Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 176 (2007) 1283–1292
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

O.R. Applications

A packet filter placement problem with application to defense against spoofed denial of service attacks

Benjamin Armbruster [a], J. Cole Smith [b,*], Kihong Park [c]

[a] Department of Mathematics, The University of Arizona, Tucson, AZ 85721, United States
[b] Department of Systems and Industrial Engineering, The University of Arizona, Tucson, AZ 85721, United States
[c] Department of Computer Sciences, Purdue University, West Lafayette, IN 47907, United States

Received 4 November 2004; accepted 23 September 2005
Available online 27 December 2005

Abstract

We analyze a problem in computer network security, wherein packet filters are deployed to defend a network against spoofed denial of service attacks. Information on the Internet is transmitted by the exchange of IP packets, which must declare their origin and destination addresses. A route-based packet filter verifies whether the purported origin of a packet is correct with respect to the current route map. We examine the optimization problem of finding a minimum cardinality set of nodes to filter in the network such that no spoofed packet can reach its destination. We prove that this problem is NP-hard, and derive properties that explicitly relate the filter placement problem to the vertex cover problem. We identify topologies and routing policies for which a polynomial-time solution to the minimum filter placement problem exists, and prove that under certain routing conditions a greedy heuristic for the filter placement problem yields an optimal solution.

Keywords: Combinatorial optimization; Internet; Route-based packet filtering; Spoofed denial of service attack; Vertex cover

1. Introduction

Given the vulnerability of communication networks to a wide array of security attacks, a number of network security measures have been proposed to counter these attacks [1,4,10,18]. One of the pressing problems facing the global Internet is distributed denial of service (DDoS) attacks, wherein a set of compromised hosts

* Corresponding author. Tel.: +1 352 392 1464x2020; fax: +1 352 392 3537.
E-mail addresses: barmbrus@stanford.edu (B. Armbruster), cole@sie.arizona.edu, cole@ise.ufl.edu (J.C. Smith), park@cs.purdue.edu (K. Park).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2005.09.031

concurrently send large amounts of traffic targeted at a server, gateway, or network [3,8]. The aim of the attack is to disrupt normal operation of the targeted network system by depleting its resources. Many DDoS attacks disguise their true origin by inscribing bogus information in the source address field of the IP (Internet Protocol) packet header, referred to as IP source address spoofing. This causes recovery to take on the order of hours and days, at which point damage has already been done. IP traceback—the problem of locating the attack source—has been an active area of research [15,17].

Whereas most DDoS defenses are reactive in nature, a proactive approach called route-based distributed packet filtering [16] is aimed at preventing spoofed DDoS packets from reaching their targets in the first place. Route-based filtering uses route constraints in transportation networks to determine whether a packet, given its source and destination address, is misrepresenting its true origin. In semi-maximal route-based filtering, only the source address is utilized to affect filtering, which enables the linear filter table size required for implementation in resource-bounded routers. Distributed route-based packet filtering applies this action at select transit nodes in the network so that with a small deployment at "checkpoints," effective discarding of spoofed packets is achieved. Although the scope of scenarios to which this filtering concept applies is broader than Internet DDoS attacks—e.g., distributed intrusion detection and sensor networks in physical transportation systems—DDoS network security is the focus area of this paper.

Consider a communication network whose connectivity is represented as a graph. The fields of a packet header, in particular, its source and destination addresses, are inscribed by the originating node. Node o can "attack" another node d by forging the source address of a sequence of packets as node s and sending them to d. The role of route-based packet filters is to identify and remove packets from the network whose source addresses can be ascertained to be spoofed, before they can aggregate and impart harm at their target. They must do so without violating the requirement of *safety*: a packet whose source address is not spoofed must not be discarded. We study the following optimal filter placement problem: Given a network and its routing, find a minimum cardinality set of nodes where route-based filters are placed such that no packet with a falsely reported origin is permitted to reach its destination.

The feasibility criterion is called *perfect security*, a special case of more relaxed security measures studied in [16] where certain triplets (o, s, d) are allowed through. The optimization criterion examined in this paper, minimizing the size of the filter node subset, is not the only meaningful criterion. For example, one may consider the number of edges as the cost function, or a notion of processing overhead that varies depending on the traffic load a node encounters. In the global inter-domain Internet context where a single node represents an entire domain, deployment of cooperative security solutions is hindered by policy barriers across different administrative boundaries. Bilateral agreements require significant effort to establish, and multilateral agreements are that much harder to come by. This provides a practical motivation, from a deployment perspective, for considering the number of filter nodes as the optimization criterion. The "processing overhead" of a node with many edges—likely a large ISP (Internet Service Provider) that provides transit service to other domains—is not localized to a single physical router since the ISP will have entry/exit/switching stations, called POPs (points-of-presence), at major cities where it interfaces with other domains. Route-based filters would need to be installed only on border routers that connect to other domains. In the intra-domain context, high-degree nodes correspond to routers with many links, and processing overhead is amplified proportionally to the number of edges.

