On the Mitigation of Traffic Correlation Attacks on Router Queues

Yan Cai†, Patrick P. C. Lee†, Weibo Gong†, Don Towsley‡
†University of Massachusetts, Amherst, ‡The Chinese University of Hong Kong, Hong Kong

Abstract—Traffic burstiness is known to be undesirable for a router as it increases the router’s queue length and hence the queueing delays of data flows. This poses a security problem in which an attacker intentionally introduces traffic burstiness into routers. We consider a correlation attack, whose fundamental characteristic is to correlate multiple attack flows to generate synchronized small attack bursts, in an attempt to aggregate the bursts into a large burst at a target router.

In this paper, we develop an analytical, fluid-based framework that models how the correlation attack disrupts router queues and how it can be mitigated. Our framework captures the dynamics of a router queue using Poisson Counter Stochastic Differential Equations (PCSDEs), and for special cases, it gives the closed-form average router queue length as a function of the inter-flow correlation. To this end, we apply our analytical framework to construct a two-stage pacing scheme that mitigates the correlation attack. The two-stage pacing scheme is composed of Markov ON-OFF pacing and rate limiting, which are respectively designed to break down the inter-flow correlation and suppress the peak rates of bursts. We verify that our fluid models conform to packet-level ns2 simulation results, and show that the two-stage pacing scheme effectively reduces the average router queue length in the face of a correlation attack.

I. INTRODUCTION

Router queues hold packets during congestion, and they play a critical role in determining the performance of packet switching networks. However, traffic burstiness is known to be undesirable for a router, as it increases the router’s queue length and hence the queueing delay of data flows.

In this paper, we analyze the impact of traffic burstiness on router queues from a security perspective, in which a router queue is a target point of attack. We consider a denial-of-service (DoS) attack called the correlation attack that intentionally introduces traffic burstiness at the target router. The correlation attack is inherently low-rate, meaning that the average rate of the attack bursts is small enough to not congest a network link. The main idea of the correlation attack is to exploit the correlation among multiple attack flows scattered across different locations and have them generate small attack bursts in a highly correlated manner. As a result of the inter-flow correlation, the aggregation of these small bursts will increase router queue lengths and hence the end-to-end transfer delays of normal flows. This is particularly annoying for real-time applications, such as interactive sessions or video streaming, where timely packet delivery is crucial for the quality of service.

Since the correlation attack is low-rate by nature, it can evade the detection of conventional counter-DoS mechanisms that defend against flooding-based DoS attacks by monitoring the volume of attack traffic. In addition, the correlation attack considered here focuses on the inter-flow correlation, while the traffic pattern within each individual attack flow can be arbitrary. Thus, it generalizes previously proposed low-rate DoS attacks that generate attack bursts at fixed periods [4], [12]. It also differs from the prior work that models traffic burstiness due to high autocorrelation (intra-flow correlation) (e.g., see [10]).

The practicality of the correlation attack is further justified with the emergence of botnets (e.g., see survey in [26]). In a botnet, an attacker (called botmaster) controls a group of compromised hosts (called bots), of which the average total number can reach 20K [22]. The botmaster can send a command to the bots and schedule them to send attack bursts to a target router in a correlated manner. While defending against botnets remains an ongoing challenge to the research community, our focus here is to design an effective defense solution that specifically mitigates the correlation attack.

The main objective of this paper is to provide an analytical framework to study the impact of the correlation attack on router queues, from both attack and defense perspectives. We model data traffic as a fluid and capture the queuing dynamics of a router using Poisson Counter Driven Stochastic Differential Equations (PCSDEs) [3], [9], [10], [15], [23].

In the attack part, we analyze how the correlation attack increases the average router queue length by exploiting the inter-flow correlation. We model attack traffic with different levels of inter-flow correlation, and derive an expression for the average router queue length as a function of the inter-flow correlation for specific cases. This enables us to understand how an attacker, while using the same expected amount of attack traffic, can maximize the inter-flow correlation and hence the damage to a router. Using packet-level ns2 simulation [21], we also show that the correlation attack significantly increases the end-to-end transfer delays of normal flows.

In the defense part, we propose to mitigate the correlation attack using pacing, a technique that absorbs traffic bursts and outputs traffic in a controlled manner. In particular, we propose a novel pacing scheme called two-stage pacing, which combines Markov ON-OFF pacing and rate limiting. Markov ON-OFF pacing leverages randomness to break down the inter-flow correlation by emitting traffic bursts at different times using an alternating Markov ON-OFF switch, while rate limiting further suppresses the peak rates of traffic bursts. Two-stage pacing seeks to minimize traffic burstiness and hence reduce the average router queue length.

To summarize, this paper makes the following contributions. First, using PCSDEs as a building block for our modeling framework, we analyze how the correlation attack increases the average router queue length by exploiting the inter-flow correlation, and analyze how our two-stage pacing
scheme mitigates the correlation attack. Second, we verify that both our fluid models and the packet-level ns2 simulation give very similar average router queue length results, while our fluid models take much less time to solve. The advantage of using fluid models to capture queueing dynamics is first suggested by [15]. In our context, the implication is that our fluid models enable us to quickly explore the relationship between the inter-flow correlation and the average router queue length.

