D-WARD: A Source-End Defense against Flooding Denial-of-Service Attacks

Jelena Mirkovic, *Member*, *IEEE*, and Peter Reiher, *Member*, *IEEE*

Abstract—Defenses against flooding distributed denial-of-service (DDoS) commonly respond to the attack by dropping the excess traffic, thus reducing the overload at the victim. The major challenge is the differentiation of the legitimate from the attack traffic, so that the dropping policies can be selectively applied. We propose D-WARD, a source-end DDoS defense system that achieves autonomous attack detection and surgically accurate response, thanks to its novel traffic profiling techniques, the adaptive response and the source-end deployment. Moderate traffic volumes seen near the sources, even during the attacks, enable extensive statistics gathering and profiling, facilitating high response selectiveness. D-WARD inflicts an extremely low collateral damage to the legitimate traffic, while quickly detecting and severely rate-limiting outgoing attacks. D-WARD has been extensively evaluated in a controlled testbed environment and in real network operation. Results of selected tests are presented in the paper.

Index Terms—Network-level security and protection, network monitoring, fault tolerance.

1 Introduction

Flooding distributed denial-of-service (DDoS) attacks send an overwhelming quantity of packets from multiple attack sites to a victim site. These packets arrive in such a high quantity that some key resource at the victim (bandwidth, buffers, CPU time to compute responses, etc.) is quickly exhausted. The victim either crashes or spends so much time handling the attack traffic that it cannot attend to its real work. The denial-of-service effect lasts for as long as the attack flood reaches the victim. When the attack is aborted or blocked by a defense mechanism, the victim usually reclaims its resources and recovers within seconds. This feature makes responsive defense approaches an attractive solution to flooding attacks.

Responsive defenses detect the occurrence of the attack and respond to it by dropping excess traffic that appears malicious. The major challenge is the differentiation of the legitimate from the attack traffic, so that all and only malicious packets can be dropped. Minimizing erroneous legitimate traffic drops is of paramount importance since a defense that drops legitimate traffic along with the attack still denies service to the victim's legitimate users.

Flooding attacks do not have to generate specific "malicious" packet content or header field values since their ability to do damage at the victim lies simply in the vast amount of traffic. This makes per-packet traffic differentiation inaccurate and easily bypassed by the attacker. Attacks that use source IP spoofing further hinder defenses that attempt to remember legitimate users and favor their traffic since the attacker easily assumes legitimate user's identity. Another approach to traffic differentiation groups all packets into higher-semantic structures (e.g., "all traffic exchanged between two IP addresses"), then gathers statistics on the structure behavior and dynamics to detect anomalous communications. Packets from suspect structures will be policed, while packets from legitimate structures will be forwarded. Semantic traffic differentiation has two main advantages over per-packet and per-user differentiation approaches: 1) It easily spots randomly generated attack traffic (with or without spoofing) since such traffic creates short-lived structures with no higher semantics. To bypass this check, the attacker must engage in nonspoofed communication with the victim, obeying protocol and application semantics. 2) It easily spots structures that are engaged in "one-way" communications, aggressively sending traffic to an unresponsive party. Since most of the Internet communications are "two-way," with one party slowing down if another party becomes unresponsive, aggressive and persistent one-way communication is an anomalous event. The per-packet processing cost and memory requirements of semantic traffic differentiation are both significant, which makes it too costly for deployment in networks that observe high-volume, high-spoofing traffic during the attacks, such as victim and core networks. Source-end deployment remains as the only alternative. Moderate traffic volumes near the sources make per-packet processing cost acceptable, and egress filtering limits spoofing and memory cost of semantic differentiation. Equipped with a reasonable attack detection capability, a source-end defense could autonomously ensure that a deploying network cannot participate in flooding DDoS attacks, while majority of the legitimate traffic could be separated from the attack and forwarded.

We note here that source-end defense is not a complete solution to flooding attacks since networks that do not deploy the proposed defense can still be misused for successful attacks. Still, source-end defense is necessary for

- *J. Mirkovic is with the Computer and Information Science Department, University of Delaware, 103 Smith Hall, Newark, DE 19716. E-mail: sunshine@cis.udel.edu.*
- *P. Reiher is with the Computer Science Department, University of California at Los Angeles, Los Angeles, CA 90095-1596. E-mail: reiher@cs.ucla.edu.*

Manuscript received 27 Apr. 2004; revised 20 June 2005; accepted 30 June 2005; published online 2 Sept. 2005.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-0061-0404.

precise differentiation of legitimate and attack traffic. In cooperation with victim-end or core-based defenses, source-end defense could ensure safe delivery to the victim of all and only legitimate traffic from the defense-deploying networks. This makes source-end defense one of the key building blocks of the complete DDoS solution and essential for promoting Internet security.

This paper describes D-WARD [1], [2], [3], a source-end DDoS defense system that achieves autonomous attack detection and surgically accurate response by deploying semantic traffic differentiation. D-WARD is an inline system installed at the exit router of an end network, a potential source of DDoS attacks. D-WARD continuously monitors incoming and outgoing traffic, collecting per-destination and per-connection statistics. Periodic profiling of those statistical records reveals anomalies in traffic dynamics that D-WARD detects as potential attack alerts. In response, D-WARD imposes a dynamic rate limit on all outgoing traffic to the alleged victim. This rate limit is selectively enforced—packets from the classified-legitimate connections are sent to the victim, while the rest of the traffic is forwarded only if there is remaining bandwidth within the limit. D-WARD has been extensively evaluated in a controlled test-bed environment, in joint tests with other research groups and in a real network operation. In all tests it has demonstrated high detection accuracy, and swift and selective response.

2 Source-End Defense

The goal of a DDoS attack is to deny some service to the legitimate clients. The primary goal of DDoS defense must therefore be to maintain good service levels between the victim and its legitimate clients in spite of the ongoing attack. We focus here on *responsive* defense approaches that detect the occurrence of the attack and respond to it by dropping traffic they perceive as malicious. To be successful, response approaches must accurately detect all attacks that inflict damage at the victim, effectively respond to these attacks (restoring services quickly to the normal level) and ensure selectiveness of the response, i.e., low collateral damage.

DDoS defenses can be deployed as autonomous defenses at the victim, core or source networks, or as cooperative defenses spanning various deployment locations. Victim-end defenses excel in attack detection but can be overwhelmed by high-volume attacks. High spoofing and high packet rates seen at the victim during some attacks allow very limited statistics gathering and per-packet processing, which hinders response selectiveness. Core-end defenses can handle (and filter) high-volume attacks but cannot spare resources for attack detection or for traffic profiling needed for selective response. The main advantage of source-end defense over core-based and victim-end defenses is in its ability to deploy sophisticated traffic profiling strategies. Routers close to the sources relay smaller amount of attack traffic than core and victim-end routers, which facilitates more complex per-packet processing. Further, source-end defenses can limit the amount of IP spoofing in the attack by deploying egress filtering [4]. Reduced traffic diversity limits memory requirements and facilitates more extensive statistics gathering and traffic profiling. Jointly, detailed statistics gathering and per-packet processing facilitate high response selectiveness. Source-end defense further preserves shared Internet resources by dropping the attack traffic as early as possible, before it reaches the core. Finally, being close to the source facilitates easier attack investigation when alerts are received.

These advantages make source-end defense an attractive solution. Even in autonomous deployment, source-end defense accurately detects many attacks (as our experiments show in Section 4) and selectively drops the attack traffic while inflicting low to zero collateral damage on the legitimate users' traffic. Further, the source end is the only deployment point that can achieve good response selectiveness in case of high-volume, high-spoofing flooding attacks, which makes it a key building block for distributed defense systems. Jointly, core and victim defenses can provide an accurate attack detection and effective response, but they inflict high collateral damage to legitimate traffic. Pushing core responses closer to the sources (such as proposed in [5]) localizes unwarranted drops to those legitimate clients that share the path to the victim with the attacker, which still may be a large portion of victim's users in case of highly distributed attacks. A source-end defense provides the means for a network to vouch for "goodness" of the traffic it originates, even if the attacker compromises some of its machines. Traffic from such networks could be regarded as guaranteed-legitimate and handled with higher priority in a distributed defense system.

3 D-WARD Design

D-WARD (**D**DoS Net**w**ork **A**ttack **R**ecognition and **D**efense) is a source-end DDoS defense system. Its goal is to autonomously detect outgoing DDoS attacks and stop them by controlling outgoing traffic to the victim, while ensuring that all legitimate traffic to the victim is detected and forwarded. In autonomous deployment, D-WARD detects attacks and responds to them without communication with any other entity. In distributed cooperative deployment, D-WARD enhances its detection by receiving attack alerts from other participants.

D-WARD is installed at the *source router* that serves as a gateway between the deploying (*source network*) and the rest of the Internet. In basic deployment, we assume that D-WARD can observe every packet exchanged between source network and the outside world. More complex deployment patterns are examined at the end of this section. D-WARD is configured with a set of local addresses whose outgoing traffic it polices—its *police address set*. This set identifies, for example, all machines in the stub network or all customers of an ISP.

D-WARD consists of *observation*, *rate-limiting*, and *traffic-policing* components, as depicted in Fig. 1. The observation component monitors all packets passing through the source router and gathers statistics on two-way communications between the police address set and the rest of the Internet. The statistics are gathered at the *aggregate flow—agflow* and *connection* granularity. An agflow describes the aggregate traffic between the police address set and one foreign

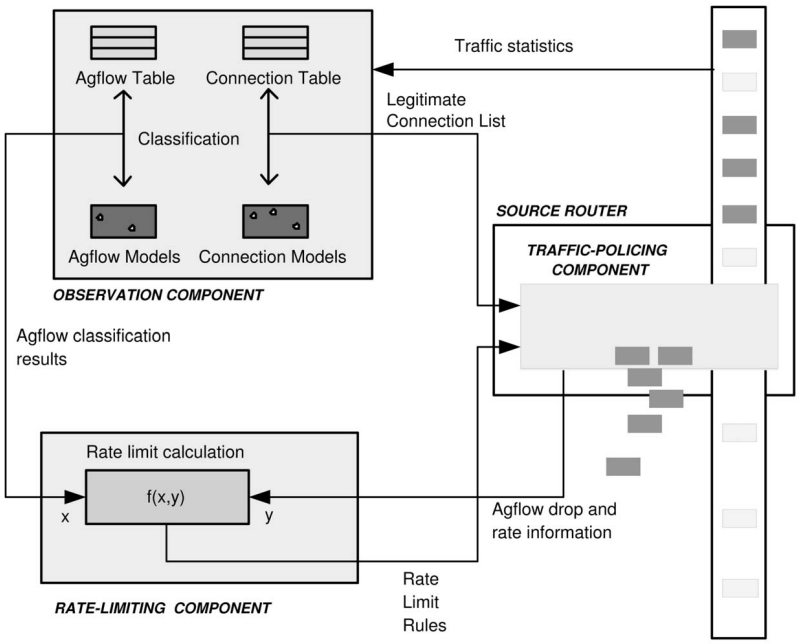

Fig. 1. D-WARD architecture.

