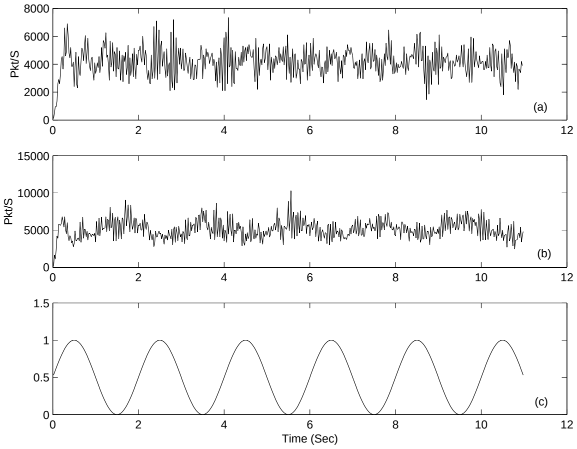

Fig. 3. a: The rate of aggregate under no perturbation, b: The aggregate rate under cosine perturbations, and c: The normalized cosine drop signature

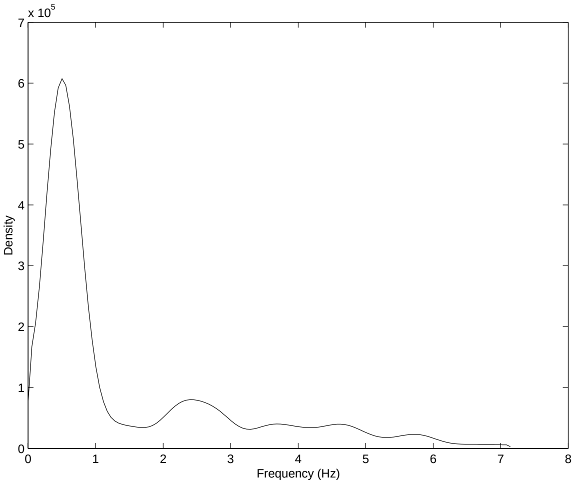

Fig. 4. The Power Spectral Density of the aggregate rate under a cosine perturbation after eliminating its DC component.

aggregate, and we compare the aggregate rate with and without perturbations. Figure 3-(a) shows the rate of traffic under no perturbations. The TCP-conformant sources generate about 4000 packets per second.

In the next experiment, we have used a single cosine drop signature at R1, while no perturbation is done at R2. The frequency of drop signature, $f_1 = 0.5Hz$, and its amplitude is $A_1 = 100$ drops per second. The normalized drop signature and the resulting aggregate rate can be seen in figure 3-(c) and figure 3-(b) respectively. By comparing the aggregate rate under cosine perturbations in figure 3-(b) and the aggregate rate in the normal conditions in figure 3-(a), it can be seen that the drop signature has modulated the aggregate rate, and the shape of the cosine drop signature appears in the aggregate rate with 180 degrees of phase shift.

Useful observations can be done by inspecting the Power Spectral Density of the aggregate under a cosine perturbation. Figure 4 shows the spectral density of the aggregate rate shown in figure 3-(b) after eliminating its DC component. One obvious observation is the high peak at $f = 0.5Hz$ which is a result of the modulation effect of cosine drop signature with frequency $f = 0.5Hz$, but another useful fact about this figure is that there are not distinguishable peaks at the multiples of $f = 0.5Hz$. This means that there is no harmonic distortion as a result of nonlinearity. Since the system only responds at the frequency it was perturbed, and the fact that cosines are eigenfunctions of linear systems confirms the linearity of the system as proved in Lemma 3.

In the next experiment we explore the typical response of aggregate when two routers perturb it simultaneously. In this experiment R1 and R2 perturb the aggregate using different CDMA drop signatures. In the simulation $T = 32$ seconds, $N = 16$, $T_c = 2$ seconds, and we have used the rectangular chip waveform. The code of R1 is $(1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1)$, and that for R2 is $(0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1)$. Under this assignment $s_1^a(t)$ and $s_2^a(t)$ are orthogonal. The resulting normalized drop signatures for R1 and R2 are shown in figure 5-(b) and 5-(c) respectively. In these figures, two periods of these drop signatures have been plotted. The amplitude of drop signatures for the two routers, A_1 and A_2, are the same and equal to 120 drops per second.

