FR-WARD: Fast Retransmit as a Wary but Ample Response to Distributed Denial-of-Service Attacks from the Internet of Things

Samuel Mergendahl
University of Oregon
Eugene, OR
smergend@cs.uoregon.edu

Devkishen Sisodia
University of Oregon
Eugene, OR
dsisodia@cs.uoregon.edu

Jun Li
University of Oregon
Eugene, OR
lijun@cs.uoregon.edu

Hasan Cam
Army Research Labs
Adelphi, MD
hasan.cam.civ@mail.mil

Abstract—**While the Internet of Things (IoT) becomes increasingly popular and ubiquitous, IoT devices often remain unprotected and can be exploited to launch large-scale distributed denial-of-service (DDoS) attacks. One could attempt to employ traditional DDoS defense solutions, but these solutions are hardly suitable in IoT environments since they seldom consider the resource constraints of IoT devices.**

We present FR-WARD, a system that defends against DDoS attacks launched from an IoT network. FR-WARD operates close to potential attack sources at the gateway of an IoT network and drops packets to throttle any DDoS traffic that attempts to leave the IoT network. However, in order to properly react to traffic too difficult to categorically label as good or bad, FR-WARD employs a novel response based on the fast retransmit and flow control mechanisms of the Transmission Control Protocol (TCP) which minimizes the energy consumption and network latency of benign IoT devices within the policed network.

Based on our mathematical analysis, simulation, and experimental evaluation, FR-WARD not only effectively mitigates DDoS traffic, but also minimizes the number of retransmitted packets and the connection durations of benign IoT devices. In fact, FR-WARD can successfully mitigate both naive flood attacks and smarter DDoS attacks that follow TCP congestion control but still reduce overhead caused by retransmitted packets for benign IoT devices by a up to a factor of 150.

I. Introduction

While the Internet of Things (IoT) rapidly expands in size and capability, IoT devices commonly become compromised, form botnets, and launch attacks. In particular, these botnets—often of a considerable size—can perform destructive distributed denial-of-service (DDoS) attacks. For example, in October 2016, an IoT botnet, Mirai, attacked and disabled a major domain name service (DNS) provider, Dyn, on which many Fortune 500 companies rely [1]. The botnet contained an estimated 100,000 malicious IoT devices and the attack achieved upwards of 1 terabits per second (Tbps) of DDoS traffic [2]. While the Dyn attack used mostly User Datagram Protocol (UDP)-based traffic, Lyu et al. also emphasize the iminent need to address Transmission Control Protocol (TCP)-based reflective attacks from IoT devices [3].

A promising approach against these types of DDoS attacks is to employ a *source-end* defense solution that detects and thwarts attack traffic before the traffic leaves its original network [4]–[6]. Such a solution can establish normal traffic profiles and check future traffic for anomalies to detect DDoS traffic initiated from inside the policed network. Moreover, in order to mitigate a suspected DDoS attack, the source-end defense solution can throttle or even completely filter any suspicious traffic.

However, IoT networks that deploy a source-end DDoS defense solution face a severe challenge; such a defense solution can mistakenly throttle or filter benign traffic. In fact, traditional DDoS defense solutions often mislabel some benign traffic as malicious while they attempt to mitigate an attack. Source-end solutions subsequently throttle or filter this mislabeled traffic, so IoT environments that deploy a source-end DDoS defense significantly suffer—especially IoT environments that cannot endure much delay or loss. For example, in a network of IoT medical devices or elderly care monitors that ensure the well-being of their users, the loss or severe delay of vital packets could prove fatal. Additionally, any retransmission required of IoT devices further increases their energy consumption [7].

In order to address this challenge for IoT, we design and evaluate a new source-end DDoS defense system, FR-WARD, specifically for IoT. Different from the previous source-end DDoS defense solutions, FR-WARD leverages the fast retransmit and flow control mechanisms of TCP to avoid dropping benign traffic but still maintains an appropriate level of attack mitigation. FR-WARD not only ensures that malicious devices send traffic at harmless rates, but it also ensures that benign devices send traffic at the fastest rate their receiver can handle. Therefore, FR-WARD not only eliminates the possibility of a DDoS attack originating from the policed network, but it also minimizes the number of retransmitted packets for a benign IoT device which reduces the benign device's connection duration

This project is in part the result of funding provided by the 2014 I3 award of the University of Oregon and the Science and Technology Directorate of the United States Department of Homeland Security under contract number D15PC00204. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Department of Homeland Security or the US Government.

978-1-5386-5156-8/18/$31.00 ©2018 IEEE

and energy consumption.

The remainder of this paper is organized as follows. Section II describes the related work on previous DDoS detection and defense systems. Section III introduces our source-end DDoS defense system, FR-WARD, which addresses the major concerns of applying previous DDoS defense solutions in an IoT environment. Section IV provides a mathematical model that estimates FR-WARD's effect on benign traffic. Section V first presents an evaluation of FR-WARD's effect on benign traffic through simulation of our models and second, presents an experimental evaluation of FR-WARD's success in preventing DDoS traffic from leaving an IoT network. Finally, Section VI concludes this paper.

II. Related Work

A typical DDoS defense solution follows two general steps: first, the defense labels current traffic as good or bad, and second, it filters the bad traffic. *Source-end* defenses attempt to both detect and filter attack traffic before the traffic leaves its original network [4]–[6], whereas *victim-end* defenses attempt to both detect and filter attack traffic near the target of the attack [8]–[11]. Because victim-end defenses can more effectively detect attacks and source-end defenses can more effectively respond to attacks, *collaborative* defenses combine the two methodologies to harness their strengths but limit their weaknesses [12]–[15].

Regardless of the placement of the DDoS solution, DDoS defense solutions typically filter traffic to mitigate attacks. For example, a network operator of a victim network can manually connect to routers in the network and install Access Control Lists (ACLs) to filter attack traffic. Because ACLs often make the victim network unreachable—even to benign traffic—the victim network can instead forward any suspected DDoS traffic to geographically distributed scrubbing centers [16], [17], which inspect the traffic and attempt to distinguish the good traffic from the bad through Deep Packet Inspection (DPI). Traffic scrubbing is a very tedious and expensive process that can often require the victim to manually describe the undesired traffic pattern to the scrubbing service.