IP address. A connection describes the aggregate traffic between a pair of IP addresses and port numbers, where one address belongs to the police address set and the other is a foreign address. Periodically, statistics are compared to legitimate traffic models, and agflows and connections are classified as attack or legitimate. The observation component passes the information on agflow classification and behavior to the rate-limiting component which decides to impose, modify or remove the rate limit on the agflow's sending rate. Rate limit rules are delivered to the traffic-policing component, which uses a list of classified-legitimate connections to selectively enforce these rules on suspicious traffic. D-WARD's rate limit decisions affect associated agflows and their future behavior. Observing how agflow statistics react to defense actions, D-WARD can quickly detect and recover from false positives. The traffic-policing component must be part of the source router, while the observation and rate-limiting components can be part of a self-contained unit that interacts with the source router to obtain traffic statistics and install rate-limiting rules.

D-WARD currently handles attacks that use the TCP, UDP, or ICMP protocols. These protocols are used by the vast majority of both the legitimate traffic and actual DDoS attacks in today's Internet. Should traffic using other protocols become commonplace in the Internet, D-WARD would need to be extended to treat such traffic properly.

3.1 Observation Component

Observation component detects outgoing DDoS attacks by monitoring agflow dynamics, looking for the following anomalies that may be signs of a DDoS attack:

1. *Nonresponsive foreign host: Aggressive sending rate coupled with low response rate*. This anomaly pertains only to two-way communications that follow a request/response paradigm such as TCP, some types of ICMP, DNS, NTP, etc. One party sends a batch of packets to the other party, and waits for a reply (an acknowledgment or a response) before sending any more. For such communications, it is anomalous to observe a persistent aggressive sending rate coupled with a low response rate. A low response rate is perceived by D-WARD as an indication that the foreign host may be overwhelmed by the attack and cannot reply, while an aggressive sending rate indicates that local hosts are likely participants in the attack. By detecting nonresponsive foreign hosts, D-WARD actually aims to detect the occurrence of the denial-of-service effect. This is a very reliable attack signal, as an attacker must create the denial-of-service effect to inflict damage on the victim.
2. *Presence of IP spoofing*. D-WARD deploys egress filtering [4] and discards, at all times, outgoing packets that do not carry local addresses. Additionally, D-WARD monitors the number of simultaneous connections between the source network and each foreign host. Those foreign hosts that are engaged in a suspiciously high number of connections are deemed to be victims of a subnet spoofing attack. This form of IP spoofing detection is not quite accurate since it may happen that a destination becomes so popular that numerous local hosts initiate legitimate connections with it (creating a "flash crowd" effect). Therefore, an incidence of a large number of connections per foreign address is perceived as an attack signal only if there are no better anomaly models. This, for instance, is done for UDP-based attack detection but not for TCP and ICMP attacks.

The observation component gathers sent and received packet and byte counts for TCP, ICMP and UDP portion of each agflow, during the last observation interval. At the end

of this interval, the agflow statistics for each protocol field are compared with corresponding legitimate traffic models. An agflow will be classified as:

1. *Attack* if at least one field's statistics do not match the corresponding model. When an attack occurs, a change in traffic statistics will first lead to attack classification of the associated agflow. The attack classification lasts as long as at least one of two conditions is met: 1) Agflow statistics mismatch the model; 2) there are packet drops on the agflow, due to rate-limiting. Since the detection signal is generated based on the agflow's traffic actually *forwarded* by D-WARD to the victim, an additional measure of rate-limit drops is needed to differentiate the case of no detection signal due to severe rate-limiting, from the case of no detection signal because the agflow is indeed well-behaving.
2. *Suspicious* if the statistics of all fields match their corresponding models, but the agflow has recently been classified as attack. When the attack stops, the corresponding agflow will be first classified as suspicious for *Compliance Period* seconds. Suspicious classification leads to a cautious, slow increase of the rate limit. If the attack repeats before *Compliance Period* has expired, the agflow will promptly be reclassified as an attack.
3. *Normal* if the statistics of all fields match their corresponding models, and the agflow has not been recently classified as attack. Suspicious agflows are promoted to normal classification after they have been well-behaved for *Compliance Period* seconds. The distinction between suspicious and normal agflows is made to minimize damage from recurring attacks. Attacks that repeat with an interval shorter than *Compliance Period* will achieve their full force only in the first attempt, and will later be constrained due to suspicious classification.

Legitimate TCP agflows are modeled by a low ratio of number of packets sent and received on the agflow. During a TCP session, data flow from source to destination host is controlled by a constant flow of acknowledgments in the reverse direction. If the flow of acknowledgments subsides, this is regarded as a sign of congestion [6] and the sending rate is promptly reduced. D-WARD defines TCP_{rto} as the maximum allowed ratio of the number of packets sent and received on the agflow; exceeding this ratio leads to attack classification. The observed packet ratio is smoothed before classification. Additionally, the sent/received packet ratio for agflows that receive no reverse packets during the observation interval, is calculated by dividing number of sent packets with a constant less than 1. This ensures detection of highly distributed attacks that affect victim's operation, while sending at a low level from each source network.

Legitimate ICMP agflows are similarly modeled by a low ratio of the number of "timestamp," "information request," and "echo" packets sent and received on the agflow, that should not exceed the maximum allowed ratio $ICMP_{rto}$. The sent/received packet ratio for agflows that receive no reverse packets during the observation interval, is calculated by dividing number of sent packets with a constant less than 1. The frequency of other ICMP messages is expected to be so small that a predefined rate limit can be used to control that portion of the traffic.

The UDP protocol [7] is used for unreliable message delivery and in general does not require any reverse packets for its proper operation. Many applications that use UDP generate a relatively constant packet rate, but the maximum rate and connection dynamics depend heavily on the underlying application. D-WARD thus defines a very broad *legitimate UDP agflow model*, attempting to detect only those UDP attacks that use heavy subnet spoofing. An alternate approach is to build a legitimate agflow model for each application that uses UDP. This approach would likely detect more UDP-based attacks, but it would be prohibitively expensive as many more statistics would have to be stored. Since it is fairly simple to defend against UDP-based attacks close to the victim, D-WARD makes only a limited attempt in this direction. D-WARD's legitimate UDP agflow model is defined as two thresholds: n_{conn}—an upper bound on the number of allowed connections per destination; p_{conn}—a lower bound on the number of allowed packets per connection. The model classifies an agflow as an attack when both of these thresholds have been breached.

In addition to attack detection, the observation component gathers per-connection statistics and periodically compares them to legitimate connection models. We assume that a connection can only carry traffic from one protocol and one application. During periodic classification, a connection will be classified as *good* if its statistics match the corresponding model, *bad* if its statistics do not match the corresponding model, and *transient* if there is not enough data to perform classification. Good connection traffic is forwarded during the rate-limit phase, bad connection traffic is dropped, and transient connections have to compete for the leftover bandwidth.

A *Legitimate TCP connection model* is similar to legitimate TCP agflow model. It also defines the maximum allowed ratio of the number of packets sent and received on the connection. The connection is classified as good if its smoothed packet ratio is below this threshold.

D-WARD does not define *legitimate ICMP connection model*[1] because the ICMP traffic does not have connection semantics—it is generated infrequently to diagnose network problems. Legitimate ICMP connection models would be of little use to legitimate ICMP traffic, as short, infrequent connections would terminate shortly after being validated.

A *legitimate UDP connection model* is built on per-application basis. We identified several main categories of applications that use UDP, which are:

1. Domain Name Service (DNS),
2. Network Time Protocol (NTP),
3. multimedia streaming,
4. voice over IP (VoIP),
5. Internet multiplayer games,
6. Network File System (NFS), and
7. chat applications.

1. ICMP is a connectionless protocol, but D-WARD regards all ICMP communication between two hosts as one connection.

We compared the selected categories with statistics from the CAIDA Web site [8] which shows the top 25 applications (regardless of underlying transport protocol). All UDP applications from this list fall among our selected categories. We designed legitimate connection models for all categories and we have implemented models for DNS, NTP and multimedia streaming traffic in the current D-WARD version. All three models (DNS, NTP, and multimedia streaming) are finite state machine models. They are described in detail in [2]. For DNS and NTP traffic, legitimate connection models capture request-reply dynamics. For multimedia streaming, legitimate connection models capture the establishment of the TCP control channel for streaming media and correlate its dynamics with UDP flow used for one-way stream traffic. As new uses of UDP become popular, new connection models will need to be generated and added to D-WARD.

According to the above description of connection classification, TCP connections that originate during an attack would be first classified as transient (after TCP SYN is sent) and TCP SYN traffic from those connections would compete for limited bandwidth with the attack traffic. This would likely result in sustained drops on new connections as attack packets are numerous and have more chance to win the bandwidth. Persistent packet drops are a serious problem for the TCP protocol and they quickly drive the sending rate of a legitimate connection to minimum. A new connection sustaining consecutive packet losses will send less traffic and do so more reluctantly as each packet is lost. Over time, it will either terminate or be severely delayed by the defense system action. On the other hand, if TCP SYN packets are always classified as legitimate by D-WARD, TCP SYN attacks are possible.

There are several strategies to ameliorate this situation. One is to assume TCP SYN cookie [9] deployment by the victim, and always forward TCP SYN packets. Since many hosts do not deploy TCP SYN cookies, D-WARD would fail to protect such hosts from the TCP SYN flood attack. Another strategy is to proxy TCP connections. D-WARD can intercept client's requests for service and reply with a TCP SYN cookie, then replay these requests to the server when the client completes the three-way handshake. Since server chooses its sequence number independently from the one chosen by D-WARD, the rest of the TCP connection would have to be proxied. This would significantly increase D-WARD's connection statistics and processing overhead. Another strategy (taken by many defense systems) is to let each TCP SYN packet through, but to reset half-open connections after a certain time. Its drawback is that D-WARD would have to keep state on half-open connections, thus becoming a potential victim of denial-of-service attack itself. The last strategy is to predict a valid range of some header fields in TCP SYN packets for a given source address, then reject packets that do not carry values within the range. The prediction function must be accurate enough so that legitimate packets will always be detected and validated. The predicted value range must also be relatively small so that packets that randomly spoof this value have a low probability of being validated as legitimate. D-WARD uses this last approach, predicting a valid range of sequence numbers on classified-transient TCP connections. Packets falling within the predicted range are considered legitimate and forwarded.

On TCP connection setup, each party chooses an initial sequence number (ISN) for this connection and numbers the bytes it sends starting from ISN. The ISN value was initially designed as a time-driven, linearly increasing value [10]. This approach generated a security hole and the opportunity for TCP connection hijacking attacks as explained in [11]. To counter these attacks, the ISN generation process was altered to include a small amount of randomness so that the ISN value cannot be guessed by the attacker, while it is still slowly increasing over time. In order to devise a sequence number prediction function, we first examined how different operating systems choose their initial sequence numbers. We initiated multiple consecutive TCP connections to a machine running Windows 2000, Windows XP with service pack 1, Windows XP with service pack 2, Red Hat Linux 7.1, Red Hat 9.0, FreeBSD 4.8, and FreeBSD 5.4. Our experiments show that FreeBSD and Windows XP with service pack 2 generate purely random initial sequence numbers while Windows 2000, Windows XP with service pack 1 and Linux generate initial sequence numbers that slowly increase with time. Windows 2000 and Windows XP with service pack 1 choose their initial sequence numbers purely as a function of time ($ISN = T$) while Linux introduces a small amount of randomness ($ISN = T + rnd$) as suggested in [11].

