

On the Economic Placement of Monitors in Router Level Network Topologies

Yongping Tang and Thomas E. Daniels
Department of Electrical and Computer Engineering
Iowa State University, Ames, Iowa 50011
Email: {tangyp, daniels}@iastate.edu

Abstract—Network monitoring systems are very important components for protecting networks. Due to economical and technical constraints, it is necessary to provide a proper monitor placement strategy. In this paper, we discuss how to place monitors for a given router level topology to maximize the observation of attack events. We set up a network model including routing strategies and a threat model for general network topology and then define the monitor placement problem. We give a simple proof to show that our problem is NP-complete and provide heuristic solutions with experimental results. Due to routing asymmetry, upstream traffic and downstream traffic often traverse different paths. We also extend the monitor placement problem to the asymmetric routing scenario and discuss how to optimize monitor placement to observe bi-directional data.

I. INTRODUCTION

A major threat on the Internet is the use of stepping stones and other types of proxies for covert illegal access to computers. In these scenarios, attackers utilize a chain of machines as stepping stones before they attack their ultimate victims thereby hiding their origin. To find the complete attack path and trace the attack to its origin, stepping stone analysis is needed. In such systems, traffic statistics and timing are recorded at many points in a network and some type of correlation analysis is done to link traffic observations corresponding to network sessions.

An obvious precondition to this analysis is sufficient attack information recorded through some network monitoring system. For example, most stepping stone analysis research needs to monitor each session in at least one direction between each node in the chain. Complicating matters, a new method of stepping stone analysis [5] has dramatically improved accuracy by requiring bi-directional traffic be monitored. Because monitors add significant cost to networks, there is the need to consider economic tradeoffs between the number of monitors, their placement, and the system's effectiveness.

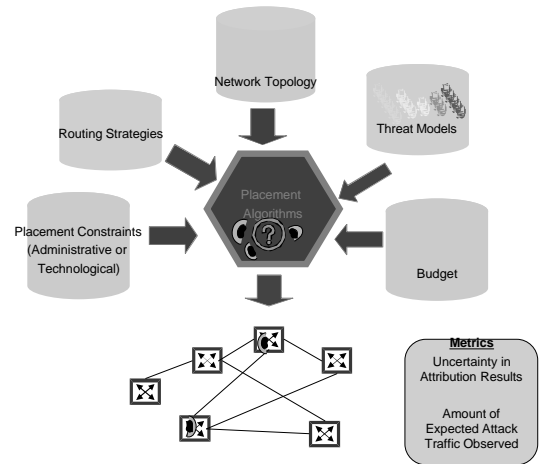


Fig. 1. General Model for Monitor Placement

Due to economic and technical constraints, it is impractical to do universal deployment. Furthermore, the power law degree distribution of networks suggest that there are highly effective, low cost placements. Finding these placements are important to making such stepping stone analysis and other types of attribution systems practical and cost effective.

Generally, there are many factors that affect how to place monitors. These include the network topology, routing strategies, threat models in the network, administrative or technological constraints for placement and the cost of deploying a monitor in a given location. Also, we can consider several different criteria to measure the effectiveness of the monitor placement. This can also drive the placement strategy. Figure 1 shows a general model for the monitor placement problem.

In our work, we take the amount of expected attack traffic observed as the metric of our monitor placement

problem and define our problem so as to maximize the observation of attack events in a router level topology network model. The basic inputs for our problem are router level network topology and the number of monitors. In this simplistic model, we consider the number of monitors as the cost, but allow that router-specific characteristics (e.g. the amount of traffic passing through a node) may affect the cost of individual monitors. In this more general case, the sum of costs may be considered.

To take into account the diversified routing policy and complex attack situations in the Internet, we create a generalized routing and threat model. These allow us to study the problem without in depth information that an ISP or other agency deploying this work would have. It should be noted that given this information, it is straightforward to use it instead of our general assumptions. We set up a routing model based on OSPF [3] and provide a reasonable threat model. Our output is an optimized placement with the goal of maximizing the observed attack events. We will show that this optimization problem is an NP-complete problem, provide some heuristic solutions, and compare experimental results.

