Stochastic Security in Wireless Mesh Networks via Saddle Routing Policy

Yanwei Wu *
Dept. of Computer Science
Illinois Institute of Technology
Chicago, IL 60616, USA
xli@cs.iit.edu

Xiang-Yang Li †
Dept. of Computer Science
Illinois Institute of Technology
Chicago, IL 60616, USA
xli@cs.iit.edu

WeiZhao Wang
Google Inc
weizhao@google.com

ABSTRACT

The security problem in multihop wireless networks is more severe than that in wired networks since its transmission media is the unprotected air. In this paper, we show how to increase the effective throughput via carefully choosing the multi-path routing for given source and destination nodes, where we call the total packets from the mesh routers to the gateway nodes that are not attacked by an attacker as effective throughput. We assume that the attacker has limited resources for attacking while attacking a node or a link will incur some certain costs. We show that it is NP-hard to find an optimum multipath routing policy even if the attacking strategy is given a prior. We model the problem as a two-player game between the routing policy designer and the attacker and propose a randomized multi-path routing protocol that achieves good effective throughputs under several possible attacking scenarios. More specifically, we theoretically prove that our routing protocols can achieve an effective network throughput (with packets which are not attacked) within a constant factor of the optimum in the worst case. Our theoretic results are confirmed by extensive simulations studies.

Keywords

Wireless ad hoc networks, security, saddle policy, stochastic routing, scheduling, optimization

1. INTRODUCTION

Wireless networks draw lots of attentions in recent years due to their potential applications in various areas, especially the wireless mesh network. A wireless mesh network is often used at the edge of the wired network to extend the wired network. Many US cities (*e.g.*, Medford, Oregon; Chaska, Minnesota; and Gilbert, Arizona) have already deployed mesh networks. These networks behave almost like wired networks since they have infrequent topology changes, limited node failures, *etc.*. Wireless mesh network is decentralized, reliable because of its mesh topology. Each node forwards the packet to its neighbor so as to relay the packet to the distant target node. If one of the neighbor fails, it will use another neighbor node instead. In addition, infrequent topology changes and limited node failure lead to the reliability of the mesh network. Even though, wireless mesh networks still have the same problems existing in wireless networks, e.g., signal interference because of sharing the channel: the radio sent out by a wireless node will be received by all the nodes within its transmission range, and also possibly causes signal interference to some other nodes that are not intended receivers. Thus, interference becomes one of the major problems which dramatically reduce the throughput in the wireless network.

Another challenging problem facing wireless networks is network security, such as eavesdropping of the wireless signal, jamming the wireless channels, intentionally dropping or inserting packets by some malicious attacks. However these problems may exist in wired networks, they are more severe in wireless networks than in the wired networks because in wireless networks the transmission media is unprotected air. We mainly define two kinds of malicious behaviors in wireless networks: non-packet-dropping attack and packet-dropping attack. In a non-packet-dropping attack, the attacker will not throw away the packets, instead the attacker will eavesdrop the packets in some nodes or links to aggregate the information so as to attack the target nodes or perform traffic analysis, or the attacker may modify some packets so as to destroy the information for the target nodes. A potentially much worse attacking is packet-dropping or packet-inserting attacking. The attacker either inserts malicious packets to jam the network transmission or drops some packets in the network so as to reduce the effective throughput achieved by target nodes. When the attacker has infinite resource and the system protector has bounded resource, the attacking will always be successful with high probability. Fortunately, this is not true in practice. The attackers always have limited resources and technologies. Thus, the attacker can only attack a part of the information in the network successfully. The routing policy maker (*i.e.*, system protector) wants to design algorithms and protocols to reduce the effect of malicious behaviors in the networks, or to avoid them finally. Notice that it is possible because the attackers only have a bounded resources and technologies.

There are a number of algorithms (*e.g.*, [8, 16, 19–21]) proposed in the literature to achieve various security methods in wireless networks. Among them, some aim to prevent or reduce the effect of the malicious attack. The stochastic routing is provided for this purpose. In traditional shortest path routing, the shortest path from the source node to the target node is always chosen in high probability to route the packet. Thus it is easy to predict the routing path if the routing protocol and the link or node cost are known to all, which results in the network prone to be attacked. The stochastic routing is designed to reduce the probability of successful prediction. Suppose the links chosen for routing will not form a cycle. In a stochastic routing, for each link e incident to a node u there is some probability p_e for the link $e = (u, v)$ to be chosen for routing. Clearly we need $\sum_{v \in V} p_{(u,v)} = 1$, $\forall u \in V$. Here (u, v) denotes a link from node u to node v. In this way, the probability that the attacker correctly chooses the routing path will be greatly reduced. The challenge now is to choose the probability p_e ap-

*The author is funded in part by National Science Foundation grant NSF CCF-0515088.

†The work of the author is partially supported by NSF CCR-0311174.

propriately so that we can guarantee certain security performance under *all* possible attacking strategies using given limited attacking resources. Recently, Bohacek *et al.* [19, 20] studied such problem for the traditional wired networks using a two persons' zero-sum game model.

Wireless networks pose some additional challenges and also additional opportunities for designing a saddle-routing policy. The challenge comes from the fact that wireless interference often makes an optimal routing problem NP-hard while the counterpart problem in the wired networks is polynomial time solvable. A typical example of such problems is to find the largest throughput using a multi-path routing between a pair of nodes; see [1, 23] for details. In this paper, we consider a multi-hop multi-channel wireless networks and assume that the routing policy maker can jointly optimize the multi-path routing, the link and channel scheduling. We assume that each node only has one radio as the majority wireless nodes only have one networking interface card. For link scheduling, we consider a synchronized TDMA since this will achieve more throughput than the CSMA contention-based approach [1, 10, 23]. Notice that TDMA based link scheduling has some implementation overhead and difficulties such as time synchronization among nodes. However, we adopt the TDMA link channel scheduling since we want to study what is the *best* scheduling that the system can achieve under the *worst* attacking scenario. To simplify our study, we assume that the attacker can know the strategy used by the routing policy maker, and vise versa. We also assume that both the attacker and the policy maker can efficiently compute their own benefits, given the attacking strategies of attacker and the routing (and scheduling) strategies of the policy maker.

Our main contributions of this paper are as follows. We mathematically formulate the joint routing, the link and channel scheduling problem by the policy maker under the possible attacks. We show that, unlike the wired network counterpart [19, 20], it is NP-hard for the policy maker to find an optimal routing strategy, given the attacking strategy. We then provide a relaxed linear programming formulation and provide a joint routing, link and channel scheduling such that the achieved effective throughput is within a constant factor of the optimum, under the *worst* possible attacks by attacker. We also studied the stability of the found strategy pair (routing strategy $\overline{\ell}$, attacking strategy $\overline{\alpha}$). We show that given $\overline{\ell}$, the attacker cannot find better attacking strategy other than $\overline{\alpha}$. When $\overline{\alpha}$ is given, we also prove that the policy maker cannot find a routing strategy that can achieve a significantly larger effective throughput. We conduct extensive simulations to study the performance of our proposed protocols. We found that our solution is also stable respect to the attacking budget: the routing strategy does not need to change if the budget B only increases by a small amount.

The rest of the paper is organized as follows. In Section 2, we present our network model, discuss the problems to be studied, show how to compute the equilibrium value for saddle attack in wireless networks. We discuss in detail our approaches in Section 3. We report our simulation results that compare the performance of our methods with existing routing methods in Section 4 and review the related work in Section 5. We conclude our paper in Section 6.

2. PRELIMINARIES AND PROBLEM FORMULATION

This paper studies two important issues in wireless networks, namely, the routing security and the network throughput, by carefully dealing with wireless interference. Wireless interference issues have been studied extensively recently because it is widely believed that reducing the interference can increase the overall throughput of a wireless network. There are different approaches to reduce the interference, *e.g.*, the scheduling on the MAC layer, route selection on the routing layer, channel assignment if multi-channels are available, and power control on the physical layer. Signal jamming, dropping packets, and eavesdropping packets will also reduce the effective network throughput. In this section, we first discuss in detail the interference models we will use and formally define the problem that we will study in this paper.