We assume that route-based filters have access to routing information to determine if a packet with source address s destined to d is spoofed, subject to safety. In the global inter-domain Internet, route asymmetry—the path from o to d is not the same as from d to o—is common, which makes maintaining accurate route-based filter tables a nontrivial challenge. The focus of this paper is on studying an optimal filter placement problem

assuming reasonably accurate route-based filter tables are available.

Our theoretical results complement the experimental results in [16] but are not directly comparable. This is so since the first part of the paper discusses general hardness of achieving perfect security, and the second part discusses polynomially solvable special cases of potential relevance to large-scale communication networks. The special cases are candidate building blocks of these empirical graphs but do not make up the whole graph. Decomposition of large-scale empirical networks into constituent components is needed to relate the optimum filter size of the building blocks to the optimum filter placement of the whole graph.

The remainder of the paper is organized as follows. In Section 2, we formalize the optimization problem and prove that it is strongly NP-hard. In Section 3, we introduce an integer programming framework for solving the general minimum filter placement problem, along with a characterization of feasible solutions for a restricted version of the problem that admits a decomposition algorithm. In Section 4, we focus our attention on optimal polynomial time algorithms for a special class of problems inspired by large-scale communication networks, such as the global inter-domain Internet [7]. We conclude with a summary of our results and a discussion of future research directions.

2. Problem description and NP-completeness

2.1. Network model

The network is given by a directed graph $G(N, A)$, where N is the set of nodes in the network, $|N| \geqslant 3$, and A is the arc set. Define the *communication set* as the set of all node pairs that engage in the exchange of packets. Formally, the communication set is given by $C \subseteq N \times N - \{\cup_{i \in N}(i,i)\}$ such that $(u,v) \in C$ if and only if packets at $u \in N$ can travel to $v \in N$ under a given routing policy R.

A set $R(u,v)$ is defined for every $(u,v) \in C$ that consists of all permissible paths that a packet may use from u to v. We allow $|R(u,v)| \geqslant 1$ for $(u,v) \in C$ to account for multi-path routing. The paths obey a *destination-based* routing scheme, wherein intermediate nodes in a path forward packets based only on the packet's destination address. A path $p \in R(u,v)$ of length k is given by a sequence of nodes $(p_0, p_1, \ldots, p_k)$ where $p_0 = u$ and $p_k = v$. We denote $R^{\mathrm{ALL}} = \cup_{(u,v)\in C} R(u,v)$. We define *nodes*$(p)$ as the set of nodes contained in path p, and *arcs*(p) as the set of edges in path p. Without loss of generality, we assume that A is *irreducible*, that is, for all $(i,j) \in A$, we have $(i,j) \in arcs(p)$ for some $p \in R^{\mathrm{ALL}}$.

Next, we define an abstract description of the network filtering system, including the filters and packets that traverse the network. Filters are allowed to be placed on nodes throughout the network. Each packet is associated with its true origin, its purported origin, and its destination. A packet filter has access to the purported origin and destination addresses of the packet, plus the arc on which the packet arrived at the filter. We consider both maximal and semi-maximal filters. A *maximal* route-based filter is defined as

$$FM(o,a,d) = \begin{cases} 0, & \text{if } a \in arcs(p) \text{ for some } p \in R(o,d); \\ 1, & \text{otherwise.} \end{cases} \tag{1}$$

That is, $FM(o,a,d)$ returns a 0 if a packet traveling from o to d could possibly use arc a under routing R, and 1 otherwise. A drawback of the maximal filter is that an $O(|N|^2)$ table must be used, which is prohibitively large for large-scale networks. As an alternative, a weaker filter function returns 0 if node o could use arc a in *any* of its routings. Formally, a *semi-maximal* is defined as

$$FS(o,a) = \begin{cases} 0, & \exists d \in N \text{ such that } a \in arcs(p) \\ & \quad \text{for some } p \in R(o,d); \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

If either filter function returns a 1, then it can be ascertained that the packet's source address has been forged, and the packet is discarded. Note that with the use of either filter mechanism, no packet with a correct address will be dropped: safety is

assured. In general, maximal filters are more powerful than semi-maximal filters, i.e., $FS(o,a) \leqslant FM(o,a,d)$ for all $(o,d) \in C$ and $a \in A$. This relationship is strict for $(o,d) \in C$ and $a \in A$ when a path $p \in R(o,b)$ exists for some $(o,b) \in C$ with $a \in arcs(p)$, but no such path exists in $R(o,d)$.