In Section 6, we further extend this problem to consider how to observe bi-directional traffic. Due to the asymmetric routing in real networks, the data back and forth between a pair of nodes may not travel through the same path. We will provide a polynomial greedy algorithm for this issue and give some evaluations on the experiment results.

The rest of the paper is organized as follows: in Section 2, we discuss some related work on placement problems; in Section 3, we set up a general routing model and a threat model and define the monitor placement problem based on these two models, and then a simple proof shows that our problem is NP-complete; in Section 4, we present heuristic solutions for our problem and compare experiment results.

In section 5, we give a brief discussion of new algorithms to trace traffic through a monitored network, and in Section 6, we discuss how to place monitors to observe bi-directional traffic using a polynomial-time greedy algorithm. We then draw some conclusions and propose future work.

II. RELATED WORK

Several placement problems have been discussed in the past work. One is the server placement problem [7] to choose k nodes in an n node network to serve other nodes acting as clients. The goal is to minimize the total weighted distance between all servers and all clients.

Similar problem such as facility locations problem is to choose locations for facilities which will provide storage service for supermarkets and the object is to minimize the total transport cost [2]. These are well known NP-Hard optimization problems.

For network monitor placement, there is some work about packet filter placement to defend against denial of service attacks [4]. The work assumes that the routing information is known for the network, and they want to place packet filters to drop all spoofed packets between any two nodes. The purpose of the packet filter placement is to minimize the number of filters and still ensure that any spoofed packets will be detected. It has proved the problem is a NP-complete problem by a reduction from Vertex Cover. The work also discussed some special cases (trees, self healing rings, and bipartite networks) that make the problem solvable in a polynomial time.

More closely related work is monitor placement in Point of Presence (POP) level topology [9]. In that work, we defined information theory metrics based on entropy to measure the system's uncertainty in the origin of the attack. We reduced the original topology graph into the Edge Observed Graph (EOG) based on a placement choice and then convert our problem to find the EOG with the minimum entropy under given number of monitors. We defined two types of entropies: average entropy which stands for uniform attack and worse case entropy which stands for attackers may have knowledge of network topology and monitor placement. We also provide heuristic solutions based on centrality heuristics and compared experimental results.

In this work, we focus on the router level topology and set up an appropriate network model to present the router level topology's features and provide corresponding solutions.

III. PROBLEM DESCRIPTION AND COMPLEXITY

We describe our monitor placement problem in the context of a given topology router level network. The monitor placement is to choose some locations (routers) where we can place monitors to record the network activities to fulfill some criteria.

For attack attribution, we need to record attack activities so that we can do correlation analysis later. Current illegal access attacks usually use several stepping-stones before finally hacking into the targets. The monitor system aims at recording as many attack activities as possible so that we can provide enough information to find these stepping-stones. We are not interested in

the locations where we can monitor maximized general traffic; we are interested in the locations where the attack traffic will pass through with greater probability and monitoring it such that we cover as much distinct attack traffic as possible.

We cannot know exact attack traffic before it happens, but we can estimate the possible attack traffic distribution based on some factors such as the number of hosts behind routers, the history of attacks found on the routers and the location or functional features of the route and its hosts.

A. Threat Model

The monitor placement problem can be modeled using the following graph-theoretic approach. Let graph $G = (V, E)$ presents the network topology, where V is the set of nodes in topology and E is the set of links connecting these nodes. For routers in network, we know some of them are connected to both hosts and routers and the others are only connected to other routers. A router level topology is the presentation of these features (For a POP level topology, one node usually presents several routers and their hosts.). So in our model, we separate the node set V into two groups: the access node set D and the relay node set R . Access nodes are routers with access sub-nets, all hosts are located in these sub-nets. So in the router level topology, we can deem all the traffic are source from access nodes and also destined to access nodes. Relay nodes are core routers that primarily transfer packets to other routers and do not have access sub-nets. Also, we assume relay nodes cannot be the attack source (stepping stone). It is clear that $D \cup R = V$ and $D \cap R = \emptyset$.

