Optimal Allocation of Filters against DDoS Attacks

Karim El Defrawy
ICS Dept.
University of California, Irvine
Email: keldefra@uci.edu

Athina Markopoulou
EECS Dept.
University of California, Irvine
Email: athina@uci.edu

Katerina Argyraki
I&C School
EPFL, Lausanne, CH
Email: katerina.argyraki@epfl.ch

***Abstract*— Distributed Denial-of-Service (DDoS) attacks are a major problem in the Internet today. During a DDoS attack, a large number of compromised hosts send unwanted traffic to the victim, thus exhausting the resources of the victim and preventing it from serving its legitimate clients. One of the main mechanisms against DDoS is filtering, which allows routers to selectively block unwanted traffic. Given the magnitude of DDoS attacks and the high cost of filters in the routers today, the successful mitigation of a DDoS attack using filtering crucially depends on the efficient allocation of filtering resources.**

In this paper, we consider a single router with a limited number of available filters. We study how to optimally allocate filters to attack sources, or entire domains of attack sources, so as to maximize the amount of good traffic preserved, under a constraint on the number of filters. First, we look at the single-tier problem, where the collateral damage on legitimate traffic is high due to the filtering at the granularity of attack domains. Second, we look at the two-tier problem, where we have an additional constraint on the number of filters and filtering is performed at the granularity of attackers and/or domains. We formulate both problems as optimization problems, and we evaluate the optimal solution over a range of realistic attack-scenarios. Our results demonstrate that efficient filter allocation significantly improves the tradeoff between the number of filters used and the amount of legitimate traffic preserved.

I. Introduction

Distributed Denial-of-Service attacks (DoS) are one of the most severe and hard to solve problems on the Internet today. During a DDoS attack, a large number of compromised hosts coordinate and send unwanted traffic to the victim thus exhausting the victim's resources and preventing it from serving its legitimate clients. For example, victims of DDoS attacks can be companies that rely on the Internet for their business, in which case DDoS attacks can result in severe financial loss or even in the company quitting the business [1]. Government sites (e.g. www1.whitehouse.gov) and other organizations can also be victims of DDoS attacks, in which case disruption of operation results in a political or reputation cost.

Several approaches and mechanisms have been proposed to deal with DDoS attacks. In this work, we focus on filtering mechanisms, which are a necessary component in the anti-DDoS solution. We consider the scenario of a bandwidth flooding attack, during which the bottleneck link to the victim is flooded with undesired traffic. To defend against such an attack, the victim must identify undesired traffic (using some identification mechanism which is not the focus of this work) and request from its ISP/gateway to block it before it enters the victim's access link and causes damage to legitimate traffic. Even assuming a perfect mechanism for identification of attack traffic, filter allocation at the victim's gateway is in itself a hard problem. The reason is that the number of attack sources in today's DDoS attacks is much larger than the number of expensive filters (ACLs) at the routers. Therefore, the victim cannot afford to selectively block traffic from each individual attack source, but instead may have to block entire domains; in that case legitimate traffic originating from that domain is also unnecessarily filtered together with the attack sources. Clearly, the successful mitigation of a DDoS attack using filtering, crucially depends on the efficient allocation of filtering resources. In this paper, we study the optimal allocation of filters to individual attackers or entire domains of attackers. Filters can be placed at a single gateways' tier, so as to maximize the preserved good traffic; the core insight in the single-tier problem is that the coarse filtering granularity makes co-located attack and legitimate traffic to share fate. We also consider filter placements at two tiers (attackers and gateways); in this case, the trade-off is between the preserved goodput and the number of filters used. We evaluate the optimal solution for three realistic attack scenarios, based on data sets from the analysis of the Code Red [16] and Slammer [17] worms, the Prolexic Zombie Report [19], and statistics on Internet users [20].

The structure of the rest of the paper is as follows. In section II, we give more background on the problem and we discuss related work. In section III, we formulate the problem of optimal filter allocation on a single tier (i.e. gateways or attackers tier) and the more general problem of filtering at both the gateway and attacker tier. We solve the problem optimally using dynamic programming. We study the properties of the optimal solution and evaluate it through simulation in section IV. In section V, we conclude the paper and discuss open issues and future work.

II. Background

A. Flooding Attacks and Filtering

In this paper, we are concerned with a DDoS attack on network bandwidth, also called flooding attack. A flooding attack is very easy to launch as it only requires sending a certain amount of traffic that overwhelms the link connecting the victim to the Internet. An example of flooding attack is shown in Fig. 1. A victim (V) is connected to the Internet through ISP-V, using an access link with bandwidth C. The victim is under a DDoS attack from several attack sources

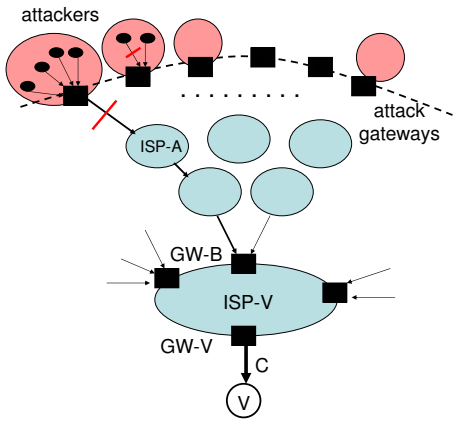


Fig. 1. The victim (V) is connected to its ISP (ISP-V) through an access router (GW-V) and an access link (with bandwidth C). GW-B is a border router of ISP-V. Attackers are located in various ASes behind the attack gateways; the total traffic exceeds the capacity C.

hosted by other ISPs, such as ISP-A. The total traffic coming from those sources exceeds the total capacity C.

Filtering is one of the mechanisms that can help to mitigate DDoS attacks and stop the unwanted traffic from reaching the victim and consuming network bandwidth along the way. For example, in Fig.1, the victim can send a filtering request to its own ISP-V to block all traffic from ISP-A to the victim. ISP-V responds by placing filters at appropriately chosen gateway(s), e.g. GW-V or GW-B. In this paper, we are not concerned with choosing the best gateway within an ISP for placing the filters; instead we look at a single gateway, say GW-V, and how to allocate filters to attackers or attack domains.

By "filters", we refer to access control lists (ACLs), which allow a router to match a packet header against rules. E.g. in the DDoS case described above, the router checks if the packet is going to victim V and coming from attacking host A; or the router might check the source IP address and filter out any packet coming from the entire ISP-A. Packet filters in routers are a scarce, expensive resource because they are stored in the expensive TCAM (ternary content addressable memory). A router linecard or supervisor-engine card typically supports a single TCAM chip with tens of thousands of entries. So, depending on how an ISP connects its clients to its network, each client can typically claim from a few hundred to a few thousand filters – not enough to block the attacks observed today and not nearly enough to block the attacks expected in the near future. We formulate two filtering problems: the *single-tier* and the *two-tier* filtering, depending on the granularity of packet filtering (or equivalently, the levels of the attack graph considered). In the single-tier case, we are interested in filtering entire attack gateways, a task for which there are enough filters today; in this context, we seek to filter out traffic so that the total traffic arriving at the victim is below the available bandwidth, while maximizing the preserved legitimate traffic. In the two-tier problem, we are interested in filtering not only attack gateways but also individual attackers, a task for which there are not enough filters in a single router today; the number of filters becomes then an additional constraint.

