Detecting Network Intrusions via Sampling : A Game Theoretic Approach

Murali Kodialam T. V. Lakshman

Bell Laboratories
Lucent Technologies
101 Crawfords Corner Road
Holmdel, NJ 07733, USA
{muralik, lakshman}@bell-labs.com

Abstract

In this paper, we consider the problem of detecting an intruding packet in a communication network. Detection is accomplished by sampling a portion of the packets transiting selected network links (or router interfaces). Since sampling entails incurring network costs for real-time packet sampling and packet examination hardware, we would like to develop a network packet sampling strategy to effectively detect network intrusions while not exceeding a given total sampling budget. We consider this problem in a game theoretic framework, where the intruder picks paths (or the network ingress point if only shortest path routing is possible) to minimize chances of detection and where the network operator chooses a sampling strategy to maximize the chances of detection. We formulate the game theoretic problem, and develop sampling schemes that are optimal in this game theoretic setting.

I. Introduction

In this paper, we consider the problem of detecting intrusions in a communication network. There is growing literature on providing security in communication networks. Two key areas of interest in security are intrusion detection and intrusion prevention. In this paper, we deal with the problem of intrusion detection. Intrusion in networks takes many forms including denial of service attacks, viruses introduced into the networks, etc. Typically, in an intrusion problem, the intruder attempts to gain access to a particular file server or website in the network. In this paper, we consider a stylized intrusion problem. In this problem, the intruder attempts to send a malicious packet to a given node in the network. The network attempts to detect this intrusion. The detection mechanism is packet sampling and examination in the network.

The idea in sampling is that some portion of packets traversing designated links (or router interfaces) are sampled and examined in detail to determine whether the packet is an intruder packet. This packet examination may be simple (limited to specific packet header fields as in packet filtering) or may involve a more detailed examination of the packet. To prevent packet mis-ordering or reduction of link throughput this examination has to be done preferably at line rates. Packet sampling has been previously proposed for a variety of networking purposes. For instance, the SRED scheme in [6] uses packet sampling to estimate the number of active TCP flows in order to stabilize network buffer occupancy for TCP traffic. Only packet headers need be examined for this scheme. The scheme proposed in [7], also uses packet sampling and it is used for fair link-bandwidth allocation. Sampling has also been proposed to infer network traffic and routing characteristics [3]. Whereas, these applications require only sampling based on packet header comparisons, intrusion detection may entail a more thorough examination of sampled packets. Also, unlike some of the sampling applications mentioned above, sampling for intrusion detection requires near line-speed packet examination since copying sampled packets or packet-headers for off-line analysis is not sufficient to prevent intruding packets from getting through. Hence, in the design of an intrusion detection scheme it is imperative to keep the sampling costs in mind.

We study this intrusion detection via sampling problem in a game theoretic setting. Game theory has been used extensively to model different networking problems. This work includes the work of Shenker for modeling service disciplines [10], Akella et. al. for TCP performance [2], and Korilis, Lazar and Orda [5] for modeling routing problems. To the best of our knowledge, this is the first attempt to model intrusion detection via sampling in communication networks using a game-theoretic framework. This work is closely related to drug interdiction models. In particular the work of Washburn and Wood [11] who considered drug interdiction in a game theoretic framework. This work differs from the drug interdiction models in two ways. First, in the drug interdiction models the objective is to deploy agents which is a discrete allocation problem. In our case, the detection is by means of sampling. Therefore the game theoretic results are much more natural than the discrete allocation models. Secondly, in our case, the game theoretic problem naturally leads to a routing problem (to

maximize the service provider's chances of detecting intruding packets) which is absent in the drug interdiction problem. The solution to the game theoretic formulation is a maximum flow problem and the routing problem can be formulated as a multi-commodity flow problem. We also consider various extensions and variants to the basic models.

II. Problem Definition

The problem set-up is outlined in three steps. First, we describe the network, then we define the adversaries in the game-theoretic framework, and finally we describe the objective of the game that is played between the adversaries.

A. Network Set-Up

We consider a network $G = (N, E)$ where N is the set of nodes and E is the set of unidirectional links in the network. We assume that there are n nodes and m links in the network. We assume that the capacity of link $e \in E$ is denoted by c_e and the amount of traffic flowing on link e is denoted by f_e. Given two nodes u and v in the network, let $\mathcal{P}_u^v$ represent the set of paths from u to v in G. Given an m-vector w, we use $M_{uv}(w)$ to denote the maximum flow that can be sent from node u to node v using w as the link capacities. We use the parameter w explicitly when we define $M_{uv}()$ to indicate that dependence of the maximum flow on the link capacities. Corresponding to this maximum flow between nodes u and v, there is a minimum cut comprising of a set of links in the network. This set of links in this minimum cut will be represented by $\mathcal{C}_u^v(c)$.

B. Network Intrusion Game

The network intrusion detection game is played on the network between two players: the *Service Provider* and the *Intruder*. The objective of the intruder is to inject a *malicious packet* from some attack node $a \in N$ with the intention of attacking a target node $t \in N$. We assume that an intrusion is successful when the malicious packet reaches the desired target t node without detection. In order to detect and prevent the intrusion, the service provider is allowed to sample packets in the network. We assume that sampling takes place on the links in the network. It is easy to modify the model to consider the case, where the sampling is done at the nodes in the network. If during the course of sampling, the service provider samples the malicious packet then the intrusion is assumed to be detected and thwarted. The game is pictorially illustrated in Figure 1.