2.1 Network System Models

We assume that there is a set $V = \{v_1, \ldots, v_n\}$ of n wireless nodes. Every node v_i has a transmission range $R_T(i)$, such that the necessary condition for a node v_j to receive correctly the signal from v_i is $\|v_i - v_j\| \leq R_T(i)$, where $\|v_i - v_j\|$ is the Euclidean distance between v_i and v_j. However, we assume that $\|v_i - v_j\| \leq R_T(i)$ is *not* the sufficient condition for $(v_i, v_j) \in E$. Some links do not belong to G because of either the physical barriers or the selection of routing protocols. We always use $\mathbf{L}_{i,j}$ to denote the directed link (v_i, v_j) hereafter. Each node v_i also has an interference range $R_I(i)$ such that node v_j is interfered by the signal from v_i whenever $\|v_i - v_j\| \leq R_I(i)$ and v_j is not the intended receiver. Typically, $R_T(i) \leq R_I(i) \leq c \cdot R_T(i)$ for some constant c. For all wireless nodes, let $\gamma = \max_{v_i \in V} \frac{R_I(i)}{R_T(i)}$. The complete communication graph is a *directed* graph $G = (V, E)$, where E is the set of possible *directed* communication links. Let $\Delta^-(u)$ (resp. $\Delta^+(u)$) denote the set of directed links that end (resp. start) at node u, *i.e.* $\Delta^-(u) = \{(w, u) \mid (w, u) \in E\}$ and $\Delta^+(u) = \{(u, v) \mid (u, v) \in E\}$.

We assume that the wireless network is multi-hop, multi-channel network. Let $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_K\}$ be the set of K orthogonal channels (typically frequency channels or CDMA codes) that can be used by all wireless nodes. For example, for 802.11 networks, $K = 11$. Each wireless node u is equipped with $\mathcal{I}(u) \geq 1$ radio interfaces. For simplicity, we assume that $\mathcal{I}(u)$ is same for all nodes. In this paper, most of studies assume that $\mathcal{I}(u) = 2$ for simplicity. Other than in the literature (*e.g.*, [1, 9, 10]) we assumed that a wireless interface card can operate on 2 channels from $\mathcal{F}$. For notational convenience, we use $\mathcal{F}(e)$ to denote the set of common channels that can be used by link e. For each link $e = (u, v)$ operating on a channel $\mathbf{f} \in \mathcal{F}(e)$, we denote by $\mathbf{c}(e, \mathbf{f})$ the rate for link e. This is the maximum rate at which a mesh router u can communicate with the mesh router v in one-hop communication using channel $\mathbf{f}$. Notice that the links are directed, thus, the capacity could be asymmetric, *i.e.*, $\mathbf{c}((u, v), \mathbf{f})$ may not be same as $\mathbf{c}((v, u), \mathbf{f})$.

We also assume that among the set V of all wireless nodes, some of them have gateway function and provide the connectivity to the Internet. For simplicity, let $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_g\}$ be the set of g gateway nodes, where $\mathbf{s}_i$ is actually node v_{n+i-g}. All other wireless routing nodes v_i (for $1 \leq i \leq n - g$) are called *ordinary* wireless nodes. We assume that the gateway nodes will *not* act as relay node for a pair of ordinary wireless nodes. Each ordinary mesh routing node u will aggregate the traffic from all its users and then route them to the Internet through some gateway nodes. We use $\ell_O(u)$ to denote the total aggregated outgoing traffic for its users by node u and $\ell_I(u)$ to denote the total aggregated incoming traffic for its users by node u. We will mainly concentrate on one of the traffics in this paper, say incoming traffics. For notation simplicity, we use $\ell(u)$ to denote such load for node u. Notice that the traffic $\ell(u)$ is not requested to be routed through a specific gateway node, neither requested to be using a single routing path. Our results can be easily extended to deal with both incoming and outgoing traffic by defining routing flows for both traffics separately.

2.2 Interference Models

In this paper we assume TDMA is used for link transmission. To schedule two links at the same time slot, we must ensure that the schedule will avoid the interference. Two different types of interference have been studied in the literature, namely, *primary interference* and *secondary interference*. In addition to these interferences, there could have some other constraints on the scheduling, *e.g.*, the radio networks that deploy the IEEE 802.11 protocol with request-to-send and clear-to-send (RTS/CTS) mechanism will pose some additional constraints. When using RTS/CTS mechanism, a transmitter first sends a RTS frame before sending a data frame. The intended receiver then responds with a CTS frame indicating that the transmitter can send the data frame. For every pair of transmitter and receiver, all nodes that are within the interference range of either the transmitter or the receiver cannot transmit. Although RTS/CTS is not the interference itself, for convenience of our notation, we will treat the communication restriction due to RTS/CTS as *RTS/CTS interference* model.

Several different interference models have been used to model the interferences in wireless networks. Some most widely used interference models are Protocol Interferences Model (PrIM) [3], Fixed Protocol Interferences Model (fPrIM) [23], RTS/CTS Model, [1], and Transmitter Interference Model (TxIM) [24].

2.3 Network Routing Game

Besides the network throughput, security is another important issue we ought to emphasize in designing wireless mesh networks. Generally, we assume that there are two players: the *Routing Policy Maker* and the *Attacker(s)*. The attacker tries to reduce the number of packets received by the target nodes by destroying packets, dropping packets, inserting garbage packets to the network, eavesdropping packets and so on. While the routing policy maker tries to design a routing policy to reduce or to prevent such malicious attack so that the effective network throughput is improved. Generally, we assume that the attacker can attack some nodes of the network (thus it has the total control of the node, which implies that all packets passing through this node are attacked), or the attacker can attack a portion of the packets passing through some links. We define two kinds of attacks in the network, the *non-packet-dropping attack* and the *packet-dropping attack*. In addition, we classify the packets in the network as *dirty packet* and *healthy packet*. We call a packet received by a target node as *dirty packet* if it is modified, eavesdropped. For simplicity, sometimes, we call an intentionally dropped packet also as a dirty packet. While we refer the packet received by the target node, which was not be modified, eavesdropped, as healthy packet.

In a non-packet-dropping attack, the attacker tries to eavesdrop some nodes or some links, which results in the leakage of the packet information, or the attacker tries to modify some packets which results in useless packets for target node. The routing policy is thus designed to reduce the number of the dirty packets produced by the attacker. If the attacker has abundant resources to support the attacking behavior, *i.e.*, it has enough resources to attack every node or every link in the network, the information in the network will be certainly filched without considering the encryption technique in the network. However, it is usually not the reality. The attacker can only *selectively* attack part of the nodes or links in the network because of the limited resources. Under a restricted budget B, the attacker will optimize its attacking strategy to maximize the number of packets attacked per unit time. Here we adopt the following simple cost model for attacking: the cost of eavesdropping a link e with capacity C_e is $\alpha_e \cdot C_e \cdot H_e$, where $\alpha_e \in [0, 1]$ is the effort the attacker put on the link e and H_e is the cost to attacking one unit of data. The budget constraint requires $\sum_{e \in E} \alpha_e \cdot C_e \cdot H_e \leq B$.

The aim of the routing policy maker here is to schedule the flow $\ell(e)$ on every link e in the network so that the number of dirty packets (eavesdropped, modified, inserted, removed) are as small as possible. Notice that the routing policy maker is also under the limited resources, such as the constrained capacity, the real-time computation and so on. Thus, it is possible that the policy maker could not find the best strategy in polynomial time when knowing the attacking strategy. Recall that for wireless networks, the flow $\ell(e)$ should be scheduled for transmitting using TDMA without causing interference among simultaneous transmitting links. It is well-known that, it is NP-hard to find a flow assignment $\ell(e)$ for wireless mesh networks that will maximize the network throughput, even without the existence of attacking [1, 10, 11, 23]. In this paper, we will show how the policy maker can find an almost optimal routing strategy such that the achieved effective network throughput is within a constant factor of optimum.

In packet-dropping attack, the attacker will insert some malicious packet, jam the network, or drop the packets in the network so as to minimize the healthy packets flowing into the target. While the routing policy maker is to minimize the effect of the malicious attack in the network. That is, the routing policy maker wants to maximize the healthy packets flowing into the target node. Thus, we can model the strategies by attacker and routing policy maker as a zero-sum game: the benefit gained by one party is the benefit lost by the other party.

For simplicity, we assume that the attacking on different links are independent although this model is little bit idealistic. Notice that in practice, when an attacker eavesdrops some link, it often could eavesdrop the packets from several links nearby. Our study here is the first step to fully study the malicious attack. It provides foundations for studying the case when the attacking costs are non-independent, *e.g.*, there is a given cost $c(S)$ for attacking some subset of links $S \subseteq E$. Then the attacker needs to distribute its resource onto attacking the subset of links $S_1, S_2, \cdots, S_k$.

We always assume that the resource of the attacker cannot attack any *cut* of the network between the source and the target nodes. Here a *cut* of a graph is a set of links whose removal will disconnect the target node from the source node. If this is not the case, the attacker can clearly put its resource to attack all links in a cut and thus all packets will be attacked. Under the assumption that no cut of the network can be fully attacked, it is easy to show that the attacker cannot fix the set of links to attack; similarly the routing policy-maker cannot fix a routing path connecting the source and the target node. Assume that the action of the opposite-one is fixed and can be predicted, the other side can choose the best action for it, *e.g.*, the routing policy-maker can also use a path avoiding the attacked links when attacker fixed the set of links to attack. Thus, instead of deciding on a definite action, the two players will assign probabilities to their respective actions, which leads to a linear programming problem with a unique solution for each player. We suppose that the two players, the attacker and the routing policy maker, both know the payoff matrix and can maximize their benefits by their strategies. The policy-maker assumes a possible distribution of attacking strategy and then makes the best routing decision; the attacker will then find the best attacking strategy based on the flow from history information; the policy-maker will adjust its flow based on observed attack, and so on. After several iterations, the two players will finally find an *Equilibrium*, which is proved by the theorem that $\min\max$ is equal to $\max\min$ in [8].