B. Related Work

A taxonomy of DDoS attacks and defense mechanisms can be found in [2]. Here we review only aspects related to our work. Our work relies on existing mechanisms to be able to identify the attack traffic, distinguish it from the legitimate traffic and trace its approximate path back to the attack source [3]. The main difficulty in path identification lies in dealing with source IP address spoofing. Other mechanisms for path identification and traceback include probabilistically sending ICMP messages [5]; mechanisms based on hashing [6] or packet marking [7]. Also, in this work, we focus on filtering at a single router, typically at the victim's gateway. Looking at the bigger picture, several mechanisms have been proposed to enable filter propagation as close to the attack source a possible. For example, Pushback [8] enables routers to propagate filtering upstream hop-by-hop, at the router-level. AITF [9] proposes to communicate filtering information from the victim upstream towards the attack domain, but at the granularity of AS, as opposed to router.

Filtering is not the only mechanism for mitigation of DDoS attacks. Some of the proposed approaches revisit the basic assumption of the Internet architecture, stating that every host can send to any other host, without requiring permission of the receiving host. For example, capabilities which propose that tokens are obtained before establishing a connection with a destination, and that these token are included in each packet [10][11] [12] This proposal requires changing the routers on the Internet and adding new servers and changes the whole Internet architecture. Other proposals use overlay mechanisms to implement a similar concept which is to restrict communication to the victim only through some known well provisioned overlay nodes which can filter and detect attacks [13] [14].

Using filtering could provide a quick solution or first line of defense to DoS attacks, until a permanent one is developed and is already used today in commercially available systems and anti-DoS services [15]. The downside of filtering is that its performance heavily depends on being able to identify attack traffic and distinguish it from legitimate traffic, which is not an easy task. However, it is an available, reactive mechanism that can be used in conjunction with other approaches.

Finally, in this paper, we rely upon data from analysis of worms, to construct realistic attack scenarios. Internet worms are older than DDoS attacks, but are relevant for studying such attacks because they are used as a tool to infect and compromise hosts on the Internet with the attack clients. The Code Red [16] worm is one well-known worm from 2001, which contained code to launch a DoS attack on the website (www1.whitehouse.gov), which did not succeed. Recently, several other worms have been caused huge financial losses, such as Slammer [17], MyDoom, Flash worms [18] and others and have attracted a lot of researcher's attention. Prolexic [15] is also regularly publishing a very informative "Zombie

Report", on the most infected hosts per country, network service-provider and other meaningful groupings [19].

III. Formulation of Optimal Allocation of Filters

A. General Discussion

In principle, one might consider allocating filters at any level of the attack graph, see Fig.1. There is clearly a trade-off between filtering granularity (to maximize goodput) and the number of filters. If there were no constraints on the number of filters, the maximum throughput of good traffic (goodput) would be achieved by allocating filters as close to individual attackers as possible. The gateway in question (GW-V) faces the following tradeoff. Ideally, GW-V would like to filter out all attackers and allow all good traffic to reach the victim. Unfortunately, in a typical DDoS attack, there are not enough filters to individually filter all IP addresses of attack hosts. A solution is to aggregate attack sources into a single filter; in practice, there are enough filters available to filter at that granularity. E.g. GW-V could summarize several attack sources coming from the same domain, e.g. behind GW-1, into a single rule and filter out the entire domains, as shown in Fig. 2. However, there is also legitimate traffic coming from each domain. Therefore, filtering at the granularity of attack gateway-tier causes "collateral" damage to legitimate traffic that falls into the range of the IP addresses described by the filter. This problem, referred to as the "single-tier filtering", is studied in section III-B so as to preserve the maximum amount of legitimate traffic while meeting the capacity constraint. This turns out to be a knapsack problem that can be solved by a greedy algorithm (shown in Algorithm 1).

In practice, there are more filters (F) than attack gateways ($N < F$), but less filters than individual attackers ($F < \sum_{i=1}^{N} M_i$) (see Fig. 3). Filtering at the gateway level is feasible but causes the collateral damage discussed above, due to its coarse granularity. Filtering at the attacker's level would preserve the maximum possible throughput but it is not realistic (due to the high number of attackers as well as due to spoofing); we still consider it as an upper bound for performance. A practical and effective compromise between the two extremes can be the two-tier filtering, shown in Fig. 3. In the two-tier filtering, we can choose to filter either at gateways' granularity (e.g. filter 1 in Fig. 3) or at attackers' granularity (e.g. filter 2 in Fig. 3). The optimal allocation of filters to individual attack sources, or to entire attack gateways, depends on the characteristics of the attack (distribution and intensity) as well as on the number of available filters. Furthermore, the successful containment of the DDoS attack crucially depends on the optimization of the filter allocation.

B. Single-Tier Filter Allocation

The single-tier scenario is shown in Fig.2. There are N attacking gateways, each generating both good (G_i) and bad (B_i) traffic toward the victim; the total traffic toward the victim exceeds its capacity C. Gateway GW-V allocates filters to block the attack traffic towards V. There are enough filters to allocate to the N gateways. The objective is to allocate filters to limit the total traffic below the available capacity, so as to maximize the amount of legitimate traffic that is getting through to the victim (because this is what the victim cares about, e.g. revenue for a web server).

Algorithm 1 Greedy Algorithm for the Single-Tier.

- Order nodes in decreasing order $\frac{G_j}{G_j+B_j}$. W.l.o.g. $j = 1, 2, ..N$ from largest to smallest efficiency.
- Find the critical node c s.t.: $\sum_{j=1}^{j=c-1} G_j + B_j < C$ and $\sum_{j=1}^{j=c} G_j + B_j > C$
- Allocate filters to nodes $i = 1, 2, ...N$ as follows:
  - $x_j = 1$ for $j = 1, 2, ..c-1$ (allow to pass)
  - $x_c = \frac{C - \sum_{j=1}^{j=c-1} G_j + B_j}{G_c + B_c}$ (rate limiter)
  - $x_j = 0$ for $j = c+1, ..n$ (filters)

Let us use $x_i = 1$ and $x_i = 0$ to indicate whether we allow or block all traffic from gateway i. The problem of optimal allocation of filters is to choose $\{x_i\}_i^N$:

$$\begin{aligned} & max \ \sum G_i \cdot x_i \\ s.t. & \sum_{i=1}^{i=N} (G_i + B_i) \cdot x_i \leq C \\ & x_i \in \{0, 1\}, i = 1, 2, ..N \end{aligned} \tag{1}$$

We noticed that the filter allocation problem is essentially a 0-1 knapsack problem [21]. Recall that in the knapsack problem, we choose some among N objects, each with profit v_i and a weight w_i, so as to maximize the total profit, subject to a total weight constraint. In our case, the objects are the attacking nodes with profits and weights G_i and $G_i + B_i$ respectively; and there is a constraint C on the victim's bandwidth. This is well-known to be a computationally hard problem. However, we need computationally efficient solutions, because the filter allocation should be decided in real-time during the attack.