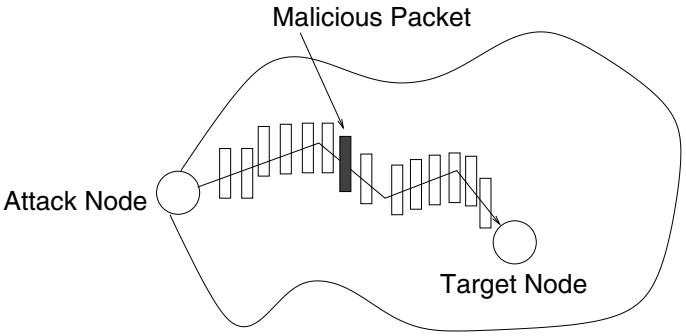

Fig. 1. Network Intrusion Game

C. The Objective and the Constraints of the Game

If there is no bound on the amount of sampling that can be done by the service provider, then the service provider can potentially inspect every packet that flows through the network and hence detect the malicious packet. Sampling the packets flowing on a link involves setting up the appropriate sampling filters and examining the packets. These can be fairly expensive operations to perform in real time. Therefore, we assume that the service provider has a sampling bound of B packets per second over the entire network. This sampling effort can be distributed arbitrarily over the links in the network. One way of implementing the sampling scheme is for each link to pick some fraction of the packets flowing through it and send it to a central intrusion detection node in the network which examines the packet in more detail. The sampling bound can be viewed as the maximum rate at which the intrusion detection node can process packets in real time. If a link e that has a traffic of f_e flowing on it, is sampled at rate s_e then the probability of detecting a malicious packet on this link is given by $p_e = s_e/f_e$. The sampling budget constraint implies that $\sum_{e \in E} s_e \leq B$. We formulate the game theoretic problems in terms of p_e. We assume that both the players have complete information about the topology of the network and all the link flows in the network. The service provider can have access to this information either from link-state routing protocols with traffic engineering extensions that distribute flow information throughout a network area or by explicit link polling from management systems. We assume that the adversary injecting intruding packets has this information available as well since this makes the service provider's detection problem more difficult. Similarly, we also assume that the intruder is capable of picking paths in the network so as to make the detection problem for the service provider more difficult. However, in Section V-A, we also consider the case where only shortest path routing is allowed in the network.

1) Strategies for the Two Players: In the case of the intruder, a pure strategy would be to pick a path from $P \in \mathcal{P}_a^t$ for the malicious packet to traverse from s to t. The intruder, in general, can use a mixed strategy. In the case of a mixed strategy, the intruder has a probability distribution q over the set of paths in $\mathcal{P}_a^t$ such that $\sum_{P \in \mathcal{P}_a^t} q(P) = 1$. Let $V = \{q : \sum_{P \in \mathcal{P}_a^t} q(P) = 1\}$ represent the set of feasible probability allocations over the set

of paths between a and t. The intruder then picks path $P \in \mathcal{P}_a^t$ with probability $q(P)$. The strategy for the service provider is to determine a set of links on which sampling has to be done. The strategy for the service provider is to choose the sampling rate s_e on link e such that $\sum_{e \in E} s_e \leq B$. If the malicious packet traverses link e with a sampling rate of s_e on a link with flow f_e results in the malicious packet being detected with probability $p_e = s_e/f_e$. Let $U = \{p : \sum_{e \in E} p_e f_e \leq B\}$ represent the set of detection probability vectors p that satisfy the sampling budget constraint. (Note that p is an m- vector.) Instead of viewing the service provider as picking the sampling rates at the links, we view the service provider as picking a set of detection probabilities at the links which belongs in the set U. Figures 2 and 3 depict the intruder's and the service provider's actions.

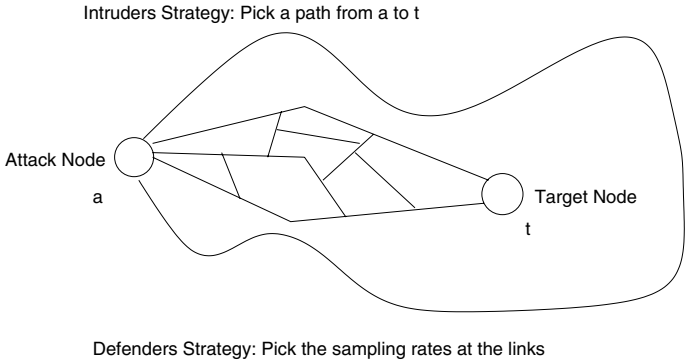

Fig. 2. Intruder's Problem

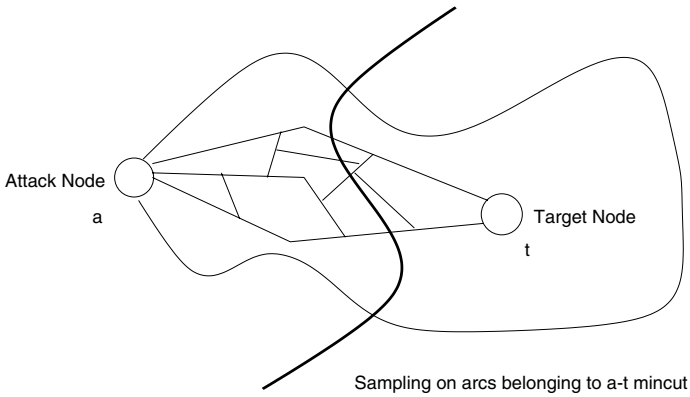

Fig. 3. Service Provider's Problem

2) Payoff Matrix: Assume that the intruder and the service provider each have chosen a strategy. This implies that the intruder has picked a probability distribution q over the set of paths in $\mathcal{P}_a^t$ and the service provider has picked a set of detection probabilities p at the links. The payoff that we consider, is the expected number of times the malicious packet is detected as it goes from a to t. For a given path $P \in \mathcal{P}_a^t$, the expected number of times that a packet is detected is given by $\sum_{e \in P} p_e$. The probability that this path P is picked by the intruder is given by $q(P)$. Therefore the expected number of times a packet is detected as it goes from the source to the destination for a fixed strategy from both adversaries is given by