Because the monitor system aims to record potential attack traffic, from now we describe the network in the "attack only mode" i.e. we think of that all traffic chosen for capture are attacks. For an access node pair (x, y) , we define the attack rate $r(x, y)$ as the number of attacks from x to y in a unit of time. Also, for the access node x , it has an attack rate $r(x, *)$ that indicates the number of attacks sourced from x in the unit of time. It is clear that $r(x, *) = \sum_{y \in D, y \neq x} r(x, y)$. It is hard to know $r(x, y)$ because x may communicate with any y in the network and guessing the attack rate between them is akin to reading the source's mind!

However, there are some clues to estimate $r(x, *)$, such as the number of hosts behind the access node x and the history of attacks related to the hosts in x 's access sub-net. To simplify the simulation in the experiment of this paper, we assume $r(x, *)$ is known for any $x \in D$ and

we assume the attacks are uniformly distributed in the network, which means $r(x, y) = \frac{r(x, *)}{|D|-1}$. Of course, if $r(x, y)$ could be measured or better approximated that would obviate this rough estimate.

For a relay node d , we define an attack observe rate $aor(d)$ as the number of attack events observed on by the node d in a unit of time. If detailed routing information is known, the attack observe rate can be decided based on it and $r(x, y)$. Unfortunately, routing information is hard to know and our problem only assumes the topology is given. So here we give a general approach to estimate the attack observe rate. We assume OSPF is the routing protocol and each node knows the topology. Then we can deduce that traffic between any access node pair will pass through the shortest path between the access node pair and if multiple shortest paths exist, the traffic will equally distribute among them. If $SP(x, y)$ is the number of shortest paths between access nodes x and y , $SP_d(x, y)$ is the number of shortest paths between access nodes x and y that pass through relay node d , we get:

$$aor(d) = \sum_{x \in D} \sum_{y \in D, y \neq x} \frac{SP_d(x, y)}{SP(x, y)} r(x, y)$$

As before, this is an estimate of routing behavior and more precise knowledge of the protocols or actual routing tables can be used to better model a given topology.

We provide the Algorithm 1 to decide the attack observe rate for each relay node.

B. Problem Definition

Based on the above routing and threat model, we can define the monitor placement problem as the following: In the router level topology $G(D, R, E)$, find a k node subset of R such that the effective attack observe rate of the k monitored relay nodes is maximized.

Because the same attack event will sometimes be observed by multiple relay nodes along an attack path, we note that the sum of k nodes' attack observe rates is not the sum of k nodes' individual attack observe rates. We call this difference an overlap. For example, consider the case where two relay nodes are connected by an edge. If both are monitored, all attacks crossing the shared edge would be observed by both of them and in this sense, they would overlap. Because of this overlap, it may not be wise to monitor both of them even if each of them individually has the greatest attack observe rate in the network.

C. NP-Completeness

To analyze the complexity of our problem, we convert the problem to its decision version: given the network

Input : Network topology $G(D, R, E)$, $r(x, *)$
Output: Each relay node's attack observe rate
begin
 for $x \in D$ **do**
 $aor(x) \leftarrow r(x, *)$;
 end
 for $d \in R$ **do**
 $aor(d) \leftarrow 0$;
 end
 for $x \in D$ **do**
 for $y \in D$ and $y \neq x$ **do**
 $aor(y) \leftarrow aor(y) + r(x, *) / (|D| - 1)$;
 $L \leftarrow$ the set of all shortest paths
 between x and y ;
 for $l \in L$ **do**
 for each relay node d on l **do**
 $aor(d) \leftarrow aor(d) + \frac{r(x, *)}{(|D|-1)*|L|}$;
 end
 end
 end
 end
end
Algorithm 1: Attack Observe Rate for Relay Node

topology and models as presented above, for k monitors, can we find a placement with at least p percent of the network wide attack observe rate? Here p is a constant.

First, we show this decision version problem is in NP. For a solution to this problem, the certifier adds up these k nodes' aor and then removes the overlapping portions to get the effective aor before verify whether the sum is at least p percent of the total. This can be done in polynomial time as showed in our Algorithm 1. So our problem belongs to the complexity class NP.