Figure 5-(a) shows the rate of aggregate when the two routers R1 and R2 perturb the aggregate simultaneously. It can be seen that the additive shape of the two drop signatures appears on top of the aggregate rate– with 180 degree phase shift again. In other words, the two drop signatures modulate the aggregate rate additively. For example, at around time $t = 40$, the amplitude of both drop signatures is zero, and this shows up as an increase in the rate of aggregate as it can be seen in 5-(a) at $t = 40$. On the other hand, at time $t = 15$ or $t = 31$, the amplitude of both drop signatures is nonzero, and this shows up as a decrease in the rate at these two times.

The purpose of next experiment is to verify that under orthogonality definition of (32), the matched filter output of a router defined by (42) is not affected by perturbations done by the other routers. We proved this fact in Lemma 4. Furthermore, we verify the linear dependence of the matched filter outputs on the amplitude of drop signature for each router. In this case we use the same CDMA drop signatures as in the previous experiment, but we change A_1 and A_2, the amplitude of the drop signatures of the two routers. Figure 6-(a) shows y_1, the output of matched filter for R1 as it is defined by (42), when A_1 changes from 0 to 160 drops per second. In this figure, each + represents a test in which R1 perturbs the aggregate with drop signature $r_1(t) = A_1 s_1(t)$ and at the same time R2 is also perturbing traffic with drop signature $r_2 = A_2 s_2(t)$, and $A_1 = A_2$. For each value of A_1 several tests have been done, and

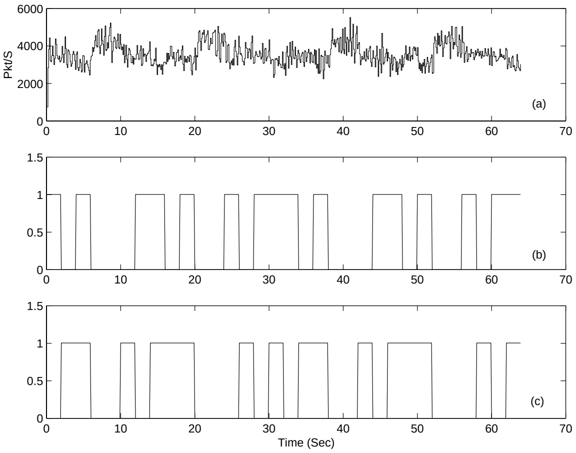

Fig. 5. a: The aggregate rate under two simultaneous perturbations, b: the normalized drop signature of R1, and c: the normalized drop signature of R2.

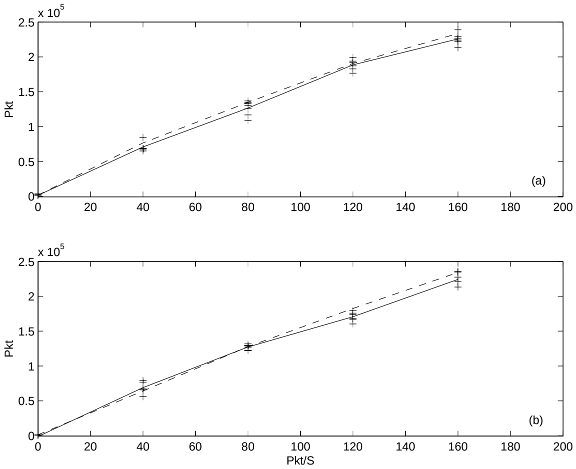

Fig. 6. Matched filter output versus the amplitude of drop signature, a: for the first router, b: for the second router.

the average over multiple tests has been plotted by the solid line. It can be seen that the deviation of y_1 for each individual test from the average value shown by solid line is relatively small; this means that the matched filter output shows a small variance. The other observation about 6-(a) is linearity in amplitude of drop signature A_1 that we proved in Lemma 3.