Also notice that our study here assumes that the game is one-shot: Saddle routing policy is to compute one attacking policy and one corresponding routing policy which is best for both attacker

and policy-maker, *i.e.*, the Nash Equilibrium of the game. If the game is played repeatedly, we would like to study the subgame perfect equilibrium (SPE) of the game.

DEFINITION 1. *Given an attacking strategy α and a routing flow ℓ by the routing policy maker, the* effective throughput $\mathcal{T}(\alpha, \ell)$ *is defined as the number of healthy packets received at the target node that are initiated by the source node.*

It is not difficult to compute such throughput $\mathcal{T}(\alpha, \ell)$ when α and ℓ are given. When the attacking strategy α is given, the routing policy maker has to find the best routing ℓ that will maximize the effective throughput. Let $\mathcal{T}_{OPT}(\alpha)$ be the effective throughput achieved when the routing policy maker uses the best routing strategy, given the attacking strategy α. Let $\mathcal{T}_{\mathcal{A}}(\alpha)$ be the effective throughput achieved when the routing policy maker finds its responding routing strategy to the given attacking strategy α using a method $\mathcal{A}$. Similarly, we define $\mathcal{T}_{OPT}(\ell)$ and $\mathcal{T}_{\mathcal{B}}(\ell)$ when the flow routing ℓ is decided in advance, where $\mathcal{B}$ is a method to find an attacking strategy by attacker when given ℓ. We say that a routing algorithm $\mathcal{A}$ is *c-efficient* (or within a constant c factor of optimum) if $\mathcal{T}_{\mathcal{A}}(\alpha) \geq c \cdot \mathcal{T}_{OPT}(\alpha)$. Notice that the routing algorithm $\mathcal{A}$ will assign flows $\ell(e)$ to each link e such that the flow can be scheduled by all links without interference among simultaneous transmissions.

2.4 Mathematical Formulation

When an attacker attacks a link, we can assume that it can choose its attacking effort $\alpha \in [0, 1]$. Depending on the effort, it will incur some cost (which is also related to the node or link), and it will have a success probability p. We can assume that the probability p is a linear function of α. So we can assume the success probability is still α after skipping the constant between them. We can also assume that the cost now is a function of effort α and the link capacity $C_e = \sum_{\mathbf{f}} \mathbf{c}(e, \mathbf{f})$. That is, the cost is $\alpha_e \cdot C_e \cdot H_e$ for link e, where H_e is the cost to attack per unit data. The total cost for the attacker is under some limited budget, $\sum_e \alpha_e \cdot C_e \cdot H_e \leq B$.

2.4.1 Interference-free Scheduling

In wireless network, the routing policy maker need also consider interference in the network. It should guarantee that links with interference will not be scheduled simultaneously while using the same channel. Let $X_{e,t,\mathbf{f}} \in \{0, 1\}$ be the indicator variable which is 1 only when e will transmit at time t using a channel f. Let e' be a link which will cause interference with link e when transmit at time t using the same channel f. Clearly, we need $X_{e,t,f} + X_{e',t,f} \leq 1$. Under a schedule $X_{e,t,\mathbf{f}}$, $\frac{\sum_t X_{e,t,f}}{T} \cdot C_{e,f} = \ell_{e,f}$ defines the flow that can be supported by a link e using channel $\mathbf{f}$. According to recent results [1, 10, 23], we know that $X_{e,t,f} + \sum_{e' \in I_{\mathcal{M}}(e)} X_{e',t,f} \leq A, \forall \mathbf{f}, e$ for a certain constant A depending on the interference model $\mathcal{M}$ used. Here $I_{\mathcal{M}}(e)$ is the set of links that will conflict with link e when they are scheduled at the same time slot using the same channel and satisfy some additional constraints depending on the interference model $\mathcal{M}$. For example, for the protocol interference model PrIM, $I_{\mathcal{M}}(e)$ is the set of links (1) that cannot be simultaneously transmitting with link e, and (2) whose Euclidean length is longer than that of e. We summarize the constraints specifying the necessary conditions on scheduling as following

$$\begin{cases} X_{e,t,f} + X_{e',t,f} & \leq 1 \quad \forall e, t, \mathbf{f} \\ \frac{\sum_t X_{e,t,f}}{T} \cdot C_{e,f} & = \ell_{e,f} \quad \forall e, \mathbf{f} \\ X_{e,t,f} + \sum_{e' \in I_{\mathcal{M}}(e)} X_{e',t,f} & \leq A \quad \forall \mathbf{f}, e \end{cases}$$

It was proved in [1, 10, 23] that, for a number of interference models, all TDMA *schedulable flows* should satisfy the above constraints. The constant A depending on the interference model underneath. Notice that the necessary condition for schedulable flow only characterizes what kinds of flows that cannot be scheduled; it does not provide methods to schedule a schedulable flow. Notice that here a flow f is TDMA schedulable if there is a time-slots assignment for every link e on the network such that (1) the achieved flow on each link e is $f(e)$, and (2) the time-slots assignment are interference free.

Given a load assignment $\ell(e)$ to every link e on the network, we would like to know whether we can schedule the transmissions of the links and channels such that the load $\ell(e)$ is achieved. Unfortunately, this has been shown to be a NP-complete problem, see *e.g.* [7]. In [1, 10, 23], various sufficient conditions are proposed for a schedulable flow and various efficient scheduling methods are designed to schedule a flow that satisfies a sufficient condition for schedulable flow. Generally, it was proved that $X_{e,t,f} + \sum_{e' \in I_{\mathcal{M}}(e)} X_{e',t,f} \leq 1, \forall \mathbf{f}, e$ is a sufficient condition for a schedulable flow. In the rest of the paper, we will adopt such sufficient condition to characterize schedulable flow.

Define $\beta_{e,f} = \frac{\sum_t X_{e,t,f}}{T}$ as the usage ratio of link e for channel f in the time period T. We transform the formula for interference-free schedulable flow ℓ to the formula as follows. If a flow assignment $\ell(e)$ is schedulable, then there must exist a positive constant A (say $C_{\mathcal{M}}$ depending on the interference model $\mathcal{M}$ used), and some non-negative real numbers $1 \geq \beta_{e,\mathbf{f}} \geq 0$ such that

$$\begin{cases} \sum_{\mathbf{f} \in F(e)} \beta_{e,\mathbf{f}} \cdot C_{e,\mathbf{f}} & = \ell_e \quad \forall e \\ \beta_{e,\mathbf{f}} + \sum_{e' \in I_{\mathcal{M}}(e)} \beta_{e',\mathbf{f}} & \leq A \quad \forall e, \mathbf{f} \end{cases}$$

On the other hand, it was proved in [1, 10, 23] that, for a number of interference models, when $A = 1$, any solution of $\beta_{e,\mathbf{f}}$ satisfying the above constraints (2.4.1) implies that there is an interference-free schedule to implement the routing flow. Notice that in practice, we found that even A is some integer larger than 1, but smaller than $C_{\mathcal{M}}$, the solution of $\beta_{e,\mathbf{f}}$ still results in some interference-free link schedule. Thus, in the rest of our paper, we will generally assume that there is an integer A specifying a necessary condition for schedulable flow.

2.4.2 Non-packet-dropping Attacking Formulation

Non-packet-dropping attack is that the attacker chooses to eavesdrop or modify some packets in part of nodes or links without dropping them or injecting new garbage packets to the network. In this model, the attacker attempts to obtain maximal information by eavesdropping certain number of packets or destroy the message in the network by modifying the packets while the routing policy maker tries to prevent the packets from being eavesdropped or modified. Because the two players are rational, they always choose the solution which will mostly benefit themselves. Accordingly, the routing policy maker always supposes that the attacker will choose the attack distribution α (when given the routing $\ell(e)$) which can maximize the flow eavesdropped or modified under certain link scheduling, *i.e.*,

$$\max_{\alpha} \sum_{e \in E} \alpha_e \cdot \ell_e.$$

Here we assume that the total number of dirty packets under attacking strategy α is $\sum_{e \in E} \alpha_e \cdot \ell_e$. Notice that it is possible that an attacker may eavesdrop the same packer when attacking different links. Thus, $\sum_{e \in E} \alpha_e \cdot \ell_e$ is the maximum number of packets that can be eavesdropped. When an attacker can choose which packets to eavesdrop, then the number of different packets eavesdropped could be exactly $\sum_{e \in E} \alpha_e \cdot \ell_e$. For the routing policy maker, it will always choose a routing policy (when given the attacking strategy α) to minimize the number of packets being eavesdropped or modified, *i.e.*,

$$\min_{\ell} \sum_{e \in E} \alpha_e \cdot \ell_e.$$

Unfortunately, it is easy to show that, under this logic, the best

action by a routing policy maker is **not** to perform any routing at all (*i.e.*, $\ell_e = 0$ for every link e), which will result in 0 packets loss.