D-WARD predicts a range of possible sequence number values in TCP packets from a given source address, using sequence number values from the two most recently validated connections and the time difference to calculate ISN, then allowing for a small range of values around this number.[2] During an attack, D-WARD will attempt to validate each outgoing TCP packet from transient connections using the predicted range. If the packet falls within the range, it will be subject to *Early Packet Rate Limit*; otherwise, it will be subject to regular rate-limiting. A stealthy attacker can still learn the predicted value range and adequately spoof his packets to fit into the predicted range. Since the stealthy attack traffic is subject to rate-limiting with *Early Packet Rate Limit*, the attacker cannot misuse the value prediction technique to perform successful attacks. Thus, the range prediction technique significantly ameliorates the situation in a randomly spoofed attack case, and does not make it any worse in a stealthy attack case.

The observation component stores statistics in the *Agflow Table* and the *Connection Table*, respectively. As spoofed attacks may generate a large number of records in these tables, table size is limited to avoid excessive memory consumption. To accommodate all the relevant information in limited-size tables, the observation component cleans the tables: 1) periodically, expelling all the stale records—expiration period is a customizable parameter, and 2) on overflow, expelling those records that are deemed less useful than others. The Agflow table's primary function is to keep data on attack agflows and the assigned rate limits;

2. Note that the degree of the sequence number prediction that is sufficient for D-WARD purposes is not sufficient to carry out connection hijacking attacks.

low-sending-rate records which are classified as normal will be termed "less useful" and deleted on overflow. The Connection table's primary function is to remember all the legitimate connections so their traffic could be forwarded; low-sending-rate connections classified as transient or bad will be termed "less useful" and deleted on overflow.

3.2 Rate-Limiting Component

When the attack has been detected, D-WARD responds by rate-limiting all outgoing traffic to the victim and, thus, relieves the victim of a heavy traffic volume. Rate-limiting is chosen instead of filtering, to allow faster recovery from false alarms. Instead of deploying a fixed rate limit, D-WARD attempts to determine (guess) the maximum sending rate that the foreign host can handle. The problem of regulating the sending rate of a one-way flow to the level manageable by the (route to the) receiver has been recognized and addressed by the TCP congestion control mechanism, at the individual connection level. D-WARD strives to solve a similar problem at a more aggregated scale, controlling all the traffic to the victim and inferring the foreign host's state from the attack detection signal. D-WARD's rate-limit strategy applies modified TCP congestion control ideas to this problem. A fast exponential decrease of the sending rate is performed when the attack is detected to quickly relieve the victim of high-volume traffic. Once the attack subsides, D-WARD performs a slow recovery of rate-limited agflows, linearly increasing the sending rate. This is done to probe the state of the receiver, and to reevaluate its ability to handle traffic. After a while, if the attack is not repeated, D-WARD performs a fast recovery of rate-limited agflows, increasing the sending rate exponentially. In addition to the attack signal, D-WARD also bases its rate-limit settings on the observed agflow behavior. If an agflow seems congestion-responsive and adapts to the rate limit, this limit will be decreased more slowly and increased more rapidly than if the agflow appears misbehaving. This policy facilitates fast recovery of misclassified legitimate (congestion-responsive) agflows while severely limiting ill-behaved, aggressive agflows that are likely part of the attack.

Rate limit values are adjusted after each observation period, based on the agflow classification results and agflow behavior history. This history is expressed through *Agflow Compliance Factor* which is calculated as:

$$acf = \frac{B_{sent}}{B_{sent} + B_{dropped}},$$

where B_{sent} represents the byte amount of agflow traffic forwarded to the victim and $B_{dropped}$ represents the byte amount of agflow traffic dropped due to rate limiting; both measured during the last observation interval. *Agflow Compliance Factor* values range from 0 to 1, where higher values indicate better compliance with the imposed rate limit.

Exponential Decrease. When an agflow is classified as an attack agflow for the first time after a long period of being normal, its rate is limited to a fraction of the offending sending rate, according to the formula

$$rl = \frac{B_{sent}}{Observation\ Interval} * f_{dec}.$$

The size of the fraction is specified by the configuration parameter f_{dec}. Subsequent classification of an agflow as an attack restricts the rate limit further, applying exponential decrease according to the formula

$$rl = min\left(rl, \frac{B_{sent}}{Observation\ Interval}\right) * f_{dec} * acf,$$

where rl is the current rate-limit. The lowest rate limit that can be imposed is defined by the *Min Rate* configuration parameter, so that at least some packets can reach the destination and trigger a recovery.

Linear Increase. When the attack-detected signal becomes negative, the associated agflow will be classified as suspicious. The agflow recovery phase is divided into slow-recovery (linear rate increase) and fast-recovery (exponential rate increase). The linear rate increase is performed according to the formula $rl = rl + rate_{inc} * acf$. The speed of the recovery is defined by the $rate_{inc}$ parameter.

Exponential Increase. When an agflow has been classified as normal, the fast-recovery phase is triggered, and the rate is increased exponentially according to the formula $rl = rl * (1 + f_{inc} * acf)$. The speed of the recovery is defined by the f_{inc} parameter, and the rate increase is limited by the *Max Rate* configuration parameter. As soon as the rate limit becomes greater than *Max Rate*, it is removed.

3.3 Traffic-Policing Component

The traffic-policing component periodically receives rate-limited agflow information from the rate-limiting component, and connection classification information from the observation component. It uses this information to reach a decision whether to forward or drop each outgoing packet. Packets from nonlimited agflows and good connections are always forwarded. TCP packets from transient connections on limited agflows, whose sequence number matches the predicted value, are forwarded if the *Early Packet Rate Limit* for the agflow is not exhausted. Other transient-connection packets are forwarded if the agflow's rate limit is not exhausted.

3.4 Parameter Values

D-WARD has many configurable parameters that guide its operation. Agflow and connection model parameters TCP_{rto}, $ICMP_{rto}$, n_{conn}, and p_{conn} are set based on the examination of legitimate traffic seen by the deploying network and choosing values that are tight around measured traffic parameters. In our experiments, we used values of 3, 1.1, 100, and 3, respectively. Table sizes and record expiration values *Agflow Table Size*, *Agflow Expiration*, *Connection Table Size*, *Good Connection Expiration*, *Transient Connection Expiration*, *Rate Limit Table Size*, *Limited Agflow Table Size*, and *Good Connection Table Size* are set empirically to satisfy a trade-off between high operation accuracy (larger table sizes and fuller tables facilitate keeping all the relevant information) and low per-packet overhead (smaller table sizes facilitate fast record search). Their values are given in the Table 1. Observation intervals for agflow and connection classification should be set to minimize detection

TABLE 1
D-WARD Memory Cost

Structure	Record Size	No. records	Total Size
User Agflow Table	332 B	1,003	325.2 KB
User Connection Table	128 B	50,003	6.1 MB
User Rate Limit Table	32 B	1,003	31.3 KB
Kernel Agflow Table (Limited Agflows)	132 B	103	13.3 KB
Kernel Connection Table (Good Connection List)	84 B	2,003	164.3 KB

and response delay. It usually takes 1-2 observation intervals to detect the attack and another 2-3 intervals to estimate and set the appropriate rate limit. In our experiments we used 1 second as the value of observation intervals. Values of *Compliance Period* and *Max Rate* are chosen to satisfy a tradeoff between fast recovery after false classifications (lower values) and resilience to repeated attacks (higher values). In our experiments we used values of 20 seconds and 80 Mbps respectively. Low values of *Min Rate* and *Early Packet Rate Limit* facilitate control of highly distributed, low-rate attacks but may inflict collateral damage if the deploying network originates new connections with high frequency. We used values of 160 bps and 800 Kbps, respectively. Finally, parameters f_{dec}, f_{inc}, and $rate_{inc}$ are used for agflow's sending rate estimation. They have to be chosen so the estimated rate is tight around the real sending rate. We used values of 0.5, 1, and 4 Kbps which we determined empirically.

3.5 Stealthy Attackers

The attacker may resort to stealthy techniques to avoid attack detection or to trick D-WARD into classifying malicious connections as legitimate.

3.5.1 Small-Rate or Nonspoofed Attacks

An attacker may try to avoid detection by distributing attack traffic over multiple source networks so that each network observes only a small portion of the attack. Let us first assume that the attack is generated by sending low-rate TCP traffic. TCP attacks can deny service in two ways: 1) sending high number of TCP SYN packets that consume the victim's memory resources and inhibit origination of new connections; existing TCP connections are not affected, and 2) sending high volume of TCP packets that consume victim's bandwidth resources; all traffic is affected. If the source network has a large number of legitimate TCP connections with the victim and if the victim is not severely affected by the attack, it will generate a sufficient number of replies to the legitimate and the attack traffic alike. A small-rate attack will thus manage to "hide itself" among legitimate packets and avoid detection but will not inflict denial of service at the victim. If, on the other hand, the victim is severely affected by the attack it will send a few or no replies to the source network. In case of a TCP SYN flood attack no new connections will be allowed to originate but sent/received packet ratio on the associated agflow will appear low as long as old connections continue to exchange traffic with the victim. When old connections terminate, the attack will be detected as the sent/received ratio will exceed the allowed maximum value. Thus, the detection of low-rate TCP SYN attacks will be delayed by the average duration of legitimate TCP connections. In case of a TCP flood attack, victim will be unable to respond to legitimate and attack traffic alike. Legitimate TCP connections will react to this by reducing their sending rate, quickly revealing the presence of the attack traffic which will continue to send packets at a fixed rate. In this case, attack traffic will dominate sent/received packet ratio and lead to attack detection.

Let us now assume that the attack is generated by sending low-rate ICMP traffic from numerous source networks. ICMP attacks deny service by sending high-volume traffic which consumes victim's bandwidth resources. If the attack is effective, victim will be unable to generate sufficient amount of replies which will trigger attack detection, regardless of how low is the attack's sending rate per network.

An interesting case occurs if the attack is generated by sending stealthy UDP traffic. UDP attacks deny service by consuming the victim's bandwidth resources. The attack detection is triggered if D-WARD detects subnet spoofing in UDP agflow to the victim. Legitimate UDP agflow models cannot detect those UDP attacks that use limited or no spoofing. However, these attacks can be detected through a secondary effect, in case when source networks originate legitimate TCP or ICMP communications with the victim with some frequency. When the UDP attack affects victim's operation, it will be unable to generate replies to the legitimate TCP and ICMP traffic. New connection requests will drive the sent/received packet ratio for TCP or ICMP traffic to high values during several observation intervals, thus triggering attack detection.

If the source network does not originate TCP or ICMP communications with the victim, stealthy UDP attacks will not be detected by D-WARD. In a distributed DDoS defense system, this could be improved by D-WARD complementing its detection with victim-end attack alerts.

3.5.2 Low-Frequency Pulsing Attacks

A sophisticated attacker could attempt to avoid detection by performing pulsing attacks—sending the attack traffic during T_{on} intervals, pausing during the next T_{off} intervals, then repeating the cycle. If all the attack machines send traffic and pause simultaneously, the victim will experience periodic but short-lived denial-of-service effect. Attacks with T_{off} value less than *Compliance Period* will have a limited effect on the victim since associated agflow will still be rate limited, but attacks will longer T_{off} values will succeed. Increasing the value of *Compliance Period* parameter will improve this situation but will delay agflow

recovery in case of false positives. If the attacking machines take turns in sending traffic to the victim, so that the victim is under sustained heavy attack but each attacking machine is inactive during more than *Compliance Period*, D-WARD will detect but will not be able to control this attack.