To show our problem is NP-complete, here we reduce the partial SET COVER problem [8] to our problem. The partial SET COVER problem can be described as: for a finite set U and a number of subsets of U , $S = \{S_1, S_2, \dots, S_n\}$, find a partial cover $S^* \subseteq S$ that covers at least p percent of U with the minimum cardinality. Here S_d can be treated as the set of attack events that can be observed on relay node d and U is the set of all attack events in a unit of time in our problem.

Assume we have the polynomial-time algorithm that solves our placement decision problem. To solve the partial SET COVER problem, we generate a network such that the elements of U are represented as access nodes, i.e. $D = U$. For each $S_i \in S$, create a relay node with edges connecting it to the appropriate nodes in D .

We then start from a random k and use the solution to find the k relay nodes that cover . If the union is larger than p percent of the total U , we decrease k ; otherwise we increase k . Repeat the above procedure until the union is near equal to p percent of the total U . Because k is between 0 and the number of the relay nodes, the above procedure is in polynomial time. According to this reduction, if our problem can be solved, the partial SET COVER problem can also be solved. So our problem is NP-Complete.

IV. EXPERIMENTS WITH HEURISTIC AND GREEDY SOLUTIONS

A. Greedy Method in Polynomial Time

For the SET COVER problem, there is a well-researched greedy algorithm [8] which has a $\ln N$ approximation where N is the cardinality of U . We modified that algorithm to fit our problem, Algorithm 2. In our case, S_r is the observed attack events set on relay node r and U is the set of all attack events in the network.

Input : Attack set $S = \{S_r | r \in R\}$ and U
Output: Monitor placement $\hat{S} = \{S_{i1}, \dots, S_{ik}\}$

begin
 $i \leftarrow 0$;
 while $U \neq \emptyset$ and $i < k$ **do**
 $max \leftarrow 0$;
 $i \leftarrow i + 1$;
 for $r \in R$ **do**
 if $S_r \neq \emptyset$ and $|S_r| > max$ **then**
 $max \leftarrow |S_r|$;
 $i^* \leftarrow r$;
 end
 end
 $U \leftarrow U \setminus S_{i^*}$;
 $\hat{S}_i \leftarrow$ the original set S_{i^*} ;
 for $r \in R$ **do**
 $S_r \leftarrow S_r \setminus S_{i^*}$;
 end
 end
 return \hat{S} ;
end

Algorithm 2: Greedy Algorithm for Monitor Placement

B. Heuristic Methods

Heuristic solutions are very useful for NP-complete optimization problems. Here we use three heuristic methods for our problem: K-max, degree and betweenness.

K-max is to choose k nodes with the highest attack observe rates, due to the existing of overlapping, the sum of K-max may not be the maximized.

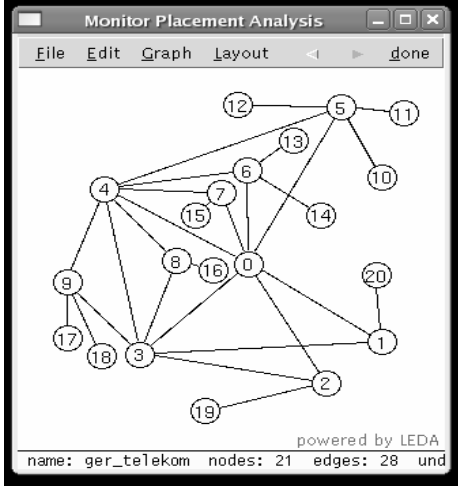


Fig. 2. A Sample Network Topology

Degree is the measurement of the number of neighbors a node has. The higher degree of a node, the more traffic may congregate at the node. The idea is that there is greater probability that the attack will pass through.

Betweenness is defined as $\sum_{s,t \in V, s \neq t \neq v} \frac{SP_v(s,t)}{SP(s,t)}$, a node with high value of betweenness means it is a common node on many shortest paths. We assume traffic go through shortest paths, so a node with high betweenness should have high probability that attack traffic will pass through it.