$$\sum_{P \in \mathcal{P}_a^t} q(P) \left[\sum_{e \in P} p_e \right].$$

Interchanging the order of summation, we get

$$\sum_{P \in \mathcal{P}_a^t} q(P) \left[\sum_{e \in P} p_e \right] = \sum_{e \in E} p_e \left[\sum_{P \in \mathcal{P}_a^t : P \ni e} q(P) \right].$$

This can be equivalently written in a matrix form as $q^T M p$ where M is an $m \times |\mathcal{P}_a^t|$ path-arc incidence matrix. Each row in M represents a link in the network and each column of M represents a path between nodes a and t. The entry corresponding to row e and column P is set to one if $e \in P$ and to zero otherwise. A more natural payoff, is the probability of detection of the malicious packet as opposed to the expected number of times the malicious packet is detected. In this case for a fixed path $P \in \mathcal{P}_a^t$, the probability of the malicious packet being detected is given by $1 - \prod_{e \in P}(1 - p_e)$. This objective is non-linear in p_e which makes the game theoretic problem intractable. However, the two payoffs that we outlined above coincide if the optimal solution for the service provider is to sample at most one link on any path $P \in \mathcal{P}_a^t$ with $q(P) > 0$. We call this strategy a *minimal sampling* strategy. We show later that for all the problems we consider, the optimal solution is a minimal sampling strategy.

3) Objective of the Adversaries: The intruder fears that if his strategy is known to the service provider then service provider will choose a strategy that $\max_{p \in U} \sum_{P \in \mathcal{P}_a^t} q(P) \left[\sum_{e \in P} p_e \right]$. Therefore the objective of the intruder is to pick a distribution $q()$ that minimizes this maximum value. In other words, the objective of the intruder is to

$$\min_{q \in V} \max_{p \in U} \sum_{P \in \mathcal{P}_a^t} q(P) \left[\sum_{e \in P} p_e \right].$$

The objective of the service provider, using a similar argument is

$$\max_{p \in U} \min_{p \in V} \sum_{P \in \mathcal{P}_a^t} q(P) \left[\sum_{e \in P} p_e \right].$$

This is a classical two person zero-sum game and the following minmax result is well known.

Theorem 1: There exists an optimal solution to the intrusion detection game where

$$\theta = \min_{q \in V} \max_{p \in U} \sum_{P \in \mathcal{P}_a^t} q(P) \left[\sum_{e \in P} p_e \right] =$$

$$\max_{p \in U} \min_{q \in V} \sum_{P \in \mathcal{P}_a^t} q(P) \left[\sum_{e \in P} p_e \right],$$

where θ is the value of the game.

In the rest of this paper, we show how this minmax optimal solution can be computed for the intrusion detection game and use that insight to route flows in the network.

III. Solution of the Game

We now consider the solution of the minmax problem formulated in the last section. The idea is to get some insight into the structure of the problem which will enable us to extend the solution to more complex cases. Consider the intruders problem.

$$\min_{q \in V} \max_{p \in U} \sum_{e \in E} p_e \left[\sum_{P \in \mathcal{P}_a^t : P \ni e} q(P) \right].$$

For a fixed $q \in V$ the inner maximization problem is the following:

$$\max \sum_{e \in E} \left[\sum_{P \in \mathcal{P}_a^t : P \ni e} q(P) \right] p_e$$

$$\sum_{e \in E} f_e p_e \quad \leq \quad B$$

$$p_e \quad \geq 0$$

Associating a dual variable λ with the budget constraint, we obtain the following dual optimization problem.

$$\min B\ \lambda$$

$$f_e \lambda \quad \geq \quad \sum_{P \in \mathcal{P}_a^t : P \ni e} q(P) \ \ \forall e \in E$$

$$\lambda \quad \geq 0$$

Substituting this optimization problem in the intruders minmax formulation makes it the following minimization problem.

$$\min B\ \lambda$$

$$\sum_{P \in \mathcal{P}_a^t : P \ni e} q(P) \quad \leq \quad f_e \lambda \quad \forall e \in E$$

$$\sum_{P \in \mathcal{P}_a^t} q(P) \quad = \quad 1$$

$$\lambda \quad \geq 0$$

Interpreting $q(P)$ as a flow on path P, the constraint

$$\sum_{P \in \mathcal{P}_a^t : P \ni e} q(P) \leq f_e \lambda$$

restricts the flow on a link e to be $f_e \lambda$. Therefore $f_e \lambda$ can be interpreted as the capacity of link e. The constraint $\sum_{P \in \mathcal{P}_u^v} q(P) = 1$ enforces one unit of flow to be sent from the source to the destination. Assume that f_e is the capacity of link e in the network. The objective then is to determine the smallest scaling factor λ, on the links in the network so that a flow of one unit can be sent from the source to the destination. This can be done as follows:

- Assume that link e has capacity f_e and determine the maximum flow, $M_{at}(f)$ from the a to t using these capacities.
- Set $\lambda = M_{at}(f)^{-1}$. By scaling the capacities by λ, note that a flow of one unit is sent from a to t.
- The value of the game $\theta = BM_{at}(f)^{-1}$.