However, virtually any procedure that categorically labels traffic as good or bad without the use of DPI struggles to correctly label all traffic [4]–[6], [18]. In order to eliminate the possibility of mislabeling good traffic as bad (i.e., false positives), the defense solution must inevitably label some bad traffic as good (i.e., false negatives), or vice-versa. If a defense solution can correctly label a subset of good traffic with very high accuracy and similarly, correctly label a subset of bad traffic with very high accuracy, the defense can create a third category of traffic called suspicious traffic for any traffic that does not fall under the good or bad labels. For example, the source-end solution, D-WARD [4], labels traffic that exhibits an outbound-inbound traffic ratio less than a threshold as good, labels traffic that exhibits an outbound-inbound traffic ratio greater than a threshold as bad, and labels any traffic in neither of these categories as suspicious. However, the crux of the problem remains; if the defense solution filters suspicious traffic, it inevitably filters some good traffic, and if the defense solution allows passage of the suspicious traffic, it fails to comprehensively mitigate attacks.

Because IoT networks have special requirements, a handful of IoT specific DDoS defense solutions have been proposed in the last few years [19]–[21]. For example, Misra et al. present a learning automata based solution that acts as a collaborative defense between IoT devices and corresponding servers for the IoT specific applications [19]. However, this approach is subject to misidentifying benign hosts as malicious and offers no special response to traffic too difficult to identify as categorically good or bad. Furthermore, Verma et al. present an efficient defense method against UDP-based flooding attacks in vehicle ad-hoc networks (VANETs) [20]. While this defense helps mitigate certain denial-of-service attacks in VANETs, it struggles to extend to other IoT environments and offers no solution to TCP-based reflective attacks from IoT devices [3].

III. FR-WARD: **F**ast **R**etransmit as a **W**ary but **A**mple **R**esponse to **D**DoS

A. Threat Model & Assumptions

FR-WARD is situated at the gateway of an IoT network and polices the network's outbound traffic. A network operator can employ FR-WARD to police a variety of different IoT environments; FR-WARD is suitable for wireless sensor networks (WSNs), mobile ad-hoc networks (MANETs), smart-homes, healthcare networks, or even multi-hop networks. We assume that a network that deploys FR-WARD is one that is extremely resource concerned and apprehensive of energy consumption. Based on the assumption that a device that follows both the congestion and flow control mechanisms of TCP cannot institute a DDoS flood attack, FR-WARD mitigates all DDoS flood attacks that originate from within the policed network—even attacks that attempt to evade detection by complying to FR-WARD. Further, FR-WARD defends against both TCP and UDP-based DDoS attacks.

FR-WARD provides an efficient *response* to connection classifications; the connection labels act as an input to the FR-WARD system. Because of FR-WARD's novel response to suspicious connections, a network can rely on any connection labeling procedure that places connections into the following three categories: good, suspicious, or bad. For example, FR-WARD can rely on a source-end labeling procedure, such as the observation component of D-WARD, or FR-WARD can participate in a collaborative DDoS defense system and receive connection labels from a collaborator.

B. Basic Design of FR-WARD

FR-WARD monitors and shapes traffic in an IoT environment in order to mitigate DDoS attacks that leave its policed network. It has two main goals; FR-WARD must throttle any DDoS traffic that leaves the source network it polices—even DDoS traffic that attempts to evade defense solutions by following congestion control, but meanwhile, FR-WARD must never throttle or filter benign traffic. Because it is not the main focus of FR-WARD to improve the detection of a DDoS attack,

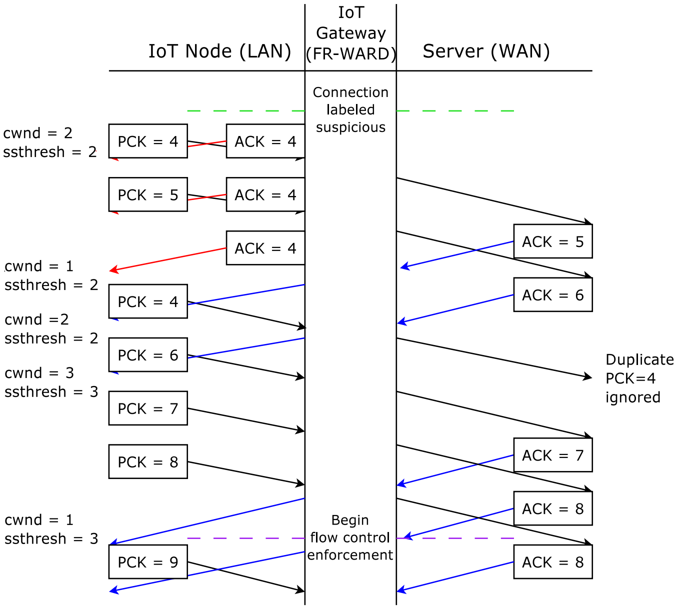


Fig. 1: The signaling mechanism of FR-WARD.

FR-WARD uses the observation component of D-WARD to monitor and label traffic. Furthermore, during an attack, FR-WARD throttles all connections labeled bad and permits passage for those labeled good. However, FR-WARD deploys a novel response to connections that are difficult to categorize as good or bad, (i.e., suspicious connections). Rather than simply dropping packets of such connections to ensure attack mitigation, FR-WARD employs the fast retransmit mechanism of TCP congestion control to reduce their sending rate which still ensures attack mitigation but reduces negative effect on benign traffic (see Fig. 1). Fig. 2 further describes the detailed architecture of FR-WARD.

The above design of FR-WARD is driven by the fundamental characteristics of an IoT environment. In particular, it follows two principles: (1) It adopts a conservative approach to avoid dropping traffic from benign devices; FR-WARD will *not* drop *any* traffic that it cannot definitively discern as malevolent. While the traffic labeled clearly as good or bad are relatively straightforward to handle, FR-WARD places a significant emphasis on the traffic or connections that are difficult to distinctly categorize as good or bad. Instead of dropping packets of such connections, FR-WARD devises a signaling mechanism to slow them down. (2) The defense cannot rely on installation of new hardware or software on IoT devices; it instead relies only on protocols and functions that the IoT devices already support. Specifically, the signaling mechanism only relies on the congestion and flow control aspects of the IoT connections.

C. Labeling Procedure

In this section, we briefly describe the observation component of the previous source-end DDoS defense system, D-WARD [4], that FR-WARD uses to label its connections in Section IV and Section V. D-WARD (**D**DoS Net**w**ork **A**ttack **R**ecognition and **D**efense) is a source-end DDoS defense solution whose goal is to detect and stop DDoS attacks from leaving a policed network. Placed at the source-end border router, D-WARD serves as a gateway between the network that deploys D-WARD and the rest of the Internet. It monitors both inward and outward traffic in order to compare current network traffic information to predefined normal flow patterns. The D-WARD system incorporates three main components: an observation component, a rate-limiting component, and a traffic policing component. FR-WARD only uses the observation component of D-WARD which we describe, as follows.