The remainder of the paper proceeds as follows. Section II formulates a fluid model for the correlation attack, and Section III analyzes the correlation attack under different levels of correlation. Section IV proposes a two-stage pacing scheme that mitigates the correlation attack. Section V reviews related work, and finally Section VI concludes.

II. PROBLEM FORMULATION

In this section, we propose a fluid model that describes the dynamics of the average router queue length, based on Poisson Counter Driven Stochastic Differential Equations (PCSDEs) [3], [9], [10], [15], [23]. Our goal is to provide insights into how the average router queue length increases as the magnitude of the inter-flow correlation increases. Finally, we formulate the correlation attack that exploits the inter-flow correlation to bring damage to a router queue. Table I summarizes the major notation used in the paper.

A. Single Router Queue

Figure 1 depicts a simple topology that guides our analysis. Suppose that there are $n$ flows, labeled $1, 2, \ldots, n$, that send traffic to a target router, where the traffic is modeled as a fluid. Each flow $i$, where $1 \leq i \leq n$, is modeled by a stochastic ON-OFF process $x_i(t)$ with peak rate $h_i$, where $x_i(t)$ equals 1 and 0 if flow $i$ is in the ON and OFF states, respectively. Also, each flow $i$ traverses a link with latency $\tau_i$. For the target router, $c$ be its outgoing link bandwidth, and let $v(t)$ be its instantaneous queue length at time $t$.

We can now model $v(t)$ with the following stochastic differential equation [3]:

$$dv(t) = -c I_v(t) dt + \sum_{i=1}^{n} h_i x_i(t) dt,$$  \hfill (1)

where $I_v$ is an indicator function that equals 1 if some value $v > 0$, or 0 otherwise. Note that Equation (1) holds for arbitrary ON-OFF distributions of $x_i(t)$.

Equation (1) assumes that the queue capacity is infinite. Although many studies (e.g., [2], [13]) advocate small router queues, today’s commercial routers still provision very large queue capacities to maximize the network throughput [2], and this motivates us to assume an infinitely large queue. Note that we can easily incorporate the finite queue condition into Equation (1) (see the technical report of [3]).

To analyze the inter-flow correlation without loss of generality, we consider a special case where each flow $i$ ($1 \leq i \leq n$) sends a stream of ON-OFF UDP traffic that is unresponsive to network congestion. According to [9], we can model $x_i(t)$ as a Markov ON-OFF process, where both ON and OFF periods are exponentially distributed. We can describe $x_i(t)$ using a PCSDE as:

$$dx_i(t) = (1 - x_i(t)) dN_{i1} - x_i(t) dN_{i2},$$  \hfill (2)

where $i = 1, 2, \ldots, n$, $x_i(0) \in \{0, 1\}$, and $N_{i1}$ and $N_{i2}$ are Poisson counters that drive the ON and OFF processes with rates $\lambda_{i1}$ and $\lambda_{i2}$, respectively. The average ON and OFF periods are given by $\lambda_{i2}^{-1}$ and $\lambda_{i1}^{-1}$, respectively. Taking expectations of both sides in (2) gives

$$\frac{d}{dt} E[x_i(t)] = (1 - E[x_i(t)]) \lambda_{i1} - E[x_i(t)] \lambda_{i2}. $$

In steady state, $E[x_i(t)] = \lambda_{i1}/(\lambda_{i1} + \lambda_{i2})$.

We define the correlation function of two processes $x_i(t)$ and $x_j(t)$ as $E[x_i(t)x_j(t)]$, which is also equal to the probability that both $x_i(t) = 1$ and $x_j(t) = 1$. We say that flows $i$ and $j$ have a higher inter-flow correlation if $E[x_i(t)x_j(t)]$ is larger. In particular, flows $i$ and $j$ are said to be independent (or uncorrelated) if $E[x_i(t)x_j(t)] = E[x_i(t)]E[x_j(t)]$.

The following theorem shows how the inter-flow correlation is related to the steady-state average queue length $E[v(t)]$ under specific conditions.
that the average queue length \(E_v\) the steady-state expected value of the sum of the pairwise inter-flow correlations of \(i\), i.e.,

\[
E[v] = \frac{1}{c - \sum_{i=1}^{n} h_i E[x_i] \left[ \sum_{i=1}^{n} \frac{h_i}{\lambda_{i1} + \lambda_{i2}} \left( (h_i - c) E[x_i] + \sum_{j=1, j \neq i}^{n} h_j E[x_i x_j] \right) \right]}
\]

**Remark**: The proof is in Appendix. Equation (1) shows that the average queue length \(E[v]\) is an increasing function of the sum of the pairwise inter-flow correlations, i.e., increasing the correlation of any two flows \(i\) and \(j\) (i.e., \(E[x_i x_j]\)) will increase \(E[v]\) while keeping \(E[x_i]\) constant.

### B. Tandem Queue Network

We next consider a tandem queue network where flows traverse more than one router. Previous work \cite{3, 10} on the tandem queue network only focuses on a single flow. We extend the analysis to multiple flows. We demonstrate how the inter-flow correlation has a cascading impact on a tandem queue network, as it increases the average queue lengths of all the router queues in the tandem queue network.