C. Experiment Results Comparison

Fig 2 is an example topology graph that has 11 access nodes and 10 relay nodes. The results based on the above four methods are showed in Fig 3.

According to these experiment results we found all methods can provide good optimization rates. And for this sample topology, greedy algorithm is a little better than heuristic methods K-max and Betweenness, but heuristic method Degree gives the best result.

If we consider algorithm complexity, we find that heuristic methods are better than the greedy algorithm. Here we assume the topology graph has n nodes and m edges. We know for method K-max, it computes attack observe rate for each relay node and needs traverse all shortest path between all access nodes, and thus the complexity is $O(n^3)$ if using the Floyd-Warshall algorithm; for method Degree, the complexity is $O(n + 2m)$ cause it only needs to visit all nodes and twice of all edges (the sum of degree is twice of the number of edges);

# Monitors	Method	Nodes Chosen	Total AOR
3	KMax	4,5,6	80%
	Degree	0,3,4	91%
	Betweenness	0,4,5	83%
	Greedy	5,6,9	89%
4	KMax	4,5,6,9	91%
	Degree	0,3,4,5	96%
	Betweenness	0,4,5,6	84%
	Greedy	5,6,9,1	95%
5	KMax	0,4,5,6,9	95%
	Degree	0,3,4,5,6	98%
	Betweenness	0,4,5,6,9	95%
	Greedy	5,6,9,1,2	98%

Fig. 3. Results Compared

for Betweenness, it runs BFS for all nodes and thus the complexity is $O(n^2 + nm)$. For the greedy algorithm, it first need to distribute attack events along all shortest paths and the complexity is $O(n^3)$ and it also need do Set Minus operation which is $O(n^4)$, so the total complexity is $O(n^4)$. Furthermore, it also simulates attack events for each access node and thus needs $O(n^2)$ space.

V. NODE CORRELATION

Stepping stone correlation algorithms take the data recorded about two network sessions from different monitors and return a similarity score indicating their similarity. When there is an effective stepping stone correlation algorithm, monitors can utilize this algorithm to decide if two suspect links in fact belong to a same stepping stone chain. But one monitor alone can only correlate link data on its local storage, to find out the whole or as much as possible of the stepping stone chain, monitors need to work together. Hence, monitors will cooperate with each other to identify as many possible stepping stones for a given attack.

In POP or autonomous system (AS) level topology, each node can be an attack source, so the chain trace back procedure can be made node by node. This method is called the Direct Search Algorithm (DSA): one node receives a found attack pattern based on an incoming link and then correlates it with all of its outgoing links. If a match is found, continue to the next node. Fig 4 shows this algorithm.

For router level topology, DSA cannot work directly because monitors are located between two access nodes and one monitor may not have all data of an access

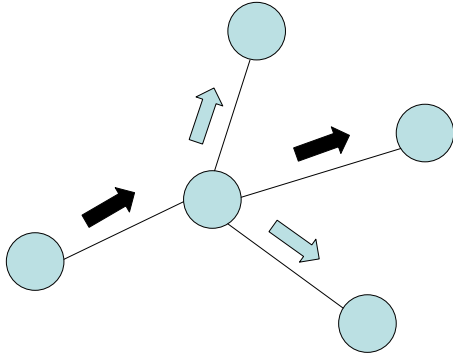


Fig. 4. DSA for AS Level Topology

node. In fact, one access node's incoming and outgoing link data are distributed in several monitors; and for one monitor, it may have several access nodes' data but not all because one monitor may be shared by several paths.

In our work, we assume monitors on the same routing path can talk to each other directly. To quickly find out the chain as much as possible, we provide the following novel algorithm to do node correlation (node trace back):

- Start from a monitor M with an attack pattern Pst.
- M shares the Pst to all monitors that it can directly talk to.
- In parallel, the monitors get the pattern:
 - Correlate the pattern with their local data.
 - Report match.
 - Share the pattern to all other monitors that they can directly talk to except the monitor where the pattern came from.
 - If repeated pattern is received, just ignore.