In the other part of this experiment we turn off the perturbations done by R2 by setting $A_2 = 0$, and do the same multiple test and measurement of y_1 for each value of A_1. The dashed line in figure 6-(a) shows the average of multiple tests for each value of A_1 for this case. It can be seen that the dashed line is very close to the solid line showing that perturbations of R2 do not affect the output of matched filter of R1. Also note that the slope of the solid line or dashed line in figure 6-(a) is K_1 defined in equation (44). We need this quantity as a parameter of estimator of the non-conformant component of an aggregate in our next experiment. Figure 6-(b) is the same as figure 6-(a) for the second router.

In the next experiment we add non-conformant traffic to the aggregate and try to estimate this non-conformant component of it by CAPM. In this experiment, we do the same CDMA-based perturbations as in the previous experiment and use CAPM, but with fixed A_1 and A_2 equal to 100 drops per second. We use the CAPM estimator introduced in equation (47) and use the values of K_1 and K_2 found in the previous experiment. In this case the non-conformant part of traffic sends about 2650 packets per second, which is about 40 percent of the average total rate of the aggregate since the TCP-conformant sources send about 4000 packet per second. Figure 7 shows the results of this experiment for the two routers. In figure 7-(a), each + shows the result of a single test and estimation of the percentage of non-conformant traffic. The solid line shows the actual value of this percentage, which is about 40 percent. The dashed line is the average over all tests. The figure shows that the result of each estimation is very close the actual value, and deviation of every single estimation is about a few percents from the actual value, and the results of multiple estimations are about the same. This shows that the estimator has a low variance. Another important observation about this experiment is that the estimator is unbiased since after several independent estimations and taking the average the error tends to zero. Figure 7-(b) is the same as figure 7-(a) for the second router.

In the last experiment we try to examine the performance of the estimator when changing the amount of non-conformant traffic. In this experiment we start from a fully conformant aggregate, but later we add non-conformant traffic to that. Then we use the CDMA based perturbations and use CAPM in order to estimate the percentage of non-conformant traffic in the aggregate. The upper limit for the amount of the non-conformant traffic is the capacity of bottleneck link. Figure 8 shows the result of the simulation for the two routers. The vertical axis in this figure is the percentage of bottleneck link utilization by the aggregate, the solid line shows the actual percentage of non-conformant traffic during the simulation. The dashed line shows the result of estimator obtained by the first router, and the + signs show those estimates of the second router. As it can be seen, the estimator is accurate to the point where the link utilization is about 80 percent. After this point the bias of estimator starts to increase. This behavior can be explained in the following way: as we get close to the congestion condition, some packets are dropped from the aggregate and this number increases as we get closer to congestion condition. So the conformant component of the aggregate shrinks its rate to respond to the drops made by congestion control, and as a result, this component becomes less responsive to the perturbations done by the APM or CAPM mechanism, and so the algorithm over-estimates the percentage of non-responsive traf-

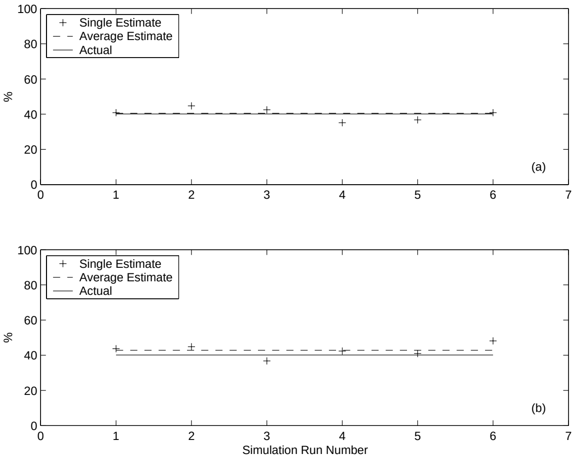

Fig. 7. Estimate of the non-conformant percentage of traffic, a: for the first router, b: for the second router.