Figure 2 depicts a tandem queue network, in which flows traverse \(m\) routers, labeled by 1, 2, \cdots, \(m\). For \(1 \leq k \leq m\), let \(c_k\) be the outgoing link bandwidth of router \(k\), and \(v_k(t)\) be the average queue length of the router \(k\) at time \(t\). Without loss of generality, we assume that \(c_1 > c_2 > \cdots > c_m > 0\) so that we observe queued packets in every router (if a downstream router has a higher link capacity than its upstream router, it can be ignored in our analysis as it has no queued packets).

To simplify our models, we assume that every flow \(i\) is a Markov ON-OFF flow and the latency of every link is zero. Note that such assumptions can be easily relaxed based on our models in Section II-A. Then we can describe the queue lengths of all routers using the following PCSDEs (where we omit the parameter \(t\) for brevity):

\[
\begin{align*}
\frac{dx_1}{dt} &= (1 - x_1) dN_{11} - x_1 dN_{12}, i = 1, 2, \cdots, n \\
\frac{dv_1}{dt} &= -c_1 I_{v_1} dt + \sum_{i=1}^{n} h_i x_i dt \\
\frac{dv_2}{dt} &= -c_2 I_{v_2} dt + c_1 I_{v_1} dt \\
&\vdots \\
\frac{dv_m}{dt} &= -c_m I_{v_m} dt + c_{m-1} I_{v_{m-1}} dt.
\end{align*}
\]

**Theorem 2**: Suppose that: (i) \(h_i > c_i > c_2 > \cdots > c_m\) for all \(i\), meaning that every ON burst of a single flow always creates queued packets in every router, and (ii) \(\sum_{i=1}^{n} h_i E[x_i(t)] < c_m < \cdots < c_1\), meaning that every router queue is stable. Then in steady state, \(E[v_1]\) is given by Theorem 1 with \(c\) replaced with \(c_1\), and for \(1 < k \leq m\),

\[
E[v_k] = \frac{(c_{k-1} - c_k)}{(c_{k-1} - \sum_{i=1}^{n} h_i E[x_i]) (c_k - \sum_{i=1}^{n} h_i E[x_i])}
\]

\[
\sum_{i=1}^{n} \frac{h_i}{\lambda_{i1} + \lambda_{i2}} \left( h_i (E[x_i] - E^2[x_i]) + \sum_{j=1, j \neq i}^{n} h_j (E[x_i x_j] - E[x_i] E[x_j]) \right).
\]

**Remark**: The proof is in Appendix. Note that even if \(E[x_i]\) is kept constant, increasing the correlation function \(E[x_i x_j]\) still increases the average queue lengths of all router queues in the tandem queue network.

### C. Correlation Attack

In the correlation attack, an attacker intentionally generates highly correlated attack flows from different sources toward a target router (e.g., the default gateway of a campus network). While the delays from different sources to the target router are generally different, the attacker can estimate the source-to-router delays using active probing \cite{12, 17} and synchronize the attack flows subject to the delay differences. The goal of the correlation attack is to increase the average queue length of the target router’s queue, thereby disrupting the normal flows that also traverse the target router. Instead of generating an aggressive amount of attack traffic, the correlation attack is characterized by a low volume of traffic, as it generates multiple streams of ON-OFF bursts. With the correlation among the attack flows, the aggregation of the ON bursts forms a high-peak-rate burst that significantly overloads a router queue.

To better control the correlation among the attack flows and the attack intensity, we assume that the attacker generates UDP ON-OFF bursts that are unresponsive to network congestion. In view of this, we model an attack flow by a Markov ON-OFF process \cite{9}. For special parameter settings, we can show via closed-form solutions, as in Theorems 1 and 2, how the inter-flow correlation increases the average queue length of a router queue. While our fluid models can be readily extended for general ON-OFF processes and parameters, closed-form solutions are not available due to the complexity of the cases where a burst of an individual flow, or an aggregate burst of multiple flows, does not create queued packets. Thus, in Section III, we resort to numerical simulation to solve the differential equations, and analyze the impact of different levels of the inter-flow correlation exploited by the correlation attack.

The fundamental characteristic of the correlation attack is to introduce traffic burstiness into the target router by exploiting the inter-flow correlation. We emphasize that the correlation attack does not restrict the distribution of each attack flow, as its goal is to introduce the inter-flow correlation across different attack flows. In fact, an attacker can also
exploit the autocorrelation (i.e., intra-flow correlation) to further aggravate the traffic burstiness. Intuitively, the attacker can generate a very long burst based on some heavy-tail ON-OFF distribution, which exhibits higher autocorrelation than an exponential ON-OFF distribution [8]. We pose the autocorrelation analysis as future work.

III. IMPACT OF THE CORRELATION ATTACK

In this section, we analyze the impact of the correlation attack. We conducted numerical simulation using our fluid modeling (Section II), and packet-level simulation using the ns2 simulator [21]. Our goals are: (i) to understand how the inter-flow correlation increases the average router queue length for general parameters, and (ii) to verify that both our fluid modeling and packet-level ns2 simulation return very similar average queue length results.