We make the following additional assumption on the network system. Regardless of the filter type being used, no packet at a *filter node*—a node where filtering is carried out—may forge its origin address. This mechanism, called egress filtering, is technically easy to carry out. A node that goes through the trouble of deploying route-based filters is assumed to disallow forged packets to emanate from its own domain. Second, a packet with a spoofed origin–destination pair $(o,d) \notin C$ will automatically be dropped at node d, irrespective of the presence of a filter at that node. This mechanism is referred to as a "trivial filter."

2.2. Optimal filter placement

The *minimum filter placement problem* (MFPP) determines the minimum number of filters required to achieve *perfect security*, i.e., all packets with forged origin addresses are discarded. We first prove that this problem is strongly NP-hard, regardless of whether maximal or semi-maximal filters are used. We define the corresponding decision problem:

FILTER: Given a connected network $G(N,A)$ with $|N| \geqslant 3$, a communication set C, a set of routes R^{ALL}, and an integer $1 \leqslant k \leqslant |N|$, does there exist a (maximal or semi-maximal) filter placement $H \subseteq N$, $|H| \leqslant k$, that achieves perfect security?

Without loss of generality (due to the use of destination-based routing), we assume that R^{ALL} is polynomially bounded in $n = |N|$ so that the problem size is polynomial in n.

Proposition 1. FILTER *is strongly NP-complete.*

Proof. First, we show that FILTER is in NP. For every pair of nodes $(o,d) \in C$, $o \notin H$, a check can be made for every possible value of $s \in N - \{o\}$ such that $(s,d) \in C$, to verify that a packet with true origin o, purported origin s, and destination d will be filtered by some node in H. (If $(s,d) \notin C$, the packet will automatically be dropped by the trivial filter rule.) This takes polynomial time.

Next, we show that FILTER is NP-hard by reducing VERTEX COVER [11] to FILTER. An instance of VERTEX COVER (VC) is specified as follows: Given an undirected, connected graph $G(V,E)$ with $|V| \geqslant 3$, and an integer $1 \leqslant c \leqslant |V|$, does there exist a subset S of c or fewer nodes such that if $(u,v) \in E$, then at least one of u and v belongs to S?

We transform an arbitrary instance of VC to FILTER as follows. The graph G is transcribed with $N = V$ and arcs (u,v) and (v,u) in A if and only if $(u,v) \in E$. We set k equal to c, and set the communication set C equal to A, that is, communication only takes place between adjacent nodes. Finally, routing R only involves single-arc routes that connect adjacent nodes.

Suppose the VC instance is a yes-instance and is verified by the solution S. Consider the selection of filters $H = S$. For every route $(u,v) \in R^{ALL}$, either $u \in H$ or $v \in H$, or both. If $u \in H$, a packet starting at u must have a valid origin address. If $u \notin H$, then $v \in H$. Since any incoming packet on the arc (u,v) must be from u, the filter at node v (whether maximal or semi-maximal) would drop any packet on this arc without source address u. Since this is true for all routes $(u,v) \in R^{ALL}$ and $|H| \leqslant k$, FILTER also yields a yes-instance.

Suppose VC is a no-instance. This implies that in the FILTER instance, for any set H containing k (or fewer) nodes, there will exist some edge $(\hat{u},\hat{v}) \in A$ such that $\hat{u} \notin H$ and $\hat{v} \notin H$, and $\hat{v}$ is adjacent to at least two nodes by connectedness and $|V| \geqslant 3$. Let b designate a node adjacent to $\hat{v}$ other than $\hat{u}$. Then a packet from $\hat{u}$ could forge its address as b and send to $\hat{v}$ without being discarded by a filter, violating perfect security. Since a solution exists to the VC instance if and only if there exists a solution to the transformed FILTER instance, we have that FILTER is NP-complete. Moreover, since no numerical data was used in the transformation, we have that FILTER is NP-complete in the strong sense. This completes the proof. □

Remark 2. Observe that the reduction in the proof of Proposition 1 does not explicitly consider the case in which C is *complete*, that is, $(u,v) \in C$ for all $u,v \in N$ and $u \neq v$. However, the foregoing transformation can be modified to demonstrate that FILTER remains strongly NP-complete when C is complete. The reduction from VC to FILTER adds one additional dummy node w, $N = V \cup \{w\}$. The arc set A is created as before, with the addition of arcs (u,w) and (w,u) $\forall\, u \in V$. Routing between two distinct nodes $u,v \in N$ remains unique (i.e., $|R(u,v)| = 1$). If $(u,v) \in A$, $R(u,v)$ consists of the single-arc path (u,v). Otherwise, the dummy node w is used as an intermediate node, and $R(u,v)$ consists of the two-arc path (u,w,v). We set k equal to $c+1$. It is easy to show that node w must be a filter node, and therefore by the same logic as given in the proof of Proposition 1, the MFPP is strongly NP-hard for arbitrary communication sets.