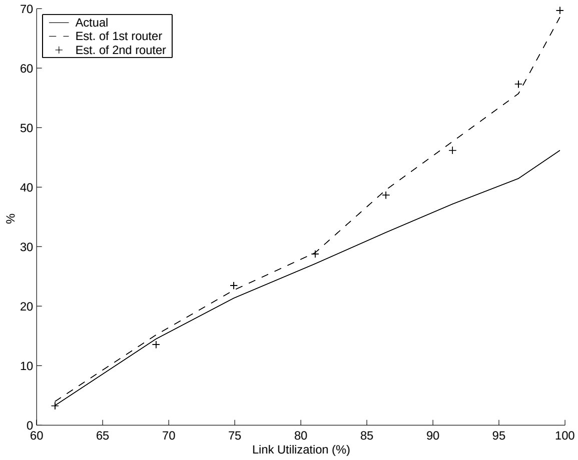

Fig. 8. Performance of CAPM based estimator of non-conformant traffic versus bottleneck link utilization.

fic. Another way of explaining this fact is to see that long term congestion causes all conformant flows decrease their window sizes, which causes a change in F_W, the PDF of the window size. This causes an error in estimator since the estimator uses K_i which depends on stationary PDF of F_W.

Although long term congestion is one factor that may degrade the performance of APM or CAPM, the APM and CAPM show a good performance in a wide range of link utilization before congestion happens. This can be one of the strong points about APM and CAPM since these methods can be applied proactively and prevent congestion that can be caused by a DDoS or any other reason.

VIII. Conclusions

In this paper, we introduced the Aggregate Perturbation Method (APM), and CDMA-based APM (CAPM) two techniques for estimating the amount of non-conforming traffic belonging to a TCP aggregate. Both algorithms perform a test on the aggregate by dropping some packets from it and observing the result. APM is the simpler test but it is not robust to simultaneous tests at different routers. So we introduced CAPM that uses some unique drop signature for each router to do the test, and the approach is similar to the Direct Sequence Spread Spectrum CDMA in communication theory.

One important advantage of APM and CAPM is that they can be implemented in a distributed manner without needing data exchange between routers, and furthermore, these methods do not need any change in the current protocols. This also permits incremental deployment. One of the strong points about APM and CAPM is that these algorithms can perform proactively, and prevent congestion or possible effects of a DDoS attack.

Proper definition of aggregates is critical in the performance of the introduced algorithms. More research can be done to find proper schemes of defining aggregates. For example, the definition of an aggregate may be based on the type of packets, source network, destination network, or any other information available in the headers of data packets. Intuitively, in order to get good performance of the estimator, the flows contained in an aggregate should have similar statistical properties in terms of their window sizes. This causes the responsiveness of the aggregate not to be dominated by a few large flows. So prior information about the window size of a flow can be useful to assign it to a proper aggregate.

References

[1] Floyd, S. and Fall, K., "Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 4, Aug 1999.

[2] Floyd, S. and Jacobson, V., "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, pp 397-413, Aug 1993.

[3] Jacobson, V., "Congestion Avoidance and Control," *Proceeding of SIGCOMM 88*, Proceeding of SIGCOMM 88, Aug. 1988.

[4] Schulzrinne, H., "Long Term Internet Traffic Statistics," available online at: http://www.cs.columbia.edu/ hgs/internet/traffic.html/.

[5] Savage, S.; Wetherall, D.; Karlin, A. and Anderson, T., "Network Support for IP Trace back," IEEE/ACM Trans. on Networking, June 2001,Vol. 9, No 3.

[6] Stevens, W., "TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms ," RFC2001, Internet Engineering Task Force, http://www.ietf.org/rfc/.

[7] Misra, V.; Gong, W. and Towsley, D., "A fluid based analysis of a network of AQM routers supporting TCP flows with an application to RED ," Proc. of SIGCOMM 2000, August 2000.

[8] Mahajan, R.; Bellovin, S.; Floyd, S.; Ioannidis, J.; Paxson, V. and Shenker S., "Controlling High Bandwidth Aggregates in the Network ," available online at: http://www.icir.org/pushback/pushback-Jul01.pdf/.

[9] Network Simulator ns2, online documentation available at: http://www.isi.edu/nsnam/ns/.

[10] Altman, E.; Avrachenkov, K. and Barakat, C., "A Stochastic Model of TCP/IP with Stationary Random Losses ," Proc. of SIGCOMM 2000, August 2000.

[11] Wu-chang, Feng; Shin K.G.; Kandlur, D.D. and Saha, D., "The BLUE active queue management algorithms," *IEEE/ACM Transactions on Networking*, Vol. 10, No. 4, pp 513- 528, Aug 2002.