The continuous relaxation of the problem (where x is no longer binary, but instead $0 \leq x_i \leq 1$) can be interpreted as placing rate-limiters: we allow ratio x_i of the traffic coming from node i to get to the victim. This corresponds to the fractional knapsack problem, which can be solved optimally using a greedy algorithm [21]. The algorithm in Algorithm 1, shown below, sorts nodes in a decreasing order of efficiency $\frac{G_j}{G_j+B_j}$,[1] and greedily accepts ($x_i = 1$) nodes with the maximum efficiency, until a critical node c, which if allowed will exceed the capacity. Traffic from all remaining nodes is filtered out ($x_i = 0$) and installs a rate-limiter to the critical element ($x_c = \frac{C - \sum_{j=1}^{j=c-1} G_j + B_j}{G_c + B_c}$) to use the remaining capacity. This requires only $O(nlogn)$ steps for sorting and $O(n)$ for filer/rate-limiters allocation.

Notice, that it is impractical to allocate rate-limiters to all attacking nodes, because rate-limiters are expensive resources and require keeping state. Fortunately, the optimal solution of the fractional problem turned out to be

[1]Technically, maximizing $\sum \frac{G_i}{G_i+B_i}$ is different from maximizing $\sum G_i$. However because the optimal solution operates at $\sum G_i + B_i \simeq C$, it is the same in practice.

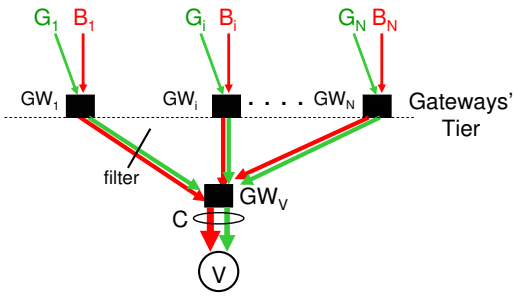


Fig. 2. Single-Tier Filtering Problem

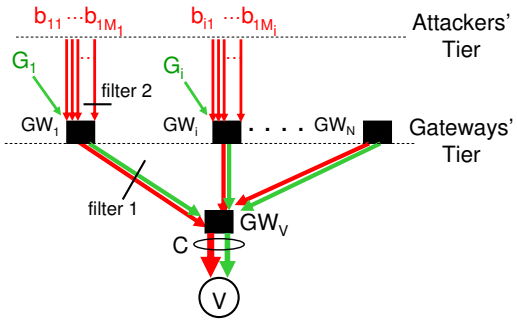


Fig. 3. Two-Tiers Filtering Problem

$(x_1, ...x_{c-1}, x_c, x_{c+1}, ..., x_N) = (1, ...1, x_{c-1}, 0, ...0)$, thus using $C-1$ filters and exactly one rate-limiter, which matches well current router resources.

C. Two-Tier Filter Allocation

The two-tier problem is the following. Consider N attack gateways and M_i attack hosts behind attack gateway i, i.e. the last two tiers in Fig.1. Each attacker contributes both good (G_{ij}) and bad traffic (B_{ij}), $i = 1, 2..N, j = 1, 2...M_j$. $x_{ij} \in 0, 1$ depending on whether we allocate a filter to attack-host j behind gateway i. $x_i \in 0, 1$ depending on whether we allocate a filter to attack-gateway i; if $x_i = 0$, then all traffic originating behind GW-i is blocked, and there is no need to allocate additional filters to attackers $(i, j), j = 1, 2, ...M_i$.

The problem is how to choose $\{x_i\}$'s, $\{x_{ij}\}$'s, given the constraints C on the victim's capacity and on the available number of filters F at the gateway:

$$
\begin{aligned}
&max \sum_{i=1}^{i=N} \sum_{j=1}^{j=Mi} G_{ij} \cdot x_i \cdot x_{ij} \\
&s.t. \sum_{i=1}^{i=N} \sum_{j=1}^{j=Mi} (G_{ij} + B_{ij}) \cdot x_i \cdot x_{ij} \leq C \\
&\sum_{i=1}^{i=N} (1 - x_i) + \sum_{i=1}^{i=N} \sum_{j=1}^{j=Mi} (1 - x_{ij}) \leq F \\
&x_i, x_{ij} \in \{0, 1\}, i = 1, ...N, \ j = 1, ..., M_j
\end{aligned}
\tag{2}
$$

The two-tier problem is harder than the single-tier one: it is a variation of the cardinality-constrained knapsack [21], and the optimal solution cannot be found efficiently. In this paper, we formulate the problem using dynamic programming and compute its optimal solution as a baseline for comparison. However, the dynamic programming algorithm is computationally expensive and cannot be used in real time; we are currently working on developing efficient heuristics.

Definitions. Consider the two-tiers configuration, shown in Fig. 3. There are N gateways. A gateway n generates legitimate traffic G_n and also attack traffic from M_n attack sources. Notice that for simplicity we depart from Eq.(2) and we consider that attacker ij generates only bad traffic b_{ij} and no goodput G_{ij}; instead we consider that the total goodput G_n comes from different hosts behind gateway n . W.l.o.g. consider that the attack sources are ordered from worst to best: $b(n, 1) > ... > b(n, M_n)$. Therefore, each gateway generates total traffic $C_n = G_n + \sum_{i=1}^{M_n} b(n, i)$. Before filtering, the total traffic exceeds the victim's access bandwidth (capacity) C: $\sum_{i=1}^{N} C_n > C$. We are interested in placing F filters across the N gateways, so as to bring the total traffic below C, while maximizing the total goodput after filtering $T_N(C, F)$. $T_N^*(C, F)$, can be computed recursively as shown in Algorithm 2. The recursion proceeds considering one more gateway at a time; the order in which gateways are considered is not important. Let $T_i^*(c, f)$, for $i \leq N$, be the maximum goodput of the smaller problem, i.e. with optimal placement of $f \leq F$ filters considering only gateways $\{1, 2, ..i\}$ and capacity up to $c \leq C$. Assume that, in previous steps, we have already obtained and stored the optimal solutions $T_i(c, f)$ considering only gateways $1, 2, ...n-1$, for all values of $c = 0, 1, ..C$ and $f = 0, 1, ...F$. Then $T_n^*(c, f)$ can be computed from the Bellman recursive equation (line 23 of Alg.2):

$$
T_n^*(c, f) = \max_{x=0,1,..f} T_{n-1}^*(c - (C_n - \sum_{j=0}^{j=x} b(n, j)), f - x) + G_n
\tag{3}
$$

Intuition. In step n, we consider gateway n together with the previous gateways $1, 2, ...n-1$. The f available filters are split among two groups of gateways: $\{1, 2, ..n-1\}$ and $\{n\}$; $x \leq f$ filters are assigned to gateway n and the remaining $f-x$ filters are assigned to the previous gateways $\{1, 2, ..n-1\}$. The x filters assigned to GW_n are used to block the x worst attackers. Therefore, $\sum_{j=0}^{j=x} b(n, j)$ bad traffic is blocked and the remaining $C(n) - \sum_{j=0}^{j=x} b(n, j) = G_n + \sum_{j=x+1}^{j=M_n} b(n, j)$ traffic goes through ($gwn_{unfiltered}$ in line 24), consuming part of the total capacity c. The remaining $f - x$ filters are

optimally assigned to gateways $1, 2, ...n-1$. Recall that we have previously obtained and stored the optimal solutions $T^*_{n-1}(c, f)$ considering only gateways $\{1, 2, ...n-1\}$, for all c and f; therefore, we already know the best allocation of $f-x$ filters across gateways $\{1, 2, ...n-1\}$ and we can get the maximum goodput $T^*_{n-1}(c-(C(n)-\sum_{j=0}^{j=x} b(n,j)), f-x)$.