Any maximum flow from a to t can be decomposed into a set of flows on paths from a to t using standard flow decomposition techniques. From network flow duality, note that corresponding to the maximum flow value there is a minimum cut. The stable operating point for the intruder and the service provider are the following:

- *Intruders Strategy:* Solve the maximum flow $M_{at}(f)$, from a to t using a capacity of f_e on link e. Using standard flow decomposition techniques, decompose the maximum flow into flow on paths $P_1, P_2, \ldots, P_l$ from a to t. with flows of $m_1, m_2, \ldots, m_l$ respectively. (Note that $\sum_{i=1}^{l} m_i = M_{at}(f)$.) The intruder introduces the malicious packet along the path P_i with probability $m_i * M_{at}(f)^{-1}$.
- *Service Providers Strategy:* The service provider computes the maximum flow from a to t using f_e as the capacity of link e. Let $e_1, e_2, \ldots, e_r$ denote the arcs in the corresponding minimum cut with flows $f_1, f_2, \ldots, f_r$. From duality $\sum_{i=1}^{r} f_i = M_{at}(f)$. The service provider samples link e_i at rate $Bf_i M_{at}(f)^{-1}$.

We now illustrate the above results on the example shown in Figure 4. The numbers next to the links are the flows on the links. How these flows are generated is discussed in detail in a subsequent section. For now assume that the flows on the links are given. Assume that there is a sampling budget B of 5 units. and $a = 1$ and $t = 5$ are the attack and target nodes respectively. The links $(1,2), (4,5)$ belonging to the minimum $a-t$ cut are shown in thick lines. The minimum cut (and hence the maximum flow) has a value of 11.5 units. The intruder's strategy is the following:

- Introduce the malicious packet along the path 1-2-5 with probability 7.0/11.5
- Introduce the malicious packet along the path 1-2-6-5 with probability 0.5/11.5

- Introduce the malicious packet along the path 1-3-4-5 with probability 4.0/11.5

The minmax strategy for the service provider is the following:

- Sample link 1-2 at rate 5/11.5 giving a total sampling rate of $(5 \times 7.5)/11.5$ on that link.
- Sample link 4-5 at rate 5/11.5 giving a total sampling rate of $(5 \times 4.0)/11.5$ on that link.

Note that $\theta = 5/11.5$ is the value of the game.

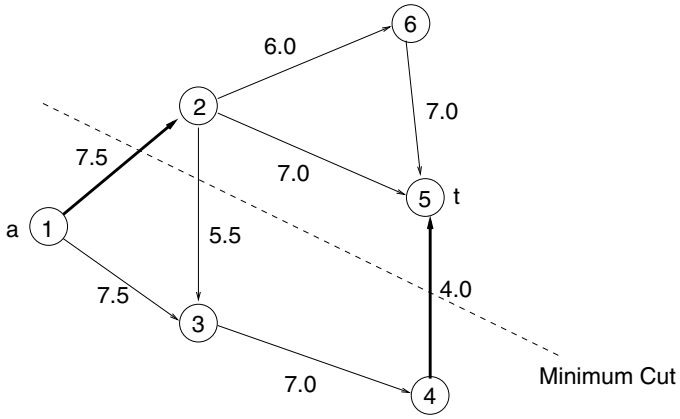

Fig. 4. Example of Network

The following observations can be made about the minmax optimal solution:

- The optimal strategy for the service provider is to sample packets on the mincut with respect to the traffic flows. This implies that along any path that the intruder would choose, the malicious packet will be sampled at most on one link. Therefore this is a minimal sampling scheme.
- If $B \geq M_{at}(f)$ then note that the malicious packet will always be detected. If $B < M_{at}(f)$ then there is a non-null probability that the malicious packet will not be detected.

IV. Routing to Improve the Value of the Game

In the last section, we showed that the value of the network intrusion game is given by $BM_{at}(f)^{-1}$. All along, we assumed that the flow f on the links is fixed. The flows on the links are a result of routing the demands (aggregate traffic between node pairs) in the network. In this section, we explore the case where the service provider adjusts the flows in the network in order to maximize the value of the game. Corresponding to each pair of nodes in the network, there could potentially be demands that have to be routed from the first node in this pair to the second node. Each node pair between which there is some demand that has to be routed is termed a source-destination pair or a commodity. We assume that there are K source-destination demand pairs (commodities) in the network. The source node for commodity k will be represented by $s(k)$, the destination node by $d(k)$ and the amount of demand (bandwidth) that has to be routed for this source-destination pair is $b(k)$. The service provider has to route these flows in the network respecting the link capacity constraints. For the example shown in Figure 4, each link is assumed to have a capacity of 10 units. The different source-destination pairs and the corresponding demands are shown in Table I. These demands have to be routed in the network such that the link capacity constraints are respected. There are several ways of routing these demands. One commonly used method is to route the demands such that the maximum link utilization in the network is minimized. (This can be solved as a maximum concurrent flow problem.) The link flows shown in Figure 4 are a result of routing the demands in order to minimize the maximum utilization in the network. The routing is given in Table II.