To avoid this trivial solution, we consider the other side of the scenario: the attacker will minimize the total flows that are not eavesdropped or modified, *i.e.*, $\min_\alpha \sum_{e\in E}(1-\alpha_e)\cdot \ell_e$. To make the attacker affect the least packets, the policy maker will choose a routing policy and a link schedule which will maximize the number of healthy packets in the network, *i.e.*, $\max_\ell \min_\alpha \sum_{e\in E}(1-\alpha_e)\cdot \ell_e$. This means that the policy maker will choose the a flow routing ℓ which provides the most healthy packets that are not eavesdropped or modified under the worst case attacking scenario. In the case of the attacker, the attacker considers that the policy maker always designs a routing method which will maximize the total healthy flow, *i.e.*, $\max_\ell \sum_{e\in E}(1-\alpha_e)\cdot \ell_e$. The attacker then will choose one attack distribution, α, to minimize the total healthy flow, *i.e.*, $\min_\alpha \max_\ell \sum_{e\in E}(1-\alpha_e)\cdot \ell_e$.

In this non-packet-dropping attacking model, each node u should satisfy the flow conservation

$$\sum_{e\in\Delta^+(u)} \ell_e - \sum_{e\in\Delta^-(u)} \ell_e = 0,$$

$\forall u \neq s, d$ where s is the source node and d is the destination node. The attack strategy should satisfy the resource constraint $\sum_{e\in E} \alpha_e \cdot C_e \cdot H_e \leq B$, where H_e is the cost of attacking a link e for one unit amount of data (using the same unit for link capacity such as Megabit). Remember that the total link capacity $C_e = \sum_{\mathbf{f}} \mathbf{c}(e, \mathbf{f})$, where $\mathbf{c}(e, \mathbf{f})$ is the capacity of link e using channel $\mathbf{f}$. Thus we summarize the formula for the non-packet-dropping attacking model as:

$$\textbf{Non-packet-dropping Model:} \quad \min_\alpha \max_\ell \sum_{e\in E}(1-\alpha_e)\cdot \ell_e, \ \textbf{s.t.} \quad (1)$$

$$\begin{cases} \sum_{e\in E} \alpha_e \cdot C_e \cdot H_e \leq B & \forall e \\ 0 \leq \alpha_e \leq 1 & \forall e \\ \sum_{f\in F(e)} \beta_{e,f} \cdot C_{e,f} = \ell_e & \forall e \\ \beta_{e,f} + \sum_{e'\in I_{\mathcal{M}}(e)} \beta_{e',f} \leq A & \forall e \\ \sum_{e\in\Delta^+(u)} \ell_e - \sum_{e\in\Delta^-(u)} \ell_e = 0 & \forall u \neq s, d \end{cases}$$

Notice that in the above LP formulation, the objective function $\min_\alpha \max_\ell \sum_{e\in E}(1-\alpha_e)\cdot \ell_e$ is the same as the objective function $\max_\ell \min_\alpha \sum_{e\in E}(1-\alpha_e)\cdot \ell_e$. Also notice that in all our formulations, we do not count the packets that could be eavesdropped multiple times by the attackers at different links or nodes: for simplicity we assume that the eavesdropped packets are different, thus, $\sum_{e\in E} \alpha_e \cdot \ell_e$ denotes the total number of packets that are dirty. Thus, in our model, the effective network throughput is

$$\sum_{e\in\Delta^-(d)} \ell_e - \sum_{e\in E} \alpha_e \cdot \ell_e.$$

Here $\sum_{e\in\Delta^-(d)} \ell_e$ stands for the number of total packets to be received by the target node d and $\sum_{e\in E} \alpha_e \cdot \ell_e$ denotes the number of eavesdropped packets among received packets. Consequently, the routing policy maker could try to maximize this value. In other words, the objective function for **Non-packet-dropping Model** could be replaced as

$$\min_\alpha \max_\ell \left(\sum_{e\in\Delta^-(d)} \ell_e - \sum_{e\in E} \alpha_e \cdot \ell_e \right) \quad (2)$$

Also notice that in our previous formulation, we do not assume that there is a traffic demand between the source and the target nodes for the routing policy maker. In some situations, it could be the case that for a source node s and a target node d, we request a data rate $\theta(s)$. The objective of the routing policy maker clearly is to maximize the number of packets from s that are not attacked (*e.g.*, eavesdropped by the attacker in the non-packet-dropping attacking model), while the objective of the attacker is to maximize the attacked packets per unit time or minimize the packets not attacked. In other words, $\min_\alpha \max_\ell \sum_{e\in E}(1-\alpha_e)\cdot \ell_e$ will be the objective function. Similarly, we can formulate the game between the policy maker and the attacker as the following joint optimization problem:

$$\textbf{Non-packet-dropping Model with Demand:} \quad \min_\alpha \max_\ell \sum_{e\in E}(1-\alpha_e)\cdot \ell_e, \ \textbf{s.t.} \quad (3)$$

$$\begin{cases} \sum_{e\in E} \alpha_e \cdot C_e \cdot H_e \leq B & \forall e \\ 0 \leq \alpha_e \leq 1 & \forall e \\ \sum_{e\in\Delta^-(s)} \ell_e - \sum_{e\in\Delta^+(s)} \ell_e \geq \theta(s) & \\ \sum_{f\in F(e)} \beta_{e,f} \cdot C_{e,f} = \ell_e & \forall e \\ \beta_{e,f} + \sum_{e'\in I_{\mathcal{M}}(e)} \beta_{e',f} \leq A & \forall e \\ \sum_{e\in\Delta^+(u)} \ell_e - \sum_{e\in\Delta^-(u)} \ell_e = 0 & \forall u \neq s, d \end{cases}$$

Observe that our formulations can be easily extended to deal with the situation when there are multiple source nodes and multiple target nodes. The simple trick is to assume that there is a virtual source node $\mathcal{S}$ and a virtual target node $\mathcal{D}$ and create virtual directed links $(\mathcal{S}, s_i)$ between the virtual source node $\mathcal{S}$ and each of the actual source node s_i with infinite capacity, and virtual directed links $(d_i, \mathcal{D})$ between each of the actual target nodes d_i and the virtual target node $\mathcal{D}$ with infinite capacity. These virtual links will not cause any interference to any other links. We also add an additional constraint that the attacker cannot attack the virtual links in our mathematical formulation.

2.4.3 *Packet-dropping Attacking Formulation*

Packet-dropping attack is that the attacker chooses to drop some packets or jam some links or nodes so as to reduce the total throughput; while the routing policy maker tries to reduce or prevent the effect from the malicious action of the attacker. For simplicity, we assume that the attacker will only *drop* the packets, *i.e.*, when it attacks a link e with effort α_e it will drop $\alpha_e \cdot \ell_e$ total packets per unit time out of total ℓ_e transmitted over link e. In the mind of the routing policy maker, the attacker is always supposed to minimize the flow to the target node d under certain flow scheduling ℓ, *i.e.*, $\min_\alpha \sum_{e\in\Delta^-(d)}(1-\alpha_e)\cdot \ell_e$. Under an attacking strategy α by the attacker, the policy maker will design the routing algorithm to maximize the flow to the target, *i.e.*, selecting ℓ for each link e such that $\max_\ell \min_\alpha \sum_{e\in\Delta^-(d)}(1-\alpha_e)\cdot \ell_e$.