We consider all possible values of x and choose the value among $0 \leq x \leq f$ that maximizes goodput (line 33 in Alg.2). There are some values of x that deserve special attention:

- $x = 0$ means that we assign no filters to gateway n, in which case our best goodput is the same as before, enhanced by the goodput of the current gateway: $T^*_{n-1}(c - C_n, f) + G_n$ ($max0$ in line 12 of Alg. 2).
- $x = 1$ means that we assign exactly one filter to gateway n, either at attacker or at gateway level. If we assign this filter to an attacker, it should be the worst attacker $b(n, 1)$ (line 16 in Alg.2). If this one filter is assigned to the entire gateway, then the entire traffic C_n from gateway n is filtered out and all goodput comes from the previous gateways $T^*_{n-1}(c, f-1)$ (line 18 of Alg.2). We compare the two options and choose the one that maximizes the overall goodput ($max1$ in line 19 of Alg.2).
- We consider increasing values of x until we either run out of filters ($x = f$) or we filter out all attackers in this gateway ($x = M_n$). Therefore, x can increase up to $min\{f, M_n\}$ (line 23 in Alg. 2).

Other technicalities in Algorithm 2 include the initializations (lines 1-3) and assigning $T^* = 0$ to infeasible problems (line 3-2^{nd} case and line 28).

Optimal Substructure. We are able to compute the optimal solution using dynamic programming (DP) because the problem exhibits the optimal substructure property.

Proposition. If a^* is the optimal solution for problem (n, c, f), then it contains a part $a^*_{\{1,...n-1\}} \subset a^*$ (corresponding to the filters assigned to the first n-1 gateways) which must also be the optimal solution for the smaller problem $(n-1, C-(C_n - \sum_{j=0}^{j=x} b(n,j)), f-x)$.

Proof: a^* is the optimal solution for problem (n, c, f), achieving maximum goodput $T^*_n(c, f)$.[2] This solution (filter assignment) must have two parts $a^* = (a^*_{\{1,2..n-1\}}, a^*_{\{n\}})$. The first part $a^*_{\{1,2..n-1\}}$ describes how filters are placed across gateways $\{1, 2, ..n-1\}$. The second part, $a_{\{n\}}$ describes how filters are assigned to gateway $\{n\}$ only. Let's look at the optimal solution a^*: it assigns some number of filters (x) to gateway n and the remaining ($f-x$) to gateways $\{0, 1, ..n-1\}$. This means that $\sum_{j=0}^{j=x} b(n,j)$ out of C_n traffic is filtered out at gateway n and the remaining $C_n - \sum_{j=0}^{j=x} b(n,j)$ is left unfiltered. The two parts contribute to the maximum throughput as follows:

$$T^*_n(c, f) := T\big|_{a^*} = T\Big|_{a^*_{\{1,2..n-1\}}} + T\Big|_{a^*_{\{n\}}}$$

[2] a^* will have the form of a vector $(1, 0, 0, ..., 0, 1)$; 0/1 describes whether an attacker or gateway has been filtered out or not; the attackers and gateways should be listed in the same order they are considered in the DP.

Algorithm 2 Dynamic Programming (DP) Formulation for the Two-Tiers Filtering Problem

```
1:  for n = 1, 2, ..N do
2:      T*_n(c = 0, :) = 0
3:      T*_n(:, f = 0) = { Σ_{n=1}^{N} G_n   if Σ_{n=1}^{N} G_n < C
                         { 0                 otherwise
4:  end for
5:
6:  for n ∈ [1, N] do
7:      for c ∈ [1, C] do
8:          for f ∈ [1, F] do
9:              /* x out of f filters are assigned to GW_n */
10:
11:             /* assign x = 0 filters to GW_n*/
12:             max0 = T*_{n-1}(c − C_n, f) + G_n
13:
14:             /*assign x = 1 filter to GW_n*/
15:             /* ...either at gateway level*/
16:             max1_gw = T*_{n-1}(c, f − 1)
17:             /* ...or at attacker level*/
18:             max1_att = T*_{n-1}(c − (C_n − b(n, 1)), f − 1) + G_n
19:             max1 = max{max1_gw, max1_att}
20:             max = max{max0, max1}
21:
22:             /* assign x ≥ 2 filters at attack level. */
23:             for x ∈ [2, min(f, M_n)] do
24:                 gwn_unfiltered := C_n − Σ_{j=1}^{x} b(n, j)
25:                 if c > gwn_unfiltered then
26:                     temp := T_{n−1}*(c − gwn_unfiltered, f − x) + G_n
27:                 else
28:                     temp := 0
29:                 end if
30:                 if temp > max then
31:                     max := temp
32:                 end if
33:                 T*_n(c, f) := max
34:             end for
35:         end for
36:     end for
37: end for
```

Assume that b, and not $a^*_{\{1,2..n-1\}}$, is the optimal filter assignment for the smaller problem $(n-1, C-(C_n - \sum_{j=0}^{j=x} b(n,j)), f-x)$. Then, by definition of the optimal filtering, it achieves larger goodput than the substructure $a^*_{\{1,2..n-1\}}$: $T^*_{n-1} := T\big|_b > T\big|_{a^*_{\{1,2..n-1\}}}$.

We can now construct another solution d for the larger problem (n, c, f) as follows. Replace the first part $a^*_{\{1,2..n-1\}}$ of a^* with b, for assigning $f-x$ filters up to gateway $n-1$, which would fit within capacity $C-(C_n - \sum_{j=0}^{j=x} b(n,j))$. Then, do exactly the same assignment as the DP would do, in Eq. 3, for assigning the x remaining filters to gateway n. This newly constructed filter assignment d has two parts $d = (b, d_2)$ that contribute to the total goodput.