In the observation component, D-WARD monitors traffic at two levels of granularity. First, in order to detect attacks, it classifies the aggregate traffic from the *entire* source network to a particular end host outside the source network as an **agflow**. For each agflow, D-WARD labels it as *attack*, *suspicious*, or *normal*, as follows. If the statistics of the agflow fail to match predefined patterns of a normal agflow, D-WARD labels it as attack (e.g., based on predefined normal traffic, D-WARD might define $TCP_{rto} = 3$ as the maximum allowed ratio of the number of packets sent and received on a normal agflow, and would label any agflow with a higher ratio as attack). Otherwise, D-WARD labels the agflow as a normal agflow, unless it was recently labeled as an attack agflow—in which case D-WARD labels it as suspicious. Second, in order to react to a detected attack, D-WARD further classifies the aggregate traffic from *one node* in the source network to a particular host outside the source network as a **connection** and labels the connection as *good*, *bad*, or *transient* according to its behavior. For example, D-WARD labels a connection as bad if the ratio between the connection's outbound traffic volume and inbound traffic volume rises above a threshold (e.g., when a device in the policed network floods DDoS traffic to a destination host outside the policed network and the target becomes too overwhelmed to respond); similarly, D-WARD labels a connection as good if the ratio remains below a threshold, or transient if the ratio fails to meet either condition or there is not enough data to perform classification (note that for simplicity, we interchangeably use the terms transient and suspicious throughout this paper). Further, an agflow or connection's label may change as their traffic behavior dynamically changes.

D. Signaling Mechanism

Rather than simply dropping packets of suspicious connections to throttle their sending rate, FR-WARD employs the fast retransmit mechanism of TCP congestion control to reduce their sending rate. Specifically, if the sender of a connection in question has sent a TCP segment but is still waiting for the acknowledgment of its delivery, FR-WARD can send the sender three duplicate acknowledgments of the TCP segment. The sender, once receiving the three duplicate acknowledgments, will cut its sending window size in half in accordance with the fast retransmit mechanism of TCP congestion control (i.e., multiplicative decrease) [22]. We refer to this set of three duplicate acknowledgments collectively as a **signal**.

We consider any device that fails to follow TCP congestion control a malicious device, so the benefit of sending a series of signals to a suspicious device is two fold. First, the signals provide FR-WARD an additional means to identify the connection's malevolence, and second, even if the device is a smart attacker that follows congestion control to evade detection, the signals reduce the attacker's transmission rate and diminish the attack. After sending a series of signals to a suspicious connection, FR-WARD monitors the connection's future traffic in order to identify compliance with TCP congestion control. When the connection fails to comply with TCP congestion control and continues to transmit at the same (high) rate, FR-WARD relabels the connection as bad and begins to throttle the traffic. Moreover, when the connection correctly follows TCP congestion control, FR-WARD has abstained from throttling a benign connection which avoids resource penalties caused by retransmission. Fig. 1 presents an example of a benign connection initially labeled suspicious and the process at which FR-WARD avoids throttling the connection.

When FR-WARD relabels a connection as bad, FR-WARD allows trace amounts of DDoS traffic to be sent to the destination, but only for a very short duration. In fact, FR-WARD can relabel a suspicious connection as bad and begin throttling within the length of a round trip time between the IoT device and the location of FR-WARD's deployment—typically a very short period of time.

E. Attackers That Follow Congestion Control

FR-WARD must also prevent smarter DDoS attacks than ones that blindly flood a victim with TCP traffic. If an attacker knows a source network utilizes FR-WARD, the attacker may design a DDoS attack correspondingly. For example, a compromised device could be programmed to follow TCP congestion control throughout its attack. If FR-WARD only sends signals to initially mitigate the DDoS attack, the compromised device could quickly return to a high sending rate but remain undetected. Therefore, for each suspicious connection that complies with TCP congestion control, FR-WARD sets an allowed outbound traffic transmission rate and enforces this allowed rate until the attack agflow is relabeled as normal. (i.e., FR-WARD initially sends signals to quickly reduce the sender's window size and when the connection complies with congestion control, FR-WARD sets an allowed rate for the connection and throttles the sender whenever the device attempts to transmit at a rate greater than the allowed rate).

FR-WARD defines the allowed rate for each connection by the flow control value advertised by the receiver. In TCP, the receiver provides a flow control service in the form of a receive window, or *recw*, that informs the sender the amount of available space in the receiver's buffer [23]. Clearly, this flow control mechanism provides a precise definition for FR-WARD's allowed transmission rate. If a sender transmits more than *recw*, the receiver's buffer will overflow, thus constituting a DDoS attack. However, in order to utilize the flow control value, FR-WARD must constantly maintain a state for each connection which could quickly become too resource intensive. Furthermore, in order to derive the *recw* value, FR-WARD must perform deep packet inspection, so rather than constantly maintain the flow control state, FR-WARD waits to observe *recw* values until after sending its initial signals. Therefore, FR-WARD must maintain the flow control state only when necessary.

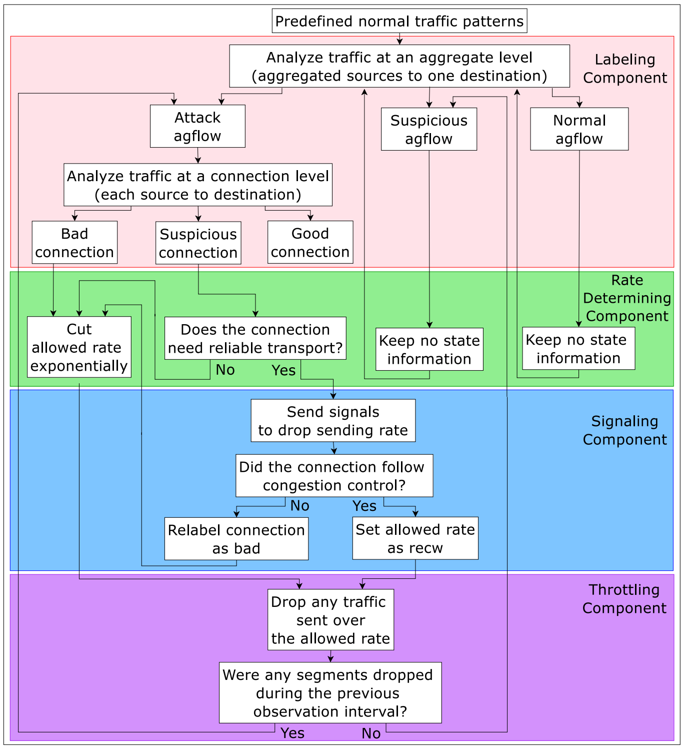


Fig. 2: The architecture of FR-WARD.