VI. BI-DIRECTIONAL TRAFFIC

Asymmetric routing [6] is common phenomenon in Internet. Due to different routing policies in different parts of Internet, it is impossible to control end-to-end routing and paths traversed. So packets may not always traverse same downstream path as they do when forwarded upstream. This can be shown as Fig 5.

In some applications, it is important to acquire bi-directional traffic. But in general, monitors cannot be placed on data source or destination due to the security reasons, instead, they usually placed on mid-way routers as in our work. When this is done, asymmetric routing may cause monitors to only capture one direction of data between two access nodes. In Sections 3 and 6, we assume all routing is symmetric. In this section, we will discuss how to place monitors under an asymmetric

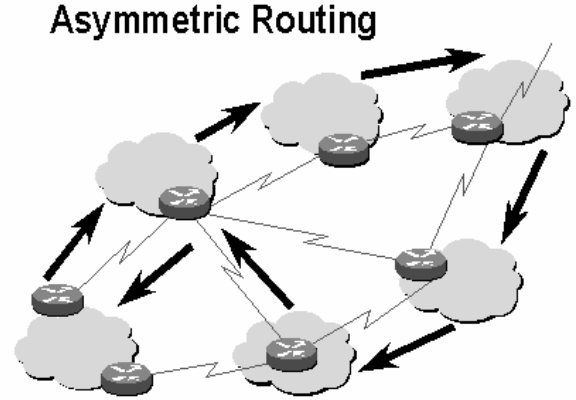


Fig. 5. Asymmetric Routing

routing scenario. Because of the asymmetric situation, we must assume routing paths are known.

It is clear that unless the forward path and the back path between an access node pair traverse a common monitor, one monitor can only get one direction of the data. According to section V, we know the monitors can correlate with each other. So we propose a straight forward idea that if there is a monitor on the forward path, we place another monitor on the back path. Based on this idea, we give a greedy algorithm (Algorithm 3) to place monitors to get bi-directional traffic data.

In this algorithm, function FindMAXNode is used to find the node that covers the most paths in a path set and cover in this context implies that the path passes through this node.

We use the data set collected by Active Measure Project (AMP) from NLANR [1] to evaluate our algorithm. This data set includes the traceroute data between 105 sites (after removing the sites unreachable in the data set). We use these traceroute paths as the routing paths to construct the network topology for our experiments. To compare the results, we also implemented the one directional traffic solutions on undirected versions of the graph: K-max and Degree.

The first experiment is based on a small topology which is derived from 6 randomly selected sites in the data set. And for this small topology we also implemented the brute-force enumeration algorithm to get ideal optimal result. The final results are shown in Fig 6. The number of access nodes is 6 and the number of

relay nodes is 57.

Input : Network Topology with Routing Information

Output: k Nodes Chosen to Place Monitors

begin

$A \leftarrow$ all paths between access nodes;

$B \leftarrow$ null;

$C \leftarrow$ null;

$Monitor \leftarrow$ null;

$numofm \leftarrow 0$;

while $A \neq \emptyset$ and $numofm < k$ **do**

$x \leftarrow FindMAXNode(A, Monitor)$;

$numofm ++$;

$Monitor \leftarrow Monitor \cup \{x\}$;

$pathA \leftarrow \{paths \in A \text{ covered by } x\}$;

$A \leftarrow A \setminus pathA$;

$C \leftarrow C \cup pathA$;

$pathA \leftarrow \{paths \in$

$A \text{ reversed with } pathA\}$;

$A \leftarrow A \setminus pathA$;

$B \leftarrow B \cup pathA$;

while $B \neq \emptyset$ and $numofm < k$ **do**

$y \leftarrow FindMAXNode(B, Monitor)$;

$numofm ++$;

$Monitor \leftarrow Monitor \cup \{y\}$;

$pathB \leftarrow \{paths \in$

$B \text{ covered by } y\}$;

$B \leftarrow B \setminus pathB$;

$C \leftarrow C \cup pathB$;

$pathA \leftarrow \{paths \in$

$A \text{ covered by } y\}$;

$A \leftarrow A \setminus pathA$;

$C \leftarrow C \cup pathA$;

$pathA \leftarrow \{paths \in$

$A \text{ reversed with } pathA\}$;

$A \leftarrow A \setminus pathA$;

$B \leftarrow B \cup pathA$;

end

end

return $Monitor$;

end

Algorithm 3: Algorithm for Bi-directional Traffic

The second experiment uses the topology based on all data and it has 105 access nodes and 1192 relay nodes. Due to the time complexity, we do not run the brute-force algorithm for this experiment. The results are shown as Fig 7.