We focus on the correlation attack that sends Markov ON-OFF UDP attack traffic that is unresponsive to network congestion [9]. We consider three attack cases that account for different levels of the inter-flow correlation: (i) independent, in which all attack flows are driven by independent ON and OFF Poisson counters, (ii) weakly correlated, in which all attack flows are driven by a single ON Poisson counter but independent OFF Poisson counters, and (iii) identical, in which all attack flows are driven by the same ON and OFF Poisson counters. Using Itô’s rule, we can derive the correlation function for any two flows $i$ and $j$ for the three attack cases as:

$$E[x_i,x_j] = \begin{cases} \frac{\lambda_{1i}}{\lambda_{1i} + \lambda_{2i}} \times \frac{\lambda_{1j}}{\lambda_{1j} + \lambda_{2j}} & \text{independent,} \\ \frac{\lambda_{1i}}{\lambda_{1i} + \lambda_{2i}} \times \frac{\lambda_{1j}}{\lambda_{1j} + \lambda_{2j}} \times f_{ij} & \text{weakly correlated,} \\ \frac{\lambda_{1i}}{\lambda_{1i} + \lambda_{2i}} & \text{identical.} \end{cases}$$

As clearly shown, among the three cases, the independent case has the smallest $E[x_i,x_j]$, while the identical case has the largest $E[x_i,x_j]$. In fact, the identical case maximizes $E[x_i,x_j]$ over all possible pairs of $x_i$ and $x_j$, as $E[x_i,x_j] = \Pr(x_i = 1, x_j = 1) = \Pr(x_i = 1) = E[x_i] = \lambda_{1i}/(\lambda_{1i} + \lambda_{2i})$. Thus, the identical case is expected to produce the highest average router queue length, as will be confirmed by our experiments.

Before describing our experiments, we point out that we implement each Markov ON-OFF flow in ns2 based on the Deterministic-2 model in [7]. That is, each flow $i$ generates the first packet when it transitions from the OFF state to the ON state, and the inter-packet time (denoted by $T_{\text{inter}}$) is fixed to be the ratio of packet size to the peak sending rate (i.e., $h_i$). Also, the flow triggers a packet transmission as long as it is in the ON state. Thus, the number of packets generated during an ON period $T_{ON}$ is given by $\lceil T_{ON}/T_{\text{inter}} \rceil$. The rounding-up effect causes the average queue length values obtained from ns2 to be slightly larger than those obtained from the fluid model, yet the difference is expected to be small.

**Default parameter settings:** Unless otherwise stated, the following experiments assume that each attack flow $i$ (which is a Markov ON-OFF UDP flow) generates 1KB packets during the ON period. We set the average ON and OFF periods to be 1s and 4s, respectively, and hence $\lambda_{1i} = 0.25s^{-1}$ and $\lambda_{2i} = 1s^{-1}$. Also, we set $h_i = 0.4\text{Mbps}$, and $c = 10\text{Mbps}$. We assume zero link latency (i.e., $\tau_i = 0$).

**Experiment A.1 (Impact of the number of attack flows):** We first study how the average router queue length varies with the number of attack flows. We focus on the single queue topology in Figure 1 (see Section II), where all flows are assumed to be attack flows due to the correlation attack. Figure 3 shows the average router queue length versus the number of attack flows. We observe that the results from both the fluid model and ns2 simulation conform to each other. Also, both the weakly correlated and identical cases significantly increase the average queue length over the independent case by introducing the inter-flow correlation. In particular, the identical case, which introduces the highest inter-flow correlation, has at least $2\times$ average queue length over the weakly correlated case.

**Experiment A.2 (Impact of delay jitter):** Since the correlation attack requires that attack flows be highly correlated when their traffic streams reach the target router, the attacker needs to estimate the link latency from each attack source to the target router so that the attack flows are synchronized at the target router. In practice, such estimation is inaccurate, and the jitter (i.e., the variation of the link latency) may reduce the inter-flow correlation at the target router. In this experiment, we study the impact of the delay jitter.

To understand the effect of the delay jitter, we can equivalently assume that the attacker estimates zero latency for each link, but actually each link has a non-zero latency. Thus, we let $\tau_i$, the link latency from the attack source $i$ to the target router, be a random variable selected from a uniform distribution from 0 to $T$ for some maximum link latency $T$. We then include $\tau_i$ in our fluid model and ns2 simulation. We again focus on the single queue topology in Figure 1.

Figure 4 depicts the average router queue length versus the maximum link latency $T$, where the number of attack flows is fixed at 60. Again, both the fluid model and ns2 simulation give very similar results. We note that the increase in the link latency decreases the average queue length, mainly because the inter-flow correlation at the target router is also reduced. However, the resulting average queue length remains significantly higher than that in the independent case. This implies that the correlation attack remains robust toward the delay jitter.