Source Dest. Pair	Demands
1-3	5.0
1-4	3.0
1-5	7.0
2-3	1.0
2-5	10.0
6-5	1.0

TABLE I
SOURCE-DESTINATION PAIRS AND DEMANDS

Source Dest. Pair	Paths	Flow
1-3	1-3	5.0
1-4	1-2-3-4	0.5
	1-3-4	2.5
1-5	1-2-6-5	6.0
	1-2-3-4-5	1.0
2-3	2-3	1.0
2-5	2-5	7.0
	2-3-5	3.0
6-5	6-5	1.0

TABLE II
FLOWS FOR BASE CASE

We now explore the case where the service provider routes these flows such that the value of the network intrusion game is maximized. In other words, the service provider routes the source-destination demands such that the maximum probability of detection of the malicious packet is increased. We first

formulate this problem and then explore different heuristics to solve the problem. Recall that $\mathcal{P}_{s(k)}^{d(k)}$ represents the set of paths between the source node $s(k)$ and the destination node $d(k)$ for commodity k. For notational simplicity, we refer to $\mathcal{P}_{s(k)}^{d(k)}$ as $\mathcal{P}_k$. Note that $\mathcal{P}_k$ represents the set of valid paths to route commodity k. Let $X = \{x(P) : \sum_{P\in\mathcal{P}_k} x(p) = d(k)\ \forall k, \sum_k \sum_{P\in\mathcal{P}_k:e\in P} x(P) \leq c_e\ \forall e \in E\}$. Note that X denotes an allocation of flow on paths in the network which meets the demand for each commodity while satisfying the capacity constraints on the links in the network. Given a feasible routing vector, $x \in X$, the flow on link e is given by $f_e = \sum_k \sum_{P\in\mathcal{P}_k:e\in P} x(P)$. From Section III, the value of the game is given by $B/M_{at}(f)$. The objective of the service provider then is to route the source destination demands such that the resulting value of $M_{at}(f)$ is as small as possible. Therefore the objective of the service provider is to solve the following optimization problem.

$$\min_{x\in X} M_{at}(\sum_k \sum_{P\in\mathcal{P}_k:e\in P} x(P)).$$

This can be written more explicitly as

$$\min_{x\in X} \sum_{P\in\mathcal{P}_a^t} y(P)$$

$$\begin{aligned} \sum_{P\in\mathcal{P}_u^v:e\in P} y(P) &\leq \sum_k \sum_{P\in\mathcal{P}_u^v:e\in P} x(P) \\ y(P) &\geq 0. \end{aligned}$$

Unfortunately this problem cannot be solved as a linear programming problem. It is possible to reformulate this problem as a non-convex optimization problem but it is not clear if there is a solution technique to solve this problem. We therefore develop two different heuristics to get good solutions to this optimization problem.

A. Flow Flushing Algorithm

Let the m-vectors c and f represent the link capacity and the flow on the link respectively. The flow on the links is a result of routing the different source- destination demands in the network. It is easy to see that

$$M_{at}(f) + M_{at}(c-f) \leq M_{at}(c).$$

This is true since the set of flows in the two terms on the left hand side of the inequality is a feasible flow for the right hand side of the inequality. Therefore $M_{at}(f) \leq M_{at}(c) - M_{at}(c-f)$. If f is the result of routing the source-destination demands then

$$M_{at}(\sum_k \sum_{P\in\mathcal{P}_k:e\in P} x(P)) \leq M(c) - M(c - \sum_k \sum_{P\in\mathcal{P}_k:e\in P} x(P)).$$

Instead of minimizing the left hand side of the inequality, we minimize the upper bound represented by the right hand side of the inequality. Since c is fixed, this is equivalent to maximizing $M(c-\sum_k \sum_{P\in\mathcal{P}_k:e\in P} x(P))$ subject to the constraint that $x \in X$. We write this more formally as:

$$\max \sum_{P\in\mathcal{P}_u^v} y(P)$$

$$\begin{aligned} \sum_k \sum_{P\in\mathcal{P}_k:e\in P} x(P) &\leq c_e \quad \forall e \in E \\ \sum_{P\in\mathcal{P}_u^v:e\in P} y(P) &\leq \sum_k \sum_{P\in\mathcal{P}_k:e\in P} x(P) \quad \forall e \in E \\ \sum_{P\in\mathcal{P}_k} x(P) &= d(k) \quad \forall k \\ x(P) &\geq 0 \quad \forall P \in \mathcal{P}_k\ \forall j\ \forall k \\ y(P) &\geq 0 \quad \forall P \in \mathcal{P}_u^v\ \forall j\ \forall k \end{aligned}$$

It is easy to view this as a multi-commodity flow problem with $K+1$ commodities. There are the original K commodities and an additional commodity between a and t. The size of the demands for the first K commodities are known. We perform a bisection search to determine the largest value of the commodity $K+1$ that still results in a feasible routing for the first K commodities. In order to develop an efficient algorithm it is better to formulate the problem as a maximum concurrent flow problem and perform the bisection search for this problem instead. We do not give the details of the solution procedure. In the case of the flow flushing algorithm, the link flows for the example in Figure 4 are shown in Figure 5 and the corresponding routing is shown in Table III.

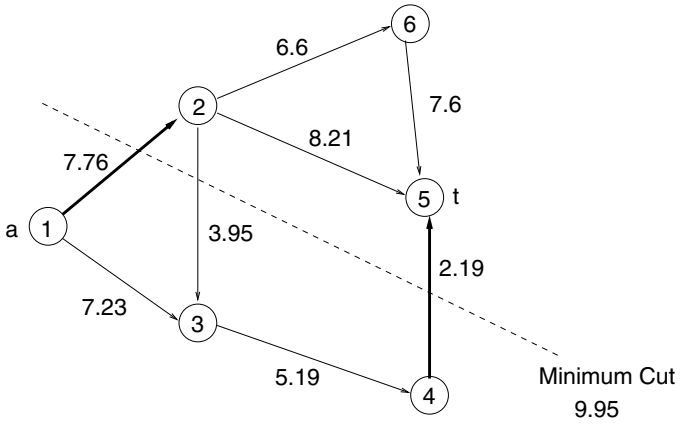

Fig. 5. Flow Flushing Algorithm

The maximum flow $M_{at}(f)$ on this network is 9.95 units. The value of the game $\theta = 5/9.95$. We now outline another

Source Dest. Pair	Paths	Flow
1-3	1-3	5.00
1-4	1-2-3-4	0.76
	1-3-4	2.23
1-5	1-2-5	7.00
2-5	2-5	1.21
	2-6-5	6.60
	2-3-4-5	2.19
2-3	2-3	1.00
6-5	6-5	1.00

TABLE III
FLOWS FOR BASE CASE

heuristic that can be used by the service provider to improve the probability of detection of the malicious packet.