3.5.3 Spoofing Acknowledgments

When classifying TCP and ICMP connections and flows, D-WARD relies on return packets from the foreign host to determine if it is under attack. If an attacker could spoof these reply packets, then D-WARD would believe that it was seeing legitimate traffic. The attacker would gather a number of cooperating slaves outside of the D-WARD-deploying network, and synchronize their spoofed acknowledgments with the attack traffic generated by the slaves residing within the source network. Due to the small size of acknowledgments relative to attack packets, the number of outside slaves could be much smaller than number of inside slaves and the synchronization could be preprogrammed (e.g., both sets of slaves use the same random number generator seed for packet sequence/acknowledgment numbers).

One possible solution is to have D-WARD allocate a small set of records and delay randomly chosen outgoing TCP packets, storing their contents in these records. Each outgoing and incoming TCP packet is then matched against these records. Duplicate outgoing packets are dropped, while the spoofed acknowledgments trigger the attack detection and invalidate the associated connection. This method is not currently implemented in D-WARD.

3.6 Security

Special care must be taken to secure D-WARD against intrusion, misuse and attacks that deny service to legitimate clients. Compromising a D-WARD router would allow an attacker to apply rate limits on any packets flowing out of the domain. However, the attacker can do much more damage with full control of the router, so adding D-WARD functionality makes the situation no worse. Further, D-WARD is unlikely to make it easier for the attacker to break into the router since it exchanges no messages with other entities.

Clever attackers will try to disguise attack traffic as normal traffic so that D-WARD will not filter it. One way to do this would be to generate a high number of legitimate-like TCP connections with the victim. These connections would have to be congestion responsive, backing off when some traffic is dropped, to be classified as legitimate by D-WARD. The attacker would thus be able to monopolize the victim's resources. D-WARD will not be able to handle this type of attack.

The attacker could aim his attack, not at a particular target machine, but at an incoming link to a target network, sending multiple small-rate flows to many target machines. If the attack is effective, some target machines will not generate sufficient responses and their associated agflows will be tagged as attacks.

D-WARD operation is based on statistics and classifications stored in agflow and connection tables. Since these tables are of limited capacity, they could overflow if the attacker generates a large amount of spoofed packets. D-WARD performs emergency clean-up of tables in case of overflow, as described in Section 3. The agflow table could overflow if the attacker generated traffic to a large number of foreign hosts. On overflow, D-WARD purges small-rate, classified-normal records from agflow table, ensuring that large-rate, suspicious-looking records remain for future observations. An attacker would need to generate large-rate flows or successful attacks to numerous destinations to keep the agflow table full, or to cause D-WARD to expel an important record and thus miss attack detection. We find this attack unrealistic. On the other hand, connection table overflow would lead to degraded service to those connections that have been expelled from the table before being classified as legitimate. As D-WARD usually takes one observation interval to validate legitimate connections, the connection table has to be large enough to accommodate all the legitimate connections plus the transient connections generated by spoofed traffic within one interval, to avoid collateral damage from the overflow.

An attacker could perform a denial-of-service attack on the source network by preventing outside responses from reaching the D-WARD. D-WARD would thus conclude that the outgoing DDoS attack is in progress and engage the rate limit. While the needless rate limiting would hurt some legitimate traffic, those connections that are well-behaved would be quickly classified as legitimate and excluded from rate-limiting. Furthermore, lack of reverse traffic would itself seriously hurt affected legitimate connections, so adding D-WARD does not make situation any worse.

An attacker could attempt to spoof a legitimate user's traffic to make D-WARD drop this user's legitimate packets. Spoofing only legitimate user's IP address with random port numbers generates separate connection record for each packet and does not affect user's legitimate connection traffic. Spoofing correct port numbers and IP addresses from a legitimate user's connection could indeed interfere with connection's classification. In case of a light attack, spoofed packets will elicit ACK responses from the victim as described in TCP protocol specification [10] and the connection will be classified as good. In case of a heavy attack, few or no responses will be generated and legitimate user's connection will be classified as bad. This may lead to legitimate packets being dropped by D-WARD. In the absence of D-WARD, this traffic would still not receive good service, as the victim is incapacitated by the heavy attack, but may receive better service than if it is dropped by D-WARD. We note here that the attacker must be able to sniff legitimate user's traffic to learn correct connection parameters.

3.7 Deployment Incentive

Current Internet has seen a limited deployment of systems that are designed to protect other participants from harmful actions of the deploying networks, e.g., egress filtering [4]. As D-WARD-deploying network does not receive substantial benefit from its operation this could be a major hurdle for widespread D-WARD deployment. We discuss below several compelling reasons that should motivate D-WARD deployment, but we note that the challenges are great. One requirement for source side deployment of a DDoS defense mechanism is strong evidence that such a defense can be effective. This paper gives the strongest such evidence to

date. Thus, the research described here is interesting even if a source side deployment of D-WARD or another DDoS defense mechanisms is not imminent.

Many people have concluded that stopping DDoS attacks completely is impossible since there is a vast number of machines whose owners are unaware of security holes or unwilling to fix them. D-WARD elevates the security requirement from individual machine to ISP or stub network level. A single D-WARD system installed at the exit router of a large University would prevent DDoS attacks originating from anywhere in the University's network in spite of unsecured machines within. Persistent traffic drops due to D-WARD response could signal the presence of compromised machines in the network.

The future may produce laws and legislations that hold DDoS-originating networks liable for harmful traffic. In this case, a network deploying D-WARD significantly lowers its risk of liability suits by following best security practices.

With DDoS attacks becoming more prevalent in the Internet, we expect widespread deployment of victim-end defenses that may be pushed to ISP level to provide paid protection to multiple users. These defenses will need accurate traffic separation to avoid collateral damage. Source-end defense, D-WARD in particular, provides means to victim's users to elevate their traffic to a privileged status. In cooperation with victim and core defenses, traffic from the networks deploying D-WARD could be detected as "guaranteed legitimate" and served with priority. We see this as the most compelling argument for D-WARD deployment.

3.8 Asymmetric Traffic

If a source network has several gateway routers (some or all of which are coupled with D-WARD) it may happen that some flow and connection traffic will exhibit asymmetric behavior, traversing different gateways in incoming and outgoing directions. This will cause problems to D-WARD's statistic gathering, creating incomplete observations and resulting in misclassifications. There are two possible solution to this problem: 1) D-WARD instances at different gateways communicate to exchange traffic statistics prior to classification. This would enable each D-WARD system to form a complete view of the traffic, but would create a lot of communication overhead as statistics would have to be exchanged prior to every classification. 2) D-WARD instances could be installed within the source network at connection points between stub subnetworks and the rest of the source network. Thus D-WARD would have a complete view of the subnetwork traffic and police it appropriately.

4 Performance Results

D-WARD is implemented in a Linux router and in an Intel IXP router. The Intel IXP router implementation was done as a part of [12], and detailed performance results are presented there. The Linux router implementation was done as a part of [2], and we present here selected results that illustrate D-WARD performance. The Linux router implementation consists of two parts: 1) the user-level implementation of the monitoring and rate-limiting components to accommodate large statistics tables, and 2) the loadable kernel module implementation of a traffic-policing component to provide high-speed packet processing.

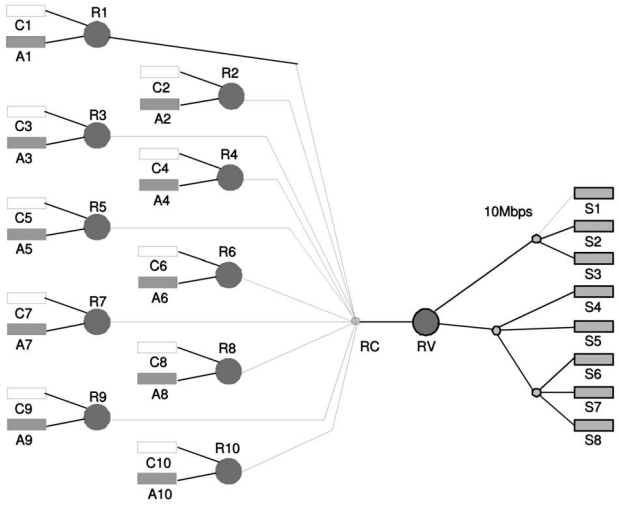

Fig. 2. A sample topology for D-WARD evaluation. Nodes C1-C10 represent legitimate clients, A1-A10 represent attacking machines (slaves) and routers R1-R10 are source routers that deploy D-WARD. Nodes S1-S10 are servers that belong to the victim network. They are connected to the rest of the Internet via the router RV. All links in the topology are 100 Mbps links except the one connecting the server S1 to the router RV, which is 10Mbps link.

4.1 Experiment Setup

To capture the variety of possible attacks and to fairly evaluate the D-WARD system, we had to allow variations in a number of different dimensions: 1) test topology, 2) background workload of legitimate traffic, and 3) attack characteristics. To test D-WARD with large topologies, we chose the Emulab [13] test facility for the experiments. We then created several large topologies, containing multiple source networks and a single victim network. One such topology is presented in Fig. 2.

To generate a realistic background traffic we use the `tcpdump` traces of traffic exchanged between the UCLA Computer Science Department and the rest of the Internet. These traces were captured during August 2001. Replaying these traces through tools such as [14], [15] would not be able to capture proper TCP connection behavior when some replayed packets are dropped, which is necessary for proper connection classification and for measuring connection delay due to collateral damage. We designed a *trace reconstruction tool*, called *tracegen* that produces *live* legitimate traffic whose characteristics resemble those from a supplied *tcpdump* file. More details about this tool can be found in [2] and the tool will be made available to public within DETER/EMIST project [16].

To test different attack scenarios, we developed a customizable DDoS attack tool, called *cleo*. It can generate all the attacks found in well-known DDoS attack tools such as trin00, TFN, TFN2K, Stacheldraht, and mstream, and it uses a master-slave architecture to coordinate attacks among multiple slaves. Attack traffic mixture (relative ratio of TCP SYN, ICMP_ECHO, and UDP packets), packet size, attack rate, target ports, spoofing techniques (random,

subnet, or none), and attack dynamics (constant, pulsing, or gradually increasing) can be customized.

The most important metric of defense success is the level of service that legitimate traffic receives during an attack. We calculate the *legitimate traffic service level* metric as the total amount of legitimate traffic that reaches the victim while the attack is ongoing. This amount is expressed as a percentage of the amount received in the baseline case, when no attack is present. The legitimate traffic service level metric takes the values between 0 and 1, with 1 being the ideal case where the attack does not hinder legitimate clients' traffic at all.

While performing the experiments with legitimate and attack traffic only, and without active defense (to measure the denial-of-service effect when the defense is not present), we noticed that the denial-of-service effect is variable. In repeated experiments, with identical scenarios, legitimate traffic had variable success in competing with the attack, resulting in a wide range of measured performance metrics. Since all connections (legitimate and attack) were initiated in an identical manner and observing the same timing, we conclude that the variability stems from packet interactions in competition for a limited resource. To capture this variability, we ran each experiment ten times and we present in graphs all the measurements, along with lines depicting minimum and maximum values.