**Experiment A.3 (Impact on a tandem queue network):**

We now analyze the impact of the correlation attack on each router queue in a tandem queue network based on the topology in Figure 2. We assume that the network contains $n = 5$ routers. We assume zero link latency between adjacent routers, and we set the outgoing link capacity of router $k$ to be $(10 - (k - 1))\text{Mbps}$, where $1 \leq k \leq 5$. Figure 5 shows the average router queue length at each router queue for a 5-router tandem queue network, where the number of attack flows is fixed at 60. Note that the change in the average queue lengths across router queues depend on the choices of parameters. In this particular parameter setting, the average queue lengths of the weakly correlated and identical cases first drop at router 2, and then keep increasing at subsequent routers. The main observation is that both weakly correlated and identical cases increase the average queue length at every router compared to the independent cases, and the results are consistent with Theorem 2.
Experiment A.4 (Impact on normal flows): We now explore how the correlation attack increases end-to-end transfer delays of normal flows, as a result of the increased average router queue length. Since our current fluid models do not consider normal traffic, we only use ns2 simulation here. We focus on the tandem queue network in Figure 2, based on the parameter setting of Experiment A.3. In addition, we create 70 flows, 10 of which are TCP flows and the remaining 60 are Markov ON-OFF UDP attack flows. Here, we measure the round-trip time (RTT) of a TCP segment, defined as the elapsed time from when the source sends the segment until the sender receives the acknowledgement for the segment. Each TCP source continuously sends 500-byte segments, one by one, and it sends the next segment only when the last sent segment has been acknowledged. We assume that all links, including those connecting the flow sources to routers and those between adjacent routers, have latency 5 ms.

Figure 6 shows the average RTT of a TCP segment versus $m$, the number of router queues in a tandem queue network, for different attack cases. Both weakly correlated and identical cases increase the RTT over the independent cases, for example, by at least 3× when $m = 5$.

Note that the RTT of a segment can significantly increase if the transmission of the segment overlaps with the attack bursts generated by the correlated attack. Figure 7 shows the RTTs of the first 5000 segments for a particular TCP flow in different attack cases. We observe spikes in both weakly correlated and identical cases. We note that the RTTs of more than 4% packets are greater than 1 s, and some are even greater than 10 s. Such spikes imply that real-time applications unexpectedly observe occasional disruptions.

IV. MITIGATING THE CORRELATION ATTACK

A fundamental characteristic of the correlation attack is to generate attack bursts from multiple attack sources in a correlated manner, such that the aggregation of the attack bursts forms a high-peak-rate burst at the target router. Thus, in order to mitigate the correlation attack, our goal is to pace high-rate bursts into smaller-rate bursts so that the target router has enough time to absorb and forward traffic bursts without queueing too many packets.

In this section, we propose one possible defense strategy called two-stage pacing that mitigates the correlation attack. In a nutshell, our two-stage pacing scheme, as its name suggests, is composed of two stages: (i) Markov ON-OFF pacing, which breaks down the correlation among multiple bursts, and (ii) rate-limiting pacing, which suppresses the peak rate of any burst. In the following, we present our two-stage pacing scheme, and address its deployment and implementation issues. We then model the two-stage pacing scheme using PCSDE-based fluid modeling. Finally, we evaluate our two-stage pacing scheme via both fluid-level and packet-level simulations.

A. Overview of Two-Stage Pacing

To mitigate traffic bursts, rate limiting has been implemented (e.g., based on the leaky-bucket algorithm) in commercial routers to pace traffic bursts while ensuring that a certain average throughput can be sustained. However, rate limiting itself is not designed to minimize the inter-flow correlation. Given that the inter-flow correlation can significantly increase the average router queue length (see Section III), we need a new pacing component, in addition to rate limiting, to break down the inter-flow correlation.

Here, we propose a two-stage pacing scheme, as shown in Figure 8, that includes Markov ON-OFF pacing in addition
to rate limiting. Markov ON-OFF pacing aims to introduce randomness when forwarding traffic, and it forwards correlated bursts at different time periods so as to break down the inter-flow correlation. To achieve this, each Markov ON-OFF pacer independently alternates between the ON and OFF states, whose durations are exponentially distributed. It forwards packets if it is in the ON state, or holds packets in a queue if it is in the OFF state. In our two-stage pacing scheme, data traffic is first paced by a Markov ON-OFF pacer, and is then forwarded to a conventional rate-limiting pacer, which bounds the peak rate of the data traffic to be forwarded to the target router.

While there could be other pacing schemes that can reduce the inter-flow correlation, Markov ON-OFF pacing offers an additional advantage in that its autocorrelation function for its ON-OFF process is light-tailed [9], [10]. Thus, although Markov ON-OFF pacing is motivated as a way for reducing inter-flow correlation, it can also reduce traffic burstiness due to high autocorrelation (intra-flow correlation), which is typically seen in practice [14], [6]. Intuitively, with a high enough switching rate, Markov ON-OFF pacing chops a long (heavy-tailed) traffic burst into small (light-tailed) bursts and forwards the bursts at different time periods. Thus, two-stage pacing still sees its potential in practical scenarios. We plan to investigate this in future work.

B. Deployment and Implementation Issues

Since our goal is to avoid having high-rate bursts at the target routers that we want to protect, it is important to pace incoming bursts before they arrive at the target routers and are aggregated to form high-rate bursts. Thus, we suggest to deploy pacing at upstream routers that connect to the downstream routers to be protected.

Figure 9 extends our prior tandem queue network in Figure 2 and illustrates the deployment points of pacing. We point out that our two-stage pacing scheme is designed to work independently, and thus pacing routers do not need to coordinate with others. Our pacing architecture is similar to the defense architecture in [24], which places rate throttling at the upstream routers and defends against flooding-based DoS attacks on downstream servers.

In the following, we address the challenges of deploying our two-stage pacing scheme in practice, and we pose these issues as future work.