In this packet-dropping model, the flow conservation is different from the former one because some packets in the network will not reach the target node. The modified flow conservation in the network is,

$$\sum_{e\in\Delta^-(u)} (1-\alpha_e)\ell_e - \sum_{e\in\Delta^+(u)} \ell_e = 0 \quad \forall u \neq s, d$$

In other words, when the routing policy maker makes a routing flow decision, it can utilize the fact that some packets from some incoming links may be dropped by the attacker (thus it will have more room for scheduling for the outgoing links). Thus we summarize the formula for the packet-dropping attacking model as:

$$\textbf{Packet-dropping Model:} \quad \min_\alpha \max_\ell \sum_{e\in\Delta^-(d)} (1-\alpha_e)\cdot \ell_e, \ \textbf{s.t.} \quad (4)$$

$$\begin{cases} \sum_{e\in E} \alpha_e \cdot C_e \cdot H_e \leq B & \forall e \\ 0 \leq \alpha_e \leq 1 & \forall e \\ \sum_{f\in F(e)} \beta_{e,f} \cdot C_{e,f} = \ell_e & \forall e \\ \beta_{e,f} + \sum_{e'\in I_{\mathcal{M}}(e)} \beta_{e',f} \leq A & \forall e \\ \sum_{e\in\Delta^-(u)}(1-\alpha_e)\ell_e - \sum_{e\in\Delta^+(u)} \ell_e = 0 & \forall u \neq s, d \end{cases}$$

2.5 Complexity Results

Given a strategy of one side, we first study the complexity of finding the best responding strategy by the other side. When the routing policy is given, *i.e.*, the load ℓ_e on every link e is known, the attacker needs to find the best strategy that will minimize the healthy packets received by the target node while satisfying the budget constraint. By checking all our formulas (see formulation (1), (2), (3), (4)) we find that every question becomes a linear programming and thus can be solved in polynomial time by the attacker.

On the other hand, when the attacking strategy α is given, the routing policy maker needs to find the best routing strategy ℓ that will maximize the effective throughput. Unfortunately this is an NP-hard problem since it is NP-hard to find a flow that maximize the network throughput even without the attacking [1,7,10,11,23]. Notice that, given a load assignment ℓ, the routing policy maker even cannot always determine if it is schedulable without causing interference. Thus, we have the following theorem.

THEOREM 1. *It is NP-hard for the routing policy maker to find the best routing flow ℓ_e that is schedulable and will maximize the effective network throughput, when given the attacking strategy α.*

We will later show how the routing policy maker can find a routing policy ℓ_e such that the achieved effective network throughput is at least a constant factor of the optimum given the attacking strategy.

3. OUR SOLUTION AND PERFORMANCE GUARANTEE

In the previous section, we provide mathematical formulations for finding an attacking strategy and a routing strategy that will result in a Nash Equilibrium when either non-packet-dropping attacking or packet-dropping attacking is used by the attacker. Several previous studies [1, 10, 23], provide various routing and scheduling methods such that the achieved network throughput is within a constant factor of the optimum when there is *no* attacking in the network. In this section, we will propose methods to find joint routing and link scheduling, which could achieve larger network throughput when there is a possible attack but the attacker has a budget constraint. We will also prove that our method will achieve networking throughput that is at least a constant factor of the optimum with attacking.

Observe that the direct formulation of optimal routing strategy (and corresponding link scheduling) under certain attacking strategy is a $\min\max$ linear programming, which cannot be solved directly using traditional linear programming. We first transform these $\min\max$ LPs into certain linear programmings with objective functions that minimize or maximize certain functions. Our approach is to transform the inner minimization problem to a maximization problem using the dual property of linear programming.

3.1 Solution for Non-packet-dropping Attacking

Remember that for the non-packet-dropping attacking, the objective function of our mathematical formulation is $\max_\ell \min_\alpha \sum_{e\in E}(1-\alpha_e)\cdot\ell_e$. Refer the **Non-packet-dropping Model** LP formulation in subsubsection 2.4.2. Given this objective function, we can not solve it directly using linear programming. Our approach is to use prime-dual formula of linear programming: we separate the formulation into two linear programmings, and then convert one of the LP to its dual such that the overall programming formulation will have a uniform objective function (wither $\max$ or $\min$). Observe that the value of the objective function will not change based on prime-dual property of LP. However, we do need convert the solution of dual back to the solution of prime LP to find the routing (or attacking) strategy. Consider the inner structure of the objective function, $\min_\alpha \sum_{e\in E}(1-\alpha_e)\cdot\ell_e$ for the fixed routing ℓ. It is trivial that we can convert this inner structure of objective function to $\max_\alpha \sum_{e\in E}(\alpha_e-1)\cdot\ell_e$. Then the original $\max\min$ linear programming is converted to a problem whose objective function is $\max$. However this conversion does not result in a linear programming since the objective function is quadratic. Thus, to solve this, we need use other approaches. We try to convert the above problem to an equivalent linear programming (with the same objective value) by using the dual of the linear programming. Let $\overline{\alpha}=1-\alpha$, this inner structure of the objective function is equal to $\min_{\overline{\alpha}} \sum_{e\in E}\overline{\alpha}_e\cdot\ell_e$.

When the routing strategy ℓ is fixed, the attacking strategy is to find $\overline{\alpha}$ such that $\min_{\overline{\alpha}} \sum_{e\in E}\overline{\alpha}_e\cdot\ell_e$ is achieved. Clearly, the attacking strategy α can be solved by using the following linear programming (when ℓ is fixed)

Prime of Inner LP of Non-packet-dropping:

$$\begin{aligned} &\min_{\overline{\alpha}} \textstyle\sum_{e\in E}\overline{\alpha}_e\cdot\ell_e && \textbf{s.t.}\\ &\textstyle\sum_{e\in E}(1-\overline{\alpha}_e)\cdot C_e\cdot H_e \le B && \forall e\\ &0\le\overline{\alpha}_e\le 1 && \forall e \end{aligned}$$

We define the dual variable x for the in-equality $\sum_{e\in E}(1-\overline{\alpha}_e)\cdot C_e\cdot H_e\le B$ and dual variable x_e for the in-equality $\overline{\alpha}_e\le 1$ for each link e. Then it is not difficult to show that the dual of the above prime Linear Programming is

Dual of Inner LP of Non-packet-dropping:

$$\begin{aligned} &\max_{x,x_e}(\textstyle\sum_{e\in E}C_e\cdot H_e - B)x - \sum_{e\in E}x_e && \textbf{s.t.}\\ &C_e\cdot H_e\cdot x - x_e\le\ell_e, && \forall e\\ &x_e\ge 0 && \forall e\\ &x\ge 0 \end{aligned}$$

Notice that the original LP for finding the pair of routing strategy and attacking strategy involves both ℓ and α. Together with the constraint conditions from the mathematical $\min\max$ formulation for **Non-packet-dropping Model**, the primary linear program is thus converted as below

Non-packet-dropping Model LP: $\max_{x,x_e}(\sum_{e\in E}C_e\cdot H_e - B)x - \sum_{e\in E}x_e$, **s.t.**

$$\begin{cases} C_e\cdot H_e\cdot x - x_e\le\ell_e & \forall e\\ \sum_{f\in F(e)}\beta_{e,f}\cdot C_{e,f}=\ell_e & \forall e\\ \beta_{e,f}+\sum_{e'\in I_{\mathcal{M}}(e)}\beta_{e',f}\le A & \forall e\\ \sum_{e\in\Delta^+(u)}\ell_e-\sum_{e\in\Delta^-(u)}\ell_e=0 & \forall u\ne s,d\\ x_e\ge 0 & \forall e\\ x\ge 0 \end{cases}$$

After we have this linear programming, clearly we can solve it using any efficient linear programming solver to find a solution for ℓ_e, x and x_e for each link e in polynomial time (since the number of variables and the number of constraints are the polynomial functions of the number of links). After we find the best routing strategy ℓ_e, we can substitute ℓ_e to the **prime of the inner LP** for non-packet-dropping model and find α in polynomial time.

Notice that when the objective function of policy routing maker and the attacker is to

$$\min_\alpha\max_\ell\left(\sum_{e\in\Delta^-(d)}\ell_e-\sum_{e\in E}\alpha_e\cdot\ell_e\right)=\max_\ell\min_\alpha\left(\sum_{e\in\Delta^-(d)}\ell_e-\sum_{e\in E}\alpha_e\cdot\ell_e\right),$$

then we can similarly convert this problem to a linear programming

as above. Recall that the prime LP of the inner structure is

Prime of Inner LP of Non-packet-dropping:

$$\min_\alpha \left(\sum_{e\in\Delta^-(d)} \ell_e - \sum_{e\in E} \alpha_e \cdot \ell_e\right) \quad \textbf{s.t.}$$
$$\sum_{e\in E} \alpha_e \cdot C_e \cdot H_e \le B$$
$$0 \le \alpha_e \le 1 \quad \forall e$$

This can be converted to

Prime of Inner LP of Non-packet-dropping:

$$\min_\alpha \left(\sum_{e\in\Delta^-(d)} \overline{\alpha}\ell_e - \sum_{e\notin\Delta^-(d)} \alpha_e \cdot \ell_e\right) \quad \textbf{s.t.}$$
$$\sum_{e\in\Delta^-(d)}(1-\overline{\alpha})\cdot C_e \cdot H_e + \sum_{e\notin\Delta^-(d)} \alpha_e \cdot C_e \cdot H_e \le B$$
$$0 \le \overline{\alpha}_e \le 1 \quad \forall e \in \Delta^-(d)$$
$$0 \le \alpha_e \le 1 \quad \forall e \notin \Delta^-(d)$$