In addition to legitimate traffic service level, we captured several other metrics: individual connection delay and failure, legitimate traffic drops (collateral damage), attack detection and response time (time of the first attack packet drop), and the attack level. The attack level metric shows the total amount of the attack traffic delivered to the victim by the defense system, relative to the amount of the attack traffic received by the victim when no defense is present. Ideally, this amount should be 0 (if D-WARD detected the attack and responded instantaneously).

In all our experiments, we tag attack packets by placing value of 255 in the *type of service* field in the IP header. As no legitimate packet will have this value, the tagging process enables us to easily tell legitimate from the attack packets when we measure traffic levels at the victim. This tagging is done only for the purposes of gathering statistics on defense system performance. D-WARD does not examine this field and does not base any decisions on the values contained in it.

4.2 Performance Results in Controlled Experiments

In our experiments, we generated a wide variety of attacks to test D-WARD performance. We present here the experiments with TCP SYN, UDP, and ICMP flooding attacks of varying strengths, using the topology presented in Fig. 2.

4.2.1 UDP Flooding Attacks

We generate UDP flooding attacks, engaging all 10 attack machines, A1-A10, simultaneously to flood the victim S1, targeting the 10 Mbps link in front of it. Attackers generate 1 KB-long packets, maximizing the chance of bandwidth consumption, and use subnet spoofing putting the stress on D-WARD's bookkeeping mechanism. The experiments last 5 minutes (300 seconds) each and the attack is generated during 100 second interval, from 20 to 120 seconds. Fig. 3a depicts the service level in these experiments in case with and without the defense. The x-axis shows the total amount of attack traffic flooding the critical link, and the y-axis shows the measured service level. Without defense, the service starts degrading when the attack traffic exceeds the link capacity—10 Mbps. It degrades severely as the attack rate increases, averaging about 5 percent for the attacks stronger than 80 Mbps. When the defense is engaged, D-WARD successfully relieves the denial-of-service effect at the victim, providing a 100 percent service level to legitimate clients during the attack.

Fig. 3b depicts the *maximum delay* experienced by a legitimate connection. Several connections were delayed due to the interaction of the legitimate traffic with the attack. Without the defense, connections start experiencing severe delay when the attack traffic exceeds the link capacity. Maximum per connection delay ranges from 61 seconds (for attack rates of 20 Mbps) up to 189 seconds (for attack rates of 60 Mbps and higher). D-WARD actions reduce maximum per connection delay to values no greater than 20 seconds for all attack rates. We note here that connections were not delayed because of D-WARD's actions but because of the legitimate packets being starved for resources by the attack in those few seconds that D-WARD takes to detect and control the attack.

Fig. 3c depicts the number of *failed* legitimate connections that did not complete within a 5-minute experimental window. Those are the connections that are so severely delayed because of the interaction with the attack traffic, that they did not manage to recover after the attack was aborted. Without the defense, connections experience failures when the attack traffic exceeds the link capacity. Number of failed connections grows with the increase in the attack rate, ranging from 10 (for attack rates of 20 Mbps) up to 82 (for attack rates of 180 Mbps). D-WARD actions drastically reduce this metrics since legitimate connections only compete with the attack in those few seconds that D-WARD takes to detect and control the attack. Only 1 to 2 connections fail, even when attack rates are high.

Due to limited space, we summarize other relevant metrics, with detailed graphs found in [2]. D-WARD detects each UDP attack within the first second and starts dropping attack packets about 4 seconds later. This 4-second gap between detection and response is used by D-WARD to estimate the appropriate rate limit value and enforce it and is implementation dependent. Introducing sampling techniques and optimizing implementation should decrease D-WARD's response time. Since response delay leads to connection delays and failures due to interaction of the legitimate with the attack traffic, decreasing it should further improve D-WARD's performance. Attack level measured in all UDP flood attacks was 4 percent when D-WARD was engaged. Detailed trace examination shows that all attack traffic passes to the victim during 4-second gap, until response is engaged. After D-WARD rate limit mechanism starts dropping attack packets, nonresponsive attack traffic quickly forces rate limit values to a very low level leading to complete filtering of the attack. Since the attack lasts for 100 seconds, 4 seconds of successful attack

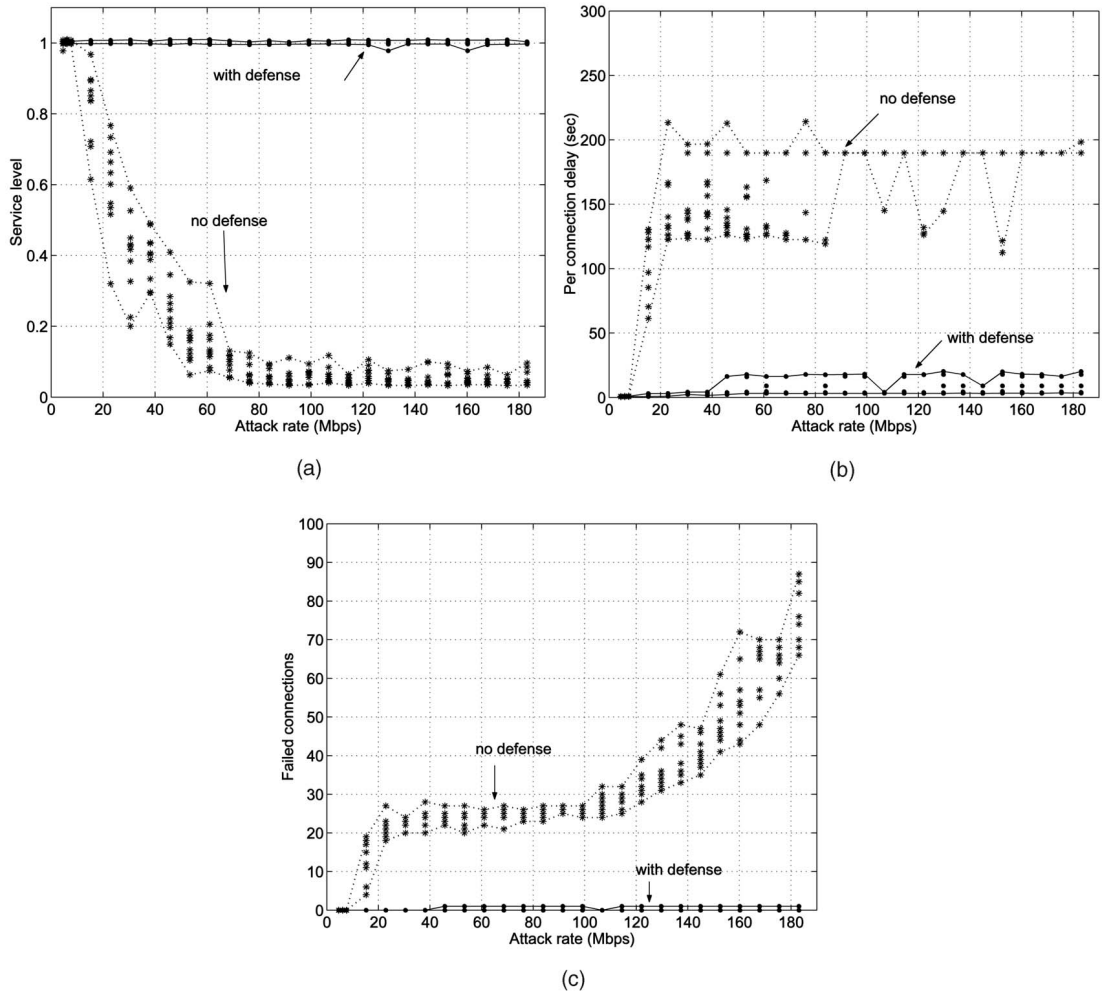

Fig. 3. UDP flood attack. (a) Service level, (b) maximum connection delay, and (c) failed connections.

lead to 4 percent attack level metric. If D-WARD response time is reduced, or the attack is prolonged, the attack level metric will decrease.

4.2.2 ICMP Flooding Attacks

The ICMP flooding attack is similar to the UDP flooding attack, as it also targets network link bandwidth. In our experiments, we generated an identical attack as in the UDP case, targeting the 10 Mbps link in front of the server S1. The detection mechanism, however, is different. Instead of detecting presence of IP spoofing like in the UDP attack case, D-WARD detects the low response rate from the victim to attacker's `ping` requests.

Fig. 4a depicts the legitimate traffic service level, with and without the defense. Just like in the UDP flood attack case, service level decreases as soon as the attack traffic starts overwhelming the bottleneck link. When D-WARD is present, service level is maintained at 100 percent, making the attack transparent to the legitimate users. Fig. 4b shows the maximum legitimate connection delay. Without defense, maximum delay ranges between 30 and 189 seconds, the higher attack rates leading to the larger delay. When D-WARD is present, maximum connection delay is less than 20 seconds, regardless of the attack rate. Fig. 4c shows the number of failed legitimate connections. Without defense, number of failed connections increases up to 79, as the attack rate increases. When D-WARD is present, at most 1 connection fails. D-WARD detects each ICMP attack within 1 to 2 seconds and starts dropping attack packets about 2 seconds later. After the initial 3 to 4 seconds, no attack traffic reaches the victim, which accounts for 4 percent value of the attack level metric.

4.2.3 TCP SYN Flooding Attacks

Unlike UDP and ICMP flooding attacks, a TCP SYN flooding attack does not target link bandwidth. Rather, it exhausts the TCP connection buffer space at the victim machine, by creating a lot of "half-open" connections with help of IP source address and port spoofing. In our experiments, we generated TCP SYN flooding attacks, targeting SSH port 22. As this port was heavily used by legitimate clients to communicate with the victim (all legitimate TCP traffic was tunnelled via SSH), attack traffic consumes a critical resource—the connection buffer at the victim. The attack spoofs random source addresses and ports in order to generate many connection records and quickly exhaust the buffer space. The number of generated connections is then the appropriate measure of the attack strength, and is proportional to the attack packet rate.