In Figure 9 and our following evaluation, we assume that every upstream router that connects to the downstream protected router has two-stage pacing deployed. We emphasize that by no means do we claim this assumption is practical. We expect that in practice, the deployment of pacing will be incremental and start with a subset of routers. It is thus important to analyze the effectiveness of two-stage pacing under incremental deployment. Nevertheless, our focus here is to explore the fundamental property of two-stage pacing in mitigating the inter-flow correlation, and hence we consider the full deployment of pacing routers.

While two-stage pacing seeks to reduce queuing delays in the protected routers, it adds delays to data transmissions in the pacing routers, as the Markov ON-OFF pacer forwards traffic only if it switches to the ON state, and the rate-limiting pacer reduces the packet transmission rate. To address this tradeoff, we propose that our two-stage pacing scheme “whitelists” certain classes of traffic that will be unpaced. Unpaced traffic is queued in a separate queue and receives a higher priority of being forwarded than paced traffic. For example, TCP flows are generally uncorrelated among themselves, as small variations of RTTs inherently break down the inter-flow correlation [2], [18]. Thus, we propose that TCP traffic be unpaced. Also, some UDP applications such as VPN and streaming likely belong to normal traffic, and should also be unpaced. We argue that the whitelisting feature of a router can be realized, as current commercial routers (e.g., [5]) already support deep packet inspection at line rate. Nevertheless, as shown in Section IV-D, even when normal traffic is paced, two-stage pacing is capable of removing delay spikes (see Section III, Experiment A.4).

C. Fluid Models for Two-Stage Pacing

We now formulate a fluid model using PCSDEs for our two-stage pacing scheme. We refer readers to Table I in Section II for additional notation defined for pacing.

We use the single router topology in Figure 1 (see Section II) to model two-stage pacing. For a two-stage pacer $i$ deployed between flow $i$ and the target router, let $N_{i3}$ and $N_{i4}$ be the Poisson counters that drive the Markov ON-OFF pacer $i$ to the ON and OFF states with rates $\lambda_{i3}$ and $\lambda_{i4}$, respectively. Thus, the ON and OFF periods of the pacer are exponentially distributed with the averages $\lambda_{i3}^{-1}$ and $\lambda_{i4}^{-1}$, respectively. Also, let $c_i^m$ and $c_i^r$ be the outgoing capacities, and $v_i^m(t)$ and $v_i^r(t)$ be the instantaneous queue lengths at time $t$, of the Markov ON-OFF pacer $i$ and the rate-limiting pacer $i$, respectively. In two-stage pacing, we assume that $c_i^m = h_i$, so that the Markov ON-OFF pacer $i$ keeps the traffic peak rate and paces traffic only via an ON-OFF process. We also assume that $c_i^m \geq c_i^r$, so that the rate-limiting pacer $i$ can actually suppress the peak rate of the incoming traffic. To simplify our models, we assume that the link latency $\tau_i = 0$. 

![Fig. 8. Two-stage pacing deployed in a router](image)

![Fig. 9. A pacing architecture: to protect the target router, pacing is deployed at the upstream routers.](image)
Now, we can formulate the PCSDEs for the two-stage pacing scheme (for brevity, we omit parameter \( t \)):

\[
\begin{align*}
(a) \quad dz_i &= (1 - z_i) dN_{i3} - z_i dN_{i4}, \quad i = 1, \ldots, n, \\
(b) \quad dv_{i}^m &= -h_i I_{v}^m z_i dt + h_i x_i dt, \quad i = 1, \ldots, n, \\
(c) \quad dv_{i}^c &= -c_i^m I_{v}^c dt + h_i I_{v}^m z_i dt, \quad i = 1, \ldots, n, \\
(d) \quad dv &= -c I_{v} dt + \sum_{i=1}^{n} c_i^m I_{v}^c dt.
\end{align*}
\]

In Equation (4a), \( z_i \in \{0, 1\} \) represents the state of the Markov ON-OFF pacer \( i \), which forwards traffic only when \( z_i = 1 \). Equations (4b), (4c), and (4d) describe the dynamics of the queues located within the Markov ON-OFF pacer \( i \) and the rate-limiting pacer \( i \), and the target router, respectively.

Note that the two-stage pacer can be reduced to one of its two pacing components. If we set \( \lambda_{i4} = 0 \), then the Markov ON-OFF pacer is always in the ON state, and hence the two-stage pacer is reduced to a rate-limiting pacer:

\[
\begin{align*}
(dv_i^c &= -c_i^m I_{v}^c dt + h_i x_i dt, \quad i = 1, \ldots, n, \\
(dv &= -c I_{v} dt + \sum_{i=1}^{n} c_i^m I_{v}^c dt.
\end{align*}
\]

On the other hand, if we set \( c_i^c > c_i^m \), the rate-limiting component is disabled. Thus, the two-stage pacer is reduced to a Markov ON-OFF pacer (note that we assume \( c_i^m = h_i \)):

\[
\begin{align*}
(dz_i &= (1 - z_i) dN_{i3} - z_i dN_{i4}, \quad i = 1, \ldots, n, \\
(dv_i^m &= -h_i I_{v}^m z_i dt + h_i x_i dt, \quad i = 1, \ldots, n, \\
(dv &= -c I_{v} dt + \sum_{i=1}^{n} h_i I_{v}^m z_i dt.
\end{align*}
\]

D. Evaluation

In this subsection, we evaluate three pacing schemes: rate limiting, Markov ON-OFF pacing, and two-stage pacing. Our goal is to verify whether they can reduce the average queue length of a router under the correlation attack.