Similarly, we can convert the above prime LP to the dual LP as follows:

Dual of Inner LP of Non-packet-dropping:

$$\max_{x,x_e}\left(\sum_{e\in\Delta^-(d)} C_e \cdot H_e - B\right)x - \sum_{e\in E} x_e \quad \textbf{s.t.}$$
$$C_e \cdot H_e \cdot x - x_e \le \ell_e, \quad \forall e \in \Delta^-(d)$$
$$C_e \cdot H_e \cdot x + x_e \ge \ell_e, \quad \forall e \notin \Delta^-(d)$$
$$x_e \ge 0 \quad \forall e$$
$$x \ge 0$$

We briefly discuss the situation when there is a data rate demand $\theta(s)$ from the source node s. It is not difficult to observe that, given a fixed routing strategy ℓ, the attacker aims to minimize $\sum_{e\in E}(1-\alpha_e)\cdot\ell_e$ subject to constraints $\sum_{e\in E}\alpha_e C_e H_e \le B$ and $0 \le \alpha_e \le 1$. Its dual LP is $\max_{x,x_e} B\cdot x - \sum_{e\in E} x_e$ subject to constraints $C_e \cdot H_e \cdot x - x_e \le \ell_e$, $\forall e \in E$, and $x, x_e \ge 0$. Then as before, we can merge this dual LP with the minimization LP characterizing the optimization problem for the routing policy maker.

3.2 Solution for Packet-dropping attacking

We then study how to find solution for packet-dropping attacking model. Remember that the objective function of the $\max\min$ mathematical formulation for **Packet-dropping Model** is

$$\max_\ell \min_\alpha \sum_{e\in\Delta^-(d)} (1-\alpha_e)\cdot\ell_e.$$

Again we will transform the inner minimization problem to a maximization problem using the dual property. For a fixed link scheduling ℓ, the inner structure of the above linear program is as below

Prime Inner LP of Packet-dropping: $\min_\alpha \sum_{e\in\Delta^-(d)} (1-\alpha_e)\cdot\ell_e$, **s.t.**

$$\begin{cases} \sum_{e\in E}\alpha_e\cdot C_e\cdot H_e \le B & \forall e \\ \sum_{e\in\Delta^-(u)}(1-\alpha_e)\ell_e - \sum_{e\in\Delta^+(u)}\ell_e = 0 & \forall u\ne s,d \\ 0\le\alpha_e\le 1 & \forall e \end{cases}$$

Let $\overline{\alpha} = 1-\alpha$. Then the prime of the inner LP for packet-dropping attacking model becomes

$$\min_{\overline{\alpha}} \sum_{e\in\Delta^-(d)} \overline{\alpha}_e\cdot\ell_e, \quad \textbf{s.t.}$$

$$\begin{cases} \sum_{e\in E}(1-\overline{\alpha}_e)\cdot C_e\cdot H_e \le B & \forall e \\ \sum_{e\in\Delta^-(u)}\overline{\alpha}_e\ell_e - \sum_{e\in\Delta^+(u)}\ell_e = 0 & \forall u\ne s,d \\ 0\le\overline{\alpha}_e\le 1 & \forall e \end{cases}$$

Define a variable x for the first constraint, define a variable x_u for each equation defined for every node $u \ne s, d$, and define a variable x_e for each link e corresponding to the inequality $\overline{\alpha_e} \le 1$. Clearly, we need $x \ge 0$, $x_e \ge 0$, and no constraints on variable x_u since its corresponding constraint is equality. Thus, we get the dual of the above linear program as follows:

Dual of Inner LP of Packet-dropping Model:

$$\max_{x,x_u,x_e}\left(\sum_{e\in E} C_eH_e - B\right)x + \sum_u\left(\sum_{e\in\Delta^+(u)}\ell_e\right)x_u - \sum_{e\in E}x_e, \quad \textbf{s.t.}$$

$$\begin{cases} C_eH_e\cdot x - x_e \le \ell_e & \forall e\in\Delta^-(d) \\ C_eH_e\cdot x + \ell_e x_u - x_e \le 0 & \forall e=(v,u)\notin\Delta^-(d) \\ x\ge 0 \\ x_e\ge 0 \end{cases}$$

Then the original mathematical $\max\min$ formulation for the packet-dropping attacking is converted to the following quadratic programming by considering all related constraints together.

Quadratic Programming of Packet-dropping Model:

$$\max_{\ell,x,x_u,x_e}\left(\sum_{e\in E} C_eH_e - B\right)x + \sum_u\left(\sum_{e\in\Delta^+(u)}\ell_e\right)x_u - \sum_{e\in E}x_e, \quad \textbf{s.t.}$$

$$\begin{cases} C_eH_e\cdot x - x_e \le \ell_e & \forall e\in\Delta^-(d) \\ C_eH_e\cdot x + \ell_e x_u - x_e \le 0 & \forall e=(v,u)\notin\Delta^-(d) \\ \sum_{f\in F(e)}\beta_{e,f}\cdot C_{e,f} = \ell_e & \forall e \\ \beta_{e,f} + \sum_{e'\in I_{\mathcal{M}}(e)}\beta_{e',f} \le A & \forall e \\ x\ge 0 \\ x_e\ge 0 \end{cases}$$

Unlike the non-packet-dropping attacking model, the above programming is quadratic instead of linear programming. For quadratic programming, typically we cannot solve it optimally [4, 14]. We could use some existing solver for quadratic programming that could find a almost optimal solution (whose objective value is at least $1-\epsilon$ of that of the optimal solution for an arbitrary small ϵ) in polynomial time.

Notice that after the routing policy maker solves this linear programming, it will perform routing using a probabilistic approach as follows. Recall that $\beta_{e,f}$ denotes the fraction of the time slots that link e will be used for routing using channel f. Thus, when a mesh router u receives data from some mesh routers, it will forward the data to a node v using channel f with probability $\beta_{e,f}$ where $e=(u,v)$. To ensure that the transmissions by different nodes will not conflict, such probability is achieved with a careful scheduling of time slots used.

3.3 Performance Guarantee and Other Variations

We first show that our solutions will find a routing strategy that will achieve an effective throughput within a constant factor of the optimum.

THEOREM 2. *Our saddle routing policy will find a multiple-path routing and corresponding link scheduling such that the total effective throughput achieved under the non-packet-dropping attack or the packet-dropping attack is within a constant factor of the optimum when the policy maker has infinite computation power.*

PROOF. Consider an optimum flow assignment defined by $\beta^*(e,\mathbf{f})$, *i.e.*, the flow supported by a link e is $\sum_{\mathbf{f}}\beta^*(e,\mathbf{f})\cdot\mathbf{c}(e,\mathbf{f})$. It was proved in [1, 10, 23] that $\beta^*(e,\mathbf{f}) + \sum_{e'\in\mathbf{I}_{\mathcal{M}}(e)}\beta^*(e',\mathbf{f}) \le C_{\mathcal{M}}$ for a constant $C_{\mathcal{M}}$ depending on the interference model $\mathcal{M}$. Define a new flow β' as $\beta'(e,\mathbf{f}) = \frac{\beta^*(e,\mathbf{f})}{C_{\mathcal{M}}}$. Obviously, $\beta'(e,\mathbf{f}) + \sum_{e'\in\mathbf{I}_{\mathcal{M}}(e)}\beta'(e',\mathbf{f}) \le 1$. It is easy to show that the new flow

β' (*i.e.*, the corresponding $\ell'_e = \sum_{\mathbf{f}} \beta'(e', \mathbf{f}) \cdot \mathbf{c}(e, \mathbf{f})$) satisfies all constraints of our mathematical formulations. Based on scheduling algorithms presented in [1, 10, 23], we know that in polynomial time, we can find a feasible time-slot scheduling to implement this new flow β'. In other words, β' is a feasible solution for both $\max\min$ formulations of the non-packet-dropping attacking models and the packet-dropping attacking models. Consequently, since our LP formulation for the non-packet-dropping attacking model and the quadratic programming formulation for the packet-dropping attacking model will find a configuration that maximizes the effective networking throughput, the *effective throughput* under the found pair of (routing strategy ℓ, attacking strategy α) is at least that of β', which is $\frac{1}{C_{\mathcal{M}}}$ of the optimum. Observe that the effective throughput for non-packet-dropping attacking model with routing ℓ' is $\sum_{e \in \Delta^-(d)} \ell'_e - \sum_{e \in E} \alpha_e \cdot \ell'_e$. This finishes the proof. □

Observe that in all our formulations, we use an integer A. When $A = 1$, it is guaranteed that we can find a link schedule for the found routing strategy ℓ in polynomial time. When $A > 1$, the polynomial time-computable link schedule might exist. When $A > C_{\mathcal{M}}$, it is known that **no** link schedule is feasible for this flow. Using a parameter A can improve the practical performance of our methods as follows. We start from an integer $A = C_{\mathcal{M}}$ and try to solve the corresponding linear programming or quadratic programming. Then we decide if we can find a correct link scheduling for the flow routing ℓ using our greedy link scheduling algorithm. If there is a scheduling, we are done. Otherwise, we reduce A by 1 and repeat. In the worst case, this will stop when $A = 1$. Assume that in general, the above procedures stop at an integer A_0. Then similar to the proof of Theorem 2, we can show that the effective throughput achieved is at least $\frac{A_0}{C_{\mathcal{M}}} \geq \frac{1}{C_{\mathcal{M}}}$ times of the optimum.