Remark 3. A filter deployment that is VC does not, in general, imply perfect security. For example, in a 5-node chain network (u,v,w,x,y) where v and x are filter nodes, w can disguise itself as u or v when sending to y. The maximal or semi-maximal filter at x is unable to distinguish such spoofed packets. The effectiveness of VC as a filter placement strategy depends on the underlying graph connectivity.

3. Solution strategies and special cases

We begin this section by providing an integer programming formulation for solving MFPP. Then we examine special cases of practical relevance and provide solutions that exploit the special structure of these problems.

3.1. Integer programming model

We formulate MFPP as a 0/1 integer programming problem. Consider a communication pair $(o,d) \in C$, along with a forgeable source address $s \in N$ such that $(s,d) \in C$. Define N_{sod} as the set of nodes i such that a filter at node i would drop any packet originating from o with destination d and spoofed source address s. By the egress filtering assumption, $o \in N_{\text{sod}}$. The following set covering problem solves MFPP:

$$\text{minimize} \quad \sum_{i \in N} y_i \tag{3a}$$

$$\text{s.t.} \quad \sum_{i \in N_{\text{sod}}} y_i \geqslant 1 \quad \forall (o,d) \in C,\ \forall s \neq o : (s,d) \in C \tag{3b}$$

$$y_i \in \{0,1\} \quad \forall i \in N. \tag{3c}$$

The objective function (3a) minimizes the number of filters placed in the network, the constraints in (3b) eliminate the possibility of any node o succeeding with a forgeable address s when sending to destination node d, and (3c) imposes an integer restriction on the filter selection variables y. The next lemma gives a necessary and sufficient condition for achieving perfect security.

Lemma 4. *A filter placement* $y = (y_0, \ldots, y_{n-1})$ *achieves perfect security if and only if* $\forall (o,d) \in C$ *and* $(p_0, p_1, \ldots, p_k) \in R(o,d)$, $k \geqslant 1$,

(i) *when* $k = 1$ *and* $(u,d) \in C$ *for some* $u \neq o \in N$, *we have that* $y_o + y_d \geqslant 1$, *and*
(ii) *when* $k > 1$, $y_{p_i} = 1$ *for all* $0 < i < k$.

Proof. We start with the "only if" claim. Part (i) was established in the proof of Proposition 1. For part (ii), suppose that for some $0 < i < k$, $y_i = 0$. By Remark 3, node p_i can attack node d by spoofing any address $s \in \{p_0, \ldots, p_{i-1}\}$ such that $(s,d) \in C$ (including $s = p_0$) which violates perfect security. The "if" claim in the lemma clearly follows from the foregoing analysis and the egress filtering property (i.e., a filter node cannot initiate an attack). □

3.2. Complete communication scenarios

Note that (3) is a general set covering formulation that may be tackled by integer programming strategies [2,5,6,9]. Instead of addressing the general problem, we restrict our attention to problems in which C is complete (i.e., all distinct node pairs

of N are present in C). Define an *endpoint* as a node that never serves as an intermediate point in routing a packet. That is, $i \in N$ is an endpoint if $i \in nodes(p)$ for some $p \in R(u,v)$, $(u,v) \in C$, implies that $i = u$ or $i = v$. The following propositions capture necessary conditions for feasible filter placement to achieve perfect security.

Proposition 5. *A solution to the MFPP with complete C must also be a vertex cover of the corresponding undirected graph.*

Proof. The proposition follows from Lemma 4, completeness, and irreducibility. □

Proposition 6. *For any solution y to MFPP, if $w \in N$ is not an endpoint then $y_w = 1$.*

Proof. If node w is not an endpoint, then there must exist a communication pair $(u,v) \in C$ and a path $p \in R(u,v)$ given by nodes $(u, p_1, \ldots, p_{k-1}, v)$ where $w = p_t$ for some $1 \leqslant t \leqslant k-1$. By Lemma 4, node w must be a filter node. □

An algorithm for MFPP could begin by determining all non-endpoint nodes by considering each path in R^{ALL}. One can then construct an *endpoint network* $\widetilde{G}(\widetilde{N}, \widetilde{E})$ where $\widetilde{N}$ consists entirely of endpoints and $(i,j) \in \widetilde{E}$ if and only if either $(i,j) \in R(i,j)$ or $(j,i) \in R(j,i)$. Let minimum vertex cover problem (MVCP) be the optimization variant of Vertex Cover. The following proposition provides an algorithm to solve the MFPP, and indicates that for the complete communication set case, no advantages are afforded by utilizing maximal filters versus semi-maximal filters.