B. Cut Saturation Algorithm

This algorithm relies on the fact that the maximum flow between a and t is upper bounded by the size of any $a-t$ cut. Let C represent the set of links in some $a-t$ cut. Given any link $e \in E$, let $\alpha(e)$ and $\beta(e)$ represent the start and end nodes of that link. The cut saturation algorithm picks some $a-t$ cut and tries to direct flow away from this cut. Once the source-destination demands are routed, this cut will be small and hence will limit the maximum $a-t$ flow. This is done as follows: Introduce two new nodes s' and t'. Introduce an arc between node s' and all nodes $\alpha(e)$ for all $e \in C$. Similarly introduce links between each node $\beta(e)$ for each $e \in C$ and the node t'. The objective now is to determine the highest flow that can be sent from s' to t' while maintaining the feasibility of routing the source-destination demands. The modification of the network is shown in Figure 6. The only links shown in the network are the cut links.

This problem can be solved almost identically to the Flow Flushing Algorithm, except that the $K+1$ commodity flows go between nodes s' and t'. One way of choosing the cut that is to be saturated is as follows: Assume that we currently have a routing of the source-destination demands resulting in a flow of $f(e)$ on link e. Determine a minimum $a-t$ cut (using these flows f as the capacities). Take this cut to be C and now attempt to saturate this cut. Continuing the example in Figure 4, assume that the cut that we saturate comprises of the links $(1,2)$ and $(4,5)$. The links flows are shown in Figure 7 and the corresponding flows are shown in Table IV.

The maximum flow $M_{at}(f)$ on this network is 8.0 units. The value of the game $\theta = 5/8$. Therefore, in this example, the cut saturation algorithm gives a better solution that the flow flushing algorithm.

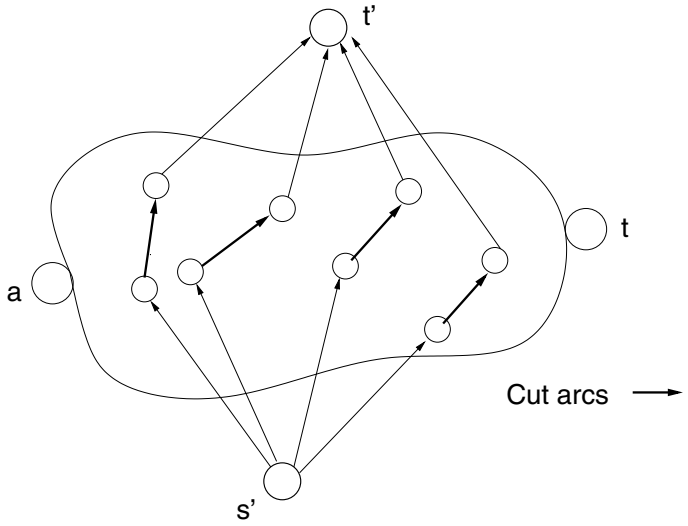

Fig. 6. Cut Saturation Algorithm Network Set-up

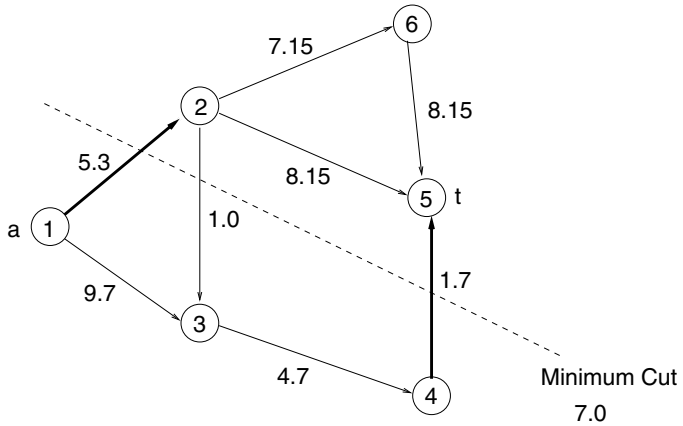

Fig. 7. Cut Saturation Algorithm

V. VARIANTS AND EXTENSIONS

We consider several variants of the problem outlined above. The first variant that we consider is the case where the intruder can introduce the malicious packet at one of a set of nodes $A \subset N$. We assume that $t \notin A$. The second variant that we consider is the case where the objective of the intruder is to reach any one of of a set of nodes $T \subset N$. We assume that $A \cap T = \emptyset$. Both these cases are easy to solve by introducing a super source node that is connected to all nodes in A and connecting all nodes in T to a super sink node. The game is now played between the super source node and the super sink node. Another variant is the case where the intruder can introduce the packet at any one

Source Dest. Pair	Paths	Flow
1-3	1-3	5.00
1-4	1-3-4	3.00
1-5	1-3-4-5	1.70
	1-2-5	5.30
2-5	2-5	2.85
	2-6-5	7.15
2-3	2-3	1.00
6-5	6-5	1.00

TABLE IV
FLOWS FOR BASE CASE

of a set of nodes A but we assume that the intruder does not have control of the routing in the network. Instead, we assume that the routing in the network is shortest path routing like in OSPF or IS- IS. We term this a shortest path routing game.