Notice that in our previous formulations, we assumed a non-packet-dropping attacking model with eavesdropping or modifying packets or a packet-dropping model with throwing away packets or jam network. In practice, the attacking could be eavesdropping, dropping, injecting, or jamming packets, or some combination of these. Our previous studies only serve as an illustration to show how to address these attacks using saddle routing policy. For example, when the attacker could inject the packets into the network, the objective function now becomes $\max_\ell \min_\alpha \sum_{e \in \Delta^-(d)} \ell_e/(1 + \alpha_e)$ and the flow conservation constraint becomes $\sum_{e \in \Delta^-(u)}(1 + \alpha_e)\ell_e - \sum_{e \in \Delta^+(u)} \ell_e = 0$ since for each incoming link e incident on a node u, the attacker could inject α_e additional malicious packets. Then similar approach can transform the $\max\min$ problem into a quadratic programming (here we fix the attacking strategy α and transform the $\max_\ell$ optimization problem of the routing policy maker into its dual linear programming, which is a minimization linear programming). If we want to combine some attacks together, we can form similar but more complicate $\max\min$ linear programming and then convert it to quadratic programming.

Another important property of the solution that we need to study is the stability[1] of the found strategy pair (routing strategy $\overline{\ell}$, attacking strategy $\overline{\alpha}$). Notice that the optimal solution (ℓ^*, α^*) is stable since it is a Nash equilibrium. However, since it is NP-hard to find the optimal strategy pair, our methods can only find an approximate strategy pair whose effective throughput is within a constant factor of the optimum. Notice that, given the routing strategy $\overline{\ell}$, the optimum attacking strategy $\overline{\alpha}$ can be found in polynomial time by solving the linear programming. Thus, we have the following lemma.

[1]A pair (ℓ, α) is stable if given the strategy of one party, the other party cannot find a better strategy.

LEMMA 3. *Given the fixed routing strategy $\overline{\ell}$, the corresponding attacking strategy $\overline{\alpha}$ is already optimal, i.e., the attacker cannot find better attacking strategy.*

On the other hand, when given the fixed attacking strategy $\overline{\alpha}$, the corresponding routing strategy $\overline{\ell}$ is *not optimal*: there may exist better routing strategy with larger effective network throughput. Our proof already showed that the routing policy maker will not gain much by finding better routing strategy since the effective network throughput $\mathcal{T}(\overline{\alpha}, \overline{\ell})$ is already within a constant factor of the best $\mathcal{T}(\overline{\alpha}, \ell)$ for any routing ℓ. Also notice that if the routing policy maker changes $\overline{\ell}$, the attacker will also quickly change its attacking strategy, which will cause the performance fluctuation of the network.

4. SIMULATION RESULTS

In this section, we will exam the impact of budget by the attacker, the impact of network size, the impact of the transmission radius, or the impact of the radios on the total throughput of the wireless mesh network under attack and without attack for non-packet-dropping model. We have discussed previously that the total throughput without attack is solved by linear program. While the throughput under attack is first formed by linear program and then solved by the dual of the linear program. For effective throughput under attack, we assume the attacker will not attack the same packet twice and the healthy throughput is calculated by

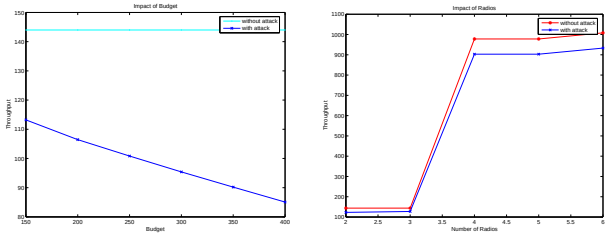
$$\max_{\ell} \min_{\alpha} \left(\sum_{e \in \Delta^-(d)} \ell_e - \sum_{e \in E} \alpha_e \cdot \ell_e \right).$$

In the simulation, the wireless network is generated by randomly choosing the position of the routing nodes and gateway nodes. The wireless network generated in the simulation is guaranteed to be connected with high probability by ensuring the follows: the parameters of the network satisfy the constraint $n\pi r^2 \geq c \log n$, where n is the number of nodes, r is the transmission radius and c is some constant. Initially, the wireless network is generated with 40 nodes. The nodes is randomly dispersed in an area of 300×400 square meters. The number of radios for each node is 2 and each radio can be used by 2 channels. For all the simulation results reported here, we use RTS/CTS interference model.

We use $802.11a$ for the link channel capacity in the wireless network, which is same as [1]. The link channel capacity thus only depends on the distance between the two nodes at the end of the link. We set the link channel capacity as $54Mbps$ when the distance of the two end nodes is within 30 meters, $48Mbps$ when the distance is within 32 meters, $36Mbps$ when the distance is within 37 meters, $24Mbps$ when the distance is within 45 meters, $18Mbps$ when the distance is within 60 meters, $12Mbps$ when the distance is within 69 meters, $9Mbps$ when the distance is within 77 meters, and $6Mbps$ when the distance is within 90 meters. Otherwise, if the distance of the two end nodes of the link is beyond 90 meters, we will set the link channel capacity as 0.

4.1 Impact of Budget

Figure 1 (a) illustrates the throughputs without attach and under attack when the budget various from 150 to 400. The upper line denotes the achieved throughput when no attack exists; the lower line denotes the throughput with attack. With the budget increases, the throughput under attack decreases. Figure 1 (a) shows that the total packets received by the target nodes changes slowly with the budget. We can see that the throughput is $113.204000Mbps$ when



(a) impact of budget (b) impact of multi-radio

Figure 1: Impact of budget on various throughputs with/without attack.

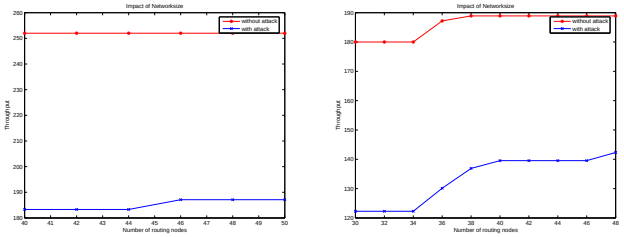
the budget is 150 while the throughput is around $95.393600 Mbps$ when the budget is 300.

We also observe an important phenomenon in our simulations that when the budget B of the attacker increases, the routing policy maker does not need to change its routing policy always. In most situations, the routing policy maker can keep the same routing policy if B is only increased by a small value. In other words, our solution is *stable* to some extent. This actually can be explained as follows. When the budget B increases, in our mathematical formulation and solutions, it means the objective function (which is a hyperplane in a high dimension) will rotate a little bit. All the constraints in **Non-packet-dropping Model LP** define a polytope, which does not depend on B. Thus, only when the hyperplane rotate enough angle, it will then be tangent on another vertex (thus a new optimal routing policy under the formulation of **Non-packet-dropping Model LP**).

4.2 Impact of Multi-Radio

In this simulation, we study how the radio can affect the throughput. We use wireless mesh networks with 40 routing nodes and 8 gateway nodes. The budget of the attack is set as 150. The radio is from 2 to 6. Figure 1 (b) illustrates the trend of the throughput when the radio varies. We can observe in Figure 1 (b) that the throughput increases with the increasing of the radio. This can be explained as following. With the radio increases, the number of common channel shared between two nodes in one period increases, which result in the increasing total capacity of the link and thus the increasing of the throughput.

4.3 Impact of Networking Size



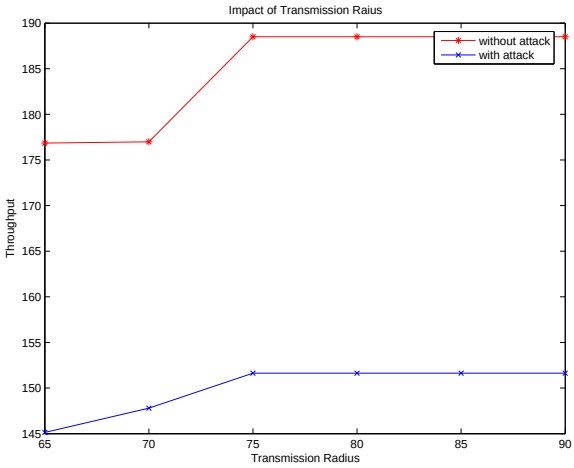
(a) routing nodes from 40 to 50 (b) routing nodes from 30 to 48

Figure 2: Impact of network size on various throughputs.