FR-WARD's main challenge is that it cannot accurately distinguish a connection from a "smart" DDoS attacker and a benign connection based solely on TCP congestion control; they may both appear as suspicious connections. While FR-WARD must shape the traffic of all suspicious connections similarly, it must maintain minimal impact on benign devices but maximum impact on malicious devices. However, whether a suspicious connection attempts to send either benign traffic or DDoS, the receiver requests traffic transmission rates no higher than its flow control value, *recw*. By design, after FR-WARD shapes each connection, all traffic transmits no faster than the requested flow control value which guarantees that benign traffic will transmit at the fastest allowable rate, but malicious traffic will always transmit at a harmless rate.

F. Extending the Signaling Mechanism

Because FR-WARD's signaling mechanism is based on aspects of TCP, we show that FR-WARD can extend its signaling mechanism to any type of connection. Traditionally, if unreliable communication is sufficient, an application used UDP as its transport layer technology. For example, if an IoT device wishes to send its location to a server, it can periodically provide the server its location with UDP datagrams. Even after a lost datagram, the server can still infer the device's location based on previous and future information. Furthermore, when FR-WARD labels such a connection as suspicious, FR-WARD can throttle the connection without apprehension. Even if FR-

WARD throttles a benign UDP connection, the connection requires no retransmission of lost packets, and thus, FR-WARD will not degrade the latency or increase the energy consumption of the IoT device.

However, while many IoT applications, such as Message Queuing Telemetry Transport (MQTT) [24], utilize TCP as their transport layer protocol, many IoT applications desire the reliability of TCP but with the overhead of UDP. For example, the Constrained Application Protocol (CoAP) [25] provides IoT devices Hypertext Transfer Protocol (HTTP) connections over UDP, and similarly, the Datagram Transport Layer Security protocol (DTLS) [26] provides the same security guarantees as the Transport Layer Security protocol (TLS) [27], but over a UDP connection. Because these connections will retransmit lost packets similar to a TCP connection, FR-WARD cannot simply throttle these connections when labeled suspicious. However, any connection that requires reliable transportation uses some type of an acknowledgment, and further, because congestion control provides improved performance, we assume these connections will utilize a variant of congestion control [28]. Therefore, FR-WARD can easily extend to provide efficient DDoS defense—even for UDP connections.

IV. Modeling Negative Effects on Benign Traffic

In this section, we mathematically model our system, FR-WARD, and the previous source-end DDoS defense system, D-WARD [4]. Because a connection will receive the same label under FR-WARD and D-WARD, and further, FR-WARD responds to known malicious or benign connections in the same manner as D-WARD, we focus our mathematical model on each system's ability to correctly handle the suspicious connections within an attack agflow. Namely, we consider both source-end solutions to handle a suspicious connection in two phases. In phase I, each system suspects the suspicious connection is an attacker, and in phase II, each system believes the suspicious connection has complied to a reasonable sending rate. See Table I for a description of the notations used in our formulas.

TABLE I: Model notations.

W	sender's congestion window at time of attack detection
f_{dec}	adjustable parameter that describes the rigor of D-WARD
A_i	allowed rate during the i^{th} observation interval
RTT	round trip time between the sender and receiver
m	number of allowed rate restrictions in phase I for D-WARD
k	the smallest integer such that $\frac{W}{2^k} \leq A_m$
n	the number of RTTs within an observation interval
S_d	number of segments sent under D-WARD in phase I
R_d	number of segments received under D-WARD in phase I
S'_d	number of segments sent under D-WARD in phase II
R_d	number of segments received under D-WARD in phase II
σ	adjustable parameter that describes the rigor of FR-WARD
t_d	the time D-WARD spends in phase I
t'_d	the time D-WARD spends in phase II
v_d	number of times D-WARD switches between phases
t_f	the time FR-WARD spends in phase I
t'_f	the time FR-WARD spends in phase II
v_f	number of times FR-WARD switches between phases
M_r	magnitude of FR-WARD's retransmission improvement
M_c	magnitude of FR-WARD's connection duration improvement

A. Modeling Packet Retransmission

We first calculate (assuming no natural packet loss) how many retransmissions D-WARD causes, as follows. During phase I, for each suspicious connection, D-WARD calculates $A_1 = W \cdot f_{dec}$ to derive the amount of traffic, A_1, allowed outside of the policed network each round trip time (RTT) where f_{dec} describes the rigor of the D-WARD system and W is the size of the sender's current congestion window. During the first RTT after D-WARD detects an attack agflow, the sender attempts to send its full congestion window, W, but D-WARD drops all segments after the A_1^{th} segment. The sender then receives A_1 ACKs and shifts its congestion window correspondingly. During the next RTT, since its congestion window has shifted, the sender delivers A_1 more segments, and because $A_1 \leq A_1$, D-WARD does not drop any segments during this RTT. However, when the receiver returns the corresponding ACKs this round, each ACK number corresponds to the first packet dropped by D-WARD in the previous RTT, and thus, due to TCP congestion control, the sender cuts its congestion window size in half. Further, D-WARD defines an observation interval in which D-WARD periodically restricts the allowed transmission rate by f_{dec} whenever the sender, on average, fails to follow the allowed transmission rate in the previous observation interval. If m is the number of times D-WARD restricts the allowed transmission rate during phase I, let $A_i = W \cdot (f_{dec})^i$, where $1 \leq i \leq m$, be the allowed amount of traffic to leave the policed network during the i^{th} observation interval of phase I. The first two RTTs of phase I repeat until the sender's congestion window falls below the allowed transmission rate; let k be this number of repetitions (i.e., k is such that $\frac{W}{2^k} \leq A_m$). See Fig. 3 for more details. Let S_d be the number of segments sent by the sender under D-WARD in phase I, and let R_d be the number of segments successfully received by the receiver under D-WARD in phase I. If n is the number of RTTs within an observation interval, we observe that:

$$S_d = \sum_{i=1}^{k} \frac{W}{2^{i-1}} + \sum_{i=1}^{m} n \cdot A_i \quad \text{and} \quad R_d = 2 \cdot \sum_{i=1}^{m} n \cdot A_i \tag{1}$$

It then follows that the number of retransmissions caused by D-WARD in phase I equals $S_d - R_d$. D-WARD then remains in phase II until the sender attempts to transmit more traffic than D-WARD's current allowed rate or D-WARD relabels the suspicious agflow as good. For each observation interval that the sender, on average, complies with D-WARD's allowed rate, D-WARD linearly increases the allowed rate by f_{inc}. If m' is the number of times D-WARD increases the allowed transmission rate during phase II, let $A_{m+i} = A_m + (i \cdot f_{inc})$, where $1 \leq i \leq m'$, be the allowed amount of traffic to leave the policed network during the i^{th} observation interval of phase II. However, D-WARD still enforces the allowed transmission rate during a suspicious agflow, so when the sender attempts to transmit more than the allowed rate, D-WARD will drop segments, and thus, cause more retransmissions. Let S'_d describe