**Experiment B.1 (Comparison of pacing schemes with different outgoing capacities):** We first evaluate the pacing schemes with different outgoing capacities based on the single router topology in Figure 1 (see Section II). Note that for pacer \( i \) deployed between flow \( i \) and the target router, its outgoing capacity is given by:

\[
\begin{align*}
\{ &c_i^c \quad \text{rate limiting only}, \\
h_i &\lambda_{i3} \quad \text{Markov ON-OFF pacing only}, \\
\min(c_i^c, h_i &\lambda_{i3} + \lambda_{i4}) \quad \text{two-stage pacing}.
\end{align*}
\]

Ideally, the outgoing capacity of a pacer should approximate the average rate of incoming traffic so as to reduce queued packets at the target router. However, the pacer's outgoing capacity is usually overestimated in practice to prepare possible increases in incoming traffic rates. We thus analyze the performance of the pacing schemes for different outgoing capacities.

Our parameter settings are based on Experiment A.1 (see Section III). We assume that all flows are attack flows with Markov ON-OFF UDP traffic. We consider 60 attack flows, each of which has \( h_i = 0.4 \text{Mbps}, \lambda_{1i} = 0.25 \text{s}^{-1} \) and \( \lambda_{12} = 1 \text{s}^{-1} \), and we set \( c = 10 \text{Mbps} \) for the outgoing link capacity of the target router. Specifically, for the Markov ON-OFF pacer and the two-stage pacer, we fix \( \lambda_{13} = \lambda_{14} = 5 \text{s}^{-1} \). Note that the inverse \( (\lambda_{13} + \lambda_{14})^{-1} \) denotes the autocorrelation time constant [10] of a Markov ON-OFF process. Intuitively, the larger the sum \( \lambda_{13} + \lambda_{14} \) is, the faster the switch alternates between the ON and OFF states. We vary the outgoing capacities of all pacers from \( h_i E[x_i] \) to \( h_i \). For fairness, all pacers use the same outgoing capacity in each comparison.

Figure 10 shows the average router queue length versus the outgoing capacities of the pacers for different attack cases (independent, weakly correlated, and identical). We observe that if the outgoing capacity of each pacer \( i \) is closer to the average rate \( h_i E[x_i] \), the average router queue length is reduced more. Also, the two-stage pacer always has a smaller average router queue length than the other two pacers, for example, by at least 60% in the identical attack case when each pacer has outgoing capacity 0.2Mbps. Also, both our fluid models and ns2 give very similar average queue lengths.

**Experiment B.2 (Evaluation of Markov ON-OFF pacing with different switch rates):** We now evaluate the effect of various Markov ON-OFF switch rates, i.e., \( \lambda_{13} \) and \( \lambda_{14} \), for Markov ON-OFF pacing and two-stage pacing. We use the parameter settings in Experiment B.1. Here, we fix the outgoing capacities of each pacer to be 0.2Mbps (note that the traffic peak rate is \( h_i = 0.4 \text{Mbps} \)), and we vary \( \lambda_{13} + \lambda_{14} \).

In the interest of space, we only consider the results of the weakly correlated and identical attack cases obtained from the fluid models. These are found in Figure 11. For both Markov ON-OFF pacing and two-stage pacing, the average router queue length increases with the sum \( \lambda_{13} + \lambda_{14} \) and hence the Markov ON-OFF switch rate. To understand
intuitively, a higher switch rate implies a shorter waiting time for a burst to be forwarded by the Markov ON-OFF pacer. If correlated bursts are forwarded at about the same time, they remain overlapped with each other and are still aggregated to a huge burst at the router. Note that regardless of the switching rate, two-stage pacing still gives a smaller average router queue length than rate-limiting pacing.

**Experiment B.3 (Pacing effect on normal traffic):** We now study whether pacing can reduce the end-to-end transfer delays of normal flows. In Section IV-B, we propose to whitelist some traffic classes, such as TCP, that will be unpaced. Here, we argue that pacing is still useful in removing the delay spikes (see Experiment A.4 in Section III) even if TCP traffic is paced.

We apply two-stage pacing to the tandem queue network in Experiment A.4, which contains 60 attack flows and 10 normal TCP flows. We pace all attack flows, and we consider the cases when TCP is unpaced, or paced by two-stage pacing. We set the outgoing capacity of each pacer to be 0.1Mbps, which is 25% higher than the average rate of each pacing. We set the outgoing capacity of each pacer to be 0.1Mbps, which is 25% higher than the average rate of each TCP flow (given by $h_{a}E[x_i] = 0.08Mbpss$). Since we also consider the case where TCP is paced, we pick $\lambda_{i3} = 20s^{-1}$ and $\lambda_{i4} = 60s^{-1}$, such that the Markov ON-OFF switch rate is high enough to minimize the waiting time of each TCP segment. Here, we only focus on the identical attack case.