Proposition 7. *Let D be the set of all non-endpoint nodes in G, and consider a solution S to MVCP over the endpoint network $\widetilde{G}$. Then filtering nodes $H = D \cup S$ results in an optimal solution to the MFPP, regardless of whether maximal or semi-maximal filters are used.*

Proof. First, let us verify that H is a feasible solution to MFPP on G, even in the presence of semi-maximal filters. Consider a pair of nodes $(u,v) \in C$ and a path $p \in R(u,v)$ given by $(u, p_1, \ldots, p_{k-1}, v)$. If u is an endpoint, then p_1 knows that the set of possible addresses emanating from u is confined to u and destination addresses are superfluous. If u is a non-endpoint, then $u \in D$, and by egress filtering, no attack can emanate from u. Only filters deployed at arcs connecting endpoints are relevant.

Next, we verify that H is a minimum cardinality feasible solution. By Proposition 6, all nodes in D must be filter nodes in any feasible solution. Now suppose by contradiction that a smaller cardinality feasible solution $\widehat{H}$ existed, with $\widehat{H} = D \cup \widehat{S}$. Since $|\widehat{S}| < |S|$, then $\widehat{S}$ must not be a vertex cover with respect to $\widetilde{G}$. The solution $\widehat{H}$ is not feasible by Proposition 5 which leads to a contradiction. □

The proof of Proposition 7 gives an optimal algorithm for solving MFPP. First, place filters on all non-endpoints. Next, create an endpoint graph $\widetilde{G}$, and solve the MVCP on it. Typically, $\widetilde{G}$ will not be connected, implying that several smaller vertex cover problems must be independently solved on the connected components of $\widetilde{G}$. (This is not always so, as indicated by Remark 2.) This decomposition requires the solution of several smaller problems rather than one large problem and allows for parallelism at the most time-consuming step of the algorithm. From a theoretical viewpoint, it also describes the precise relationship between the vertex cover and filter placement problems and leverages known (in)approximability properties of VC. They include the well-known 2-factor approximation scheme [13,14], and, on the negative side, the 7/6-factor lower bound which holds unless P = NP.

Remark 8. The foregoing analysis assumes that the objective of the MFPP is to minimize the total number of deployed filters, rather than weighting filter deployments by overhead required to carry out their filtering operations. While this assumption is made for the sake of convenience in exposition, it is straightforward to extend this analysis to the case in which our objective is to minimize a weighted number of filter deployments. The decomposition algorithm remains the same, except that we would now solve a weighted minimum vertex cover problem over each endpoint network remaining after filters are assigned to the non-endpoints.

3.3. Polynomially solvable cases

One further advantage of the algorithm suggested by Proposition 7 is that it identifies a class of filter placement problems that are polynomially solvable. For example, problems with complete communication sets that yield no vertex cover problem phase, or yield a set of vertex cover problems that are themselves polynomially solvable or are bounded by some constant, are solvable in polynomial time via this algorithm. We list specific network topologies below.

3.3.1. Tree networks

Suppose the network G is given by a tree with bidirectional edges. There is a unique path between every pair of nodes and the endpoints of the network consist of the leaf nodes in G. Filtering all other nodes yields an endpoint graph containing no edges ($|N| \geqslant 3$). Hence, the optimal solution is to filter all nodes with degree at least two. Note that a star topology is a special case of a tree network with a single transit (i.e., non-endpoint) node where a single filter node suffices to achieve perfect protection.

3.3.2. Cycle (self healing ring) networks

Suppose that $G(N, A)$ is given by $N = \{1, 2, \ldots, n\}$, with A consisting of bidirectional edges $(i, i+1)$ for $i = 1, \ldots, n-1$ and $(n, 1)$. For each communication pair $(u, v) \in C$, $u < v$, there exist two paths in $R(u, v)$: a clockwise path $p^+_{uv} = (u, u+1, \ldots, v)$, and a counterclockwise path $p^-_{uv} = (u, u-1, \ldots, 1, n, n-1, \ldots, v)$. No node is an endpoint in this scenario, and thus all nodes must be filter nodes.

3.3.3. Bipartite networks

Consider a graph $G(N, A)$, where $N = N_1 \cup N_2$ and $N_1 \cap N_2 = \emptyset$, and each arc (i, j) is bidirectional with either $i \in N_1$ and $j \in N_2$, or vice versa. Regardless of the routing scheme, after preprocessing by filtering non-endpoints, the remaining problem seeks to find a minimum vertex cover on a bipartite network. This problem is polynomially solvable.

From the aforementioned networks, one may compose more complex networks such as a ring of trees (or stars)—a common configuration in metropolitan area networks where the ring serves as a backbone and the tree serves as an access network—and peering connections between two tier-1 Internet Service Providers (ISPs) whose connectivity resembles that of a bipartite graph.

4. Special cases arising in large-scale communication

Most large-scale communication topologies consist of a set of *client* nodes, which connect to *router* nodes that serve as hubs. The routing nodes communicate with one another via some mesh connectivity providing survivability in the event of equipment failure. In this section we analyze the MFPP on such topologies, describing cases in which the decomposition approach given in the previous section executes in polynomial time and showing a case in which a greedy vertex cover heuristic solves the MFPP to optimality.