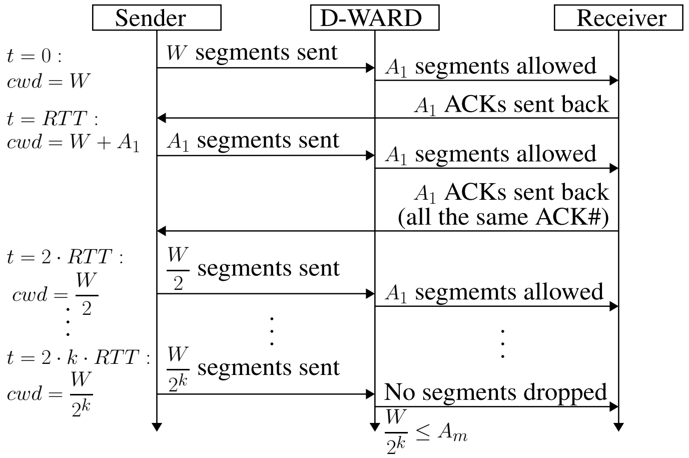


Fig. 3: Phase I of D-WARD.

the natural growth of the sender's congestion window under TCP congestion control and R'_d describe the growth of D-WARD's allowed rate under a suspicious agflow; the number of retransmissions caused by D-WARD in phase II equals $S'_d - R'_d$. If k' is the pre-set number of RTTs D-WARD keeps an agflow labeled as suspicious before relabeling the agflow as good, S'_d and R'_d can be calculated as follows:

$$S'_d = \sum_{i=1}^{k'} \frac{W}{2^k} + \frac{i}{W/2^k} \quad \text{and} \quad R'_d = \sum_{i=1}^{m'} n \cdot A_{m+i} \tag{2}$$

On the other hand, FR-WARD never drops a suspicious connection's segments, and instead, the amount of retransmissions required of a suspicious connection directly correlates with the number of signals sent by FR-WARD. We calculate the number of signals sent by FR-WARD (and thus, the number of retransmitted segments), as follows. FR-WARD initially allows the sender's entire congestion window, W, through to the sender, and instead of dropping any segments, FR-WARD immediately delivers a series signals at the time of the attack agflow to reduce W. When the sender receives a signal from FR-WARD, in addition to reducing its congestion window in half, it must immediately retransmit the "lost" segment. Let σ be the number signals FR-WARD sends during phase I; similar to f_{dec} for D-WARD, σ describes the rigor of FR-WARD. The network operator of FR-WARD can choose a value for σ such that:

$$1 \leq \sigma \leq \lfloor \log_4(W) \rfloor \tag{3}$$

where $\sigma = \lfloor \log_4(W) \rfloor$ is the most conservative response to attack detection and $\sigma = 1$ is the least conservative—but still effective—response to attack detection. After receiving FR-WARD's signals in the first RTT, the sender's congestion window will reduce to $\frac{W}{2^\sigma}$. However, the sender then receives the initial W ACKs back from the receiver and increases its window to $\frac{W}{2^\sigma} + \frac{W}{W/2^\sigma} = \frac{W}{2^\sigma} + 2^\sigma$. See Fig. 4 for more details. Furthermore, during phase II, FR-WARD will never cause retransmissions because it never increases its allowed amount of traffic above $recw$, and we assume that a benign sender will never transmit above $recw$.

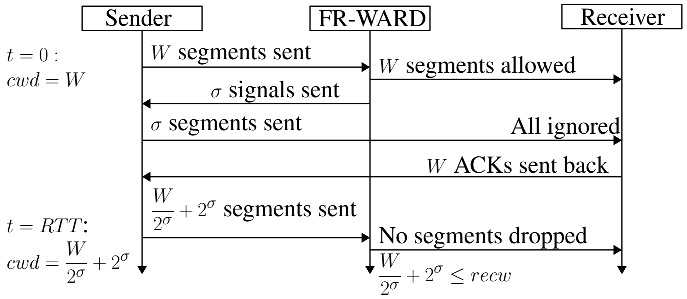


Fig. 4: Phase I of FR-WARD.

If we let v_d and v_f be the number of times D-WARD and FR-WARD respectively interchange between phase I and II, then we can combine equations (1), (2), and (3) in order to derive a model for the magnitude of improvement, M_r, FR-WARD has over D-WARD in terms of retransmission, as follows:

$$M_r = \frac{v_d[(S_d - R_d) + (S'_d - R'_d)]}{v_f \cdot \sigma} \tag{4}$$

B. Modeling Connection Duration

We now calculate (assuming no natural packet loss) the duration of a connection under D-WARD and FR-WARD, as follows. From our previous analysis, we saw that during phase I, D-WARD repeats its first two RTTs k times where k is the smallest integer such that $\frac{W}{2^k} \leq A_m$. Further, k' was the number of RTTs D-WARD stays in phase II, so we can observe the length of time, t_d, that D-WARD is in phase I and the length of time, t'_d, that D-WARD is in phase II:

$$t_d = 2 \cdot k \cdot RTT \quad \text{and} \quad t'_d = k' \cdot RTT \tag{5}$$

However, because a benign connection responds to FR-WARD's signals, FR-WARD remains in phase I for only one round trip time. Further, FR-WARD remains in phase II for the remainder of the transmission because its allowed amount of traffic is $recw$, and we assume that a benign sender will never transmit above $recw$. Moreover, if k_f is the number of RTTs to naturally send the remainder of data, we can observe the length of time, t_f, that FR-WARD is in phase I and the length of time, t'_f, that FR-WARD is in phase II:

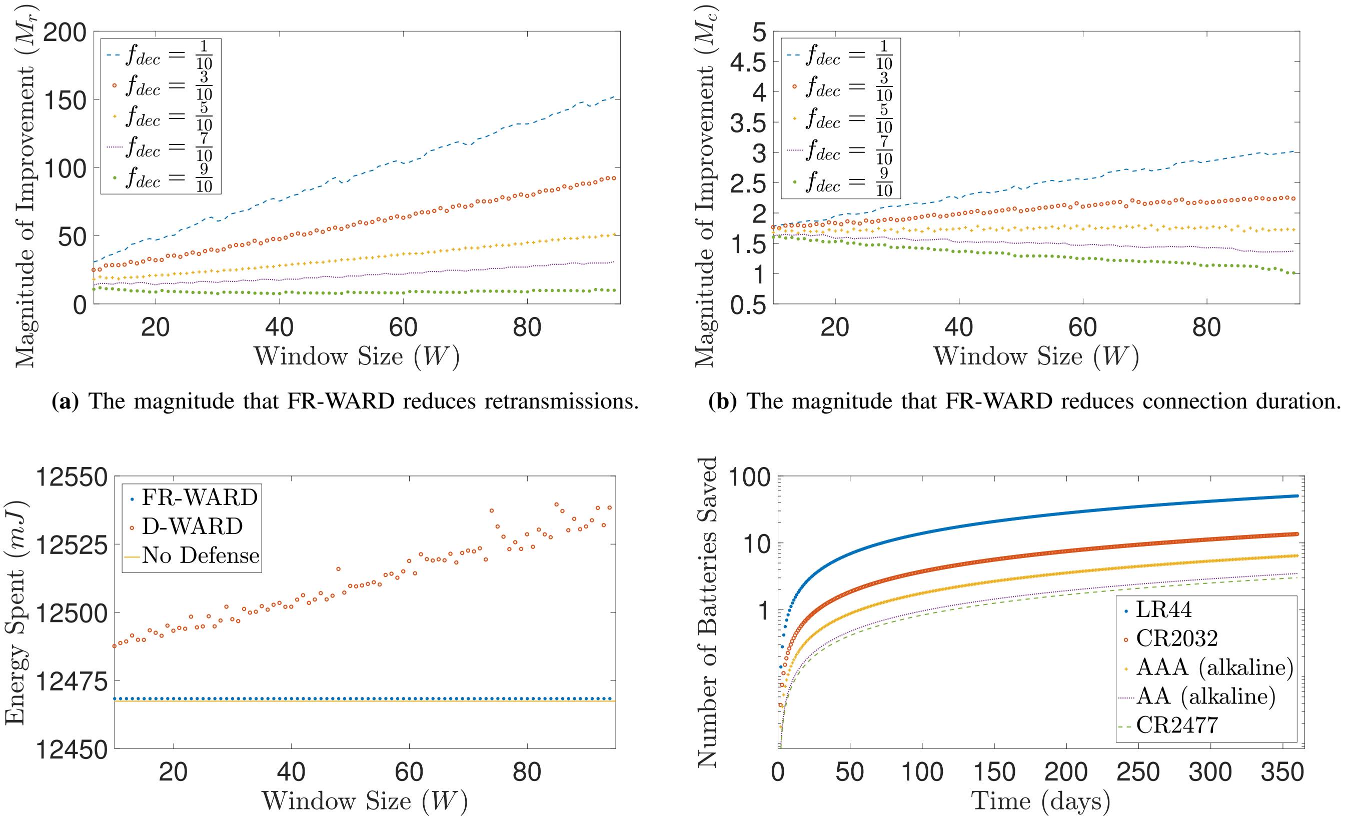
$$t_f = 1 \cdot RTT \quad \text{and} \quad t'_f = k_f \cdot RTT \tag{6}$$

Using v_f and v_d from our above analysis, we can combine equations (5) and (6) in order to derive a model for the magnitude of improvement, M_c, FR-WARD has over D-WARD in terms of connection duration, as follows:

$$M_c = \frac{v_d(t_d + t'_d)}{v_f(t_f + t'_f)} \tag{7}$$

V. Evaluation

In this section, we use real-time experiments and our models from Section IV to investigate FR-WARD's performance across the following five metrics: (1) the retransmission required of benign connections, (2) the connection duration of benign connections, (3) the energy consumption of benign connections,



(a) The magnitude that FR-WARD reduces retransmissions.

(b) The magnitude that FR-WARD reduces connection duration.

(c) The energy consumption for a benign IoT device.

(d) The number of batteries that FR-WARD saves during one year.

Fig. 5: FR-WARD reduces the energy consumption of benign IoT devices and improves their battery life.

(4) the throughput of an attacker who attempts to perform a naive TCP SYN-flood DDoS attack, and (5) the throughput of an attacker who attempts to perform a smarter DDoS attack that evades detection by following TCP congestion control. We discover that regardless of the connection state at the time of attack detection, benign hosts under FR-WARD maintain less packet retransmission, shorter connection durations, and less energy consumption than benign hosts under D-WARD. Further, we show that FR-WARD successfully mitigates both the naive DDoS attack and the smarter DDoS attack that attempts to evade detection. Note, we expect FR-WARD to have the same accuracy and detection as D-WARD, so we do not evaluate the systems on these metrics in this paper.

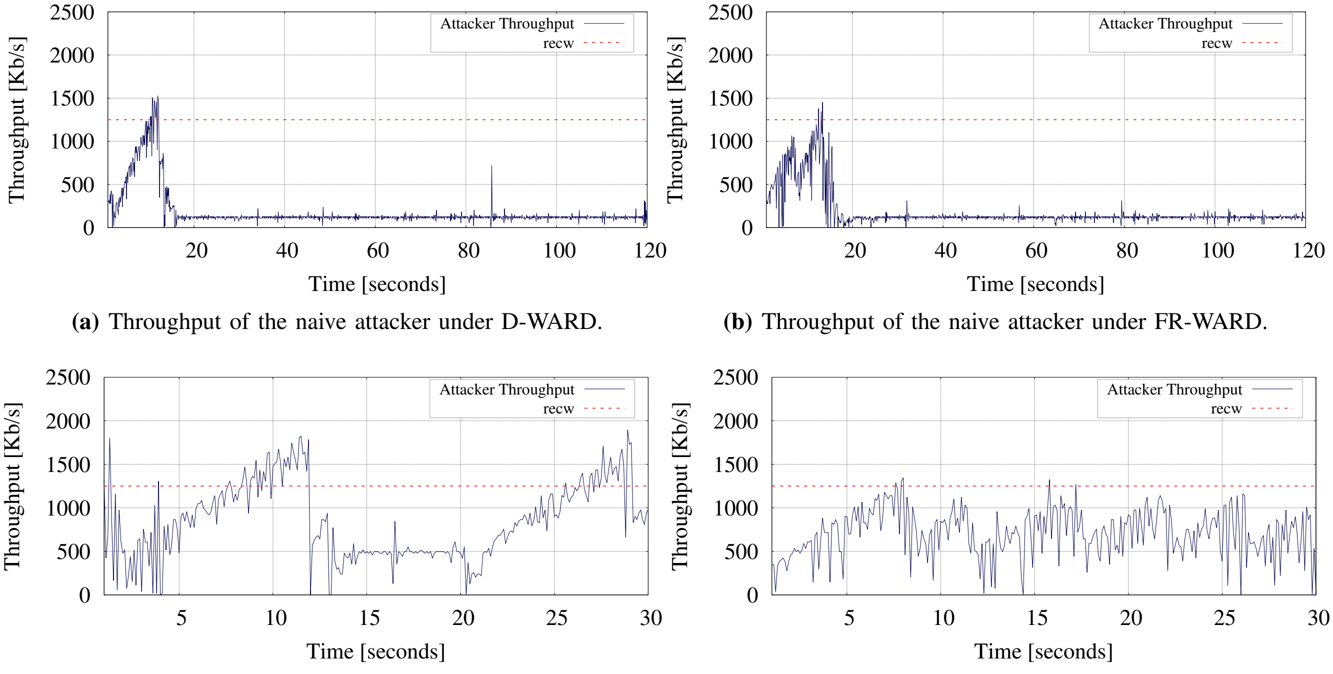
Throughout our real-time experiments, we constructed a wireless client-server TCP application where the client attempts to send 2.5 MB of data to the server. Both the client and server applications were deployed on separate 2012 Macbook Pro laptops with 2.3 GHz Intel Core i5 processors and 4GB of RAM. Both the client and server utilized the TCP New Reno algorithm to accurately perform congestion control. Also, between the client and server we employed a router that is a 2015 Dell XPS with 2.2 GHz Intel Core i5 processor and 8GB of RAM. We then ran FR-WARD on the router and compare its performance against D-WARD, the source-end DDoS-defense solution that is the most related to FR-WARD.