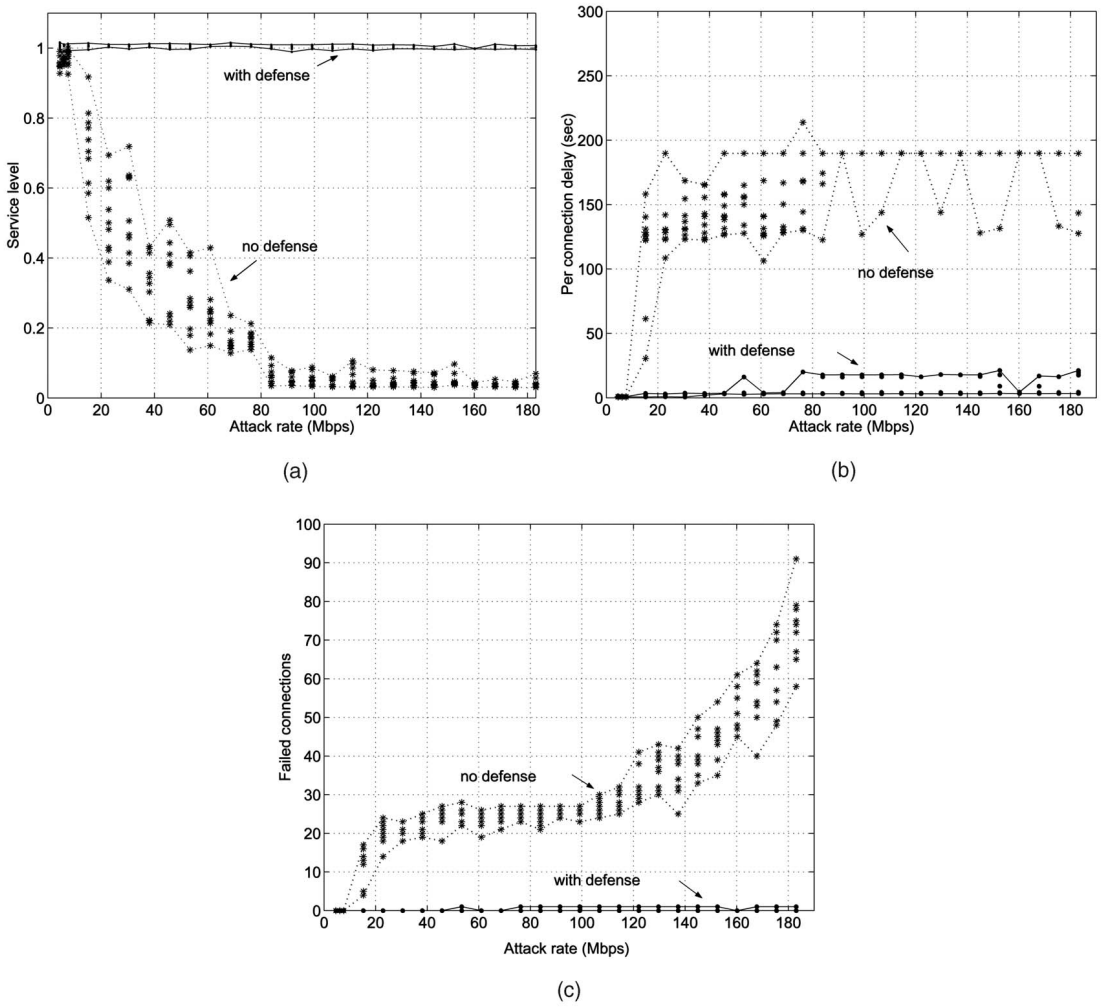

Fig. 4. ICMP flood attack. (a) Service level, (b) maximum connections delay, and (c) failed connections.

Fig. 5a depicts the service level in a case of TCP SYN flood attack. When no defense is present, the service level is quickly reduced to 90 percent at attack rates higher than 100 packets per second. However, unlike previous attack cases, service level is not affected by the attack rate increase. The reason for this is that TCP SYN attack exhausts connection buffer space, prohibiting setup of *new* connections, but it does not affect the connections established prior or after the attack. The service will be degraded by the amount of traffic belonging to those connections that originate while the attack is ongoing. In our tests, this amount is around 10 percent. When D-WARD is engaged, service level is maintained at 100 percent. Fig. 5b depicts the maximum legitimate connection delay. When no defense is present, legitimate connections experience the maximum delay of 190 seconds. D-WARD action brings the maximum connection delay down to 21 seconds, regardless of the attack rate. Fig. 5c shows the number of failed connections. When no defense is present, about 16 legitimate connections fail to complete. D-WARD action brings the number of failed connections down to 2.

D-WARD attack detection takes up to 20 seconds for the smallest attack rate (100 packets per second) but becomes prompt for higher attack rates (around 1 to 2 seconds). The response starts around 5 seconds after the detection. The detection delay for small attacks occurs because small attack rates do not create the denial-of-service effect promptly, but rather do so after some time. D-WARD cannot observe any anomalous events until connection buffer fills up; and detection lags after the actual onset of the attack. Response becomes active 1 to 2 seconds after the detection, leading to complete filtering of the attack traffic after this time, and the 7 percent attack level.

4.2.4 Low-Rate and Distributed Target Attacks

We test D-WARD's ability to detect low-rate attacks by deploying D-WARD only at $R1$ and generating low-rate attacks from the machine $A1$ in topology shown in Fig. 2. Machines $A2 - A10$ generate high attack volume. We generate UDP, TCP SYN, and ICMP floods varying the amount of traffic generated from $A1$, and test two attack cases: 1) moderate attack that lowers service level metrics to 70-90 percent and 2) heavy attack that lowers service level metrics to less than 40 percent. Figs. 6a, 6b, and 6c show the detection time of small rate UDP, TCP SYN, and ICMP attacks, with value of 300 seconds depicting a not-detected case. UDP and ICMP attack packets were 100 B long. We note that moderate attacks can avoid or postpone detection

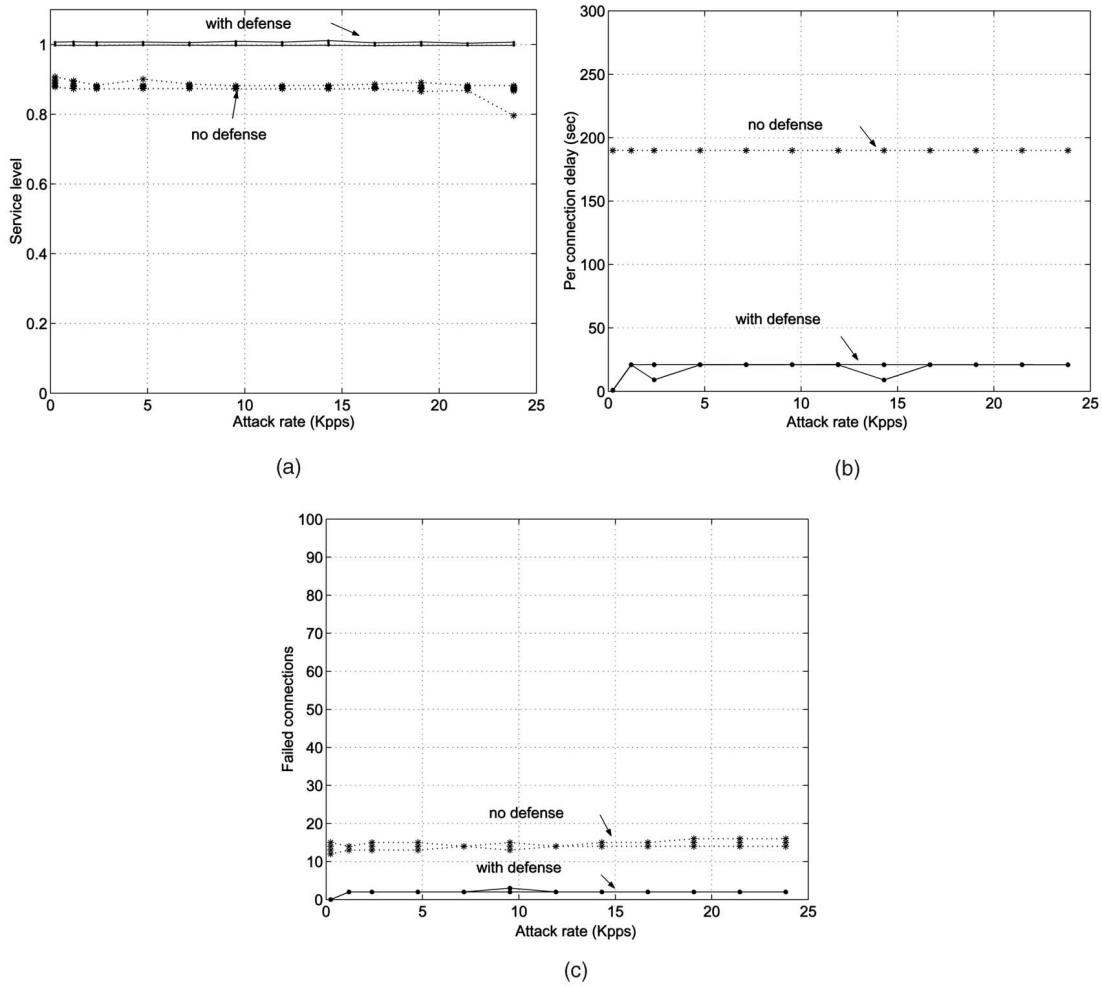

Fig. 5. TCP SYN flood attack. (a) Service level, (b) maximum connections delay, and (c) failed connections.

by deploying small-rate, distributed attack. In our experiments attacks sending less than 10 packets per second from $A1$ were occasionally missed by D-WARD. Heavy attacks were always detected regardless of the small sending rate.

We next test D-WARD's ability to detect distributed-target attacks by deploying D-WARD only at $R1$ and generating low-rate attacks targeting machines $S1 - S8$. The topology is shown in Fig. 2, but the bandwidth of the link $S1 - RV$ is set to 100 Mbps and the bandwidth of the link $RV - RC$ is set to 50 Mbps. We generate same attacks as in the previous experiment. For brevity, we summarize the results. Detection times resemble the ones from the previous experiment. Moderate attacks can avoid detection if they generate low-volume traffic from each source network. Attack rates of 1-100 packets sent from an individual network to all target machines were occasionally missed by D-WARD. Heavy-rate attacks were always detected. We conclude that D-WARD successfully detects and controls low-rate and distributed-target attacks if they inflict DoS effect at the victim.

4.2.5 False Alarms and Legitimate Packet Drops

In all of the above experiments D-WARD had no false alarms and no legitimate packet drops, when the attacks were not present. During the attack, D-WARD inflicted an extremely low collateral damage to the legitimate traffic. Below 0.01 percent of legitimate traffic was dropped in some of the experiments, while others generated no legitimate packet drops.

In order to stress-test D-WARD's false alarm rate and measure the range of collateral damage on legitimate traffic, we generated a "flash crowd" effect. In addition to the legitimate traffic generated in previous tests, we also generate simultaneous FTP requests for a same file, residing on the server S1, from all ten legitimate clients. In each run, a client generates N requests for a given file, spaced 1 second apart. The file is 1 MB large, thus requests and replies easily overwhelm the bottleneck link. We vary N from 1 to 100, thus generating from 10 to 1,000 simultaneous requests per second.

Fig. 7 shows the time when D-WARD falsely detects the attack, measured from the start of the flash crowd, and the amount of legitimate traffic dropped due to this detection. The detection occurs between 13 and 26 seconds from the start of the FTP request series, as shown in the Fig. 7a. The amount of legitimate traffic drops is given in bytes in the Fig. 7b, along with the maximum and median drop values. We observe that the highest drop is 305 B which is negligible compared to the 112 MB to 1.1 GB of legitimate traffic being transferred during test runs.