The first part b is over gateways $\{1, 2, ..n-1\}$. We constructed this part to be the same as the optimal assignment of $f-x$ filters over gateways $\{1, 2, ..n-1\}$, with available capacity $C-(C_n - \sum_{j=0}^{j=x} b(n,j))$. Therefore it achieves optimal goodput $T\big|_b := T^*_{n-1} \geq T\big|_{a^*_{\{1,2..n-1\}}}$. The second part d_2 is an assignment over only gateway $\{n\}$. We constructed it to do exactly what the DP would do at step n with x available filters: either filter out the worst x attackers of gateway n (i.e. attackers $b(n,1)...b(1,x)$) or filter out the entire gateway

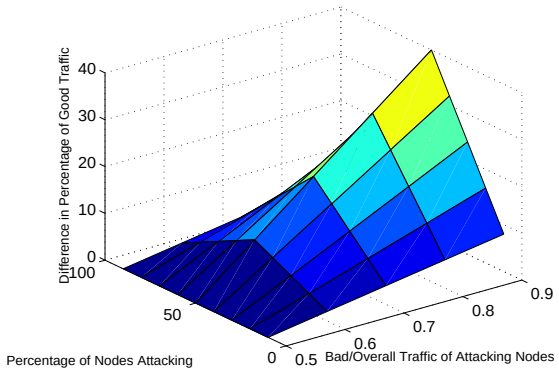


Fig. 4. Improvement from using optimal filtering for various attack intensities (% of attacking nodes, $B/(G+B)$). We consider $n = 1000$ attacking nodes, all sending at the same rate (10Mbps).

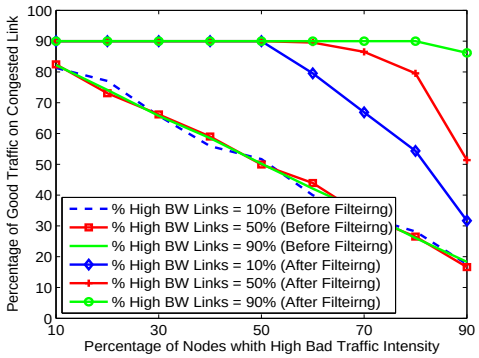


Fig. 5. Improvement from using optimal filtering for various % of nodes with high intensity, and various % of nodes sending at higher rate (100Mbps). $H = 0.9$ is fixed for all attacking nodes.

(if $x = 1$ is assigned at gateway level). Therefore, d_2 is by construction the same assignment as the DP's: $d_2 = a_{\{n\}^*}$ and results in the same goodput: $T\big|_{d_2} = T\big|_{a^*_{\{n\}}}$.

Therefore, we constructed a solution $d = (b, d_2)$ which performs better than the DP solution $a*$.

$$T\Big|_d = T\Big|_b + T\Big|_{d_{\{n\}}} > T\Big|_{a^*_{\{1,2..n-1\}}} + T\Big|_{a^*_{\{n\}}} = T\Big|_{a^*} \quad (4)$$

This is a contradiction because we assumed that $a*$ was an optimal solution for the bigger problem (n, c, f). Therefore the substructure $a^*_{\{1,...n-1\}}$ of the optimal solution a^* has to be the optimal solution for the smaller problem. ■

Cost of the Dynamic Programming. Computing the optimal solution value $T^*_n(C, F)$ using dynamic programming (DP) can be seen as filling up a table of size $N \cdot C \cdot F$. For realistic (large) values of N, C, F this can be prohibitively large, both from a run-time and from a memory point of view. While the number of gateways N can be moderate, C (normalized in units of the smallest attack rate) and F (in the order of thousands or tens of thousands) can be quite large in practice. An idea might be to work with coarser increments of C and F - which brings us already in the realm of heuristics for the DP, not addressed in this paper. Nevertheless, computing the optimal solution is still important as a benchmark for evaluation of any proposed heuristic.

Properties of the Optimal Solution. From the simulations in section IV-D, we made some preliminary observations. E.g., we compared the two-tier with the attack-tier-only and the gateway-tier-only filtering. Let $T_N(C, f)$, $G_N(C, f)$, $A_N(C, f)$ be the maximum goodput achieved by the optimal placement of f filters across N gateways, considering two-tier placement, single-tier placement at gateways and single attackers respectively. For the same attack scenario:

- $T_N(C, f) \geq A_N(C, f)$ and $T_N(C, f) \geq G_N(C, f)$. This is expected, because by definition, the optimal solution of the two-tier problem considers placing all filters at gateway and attack level, as special cases.
- $G^* < T < A^*$ where: G^* is the optimal filtering for (single) gateway tier and A^* is the optimal (single) attack tier filtering. Single-tier filtering does not have a constraint on the number of filters and is only constrained by the "collateral damage" on legitimate traffic.
- As $f \uparrow$, T converges to A^*, the optimal solution for attacker's single tier, without a constraint on f.

IV. Simulations

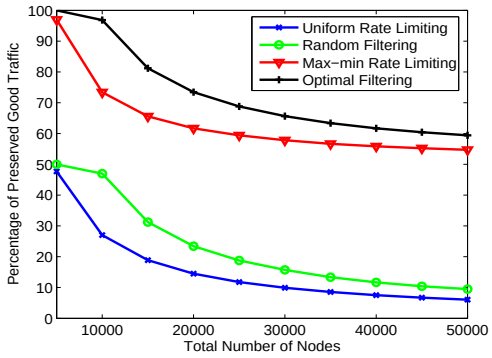
A. Single-Tier Artificially Generated Scenarios

We considered a wide range of scenarios and here we showcase some representative results. Let us fix the number N of attack nodes; we considered $N = 10, 100, 1000$. We control the intensity of the attack through a simple model with three parameters. (i) the bandwidth at which each node sends is a configurable parameter. (ii) x% of the nodes that are attacking and the remaining (100-x)% send legitimate traffic (iii) attacking nodes have all the same bad-to-overall traffic ratio $H = \frac{B}{B+G}$; the legitimate nodes have ratio $1 - H$ of bad to overall.

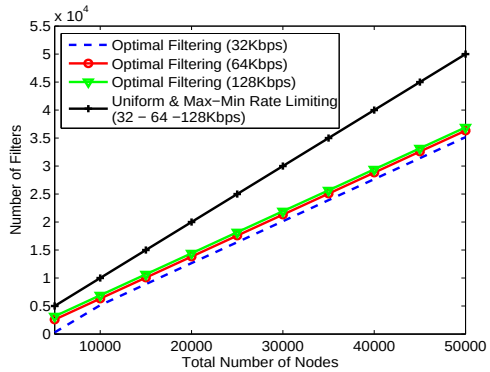
Fig.4 shows the results for $N = 1000$ nodes, which all send at the same rate (10Mbps). We consider all combinations of $x \in \{0, 100\}\%$ and $H \in (0.5, 0.9)$ and we look at the difference in the % of good traffic on the congested link, before and after optimal filtering. The figure shows that there is always improvement, with the best improvement (40%) achieved when 50% of all nodes are attackers, sending at $H = \frac{B}{B+G} = 0.9$.

Then, we also vary the sending rate of each node. We randomly pick 10%, 50% or 90% of the nodes to have 10 times more bandwidth than the rest (i.e. 100Mbps). The reason we look at heterogeneous bandwidths is that a node should be filtered based not only on the ratio $\frac{B}{B+G}$, but also on its total contribution $B+G$ to the capacity of the congested link. Fig.5, shows that optimal filtering significantly helps in this case.