Consider a topology described by a set of client nodes N_c and router nodes N_r ($N = N_c \cup N_r$, and $N_c \cap N_r = \emptyset$). Each arc is bidirectional, and either links a client to a router, or links two routers. We call such topologies *semi-bipartite*. This structure is inspired by a typical communication topology (either small or large scale), in which each client node is connected to the larger network by at least one hub node. All routing from one client node to another goes from the client node to one of its hub nodes, through a (usually brief) series of hub node links, and finally to its destination. In our framework client nodes do not connect to one another directly.

The resulting topology is not a bipartite graph, since there exist arcs whose endpoints both lie within N_r. However, the properties of bipartite graphs mentioned in Section 3.3 play a role in permitting problems of this nature to be solved to optimality.

Proposition 9. *Consider an MFPP instance with complete routing on a semi-bipartite graph $G(N, A)$, where N is partitioned into N_c and N_r, $|N_c| \geqslant |N_r|$. Define $\widehat{A} = \{(i, j) \in A : i \in N_c \text{ or } j \in N_c\}$. If there exists a matching on the network $\hat{G}(N, \widehat{A})$ of size N_r, then an optimal solution H exists in which $H = N_r$.*

Proof. If a matching of cardinality $|N_r|$ exists to $\widehat{G}$, then the minimum cardinality vertex cover on $\widehat{G}$ has $|N_r|$ nodes. Clearly, filtering the set of nodes N_r is one such vertex cover to both $\widehat{G}$ and G. Also, since all non-endpoints lie in N_r, the necessary conditions given by Propositions 5 and 6 hold true, and $H = N_r$ is an optimal solution. This completes the proof. □

It is common for the conditions of Proposition 9 to hold true in large-scale communication networks, since $|N_r|$ is normally much smaller than $|N_c|$, and the degree of nodes in N_r is generally much larger than that of the nodes in N_c. In fact, either of the following two conditions are sufficient for a matching of cardinality $|N_r|$ to exist in $\widehat{G}$.

Condition 1. For each $(u,v) \in C$ with $u \in N_c$, there commonly exists a primary path and a set of backup paths. Suppose that for every $i \in N_r$, there exists a primary path $p \in R(u,v)$ for some $(u,v) \in C$, $u \in N_c$, such that arc (u,i) is the first arc in p. (A "primary" path may serve as the path taken during normal system operation, for instance.) Clearly, this would satisfy the matching condition of Proposition 9. If in addition we assume complete routing, then the hypotheses of Proposition 9 are fulfilled.

Condition 2. Define $d_r^{\min}$ to be the minimum degree of a node in N_r on the bipartite graph $\widehat{G}$, and define $d_c^{\max}$ to be the maximum degree of any node in N_c on $\widehat{G}$ (or G). If $d_r^{\min} \geqslant d_c^{\max}$, then Hall's Theorem [12] can be used to guarantee a matching of cardinality $|N_r|$. This condition is satisfied in many practical communication networks, since the degree of hub nodes is typically much larger than the degree of client nodes. Hall's Theorem guarantees a matching of size $|N_r|$ if and only if for every subset $B \subseteq N_r$, we have that $|\mathcal{N}(B)| \geqslant |B|$, where $\mathcal{N}(B)$ represents the set of nodes $\{k \in N_c: (i,k) \in A \text{ for some } i \in B\}$. To see that the conditions of Hall's Theorem hold under Condition 2, suppose by contradiction that a $B \subseteq N_r$ existed such that $|\mathcal{N}(B)| < |B|$. Then since $\mathcal{N}(B) \subseteq N_c$, the total number of edges emanating from $\mathcal{N}(B)$ is no more than $d_c^{\max}|\mathcal{N}(B)|$, and since $B \subseteq N_r$, the total number of edges emanating from B in graph $\widehat{G}$ is no fewer than $d_r^{\min}|B|$. Since $|\mathcal{N}(B)| < |B|$ and $d_c^{\max} \leqslant d_r^{\min}$, we have that $d_c^{\max}|\mathcal{N}(B)| < d_r^{\min}|B|$. However, since edges incident to nodes in B are connected only to nodes in $\mathcal{N}(B)$, while edges incident to nodes in $\mathcal{N}(B)$ may connect to nodes other than those in B, we must have that $d_c^{\max}|\mathcal{N}(B)| \geqslant d_r^{\min}|B|$. This contradicts the assumption that $|\mathcal{N}(B)| < |B|$ and $d_c^{\max} \leqslant d_r^{\min}$, and hence, a matching must exist.