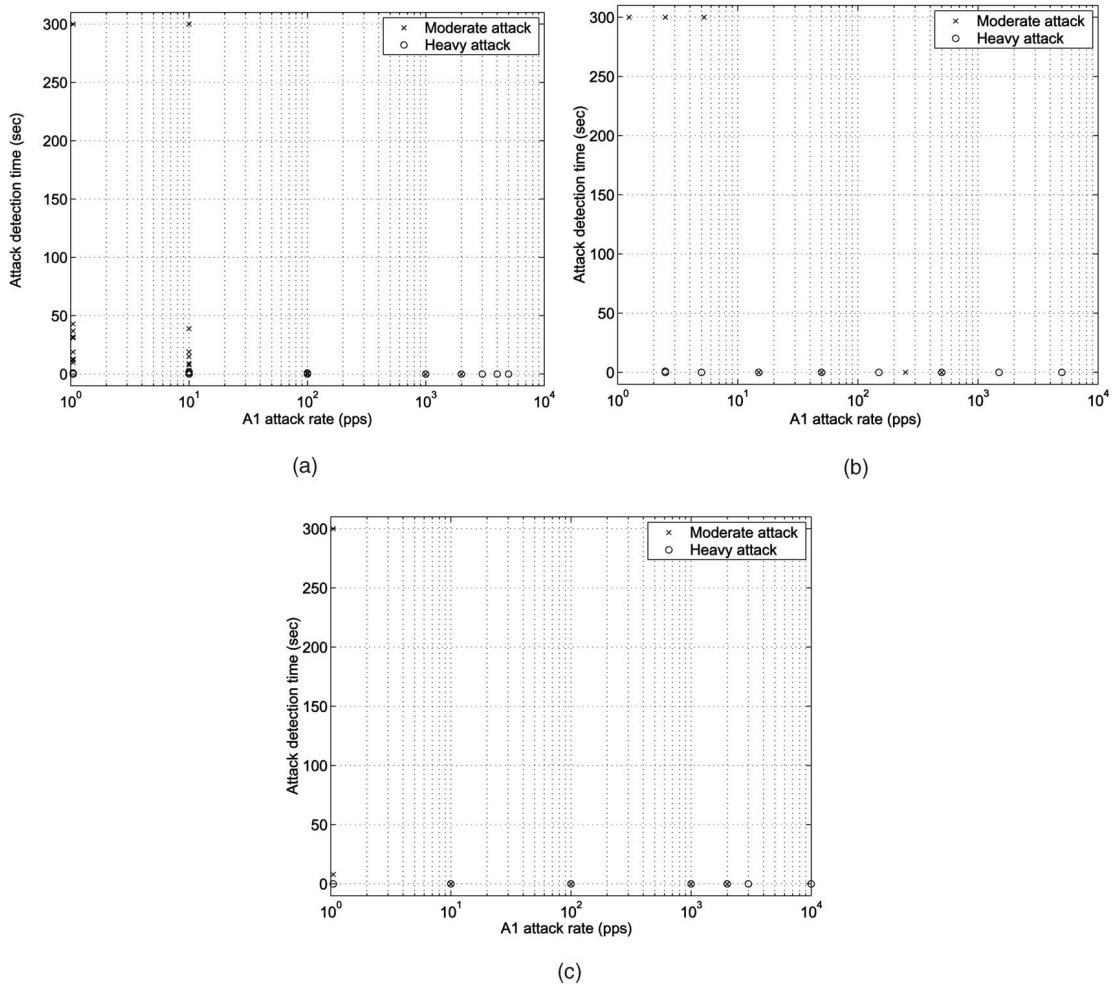

Fig. 6. Small rate attacks. (a) Detection time of small-rate UDP, (b) detection time of small-rate TCP SYN attacks, and (c) detection time of small-rate ICMP attacks.

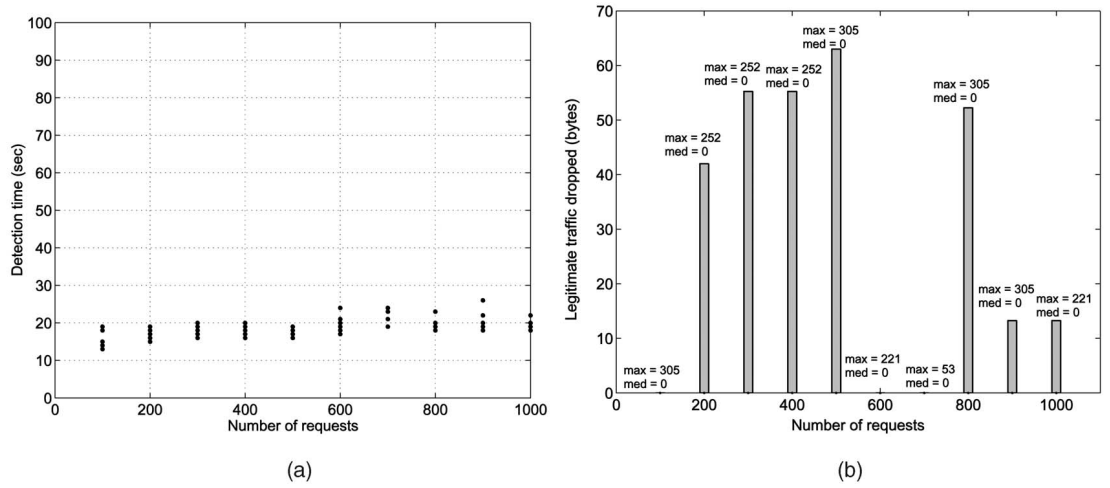

Fig. 7. False detection time and legitimate traffic drops in the flash crowd case. (a) False detection time in the flash crowd and (b) legitimate traffic drops in the flash crowd case.

To test D-WARD's performance with realistic traffic, we modified the system to read packet header data from a `tcpdump`-generated trace file instead of sniffing it from the network. We used nine packet traces gathered from the UCLA Computer Science network during August 2001. The network has approximately 800 machines and experiences an average of 5.5 Mbps (peak 20 Mbps) of outgoing traffic and 5.8 Mbps (peak 23 Mbps) of incoming traffic. We assume that no attack has occurred during the trace-gathering process.

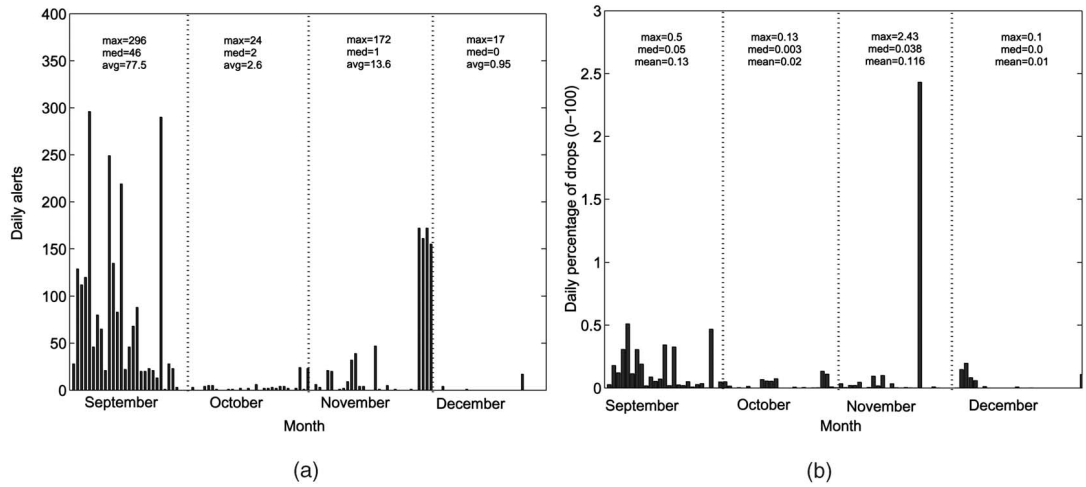

Fig. 8. D-WARD alarm rate and traffic drops in real operation. (a) Alarm rate and (b) traffic drops.

We define the following metrics for measuring the level of false positives:

1. the percentage of unique flows that were misclassified as attack or suspicious,
2. the percentage of unique connections that were misclassified,
3. the percentage of flow misclassifications as attack or suspicious, and
4. the percentage of connection misclassifications as bad.

Metrics 1 and 2 show what portion of the traffic will possibly be affected by a bad decision. Metrics 3 and 4 show how good the detection process is. We had between 0.23-1.36 percent unique misclassified flows and 0.41-0.77 percent unique misclassified connections. Overall there were 0.10-0.43 percent flow misclassifications and 0.003-0.085 percent connection misclassifications. A low false positive measure in the flow classification case indicates that D-WARD is very unlikely to perform false detection and inflict damage on legitimate traffic. Furthermore, even if false detection occurs, D-WARD will preserve correctly classified legitimate connections, thus further minimizing the damage. As connection classification accuracy is very high, we conclude that D-WARD would inflict extremely low collateral damage in real operation.

4.3 Performance Results in Real Operation

D-WARD has been continuously deployed from 09/2003 to 01/2004 between the machines in the LASR lab at UCLA (20 Linux/FreeBSD/Windows machines, 2 multiuser machines) and the rest of the Internet. D-WARD policed all traffic (including NFS file sharing) exchanged between LASR lab and the Internet. All users were aware of D-WARD's deployment and were instructed to disable D-WARD and file complaints if they experience traffic drops. Additionally, authors examined attack detection, connection classification, and packet drop logs on regular basis and investigated all alerts.

Figs. 8a and 8b summarize daily attack alerts and traffic drops, on month basis. Through trace and log examination we were able to establish that our network was not misused for DDoS attacks during these four months, i.e., Figs. 8a and 8b really represent false alarms and collateral damage. Out of all attack alerts, 50.4 percent were triggered by aggressive TCP implementations, 43.1 percent by aggressive portscans, 4.6 percent by file sharing traffic using *bittorrent*, and 1.9 percent by other traffic. During September we had a high number of false alerts (and user complaints), caused by aggressive TCP implementations on some of our machines. Those implementations rapidly open the congestion window in the slow-start phase without waiting for recipient's acknowledgments, which triggers D-WARD's attack detection. We tuned D-WARD to accommodate for aggressive-TCP machines by applying different TCP thresholds to those machines' traffic, which significantly reduced our false alarms in the following months. No user complaints were registered in the last three months of operation.

Operation Cost. We measure deployment cost as memory cost (for D-WARD statistics storage) and the packet handling overhead. Table 1 gives the storage cost and the configuration parameters used in the experiments.

Packet handling overhead under normal operation is measured using the ping utility in the topology shown in Fig. 2. A total of 1,000 ICMP ECHO packets is sent from C1 to S1, with D-WARD engaged at R1. The average additional one-way delay imposed by D-WARD was $3\mu s$ per packet. The kernel-level implementation incurs additional overhead for spoofed packet handling, which becomes critical at rates higher than 12,000 packets per second.

5 Related Work

There are numerous approaches to DDoS defense and network security in general. For brevity, we provide an overview of the approaches similar to D-WARD, and summarize the rest.

Most systems for combating DDoS attacks, such as [17], [18], [19], [20], [21], [22], [23], [24], [25], work on the victim side, detecting anomalous behavior. While victim-end defense cannot provide complete protection from heavy DDoS attacks, it could be possible to deploy a victim-end defense system at the source-end, reversing its functions so

that it examines and polices the outgoing traffic. Since the attack traffic does not create the same set of anomalies at the source-end as it does at the victim-end, it is likely that the sensitivity of most victim-end defenses will be inappropriate for autonomous source-end defense. However, we expect that most of the victim-end attack detection techniques can successfully be combined with D-WARD to improve its attack detection.

Resource accounting mechanisms, such as [26], [27], [28], [29], [30], police the access of each user to resources based on the privileges of the user and his behavior. As DDoS is in fact a resource overloading problem, resource accounting approaches are well suited to address it. However, they impose a burden on the defense system to ensure user identity, keep state per user, and are not well suited to handle large-scale spoofing attacks. Resource accounting techniques could be combined with D-WARD, to assure fair bandwidth distribution among legitimate connections.