1) Varying the number of attacking nodes: In this section, we increase the number of nodes and we are interested not only in the % of good traffic preserved, but also in the number of filters required. We compare *optimal filtering* to 3 benchmark policies:



(a) % Good Traffic Preserved after Filtering



(b) Number of filters used.

Fig. 6. Performance of Optimal Filtering for the attackers' one tier.

- *Uniform rate limiting*: rate-limit all nodes by $\frac{C}{total\ traffic}$, to make sure the total traffic does not exceed the capacity. Notice, that this policy is equivalent to *no filtering* in terms of percentage of good to overall traffic on the congested link.
- *Random filtering*: randomly place the same number of filters as the optimal policy.
- *Max-min rate limiting*: admit the low-rate nodes first while allocating the same bandwidth to the high rate ones; then distribute the excess capacity fairly among the unsatisfied remaining nodes.

We vary the number of attackers (from 1000 to 50000) and we allocate filters to individual attackers (attackers' one-tier problem)[3]. In Fig6, optimal filtering clearly outperforms the other policies: it preserves more good traffic using less filters. However, the number of filters increases linearly with the number of attackers, which clearly does not scale for a large number of attackers.

To deal with this scalability issue, we solve the one-tier problem at the gateway level. We consider again an increasing number of attackers (from 1000 to 50000), but this time

[3]In this simulation scenario, we vary N, but we make sure that the total good traffic is below the capacity (in particular $\sum_1^N G_i = \frac{C}{2}$), because this is the practical case of adequate provisioning. To construct such an assignment, we assign $\frac{C}{2}$ over half of the nodes assigning $\frac{C}{N}$ to every other node and we randomly pick $N/2$ nodes and assign them bad traffic. We make sure that the total traffic emitted by each node is no more than its maximum rate (32, 64, or 128 kbps)

TABLE I

CODE-RED SCENARIOS

Country	GW	Code Red I		Code Red II	
		% of Good Traffic from [20]	% of Bad Traffic from [16]	% of Good Traffic	% of Bad Traffic
USA	1	36.27	43.9	36.2	45.9
Korea	2	5.8	11.5	0	12
China	3	18.35	10.3	24.1	0
Taiwan	4	2.46	6.1	2.4	16.7
Canada	5	3.64	5.4	3.6	5.4
UK	6	6.74	5.2	6.7	5.3
Germany	7	8.4	5.1	8.4	5.2
Australia	8	2.5	4.3	2.5	1.1
Japan	9	13.91	4.2	14.2	0
Netherlands	10	1.93	4.1	1.9	8.4

attackers are evenly spread behind $n = 1000$ gateways (as in Fig. 1) and we allocate filters to gateways, not to individual attackers. The results are shown in Fig. 7. The optimal policy again outperforms the others: it preserves significantly more good traffic while using much less filters. However, there are several differences from filtering at the attackers' tier, all due to the coarser filtering granularity. First, we need less filters, but the % of preserved traffic drops below 50% in the case of larger number of attackers. Second, the number of filters used by the optimal policy increases fast up to around 90% and then saturates, because otherwise all traffic would be blocked. Third, the max-min policy performs much worse now; also from a practical point of view, the uniform and max-min policies are less attractive, because they use rate-limiters on all nodes, which is unrealistic.

B. Realistic Attack Scenarios

First, we used data from the analysis of two recent worms, Code-Red [16] and Slammer [17] to construct realistic attack distributions as in the single-tier section. Another source of data we used for the attack traffic distribution is Zombie Report [19] published by Prolexic [15]. This report contains the percentage of bots, grouped per country, network, ISP and other meaningful groupings; we use the data referring to the number of infected hosts per country. We assume that if a victim is under attack that traffic would come from ten countries. We consider the ten first countries and assume that they are behind ten different gatewaysThe distribution of attack traffic for the Code-Red, Slammer Zombie scenario is summarized in the last column of Tables I, II and III.

We consider a typical victim – a web-server with 100Mbps access link. We also consider that each country is in a different AS, thus is behind a different gateway; we then use the number of attack sources per gateway, as reported in [16], [17], [19] [4] and shown in the 4^{th} column of Table I. For the legitimate traffic, we use the breakdown of Internet users per country reported in [20] and shown in the 3^{rd} column of Table I, II and III. We consider that both attackers and

[4]In [16] 80% of the total attack comes from 10 countries; we distributed the rest 20% of the attack uniformly across the lower 8 countries.

TABLE II

SLAMMER SCENARIO: ATTACK LAUNCHED BY A POPULATION OF HOSTS INFECTED BY A WORM SIMILAR TO SLAMMER.

Country	GW	% Good Traffic	% Bad Traffic
USA	1	36.3%	44.6%
South Korea	2	5.8%	13.6%
China	3	18.5%	8%
Taiwan	4	2.4%	5.7%
Canada	5	3.6%	4.6%
Australia	6	2.5%	4.2%
UK	7	6.7%	3.8%
Japan	8	13.9%	3.5%
Netherlands	9	1.9 %	3.3%
Unknown	10	8.4%	8.7%
Total		100%	100%

TABLE III

PROLEXIC SCENARIO: ATTACK LAUNCHED BY A BOTS POPULATION, SIMILAR TO THE ONE IN THE PROLEXIC ZOMBIE REPORT.

Country	GW	% Good Traffic	% Bad Traffic
US	1	36.5%	21.5%
China	2	18.5%	14.5%
Germany	3	8.5 %	13.5%
UK	4	6.78%	8.5%
France	5	4.59%	8.5%
Brazil	6	4%	7.5%
Japan	7	13.99%	7.5%
Phillippines	8	1.4%	6.5%
Russia	9	13.94%	6.5%
Malaysia	10	1.8%	5.5%
Total		100%	100%

legitimate users send at the same rate (32kbps, 64kbps or 128kbps), corresponding to upstream dialup/dsl. Therefore, if the total number of legitimate users is N and that of attackers is M, then the amount of good and bad traffic coming from gateway i is $G_i = N\cdot$ (% users behind gateway i) $\cdot$ (rate) and $B_i = M\cdot$ (% users behind gateway i) $\cdot$ (rate). Our rationale is that the number of legitimate users is representative of the legitimate traffic coming from each country. We use the number/percentage of legitimate users, to compute the % of total good traffic generated by each gateway. The result is summarized in the third column in each attack scenario.

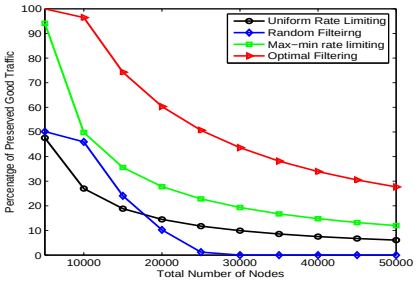
In summary, we construct three realistic attack scenarios using data from [20] as well as from the analysis of code-red worm, slammer worm and the zombie report.