A. Shortest Path Routing Game

We now consider the problem where the routing in the network is along shortest paths. We assume that each link has a length and packets are routed from the source to the destination along shortest paths according to this length metric. We assume that ties are broken arbitrarily. Therefore given any two nodes in the network, there is a unique path from one node to the other. Given a target node, all packets arriving at this node traverse the shortest path tree. Shortest path routing implies that there is a unique tree rooted at the destination. A packet introduced at any node in the network traverses the unique path from that node to the destination along the links in the shortest path tree. We use A to represent the set of nodes that the intruder can introduce a malicious packet into the network. The objective of the intruder is to determine which node of this set A to introduce the packet into and the objective for the service provider is to determine the sampling rate at the links subject to a sampling budget of B. The main difference between this problem and the problem that we originally studied is the fact that it is easy to compute the maximum flow and hence the minimum cut on a tree. The algorithm for solving this problem is the following:

- Eliminate all leaf nodes in the routing tree that do not belong to A. Let T represent this tree. Let $P(i)$ represent the predecessor of node i on T.
- Set $L(i) = \infty$ for all leaf nodes.
- While there are no leaf nodes do
  - Pick a leaf node i. Let e be the edge connecting i to $P(i)$. Set $L(P(i)) \leftarrow L(P(i)) + \min\{L(i), f_e\}$.
- Output $L(t)$.

Note that $L(d)$ represents the maximum flow that can be sent from all the nodes in A to the destination node d. The value of the game is $B/L(d)$.

VI. EXPERIMENTAL RESULTS

In this section we evaluated the algorithms developed on two networks. The first network is shown in Figure 8. Each undirected link in the figure represents two directed links each having a capacity of 10 units. We performed the following experiments:

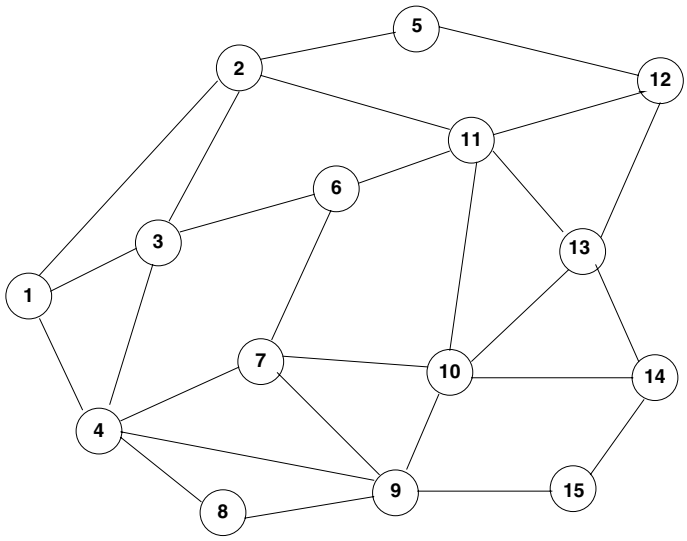

Fig. 8. Experimental Network 1

- Single attack node and single target node. (3 problems).
- Multiple attack node and single target node. (1 problem).
- Multiple attack node and multiple target node. (1 problem).

For each of the cases, we ran three different algorithms.

1) Routing to minimize the highest utilized link with f_1 representing the m-vector of link flows as a result of this routing algorithm.
2) Routing with flow flushing algorithm with f_2 representing the m-vector of link flows as a result of this routing algorithm.
3) Routing with cut saturation algorithm with f_3 representing the m-vector of link flows as a result of this routing algorithm.

Let $M(f_i)$ for $i = 1, 2, 3$ represent the maximum flow that can be sent from node a to t using f_i as the link capacities. If B is the sampling budget, then the value of the game $\theta = B/M()$. Table V shows the values of $M()$ instead of θ. The smaller that value of M, the better the chances of detection for a given sampling budget.

Attack Node(s)	Target Node(s)	$M(f_1)$	$M(f_2)$	$M(f_3)$
1	13	13.2	8.9	9.1
5	7	9.2	7.55	7.55
7	11	16.4	7.3	7.1
1,2,4,8	13	16.2	9.4	8.7
1,2,4,8	12,13,14	24.88	19.5	18.9

TABLE V
COMPARISON OF DIFFERENT ROUTING ALGORITHMS

From the table, note that the maximum flow value and hence the value of the game can be changed significantly by changing the routing in the network. In most of the examples the performance of the flow flusing algorithm and the cut saturation algorithm are quite similar, and better than the simple minimization of maximum link utilization algorithm

A. *Effect of Capacity on the Value of the Game*

As the amount of spare capacity in a network increases, the opportunity to reroute flows increases. This implies that the service provider can improve the probability of detection by exploiting the spare capacity to reroute flows. We illustrate this in the following set of experiments, using a second example network, where the capacity of the links in the example network are fixed at some constant value C. If the value of C increases, then the opportunity to reroute flows goes up. We consider the intrusion detection game between nodes $a = 1$ and $t = 13$. The demands in the network are uniformly distributed between zero and one. We first run the algorithm to route the flow such that maximum utilization of any link is minimized. This maximum utilization value versus the link capacity C is shown in Figure 9. As the maximum utilization becomes lower, the amount of spare capacity to reroute flows increases in the network. This implies that both the flow flushing algorithm as well as the cut saturation algorithm will have more alternate paths. In Figure 10, we show the performance of the flow flushing algorithm as the value of C increases. The straight line in the plot shows the performance of the base case which is the routing algorithm that minimizes the maximum utilization. The $a - t$ maximum flow is independent of the value of C. In the case of the flow flushing algorithm, the $a - t$ maximum flow value decreases with increasing link capacity. It asymptotes at about 8.8. The same kind of performance was observed in the case of other attack-target pairs as well as the case for multiple attack sites.