Distributed defense systems, such as [5], [31], [32], [33], [34], [35], [36], [37] locate defense nodes at several points at edge (victim and source) networks, or at the edge and in the core. Distributed defense has a definite advantage over single-point defense, and we believe that in the future it will provide a complete answer to DDoS problem. We see D-WARD as one of the main building blocks of the future distributed DDoS defense systems. Distributed systems usually attempt to detect an attack near the victim, then engage defense actions as close to the sources as possible to minimize collateral damage. Combining D-WARD with a distributed DDoS defense can greatly enhance performance of both parties. D-WARD can improve its detection by receiving victim-end attack alerts and advisories on desired actions, while distributed defense can improve its selectiveness by deploying D-WARD's source-end response. This effect was also demonstrated in the integration of the D-WARD system with COSSACK [33] and with DefCOM [32]. More details about this can be found in [2].

We are aware of two source-end defense systems—MULTOPS [38] and Reverse Firewall [36]. MULTOPS [38] is a heuristic and a data-structure that network devices can use to detect DDoS attacks. Each network device maintains a multilevel tree, monitoring certain traffic characteristics and storing data in nodes corresponding to subnet prefixes at different aggregation levels. The attack is detected by abnormal packet ratio values, and offending flows are rate-limited. The system is designed so that it can operate as either a source-end or victim-end DDoS defense system. While the high-level design of this system has much in common with D-WARD, the details are different. MULTOPS models normal flows at a lower granularity than D-WARD, using only the aggregate packet ratio, which may lead MULTOPS to miss some attacks that D-WARD catches. Non-TCP flows require special handling by MULTOPS. It further imposes a fixed, nonselective rate limit on outgoing or incoming traffic (based on its deployment at source or victim-end), thus likely inflicting collateral damage on legitimate traffic. D-WARD offers selective and dynamic response with almost no legitimate traffic drops.

Reverse Firewall (RF) [36] prevents DDoS attacks by limiting the rate of "unexpected" TCP packets at a network's exit router. The sequence numbers of expected packets are derived from the foreign-peer acknowledgements to previously sent traffic. Only the outgoing packets matching the expected sequence number range will be forwarded. This technique is similar to ISN prediction but it requires storing extensive TCP data for each legitimate connection, whereas ISN prediction stores data per live local machine. While RF is likely to be more accurate in distinguishing legitimate from attack traffic, this improvement can only be marginal compared to D-WARD's already excellent traffic separation. The improvement is gained at a large storage and processing cost that may make the system unable to handle a large traffic volume. RF deploys a fixed rate limit on "unexpected" packets and on non-TCP traffic and, thus, cannot adapt to dynamic changes in network conditions. Rate limits are engaged at all times, which leads to a difficult trade-off for setting the limit. Small values lead to poor resource utilization, as they cannot accommodate traffic bursts. Large values, on the other hand, do not control highly distributed attacks, as their traffic passes below the rate limit. D-WARD successfully addresses this problem by dynamically adjusting rate limits based on current traffic observations.

6 Conclusions

DDoS is a complex and difficult problem. Many solutions have been proposed to handle this threat, usually with variable effectiveness and cost, depending on the attack characteristics. In this paper, we propose D-WARD, a source-end DDoS defense system, which can work both as an autonomous system or as a source-end component in a distributed defense. More than being just another DDoS solution, we see D-WARD as a missing piece on DDoS defense scene. It provides a dynamic and selective response to an attack, maximally preserving legitimate traffic and adjusting to varying network conditions. Both the dynamics and the selectiveness of the response have been missing from current DDoS solutions accounting for large collateral damage to legitimate traffic at times of the attack.

D-WARD has demonstrated an excellent performance both in the test-bed experiments and in the real operation. It autonomously detects and effectively stops a wide range of DDoS attacks, ensuring good service to legitimate clients, and incurs low operation cost. It can also be integrated with other defense systems, achieving a synergistic effect. A complete DDoS solution is likely to require some kind of a distributed system where attack detection is performed near the victim and the response is deployed close to the sources of the attack. In such a system, D-WARD would be one of the crucial building blocks, as it would provide a dynamic and selective source-end response, ensuring a fair treatment even of those clients that share a network with an attacker.

Acknowledgments

The research on D-WARD project was supported by DARPA under contract number N66001-01-1-8937. The implementation of D-WARD in Intel IXP router was made possible thanks to an equipment grant from Intel Corporation.

REFERENCES

[1] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the Source," *Proc. Int'l Conf. Network Protocols,* Nov. 2002.

[2] J. Mirkovic, "D-Ward: Source-End Defense against Distributed Denial-of-Service Attacks," PhD dissertation, Univ. of California Los Angeles, Aug. 2003, http://lasr.cs.ucla.edu/ddos/dward-thesis.pdf.

[3] J. Mirkovic, G. Prier, and P. Reiher, "Challenges of Source-End DDoS Defense," *Proc. Int'l Symp. Network Computing and Applications,* Apr. 2003.

[4] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing," *RFC 2827,* May 2000.

[5] R. Mahajan, S. Bellovin, S. Floyd, V. Paxson, and S. Shenker, "Controlling High Bandwidth Aggregates in the Network," *ACM Computer Comm. Rev.,* vol. 32, no. 3, July 2002.

[6] V. Jacobson, "Congestion Avoidance and Control," *ACM Computer Comm. Rev.; Proc. Sigcomm '88 Symp.,* vol. 18, no. 4, pp. 314-329, Aug. 1988.

[7] J. Postel, "User Datagram Protocol," *RFC 768,* Aug. 1980.

[8] Characterization of Internet Traffic Loads, Segregated by Application, CAIDA, http://www.caida.org/analysis/workload/byapplication/, 2002.

[9] C. Schuba, I. Krsul, M. Kuhn, G. Spafford, A. Sundaram, and D. Zamboni, "Analysis of a Denial of Service Attack on TCP," *Proc. 1997 IEEE Symp. Security and Privacy,* May 1997.

[10] I.S. Institute, "Transmission Control Protocol," *RFC 793,* Sept. 1981

[11] S. Bellovin, "Defending against Sequence Number Attacks," *RFC 1948,* May 1996.

[12] G. Prier, "iDward: Implementing D-WARD in the IXP," Master's thesis, Univ. of California Los Angeles, 2003.

[13] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, "An Integrated Experimental Environment for Distributed Systems and Networks," *Proc. Fifth Symp. Operating Systems Design and Implementation,* pp. 255-270, Dec. 2002.

[14] Sourceforge, "tcpreplay Tool," http://tcpreplay.sourceforge.net/, 2000.

[15] H. Bos, "tcpreplay-Lite Tool," http://www.cs.vu.nl/~herbertb/misc/replay/, 2004.

[16] DETER/EMIST project Web page, http://www.isi.edu/deter, 2004.

[17] NetRanger Overview, Cisco, http://www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids1/csidsug/overview.htm, 2004.

[18] Network Intrusion Detector Overview, Computer Incident Advisory Capability, http://ciac.llnl.gov/cstc/nid/intro.html, 2004.

[19] Intrusion Detection Security Products, Internet Security Systems, http://www.iss.net/securing_e-business/security_products/intrusion_detection/index.php, 2005.

[20] NFR Sensitivist Intrusion Detection System, NFR Security, http://www.nfr.com/solutions/sentivist-ids.php, 2003.

[21] P.G. Neumann and P.A. Porras, "Experience with EMERALD to DATE," *Proc. First USENIX Workshop Intrusion Detection and Network Monitoring,* Apr. 1999.

[22] S. Liu, Y. Xiong, and P. Sun, "On Prevention of the Denial of Service Attacks: A Control Theoretical Approach," *Proc. IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop,* June 2000.

[23] R. Thomas, T. Johnson, J. Croall, and B. Mark, "NetBouncer: Client-legitimacy-based High-performance DDoS Filtering," *McAfee Security J.,* vol. 6, no. 1, 2004.

[24] Mazu Technical White Papers, Mazu Networks, http://www.mazunetworks.com/solutions/white_papers/, 2005.

[25] The Peakflow Platform, Arbor Networks, http://www.arbornetworks.com, 1999.

[26] A. Juels and J. Brainard, "Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks," *Proc. 1999 Networks and Distributed System Security Symp.,* Mar. 1999.

[27] Y.L. Zheng and J. Leiwo, "A Method to Implement a Denial of Service Protection Base," *Information Security and Privacy,* 1997.

[28] O. Spatscheck and L.L. Petersen, "Defending against Denial of Service Attacks in Scout," *Proc. Third Symp. Operating Systems Design and Implementation,* Feb. 1999.

[29] A. Garg and A.L. N. Reddy, "Mitigation of DoS Attacks through QoS Regulation," *Proc. IWQOS Workshop,* May 2002.

[30] F. Lau, S.H. Rubin, M.H. Smith, and L. Trajkovic, "Distributed Denial of Service Attacks," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics,* pp. 2275-2280, Oct. 2000.

[31] A.D. Keromytis, V. Misra, and D. Rubenstein, "SOS: Secure Overlay Services," *Proc. SIGCOMM 2002,* 2002.

[32] J. Mirkovic, M. Robinson, and P. Reiher, "Forming Alliance for DDoS Defenses," *Proc. New Security Paradigmes Workshop,* Aug. 2003.

[33] C. Papadopoulos, R. Lindell, J. Mehringer, A. Hussain, and R. Govindan, "Cossack: Coordinated Suppression of Simultaneous Attacks," *Proc. DARPA Information Survivability Conf. and Exposition (DISCEX) III,* 2003.

[34] R. Canonico, D. Cotroneo, L. Peluso, S.P. Romano, and G. Ventre, "Programming Routers to Improve Network Security," *Proc. OPENSIG 2001 Workshop Next Generation Network Programming,* Sept. 2001.

[35] D. Schnackenberg, K. Djahandari, and D. Sterne, "Infrastructure for Intrusion Detection and Response," *Advanced Security Research J.,* vol. 3, no. 1, 2001.

[36] MANAnet DDoS White Papers, Cs3. Inc, http://www.cs3-inc.com/mananet.html, 2002.

[37] T. Peng, C. Leckie, and K. Ramamohanarao, "Defending against Distributed Denial of Service Attack Using Selective Pushback," *Proc. Ninth IEEE Int'l Conf. Telecomm. (ICT 2002),* 2002.

[38] T.M. Gil and M. Poletto, "MULTOPS: A Data-Structure for Bandwidth Attack Detection," *Proc. 10th Usenix Security Symp.,* Aug. 2001.

Jelena Mirkovic received the PhD degree from the University of California, Los Angeles in 2003 and is currently an assistant professor at the Univeristy of Delaware. Her research interests include all the areas of network security and specificaly focus on denial-of-service characterization, detection and defense, worm simulation and defense, and IP spoofing defense. She is a member of the IEEE.

Peter Reiher received the PhD degree from University of California, Los Angeles (UCLA) in 1987. After working on the Time Warp Operating System at the Jet Propulsion Laboratory for five years, he returned to UCLA, where he is an adjunct associate professor. His research interests include network security, operating systems, distributed systems, and ubiquitous computing. He is a member of the IEEE.