From the above results, we found our greedy bi-directional algorithm found placements with 60-65% of attacks observed for the topology constructed by AMP

# Monitors	Method	Node Chosen	Total AOR	Approximation
4	K-max	33,25,42,34	73.33%	88%
	Degree	33,11,3,42	76.67%	92%
	Brute-force	42,32,11,3	83.33%	100%
	Bi-directional	6,10,25,33	73.33%	88%

Fig. 6. Results for 6 Sites

# Monitors	Method	Total AOR
30	K-max	82.10%
	Degree	74.06%
	Bi-directional	60.32%
40	K-max	86.54%
	Degree	77.76%
	Bi-directional	62.68%
50	K-max	88.16%
	Degree	81.66%
	Bi-directional	65.92%

Fig. 7. Results for all Sites

traceroute data.

In the above tables we found that the heuristic methods K-max and Degree get more aor than our bi-directional algorithm. This because heuristic methods only get one directional data but our greedy algorithm will ensure all data captured are bi-directional. So the less aor got compare to those heuristic methods here is the price we pay to get bi-directional data. Another reason is that the topology in experiments constructed using traceroute paths, it has some special features: not a real Internet topology, has small number of access nodes and all forward paths are totally different with back paths (each pair of paths share no common nodes except the source and the destination). These features make the results worse. In a real Internet topology, the number of access nodes will be greater than the number of relay nodes and most forward paths and back paths should share common nodes. So in a real Internet topology, the result of our algorithm should be much better.

VII. CONCLUSIONS AND FUTURE WORK

Using our approach, we can easily find reasonable locations to place monitors for a router-level topology and also find a way to minimize the number of monitors deployed in a network while maintaining effective network attack attribution. Our bi-directional algorithm is a new and effective approach to solve the problem of observing two-way traffic via monitor placement.

There are still many aspects of this work we need to consider. In section V, we just provide a simple idea to do the node correlation in router-level topology without evaluation. We need provide more formal algorithm and experiments to evaluate it. The routing model and threat model in section III and VI are simplistic. In the future, we need provide more realistic models for router-level topology. Also, we need real asymmetric routing Internet topologies to further evaluate our bi-directional greedy algorithm.

ACKNOWLEDGMENT

This research is supported by DOI under contract No. NBCHC030107.

REFERENCES

- [1] Active measure project. <http://amp.nlanr.net/>.
- [2] Rfc 2328. <http://www.ietf.org/rfc/rfc2328.txt>.
- [3] Benjamin Armbruster et al. A packet filter placement problem with application to defense against spoofed denial of service attacks. <http://www.cs.purdue.edu/nsl/ejor-dpf05.pdf>.
- [4] Jianqiang Xin et al. An effective scheme for detecting stepping stone attacks in real-world networked systems. Submitted to InfoComm 07.
- [5] Yihua He et al. On routing asymmetry in the internet. <http://www.cs.ucr.edu/krish/yhe-gcom05.pdf>.
- [6] Yu-Ju Kuo and Hans D. Mittelman. Facility location problem. <http://plato.asu.edu/papers/paper97/node5.html>.
- [7] L Qiu, V. Padmanabhan, and G. Voelke. On the placement of web server replicas. In *IEEE Infocom*, April 2001.
- [8] Petr Slavik. *Approximation Algorithms for Set Cover and Related Problems*. PhD thesis, University of New York at Buffalo, Buffalo, NY, 1998.
- [9] Yongping Tang, R. Y. Liverpool, and T. E. Daniels. Monitor placement for stepping stone analysis. In *Proceedings of the 25th IEEE International Performance Computing and Communications Conference (Workshop on Information Assurance WIA-2006)*, Phoenix, AZ, April 2006.