In the simulation of our models, a benign IoT device attempts to send 2.5 MB of data outside the policed network. We observed the number of retransmissions and connection duration each DDoS defense system required of the IoT device over the two main parameters of equations (4) and (7): the sender's congestion window size, W, at the time D-WARD or FR-WARD detects an attack agflow, and the pre-set fraction, f_{dec}, of traffic that D-WARD allows to leave the source network during a suspected DDoS attack. Throughout our experiments, we set $\sigma = 1$, and furthermore, Mirkovic et al suggest $f_{dec} = \frac{1}{2}$ for a typical deployment of D-WARD [4].

A. Effects on Benign Traffic

1) Retransmissions: Based on equation (4), Fig. 5a estimates how many more retransmissions D-WARD requires than FR-WARD for a benign IoT device. Because f_{dec} determines how strictly D-WARD throttles during an attack agflow (i.e., D-WARD drops $W - W \cdot f_{dec}$ segments every $2RTT$), as W increases or f_{dec} decreases, D-WARD drops more segments and thus, causes more retransmissions. On average, across all possible f_{dec}, FR-WARD will reduce retransmissions for benign IoT devices by a factor of 33.

2) Connection Duration: Based on equation (7), Fig. 5b estimates how much more D-WARD increases a benign connection's duration than FR-WARD. As f_{dec} decreases (and D-WARD becomes stricter), the sender's average transmission rate decreases which results in longer connection durations. Furthermore, FR-WARD uses $recw$ rather than f_{dec} to derive



(a) Throughput of the naive attacker under D-WARD.

(b) Throughput of the naive attacker under FR-WARD.

(c) Throughput of the smart attacker under D-WARD.

(d) Throughput of the smart attacker under FR-WARD.

Fig. 6: FR-WARD defends against both naive and smart DDoS attacks while D-WARD fails to defend against smart attackers.

an allowed rate, so as W increases, if f_{dec} is too strict (i.e., $W \cdot f_{dec}$ is much less than $recw$), FR-WARD allows a higher throughput for the benign connection which decreases the overall connection duration. On average, across all possible f_{dec}, FR-WARD will reduce connection duration for benign IoT devices by a factor of 1.77.

3) Energy Consumption: Based on the analysis presented by Feeney et al. [7] that estimates the microwatt seconds consumed by a wireless device with respect to the amount of data transmitted, and our aforementioned mathematical models, Fig. 5c estimates the energy consumption each DDoS defense system requires from a benign IoT device. Clearly, with FR-WARD a benign IoT device consumes much less energy than a device with D-WARD. In fact, FR-WARD consumes virtually the same amount of energy as when there is no defense.

We can further extend the result from Fig. 5c to conduct an economic study of the cost of electricity or the battery life that can be saved if a benign IoT device uses electricity or battery, respectively. Fig. 5d shows the result of such a study for the latter. We examine the battery life of a benign IoT device under FR-WARD and D-WARD across five popular IoT batteries (CR2032, CR2477, AAA, AA, LR44) and present how many more batteries an IoT device will consume under D-WARD throughout a year of deployment. As W increases (see Fig. 5a), D-WARD causes more retransmissions, and therefore, causes a benign IoT device to spend additional energy. On average, across the batteries tested, a benign IoT device will consume 15.55 less batteries every year under FR-WARD.

B. Effects on Malicious Traffic

1) Naive Attack: Fig. 6a and Fig. 6b examine the throughput of an attacker that attempts to perform a TCP-SYN flood attack under each of the DDoS defense systems. The red, horizontal, dotted line on each graph represents the receiver's advertised $recw$ converted to an equivalent throughput. We used the `hping3` command-line tool to perform the TCP-SYN flood as fast as possible. Both DDoS defense systems used the same observation component and detected the attack at around 12 seconds. After detecting the attack, both systems successfully throttled the attackers throughput to the minimum sending rate of 125 Kbps within a few seconds. The graphs look almost identical, but under FR-WARD, a negligible extra *instant* of DDoS traffic was allowed to reach the receiver.

2) Smart Attack: In this section, we describe D-WARD's vulnerability to a "smart" TCP flood attack that follows congestion control, but argue that FR-WARD successfully mitigates such an attack. Further, we present experimental evidence to reinforce our claims.

During phase II of D-WARD, the allowed transmission rate linearly increases as long as the sender never attempts to send more than the allowed rate. Because D-WARD maintains no state of the receiver's flow control value, $recw$, D-WARD's allowed rate can increase above $recw$. While the attacker complies with TCP congestion control, the attacker does not comply with TCP flow control; it only follows D-WARD's estimated allowed rate. During this period, the attacker can send DDoS traffic until D-WARD detects another attack agflow and returns to phase I. However, the design of FR-WARD prevents the smart attack from succeeding. Because FR-WARD only labels a connection benign after compliance with *both* TCP congestion and flow control, the smart attacker's transmission rate never surpasses the receiver's advertised $recw$.