C. Results for Single-Tier

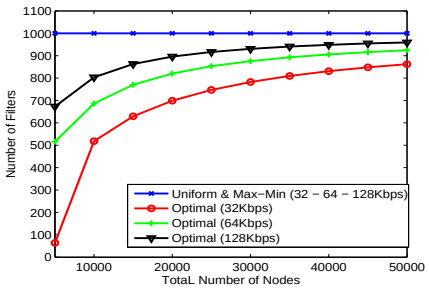
We simulated the Code-Red scenario, (*CodeRed I* – columns 3 and 4 of Table I) for a number of legitimate users N from 1000 to 10000 and attackers M from 1000 to 10000, and we compared the amount of good traffic preserved without any filtering and with optimal filtering. The results are shown in Fig. 8 for 32Kbps sending rate. (The results for 64kbps and 128kbps show similar trends and are omitted here).

Fig. 8(a) shows the % of good traffic preserved. When the total good traffic is less than the capacity of the congested link,[5] and the number of attackers was between 1000 and 2000,

[5]When the good traffic exceeds the capacity, we cannot preserve 100% of it. This can be seen as a combination of a flash-crowd and a DDoS attack. In the rest of the paper, we focus on cases where the good traffic does not exceed the capacity (which is the case with normal operation and good provisioning).



(a) % Good Traffic Preserved after Filtering



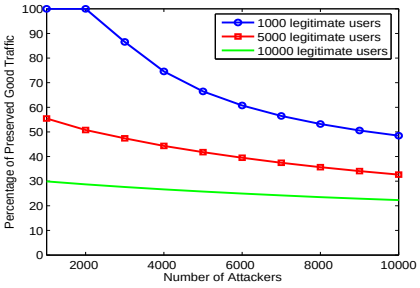
(b) Number of filters used.

Fig. 7. Performance of Optimal Filtering for the gateways' one tier (1000 gateways, same number of attackers behind each gateway).

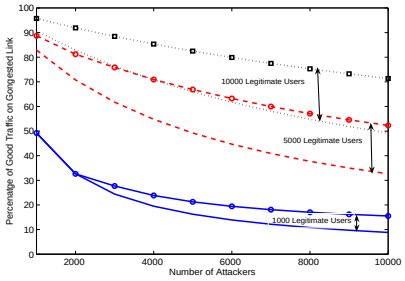
optimal filtering preserves 100% of the good traffic. As the number of attackers increases, the % of good traffic preserved drops; e.g. for 1000 users and 10000 attackers, optimal filtering preserves 55% of the good traffic. This is because filtering at the gateway level is based on destination address and domain source address; better results could be achieved if a finer granularity of filtering could be applied (i.e. source address of individual attackers), as in the multi-tier case later.

Fig. 8(b) shows the % of the capacity of the congested link that consists of good traffic. When the number of attackers is comparable to the number of legitimate users (e.g. 10000 attackers and 10000 users), we observe that the optimal filtering preserves 25% more good traffic, which increases the percentage of good traffic on the congested link from 50% to 75%. However, at the extremes where the number of legitimate users is much smaller (e.g. 1000 users and 10000 attackers) or much larger (e.g. 10000 users and 1000 attackers) than the number of attackers then the optimal filtering increases the preserved good traffic only marginally (10%).

The improvement achieved above was moderate, because all gateways had both good and bad traffic. If there are gateways that carry only good or bad traffic, then filtering would be able to better separate good from bad traffic and further improve performance. To demonstrate this, we modify the distributions of good and bad traffic per gateway, as shown in scenario *Code Red II* – 5^{th} and 6^{th} column of Table I. This modification maps to real life scenarios in which a certain website has only customers in some, but not all, countries (ASes). Also it is reasonable to assume that the attacker will not be able to compromise hosts in ASes that span all countries, thus there are some gateways with only bad or good traffic.



(a) % Good Traffic Preserved after Optimal Filtering



(b) Good % Link BW before/after Optimal Filtering.

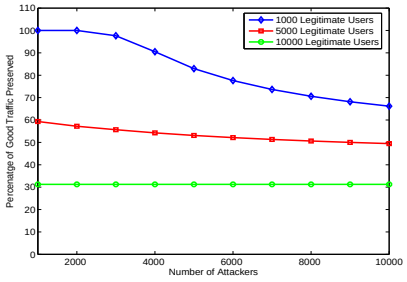
Fig. 8. Performance of Optimal Filtering for scenario Code Red I.

We used the same simulation setup as for the *Code-Red I* scenario and the results are shown in Fig. 9. The trends are similar but the improvement is more substantial: the capacity of the congested link used by good traffic improves up to 50%. In the case of 10000 legitimate users, optimal filtering allows only good traffic through the congested link until the capacity is used. The same behavior can be observed for 5000 legitimate users with 64Kbps and 128Kbps rates.

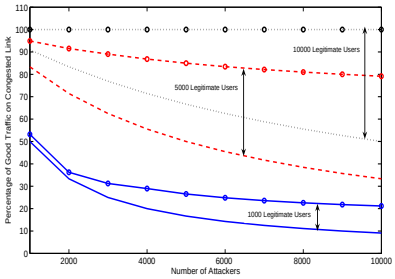
D. Results For Two-Tiers

Figures 10, 11, and 12 show the performance of optimal two-tier filtering for the Code-Red scenario, Slammer and Zombie scenario respectively. In all three cases, we increase the number of attackersand we look at how well filtering can handle the increasing attack traffic. The performance metrics of interest are (a) the % goodput preserved after filtering and (b) the number of filters used in the process. As a baseline for comparison, we also show the performance of the optimal single-tier filtering at gateway and attack level.

In all three figures, the optimal solution performs similarly, although they are based on different distributions of good and bad traffic. As expected, filtering at attackers' level (plain red line) gives the upper bound for the preserved goodput. Indeed, one can preserved 100 % of the good traffic by filtering out each individual attacker (assuming there are no hosts that produce both good and bad traffic) but requires as many filters as the number of attackers, which is impractical. Filtering at the gateway level (shown in dashed green line) provides a lower bound to the preserved goodput (because it filters out together both good and bad traffic behind the same gateway) but uses a small number of filters. Two-tier filtering lies in the middle (blue curves): it provides a graceful degradation of

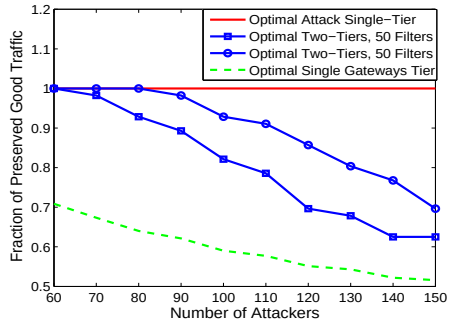


(a) % Good Traffic Preserved after Optimal Filtering

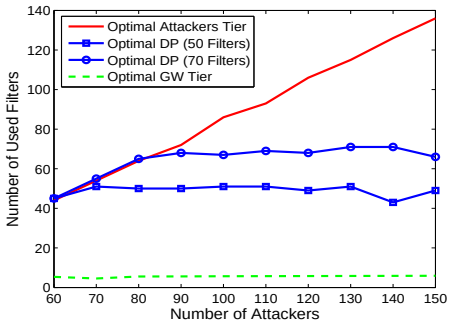


(b) Good % Link BW before/after Optimal Filtering.