Figure 12 shows the average RTT of a TCP segment versus $m$, the number of router queues in a tandem queue network. As expected, if we can keep TCP unpaced, we can reduce the average router queue length by a higher percentage. As compared to the no-pacing case, two-stage pacing reduces the average RTT of a TCP segment in most cases, for example, by 30% and 10% when TCP is unpaced and paced when $m = 5$, respectively.

Most importantly, the two-stage pacer can remove the delay spikes observed in Experiment A.4. Figure 13 shows the RTT distribution of the first 5000 TCP segments of a particular TCP flow. It shows the absence of spikes when TCP flows are unpaced or paced, mainly because two-stage pacing paces the attack flows and reduces the average router queue length.

V. RELATED WORK

Traditional DoS attacks are flooding-based and generate high-rate attack traffic. This high-rate nature is exploited by defense mechanisms [11], [19], [24]. A low-rate DoS attack is first proposed in [12] and is characterized by the low average volume of attack traffic. It uses periodic, high-rate bursts to synchronize all flows to enter the same retransmission timeout (RTO) state. Another low-rate attack [4] uses periodic bursts to cluster together server requests to overload a server. They can be mitigated by randomizing the RTO value or the server request interval [12], [4], or by throttling the sources that generate periodic bursts at some suspicious rate [20]. On the other hand, the correlation attack considered here does not require periodicity of attack bursts, provided that the bursts from multiple attack flows are correlated. Also, previous studies [12], [4] mainly study the low-rate attacks using empirical studies, while we provide an analytical framework that formally models the impact of the inter-flow correlation on a router queue.

Our work leverages PCSDE, which has been used to model queueing dynamics (e.g., [3], [10], [23]) and TCP flows (e.g., [9], [15]). Our work also models queueing dynamics, but we focus on how the inter-flow correlation affects the queueing dynamics in the security context.

Pacing is also used in the TCP context to correct the compression of acknowledgements due to cross traffic [25] or to avoid slow start [16]. Aggarwal et al. [1] study the performance of TCP pacing over different networks. As opposed to prior work on TCP pacing, we propose to use pacing to reduce the inter-flow correlation.

VI. CONCLUSIONS

This paper considers a low-rate correlation attack that exploits the inter-flow correlation to introduce traffic burstiness into the attacked router so as to increase the average router queue length. We develop a fluid model for the correlation attack based on Poisson Counter Stochastic Differential Equations. We verify the correctness of our fluid model using ns2 simulation. Also, we show that the average router queue length with highly correlated attack flows is significantly higher than that with uncorrelated flows. Finally, we propose
and model a two-stage pacing scheme that mitigates the correlation attack.

**APPENDIX**

A. Proof of Theorem 1

For brevity, we omit $t$ in our derivation. To compute $E[v]$, we start by deriving $dv^2$:

$$dv^2 = 2vdv = -2cvI_v dt + 2 \sum_{i=1}^{n} h_i x_i vd t.$$  

We take expectations on both sides and assume that the steady state exists, i.e., $\frac{dE[v^2]}{dt} = 0$. Note that $E[v I_v] = E[v]$. Thus,

$$cE[v] = \sum_{i=1}^{n} h_i E[x_i v_i].$$

We next derive $E[x_i v]$:

$$dx_i v = x_i dv + vdx_i = x_i(-cI_v dt + \sum_{j=1}^{n} h_j x_j dt) + v((1 - x_i) dN_{i1} - x_i dN_{i2})$$  

$$= -cx_i dt + \sum_{j=1}^{n} h_j x_j dt + vdN_{i1} - v x_i(dN_{i1} + dN_{i2}).$$

Note that $x_i I_v = x_i$, as whenever a burst arrives (i.e., $x_i = 1$), the queue is built up (i.e., $I_v = 1$). Again, we take expectations on both sides and assume that we have the steady state $\frac{dE[x_i v]}{dt} = 0$. Thus,

$$E[x_i v] = \frac{1}{\lambda_1 + \lambda_2} (\lambda_1 E[v] - cE[x_i] + \sum_{j=1}^{n} h_j E[x_i x_j]).$$

Substituting it into the equation of $cE[v]$, we have

$$cE[v] = \sum_{i=1}^{n} \frac{h_i}{\lambda_1 + \lambda_2} (\lambda_1 E[v] - cE[x_i] + h_i E[x_i^2] + \sum_{j=1, j \neq i}^{n} h_j E[x_i x_j]).$$

Through algebraic operations, and observing that $E[x_i^2] = E[x_i]$, the theorem follows.

B. Proof of Theorem 2

By summing all $dv_k$’s, where $1 \leq k \leq m$, in Equation (3) (see Section II-B), we have

$$\begin{align*}
\frac{dx_i}{dt} &= (1 - x_i) dN_{i1} - x_i dN_{i2}, i = 1, 2, \ldots, n \\
\frac{dv_i}{dt} &= -cm v_i dt + \sum_{i=1}^{n} h_i x_i dt.
\end{align*}$$

The expected value $E[\sum_{i=1}^{n} v_i]$ can be computed using the result of the single queue case in Theorem 1, with the outgoing link capacity set to be $c_m$. Thus, we can compute $E[v_1]$ by setting $m = 1$, and for $1 < k \leq m$, we can set $E[v_k] = E[\sum_{i=1}^{n} v_i] - E[\sum_{i=1}^{n} v_i]$. The theorem then follows.

**REFERENCES**