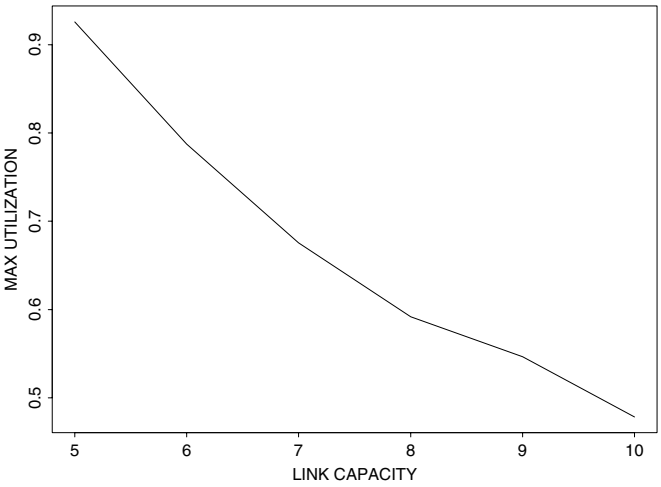

Fig. 9. Max. Utilization vs. Link Capacity for flow routing to minimize maximum link utilization.

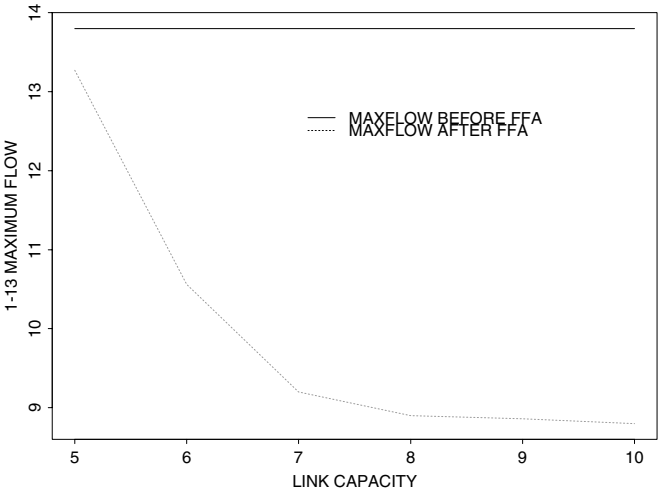

Fig. 10. Performance of flow flushing algorithm for different link capacities

VII. CONCLUDING REMARKS

We considered the problem of detecting intruding packets in a network by means of network packet sampling. Since packet sampling and examination in real-time can be expensive, the network operator must devise an effective sampling scheme to detect intruding packets injected into the network by an adversary. We considered the scenario where the adversary has considerable information about the network and can either pick paths to minimize chances of detection or can pick a suitable network ingress-point if only shortest path routing is allowed. The detection via sampling problem was formulated in a game-theoretic framework. The solution to this game-theoretic problem is a max-flow problem from which the stable operating points are obtained. We also considered the network op-

erator's problem of routing aggregate traffic between ingress-egress pairs as to to maximize the chances of detection within a given packet sampling budget. We proposed two heuristic algorithms for solving this problem. Finally, we evaluated the performance of the developed algorithms on some sample networks.

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, 1993.

[2] Akella, A., Karp, R., Papadimitriou, C., Seshan, S., Shenker, S., "Selfish Behavior and the Stability of the Internet: A Game Theoretic Analysis of TCP", *Proceedings of SIGCOMM 2002*, 2002.

[3] Duffield, N., Greenberg, A., Grossglauser, M., Rexford, J., "A Framework for Passive Packet Measurement". IETF Draft, work in progress, draft-duffield-framework-papame-01, February 2002.

[4] Garg, N., and Könemann, J., "Faster and Simpler Algorithms for Multicommodity Flow and other Fractional Packing Problems", *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pp.300-309, 1998.

[5] Korilis, Y., Lazar, A., Orda, A., "Architecting Noncooperative Networks", *IEEE Journal on Selected Areas in Communications*, pp. 1241-1251, September 1995.

[6] Ott, T. J., and Lakshman, T. V., and Wong, L. H., "SRED: Stabilized RED", *Proceedings of Infocom 1999*, pp. 1346-1355, 1999.

[7] Pan, R., Prabhakar, B., Psounis, K., "CHOKE, A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation", *Proceedings of Infocom 200*, pp. 942-951, 2000.

[8] Owen, G., *Game Theory*, Academic Press, New York.

[9] Shahrokhi, F., and Matula, D., "The Maximum Concurrent Flow Problem", *Journal of the ACM*, 37, pp. 318-334, 1990.

[10] Shenker, S., "Making Greed Work in Networks: A Game-Theoretic Analysis of Switch Service Disciplines", *IEEE/ACM Transactions on Networking*, 1995.

[11] Washburn, A., and Wood, K., "Two-Person Zero-Sum Games for Network Interdiction", *Operations Research*, 43, pp. 243-251, 1995.