Fig. 9. Performance of Optimal Filtering for scenario Code Red II.

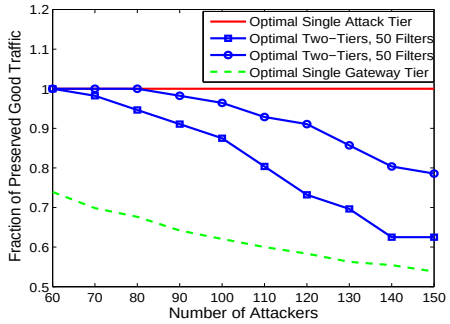


(a) % Good Traffic Preserved

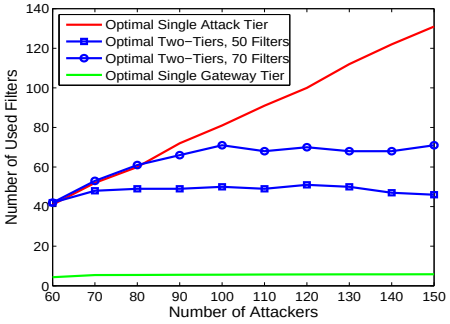


(b) Number of filters used.

Fig. 10. Performance of Optimal Two-tier Filtering for the CodeRed scenario.

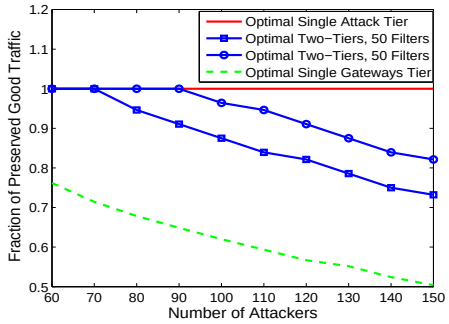


(a) % Good Traffic Preserved

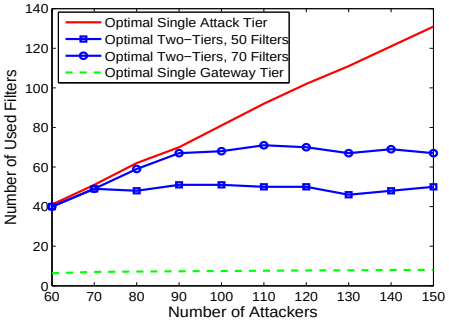


(b) Number of filters used.

Fig. 11. Performance of Optimal Two-tier Filtering for the Slammer scenario.



(a) % Good Traffic Preserved



(b) Number of filters used.

Fig. 12. Performance of Optimal Two-tier Filtering for the Zombie scenario.

preserved goodput using a small number of filters. The larger the number of filters available to multi-tier filtering, the closer the preserved goodput to the upper bound.

V. Conclusion

In this paper, we studied the optimal allocation of filters against DDoS attacks. We formulated the single-tier allocation as a knapsack problem and the two-tier problem using dynamic programming. We simulated the optimal solution using realistic attack scenarios and showed that optimal filtering can significantly improve the trade-off between preserved good traffic and number of filters. We are currently working on several issues including efficient heuristics to achieve near-optimal performance at lower complexity, filter allocation under uncertainty in the identification of attackers, and allocation across many routers and victims.

Acknowledgment

We would like to thank the Center for Pervasive Communications and Computing (CPCC) and the Emulex Fellowship from Calit2 for sponsoring K. El Defrawy at U.C. Irvine.

References

[1] "Blue Security folds under spammer's wrath", *http://www.securityfocus.com/news/11392*.
[2] J. Mirkovic and P.Reiher, "A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms", *ACM CCR Vol.34(2)*, 2004.
[3] A.Yaar, A. Perrig and D.Song, "A Path Identification Mechanism to Defend against DDoS Attacks", *in Proc. of IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 2003.
[4] P. Furgeson and D. Denie, "Network Ingress Filtering: Defeating DoS Attacks that Employ IP Source address spooofing", *RFC 2827*, Jan. 1998.
[5] S. Savage, D. Wetherall, A. Karlin, T. Anderson, "Practical Network Support for IP Traceback", *in ACM SIGCOMM Computer Communication Review*, Vol. 30(4), pp. 295 - 306, Oct. 2000.
[6] A. Snoeren, , "Hash-Based IP Traceback", *in Proc. ACM SIGCOMM*, Aug. 2001, San Diego, CA.
[7] Michael T. Goodrich, "Efficient Packet Marking for Large-Scale IP Traceback", *in Proc. of CCS'02*, pp.117-126.
[8] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson and S. Schenker, "Controlling High Bandwidth Aggregates in the Network", *in ACM Computer Comm. Review*, Vol. 32(3), pp. 62-73, July 2002.
[9] K. Argyraki and D.R.Cheriton, "Active Internet Traffic Filtering: Real-time Response to Denial-of-service Attacks", *in Proc. of USENIX Security 2005*.
[10] T.Anderson, T.Roscoe, D.Wetherall, "Preventing Internet Denial-of-Service attacks with Capabilities", *Internet Denial of Service*, 2003.
[11] X.Yang, D.Wetherall, T.Anderson, "A DoS-limiting Architecture", *in Proc. ACM SIGCOMM*, Aug. 2005, Philadelphia, PA, USA.
[12] X. Liu, X. Yang, D. Wetherall, T. Anderson, "Efficient and Secure Source Authentication with Packet Passports", *in Proc. of USENIX SRUTI*, San Jose, CA 2006.
[13] A. Keromytis, V. Misra, D. Rubenstein, "SoS: Secure Overlay Services", *in Proc. of ACM SIGCOMM'02*, pp. 61-72, Aug. 2002, Pittsburgh, PA.
[14] D.Andersen, "Mayday: Distributed Filtering for Internet Services", *in Proc. 4th USENIX Symposium (USITS)*, Seattle, WA, Mar. 2003.
[15] Prolexic Technologies, *http://www.prolexic.com*
[16] D. Moore, C. Shannon, J. Brown, "Code-Red: a case study on the spread and victims of an Internet worm", *in Proc. ACM IMW 2002*.
[17] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford and N. Weaver, "Inside the Slammer Worm", *in Proc. of IEEE Security and Privacy*, 1(4):33-39, July 2003.
[18] S. Staniford, D. Moore, V. Paxson and N. Weaver, "The Top Speed of Flash Worms," *in Proc. ACM CCS WORM*, Oct. 2004.
[19] "The Prolexic zombie report", *http://www.prolexic.com/zr/*, Q1-Q2 2005.
[20] "Internet World Stats", *http://internetworldstats.com/list2.htm*.
[21] H. Kellerer, U. Pferschy, D.Pisinger, "Knapsack Problems", *Springer* 2004.