In this simulation, we generate networks with different number of routing nodes to study how the network size can affect various throughputs in a wireless mesh network. We expect the throughput will increase with the network size increases and then saturate at some point due to the limit of gateway nodes and the attacking from attackers.

In Figure 2 (a), we show the results when the network is generated randomly from 40 routing nodes to 50 routing nodes and 8 gateway nodes. The number of channels per radio is 2, the number of radios per node is 2 and the maximal transmission radius is 90 meters. We add routing nodes by 2 each time. We find that the throughput of the network has already reached the saturate point when the number of routing nodes is 40 without attack. So the throughput without attack keeps the same when the routing nodes increase from 40 to 50. While the more routing nodes imply the more choices for the routing, so the throughput under attack increases when the routing nodes increase from 44 to 46 and then remains the same as the routing nodes are added, which implies that the throughput reaches the saturated point. We generate the other network in Figure 2 (b), to show the results when the network is generated randomly from 30 routing nodes to 48 routing nodes. Similarly, we add 2 routing nodes each time. This figure shows more clearly that the adding of routing nodes brings more choices for the routing and then results in more throughput. Finally, the throughput in the network will arrive at the saturated point.

4.4 Impact of Transmission Radius



transmission radius

Figure 3: Impact of network size on various throughputs.

In this simulation, we study how the radius can affect various throughputs. We use wireless mesh networks with 40 routing nodes and 8 gateway nodes. The budget of the attack is set as 150. The radius varies from 65 meter to 90 meter. Figure 3 illustrates the trend of the throughput when the radius varies. We can observe in Figure 3 that the throughput either in the network without attack or in the network under attack increases until it reaches the saturated point after the transmission radius increases to some value. With the increase of the transmission radius for a node, such as node u, there are more nodes falling into node u's transmission radius and becoming its neighbors, which forms additional outgoing links for node u. Every node in the network has the same situation, so this makes the policy maker more choices for the routing policy. The policy maker then can schedule a better routing or scheduling method and then improve the throughput until the throughput saturates at some value. That is, the increase of the transmission radius provides more choices for the routing policy and then results in the increase in throughput. After the throughput attains to the saturated point, it will remain the same whenever the transmission radius increases.

5. LITERATURE REVIEW

Usually, the conventional routing protocols are based on shortest path, such as OSPF [15] and RIP [13]. This makes the path predictable and results in the interception or eavesdropping attack. Multi-Path can ameliorate this matter while make packet-reordering more complicated [22]. Some techniques can solve it, such sophisticated coding technique [6], standard pre-buffering technique [12] and so on. In [16], the author proposed a distributed secure multipath solution so that the data is routed by multiple paths.

There are also some techniques focusing on routing-level security to detect the DoS attack such as CenterTrack [21], IP Ttraceback [18]. RON (Resilient Overlay Networks) [2] is an architecture which improves current network for allowing the network to recover from outages within several seconds. The author in [8] detects network attack by sampling. The idea is to examine partial packets in the network are sampled and examined. The author formulates it as a game theory problem and resolves it by dual of the linear program.

The most related results presented in the literature are [5, 17, 19, 20]. In [17], users send their request either to the server or to the other users rather than directly to the server. In [5], the author first proposed SSR (Security Stochastic Routing), which takes multiple paths with some probability instead of single path routing. In [19, 20], the author extends the result in [5] by considering more general attacks. However, all these results are based on wired networks, which do not have the interference constraints when scheduling links for transmission. Notice that as observed in the literature, interference constraints often make many problems intractable such as network throughput maximization and link scheduling.

6. CONCLUSION

In this paper we study how multiple-paths routing can be simultaneously used to improve the network throughput and the routing security. We specifically study the non-packet-dropping attacking and the packet-dropping attacking models by attackers. We mathematically formulate the problem as a $\max\min$ optimization problem and then convert them to either an equivalent linear programming or quadratic programming problem. We theoretically prove that the effective network throughput achieved under the found routing strategy and attacking strategy is at least a constant factor of the optimum. We also show that the strategy pair is stable for the attacker in the sense that, if the routing policy remains the same, the attacker cannot further reduce the achieved networking throughput. When the attacking strategy is fixed, it is NP-hard for the routing policy maker to find the best routing strategy to maximize the achieved effective throughput; our routing strategy will find a routing whose effective throughout is at least a constant factor of the optimum. Notice that although we considered the non-packet-dropping attacking and the packet-dropping attacking separately, it is not difficult to show that our results can be easily extended to situations when both non-packet-dropping attacking and packet-dropping attacking happen. In that case, the mathematical formulation will be quadratic programming. The details of this study is omitted due to space limit. There are clearly many interesting situations we did not address here. For example, how to address the situation when the attacking cost of nearby links (or nodes) are not independent; how to implement the routing strategy in a distributed manner efficiently and how to dynamically adapt to possible varying attacking strategy by the attackers.

7. REFERENCES

[1] Mansoor Alicherry, Randeep Bhatia, and Li (Erran) Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *MobiCom '05*, pages 58–72.

[2] M. F. Kaashoek D. G. Andersen, H. Balakrishnan and R. Morris. Resilient overlay networks. *in Proc. 18th ACM SOSP*, 2001.

[3] P. Gupta and P. Kumar. Capacity of wireless networks. Technical report, University of Illinois, Urbana-Champaign, 1999.

[4] Christoph Helmberg. Semidefinite programming for combinatorial optimization, 2000. lecture notes.

[5] J. P. Hespanha and S. Bohacek. Preliminary results in routing games. *in Proceedings of the 2001 American Control Conference*, June, 2001.

[6] M. Luby J. Byers and M. Mitzenmacher. Accessing multiple mirror sites in parallel: Using tornado codes to speed up downloads. *IEEE INFOCOM*, March 1999.

[7] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. In *MobiCom '03*, pages 66–80.

[8] Murali Kodialam and T. V. Lakshman. Detecting network intrusions via sampling : A game theoretic approach. *INFOCOM*, 2003.

[9] Murali Kodialam and Thyaga Nandagopal. Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem. In *MobiCom '03*, pages 42–54.

[10] V. S. Anil Kumar, Madhav V. Marathe, Srinivasan Parthasarathy, and Aravind Srinivasan. Algorithmic aspects of capacity in wireless networks. *SIGMETRICS Perform. Eval. Rev.*, 33(1):133–144, 2005.

[11] Xiang-Yang Li, YanWei Wu, and WeiZhao Wang. Throughput optimization in multihop multiradio multichannel wireless networks, 2006. Manuscript.

[12] D. Loguinov and H. Radha. End-to-end internet video traffic dynamics: Statistical study and analysis. *IEEE INFOCOM*, June 2002.

[13] G. Malkin. Rip version 2. *RFC 2453*, November 1998.

[14] Renato D. C. Monterio. First and second order methods for semidefinite programming, 2002.

[15] J. Moy. Ospf version 2. *RFC 2328*, April 1998.

[16] D Rubenstein PPC Lee, V Misra. Distributed algorithms for secure multipath routing. *IEEE INFOCOM*, 2005.

[17] M. K. Reiter and A. D. Rubin. Crowds: Anonymity forweb transactions. *ACM Trans. on Information and System Security*, 1998.

[18] A. Karlin S. Savage, D. Wetherall and T. Anderson. Practical network support for ip traceback. *ACM SIGCOMM*, 2000.

[19] Katia Obraczka Stephan Bohacek, J. P. Hespanha. Enhancing security via stochastic routing. *Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference*, 2002.

[20] Katia Obraczka Stephan Bohacek, J. P. Hespanha. Saddle policies for secure routing in communication networks. *Decision and Control, 2002, Proceedings of the 41st IEEE Conference*, 2002.

[21] R. Stone. Centertrack: An ip overlay network for tracking dos floods. *9th USENIX Security Symposium*, 2000.

[22] D. Thaler and C. Hopps. Multipath issues in unicast and multicast next-hop selection. *RFC 2991*, November 2000.

[23] WeiZhao Wang, Yu Wang, Xiang-Yang Li, Wen-Zhan Song, and Ophir Frieder. Efficient interference aware tdma link scheduling for static wireless mesh networks. In *ACM MobiCom*, 2006.

[24] S. Yi, Y. Pei, and S. Kalyanaraman. On the capacity improvement of ad hoc wireless networks using directional antennas. In *4th ACM MobiHoc*, pages 108–116, 2003.