Remark 10. Proposition 9 addresses the MFPP with complete routing on a semi-bipartite graph when there exists a matching of size N_r. However, the general problem on such graphs in which no guarantee exists on the size of the matching is strongly NP-hard. While a proof of this claim is omitted for the sake of brevity, the required transformation copies the nodes and edges of a VC instance onto a subgraph, and adds a dummy node w that is adjacent to each of the transcribed nodes. The nodes in this subgraph form the set of routing nodes, with connectivity as described above. The cardinality of the client node set equals the number of nodes in the routing set, and each client node is connected to node w. Routing between each pair of nodes is done by a single-link path if possible, and otherwise by a two-link path via node w. After putting a filter on node w as required by Lemma 4, the problem now reduces to a minimum vertex cover problem on the graph transcribed from the VC instance.

One means of estimating the solution of the MFPP has been by executing the following well-known greedy vertex cover heuristic on the graph G.

4.1. Greedy vertex cover heuristic (GVCH)

For each $i \in N$, initialize $\widehat{H} := \emptyset$, and let d_i equal the number of nodes $j \in N$ such that either $(i,j) \in A$ or $(j,i) \in A$. We call d_i the *uncovered degree* of node i, since it represents the number of edges incident to i that have not yet been covered.

Step 1. Choose node i such that $d_i = \max_{k \in N} d_k$. If $d_i = 0$, stop and return the heuristic solution $\widehat{H}$. Otherwise, proceed to Step 2.

Step 2. Add node i to $\widehat{H}$. Set $d_i = 0$, and for all arcs $(i,j) \in A$ or $(j,i) \in A$, reduce d_j by one. Return to Step 1.

Since this algorithm is an incumbent methodology for solving the filter placement problem, it is interesting to characterize sufficient conditions under which GVCH yields an optimal solution for MFPP instances.

Lemma 11. *Consider a network with complete (and destination-based) routing and a semi-bipartite topology, and define* $d_r^{\min}$ *and* $d_c^{\max}$ *as in Condition 2. Then if* $d_r^{\min} > d_c^{\max}$, *GVCH will identify the optimal filter placement problem solution to MFPP.*

Proof. Under the assumptions of Lemma 11, Condition 2 holds true, which implies by Proposition 9 that an optimal solution $H = N_r$ exists to MFPP. At each step of GVCH, the node with the largest uncovered degree is chosen. The first vertex chosen must clearly belong to N_r. By induction, suppose that the first $k < |N_r|$ vertices are selected from N_r to belong to $\widehat{H}$. Then each unselected node in N_r must still have an uncovered degree of at least $d_r^{\min}$, since no nodes from N_c belong to $\widehat{H}$. Since the uncovered degree of a node is non-increasing through the algorithm, the maximum uncovered degree of any node in N_c is no more than $d_c^{\max}$, and hence the $k+1$st node selected in $\widehat{H}$ must also belong to N_r. Finally, this process will repeat exactly $|N_r|$ times until all nodes in $|N_r|$ are covered, or else $d_r^{\min} = 0$ which would contradict the assumption that $d_r^{\min} > d_c^{\max}$. This completes the proof. □

It is interesting to show that the condition of Lemma 11 is tight; that is, no guarantee on an optimal solution via the GVCH exists if $d_r^{\min} = d_c^{\max}$. Fig. 1 displays a network where $N_c = \{1,2,3,4,5\}$ and $N_r = \{6,7,8\}$. By Proposition 9, the optimal MFPP solution would filter all nodes in N_r. Observe that $d_c^{\max} = 1$ and $d_r^{\min} = 1$. GVCH would select nodes 6 and 7 first (in some order), but would have an arbitrary choice to make between nodes 5 and 8, since the uncovered degree of nodes 5 and 8 would equal 1 and all other uncovered node degrees would be zero. If node 5 were chosen, the resulting solution would indeed be an optimal vertex cover, but not a feasible filter placement since node 8 could forge a packet's address as node 5 when sending to any node other than 5.

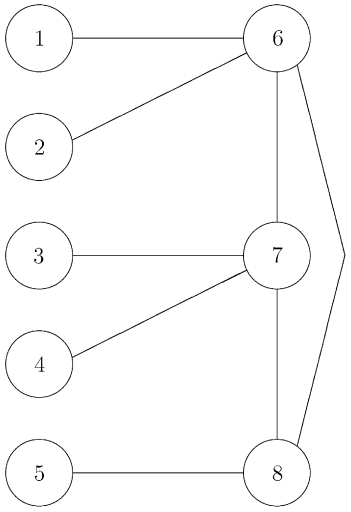

Fig. 1. Example topology demonstrating the tightness of Lemma 11.

Furthermore, note that it is common for nodes to exist whose purpose is to perform routing operations among the routers themselves, and only limited service to client nodes (or perhaps none at all). The presence of these "super-routers" violates the assumptions of Lemma 11, since $N_r^{\min}$ would likely be smaller than $N_c^{\max}$. In this case, the GVCH may fail to place filters on super-routers, despite the fact that these nodes can be used to attack a broad array of client nodes.