Fig. 6c and Fig. 6d examine the throughput of the smart attacker under each system. The red, horizontal, dotted line on each graph represents the receiver's advertised $recw$ converted to an equivalent throughput. First, Fig. 6c shows the attacker's throughput under D-WARD. After D-WARD detects the attack agflow, it restricts the allowed rate twice until the attacker complies to an average sending rate of 500 Kbps. D-WARD then switches to phase II; the smart attacker linearly increases its sending rate—even past $recw$—and continues to send DDoS traffic until D-WARD detects another attack agflow. While the test ends after the attacker has transmitted 2.5 MB of data, one can assume the graph will continue to repeat as long as the attacker wishes. Second, Fig. 6d shows the smart attacker's throughput under FR-WARD. Because FR-WARD uses $recw$ as its allowed transmission rate, each time the smart attacker attempts to transmit more than $recw$, FR-WARD drops the excess traffic. Further, because the smart attacker follows congestion control, the attacker subsequently must reduce its throughput. Therefore, unlike D-WARD, after FR-WARD detects the attack agflow, the attacker is unable to achieve a successful DDoS attack for the rest of the transmission.

VI. Conclusion

FR-WARD is a source-end DDoS defense system designed to defend against DDoS attacks launched from an IoT network. While past DDoS defense solutions seldom consider the resource constraints of IoT devices and can cause IoT environments to significantly suffer, FR-WARD leverages the fast retransmit and flow control mechanisms of TCP as a novel response to suspicious traffic which not only limits DDoS traffic that leaves the policed network but also minimizes the energy consumption and network latency of benign IoT devices within the policed network. Our mathematical analysis, simulation, and experimental evaluation show that while past source-end DDoS defense solutions may hurt benign IoT traffic and fail to handle "smart" DDoS attacks, FR-WARD allows benign IoT devices to transmit traffic at the fastest rate possible but forces malicious IoT devices to transmit traffic at a harmless rate.

References

[1] S. Hilton, "Dyn Analysis Summary Of Friday October 21 Attack." https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack, 2016.

[2] B. Krebs, "DDoS on Dyn Impacts Twitter, Spotify, Reddit." https://krebsonsecurity.com/2016/10/ddos-on-dyn-impacts-twitter-spotify-reddit, 2016.

[3] M. Lyu, D. Sherratt, A. Sivanathan, H. H. Gharakheili, A. Radford, and V. Sivaraman, "Quantifying the Reflective DDoS Attack Capability of Household IoT Devices," in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 46–51, ACM, 2017.

[4] J. Mirkovic and P. Reiher, "D-WARD: A Source-end Defense Against Flooding Denial-of-Service Attacks," in *IEEE Transactions on Dependable and Secure Computing*, vol. 2, pp. 216–232, 2005.

[5] S. Abdelsayed, D. Glimsholt, C. Leckie, S. Ryan, and S. Shami, "An Efficient Filter for Denial-of-Service Bandwidth Attacks," in *Global Telecommunications Conference*, vol. 3, pp. 1353–1357, 2003.

[6] T. M. Gil and M. Poletto, "MULTOPS: A Data-structure for Bandwidth Attack Detection," in *Proceedings of the 10th Conference on USENIX Security Symposium*, vol. 10, 2001.

[7] L. M. Feeney and M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad hoc Networking Environment," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1548–1557, IEEE, 2001.

[8] S. Lim, J. Ha, H. Kim, Y. Kim, and S. Yang, "A SDN-oriented DDoS Nlocking Scheme for Botnet-based Attacks," in *Ubiquitous and Future Networks (ICUFN), 2014 Sixth International Conf on*, pp. 63–68, IEEE, 2014.

[9] B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, "DDoS Attack Protection in the Era of Cloud Computing and Software-defined Networking," *Computer Networks*, vol. 81, pp. 308–319, 2015.

[10] A. Kalliola, K. Lee, H. Lee, and T. Aura, "Flooding DDoS Mitigation and Traffic Management with Software Defined Networking," in *Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on*, pp. 248–254, IEEE, 2015.

[11] S. K. Fayaz, Y. Tobioka, V. Sekar, and M. Bailey, "Bohatei: Flexible and Elastic DDoS Defense," in *USENIX Security Symposium*, pp. 817–832, 2015.

[12] J. Li, S. Berg, M. Zhang, P. Reiher, and T. Wei, "Drawbridge: Software-defined DDoS-resistant Traffic Engineering," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 591–592, 2015.

[13] J. Ioannidis and S. M. Bellovin, "Implementing Pushback: Router-Based Defense Against DDoS Attacks," in *NDSS*, vol. 2, 2002.

[14] E. Kline, M. Beaumont-Gay, J. Mirkovic, and P. Reiher, "RAD: Reflector Attack Defense Using Message Authentication Codes," in *Computer Security Applications Conference, 2009. ACSAC'09. Annual*, pp. 269–278, IEEE, 2009.

[15] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett, "SDX: A Software Defined Internet Exchange," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 551–562, 2015.

[16] A. Networks, "DDoS Protection by Arbor Networks APS." http://www.arbornetworks.com/ddos-protection-products/arbor-aps, 2016.

[17] A. Technologies, "Akamai DDoS Protection Service." http://www.arbornetworks.com/ddos-protection-products/arbor-aps, 2016.

[18] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS Flooding Attack Detection using NOX/OpenFlow," in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pp. 408–415, IEEE, 2010.

[19] S. Misra, P. V. Krishna, H. Agarwal, A. Saxena, and M. S. Obaidat, "A Learning Automata Based Solution for Preventing Distributed Denial of Service in Internet of Things," in *Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, pp. 114–122, IEEE, 2011.

[20] K. Verma, H. Hasbullah, and A. Kumar, "An Efficient Defense Method Against UDP Spoofed Flooding Traffic of Denial of Service (DoS) Attacks in VANET," in *Advance Computing Conference (IACC), 2013 IEEE 3rd International*, pp. 550–555, IEEE, 2013.

[21] C. Zhang and R. Green, "Communication Security in Internet of Thing: Preventive Measure and Avoid DDoS Attack Over IoT Network," in *Proceedings of the 18th Symposium on Communications & Networking*, pp. 8–15, Society for Computer Simulation International, 2015.

[22] V. Jacobson, "Congestion Avoidance and Control," in *ACM SIGCOMM Computer Communication Review*, vol. 18, pp. 314–329, 1988.

[23] M. Allman, V. Paxson, and E. Blanton, "TCP Congestion Control," RFC 5681, IETF, 2009.

[24] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S - A Publish/Subscribe Protocol for Wireless Sensor Networks," in *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on*, pp. 791–798, IEEE, 2008.

[25] C. Bormann, A. P. Castellani, and Z. Shelby, "Coap: An Application Protocol for Billions of Tiny Internet Nodes," *IEEE Internet Computing*, vol. 16, no. 2, pp. 62–67, 2012.

[26] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security Version 1.2," RFC 6347, IETF, 2012.

[27] T. Dierks, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, IETF, 2008.

[28] A. Betzler, C. Gomez, I. Demirkol, and J. Paradells, "CoAP Congestion Control for the Internet of Things," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 154–160, 2016.