5. Summary and future research

In this paper, we introduce a problem arising in the strategic deployment of packet filters to reduce the susceptibility of computer networks to DDoS attacks. We prove that the problem of providing perfect security against forged address attacks using the least number of packet filters is strongly NP-hard, and under common network routing assumptions, can be solved by decomposing the network into multiple vertex cover problems.

However, for several special topological structures, this minimum filter placement problem is polynomially solvable. The greedy vertex cover heuristic commonly used in practice is guaranteed to provide the optimal solution to the minimum filter placement problem under some basic topologies and routing policies, but not necessarily under those that commonly arise in more complex large-scale communication systems (such as the Internet).

A large number of related problems may be investigated stemming from the results of this study. One project that we will undertake is a scenario in which imperfect security is required, since perfect security often requires a rather large deployment of packet filters. Such an approach might try to limit the total number of possible attacks, the maximum cardinality of attacks that can commence from a particular node, or the maximum number of origins that can attack any node. The latter metric is perhaps the most useful, since it allows a node under attack to focus the list of suspected attackers to a limited subset of candidates. A different objective may minimize the expected number of filters through which legitimate packets must traverse, given a probability distribution on how often certain origin–destination pairs exchange packets, rather than the cardinality of filters placed in the network. Similarly, an expectation can be placed on the frequency of attacks from various sources, given a degree of trust established for each node.

Acknowledgements

The authors are grateful to the anonymous referees, whose comments helped improve the paper. B. Armbruster acknowledges the support of an Undergraduate Research Assistantship from the University of Arizona. J.C. Smith acknowledges the support of the *Defense Advanced Research Projects Agency* (DARPA) under Grant No. N66001-01-1-8925. K. Park was supported, in part, by grants from DARPA (AFRL F30602-01-2-0539), ETRI, NSF (ANI-9875789, EIA-9972883, ANI-0082861), and Xerox.

References

[1] S. Axelsson, Research in intrusion detection systems: A survey, Tech. Rep. TR 98-17, Chalmers University of Technology, 1999.
[2] E. Balas, A.C. Ho, Set covering algorithms using cutting planes, heuristics, and subgradient optimization, Mathematical Programming Study 12 (1980) 37–60.
[3] Computer Emergency Response Team (CERT), Advisory CA-2000-01 Denial-of-service developments, 2000. Available from: <http://www.cert.org/advisories/CA-2000-01.html>.
[4] CERT/CC, SANS Institute, and CERIAS, Consensus roadmap for defeating distributed denial of service attacks, 2000. Available from: <http://www.sans.org/ddos_roadmap.htm>.
[5] M. Conforti, G. Cornuéjols, A. Kapoor, K. Vuskovic, Perfect, ideal and balanced matrices, European Journal of Operational Research 133 (3) (2001) 455–461.
[6] E. El-Darzi, G. Mitra, Solution of set-covering and set-partitioning problem using assignment relaxations, Journal of the Operational Research Society 5 (1992) 483–493.
[7] M. Faloutsos, P. Faloutsos, C. Faloutsos, On power-law relationships of the Internet topology, in: Proc. ACM SIGCOMM, 1999, pp. 251–262.
[8] P. Ferguson, D. Senie, Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing, RFC 2827, 2000.
[9] M.L. Fisher, P. Kedia, Optimal solution of set covering/partitioning problems using dual heuristics, Management Science 36 (1990) 674–688.
[10] L. Garber, Denial-of-service attacks rip the Internet, Computer 33 (4) (2000) 12–17.
[11] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, San Francisco, 1979.
[12] P. Hall, On representatives of subsets, Journal of the London Mathematical Society 10 (1935) 26–35.
[13] D.S. Hochbaum, Approximation algorithms for the set covering and vertex cover problems, SIAM Journal on Computing 11 (3) (1982) 555–556.
[14] D.S. Hochbaum, Solving integer programs over monotone inequalities in three variables: A framework for half integrality and good approximations, European Journal of Operational Research 140 (2) (2002) 291–321.
[15] K. Park, H. Lee, On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack, in: Proc. IEEE INFOCOM, 2001, pp. 338–347.
[16] K. Park, H. Lee, On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets, in: Proc. ACM SIGCOMM, 2001, pp. 15–26.
[17] S. Savage, D. Wetherall, A. Karlin, T. Anderson, Practical network support for IP traceback, in: Proc. of ACM SIGCOMM, 2000, pp. 295–306.
[18] M. Schultz, E. Eskin, E. Zadok, S. Stolfo, Data mining methods for detection of new malicious executables, in: Proc. IEEE Symposium on Security and Privacy, 2001, pp. 178–184.