# Linear Programming Models for Jamming Attacks on Network Traffic Flows


Patrick Tague*, David Slater*, Guevara Noubir†, and Radha Poovendran*

*Network Security Lab (NSL), Department of Electrical Engineering, University of Washington, Seattle, WA, USA. Email: {tague, dmslater, rp3}@u.washington.edu

†College of Computer and Information Science, Northeastern University, Boston, MA, USA. Email: noubir@ccs.neu.edu



***Abstract*—We present a new class of network attacks, referred to as *flow-jamming attacks*, in which an adversary with multiple jammers throughout the network jams packets to reduce traffic flow. We propose a linear programming framework for flow-jamming attacks, providing a foundation for the design of future protocols to mitigate flow-jamming. We propose metrics to evaluate the effect of a flow-jamming attack on network flow and the resource expenditure of the jamming adversary. We develop, evaluate, and compare a variety of flow-jamming attacks using the proposed metrics and the linear programming formulation. In addition, we formulate a distributed flow-jamming attack algorithm for a set of jammers operating without centralized control and compare the performance to the centralized attacks using the linear programming formulation.**


## I. Introduction

The nature of wireless communication using an open and shared physical medium makes it vulnerable to denial-of-service (DoS) attacks [1]. A jamming adversary can perform a variety of DoS attacks, such as transmitting wide-band noise, high-power narrow-band pulses, or interfering waveforms [2]. Anti-jamming communication systems typically rely on the use of spread-spectrum techniques, forcing the adversary to jam a wider frequency band and significantly increasing the jamming power [2], [3]. Such techniques are especially effective against resource-constrained jamming adversaries, as the required energy to jam each bit is drastically increased.

A resource-constrained jamming adversary can, however, counteract the impact of an anti-jamming system such as spread-spectrum by incorporating information of higher-layer communication or networking protocols. For example, intelligent jamming techniques have recently been developed for DoS attacks targeting certain wireless link layer and MAC protocols [4]–[6] and link layer error correction protocols [7], leading to significant energy savings over continuous jamming.

We suggest that DoS attack efficiency can be further improved by incorporating network layer information. Since a single packet traverses multiple wireless network links, the adversary can choose to jam each packet when minimal energy is required, effectively jamming the traffic flow [8]. An adversary in control of multiple jammers can thus balance the total energy expenditure required to jam network flows over the jammers, optimizing an objective function such as the total energy and prolonging jammer lifetime. Hence, the efficiency of the attack can be optimized by intelligent assignment of jammers to flows. We refer to this efficient DoS attack as a *flow-jamming attack*.

The first step toward defending against flow-jamming attacks is the ability to model them in the context of network protocol design. To the best of our knowledge, incorporating the effects of jamming into network protocol design is a new research area. We make the following contributions toward this problem.

- We show that flow-jamming attacks can be formulated using a linear programming framework, often used for network resource allocation problems.
- As the basis of our formulation, we propose metrics to evaluate the effect of flow-jamming attacks on network traffic flows and the resource expenditure of the jamming adversary with respect to a finite resource constraint.
- We develop, evaluate, and compare a variety of flow-jamming attacks which are optimal with respect to the proposed metrics.
- We propose a distributed flow-jamming attack algorithm for a set of jammers without centralized control and compare the performance of the distributed and centralized attacks.

The remainder of this work is outlined as follows. In Section II, we state our assumptions about the wireless network and jamming adversary and propose evaluation metrics for flow-jamming attacks. In Section III, we formulate optimal centralized flow-jamming attacks using linear programming. In Section IV, we develop a distributed algorithm for flow-jamming in the absence of a centralized adversary. In Section V, we evaluate the performance of the centralized and distributed flow-jamming attacks using the proposed metrics. In Section VI, we summarize our results.

## II. Model Assumptions

In this section, we state our assumptions and provide notation and definitions for the wireless network and jamming adversary. In addition, we provide metrics for the evaluation of flow-jamming attacks. A summary of notation is provided in Table I.


This work was performed while G. Noubir was visiting NSL at UW.

TABLE I
A SUMMARY OF NOTATION IS PROVIDED.

| Symbol | Definition |
|---|---|
| $\mathcal{N}$ | Set of wireless network nodes |
| $\mathcal{F}$ | Collection of network flows |
| $r_f$ | Flow rate of flow $f \in \mathcal{F}$ |
| $\mathcal{J}$ | Set of jammers |
| $c_j$ | Jamming resource supply for jammer $j \in \mathcal{J}$ |
| $c_{jf}$ | Cost per unit flow rate for $j \in \mathcal{J}$ and $f \in \mathcal{F}$ |
| $x_{jf}$ | Jammer-to-flow assignment for $j \in \mathcal{J}$ and $f \in \mathcal{F}$ |
| $\mathbf{x}_j$ | Jammer-to-flow assignment vector for jammer $j \in \mathcal{J}$ |
| $\mathbf{x}_f$ | Jammer-to-flow assignment vector for flow $f \in \mathcal{F}$ |
| $\mathbf{x}$ | Jammer-to-flow assignment vector |
| $\lambda_j(\mathbf{x}_j)$ | Resource expenditure of jammer $j \in \mathcal{J}$ |
| $\Lambda(\mathbf{x})$ | Vector of resource expentiture variables |
| $I(\mathbf{x})$ | Jamming impact, see Definition 1 |
| $E(\mathbf{x})$ | Jamming efficiency, see Definition 2 |
| $V(\mathbf{x})$ | Jamming resource variation, see Definition 3 |

### A. Network Model

The wireless network consists of a set of nodes $\mathcal{N}$. Data traffic between source and destination nodes in $\mathcal{N}$ is modeled by a set of flows $\mathcal{F}$. We let $r_f$ denote the flow rate of each flow $f \in \mathcal{F}$. We assume that the nodes in $\mathcal{N}$ are fixed and the flows in $\mathcal{F}$ are fixed for the duration of the flow-jamming attack. We further assume that the flows in $\mathcal{F}$ do not interfere with each other.

### B. Adversarial Model

We let $\mathcal{J}$ denote the set of jammers deployed throughout the wireless network. We assume that each jammer is constrained by a finite energy supply and can jam a given packet with minimum energy expenditure by appropriately adjusting its transmission power or waveform. The minimum transmission power required to jam a packet can be computed as a function of the Jamming-power to Signal-power Ratio (JSR) [2], yielding the required power to increase the bit-error rate to a sufficient threshold. The JSR is computed as a function of the transmitter and jammer power, the network topology, and the antenna properties of each node and jammer. For simplicity, we ignore the randomness in channel variation and assume that the jamming power computed using the JSR is sufficient to jam each packet with probability one.

Each jammer $j \in \mathcal{J}$ can thus compute the JSR for each packet transmission along a flow $f$ and determine the receiving node at which the packet can be jammed with minimum resource expenditure. We let $c_{jf}$ denote the associated resource cost per unit flow rate to jam flow $f$, yielding a total resource cost of $c_{jf}r_f$ to jam every packet in flow $f$. However, since it is not necessary for a single jammer $j$ to jam every packet in a flow $f$, we define the *jammer-to-flow assignment* $x_{jf} \in [0, 1]$ as the fraction of packets in flow $f$ assigned to jammer $j$. A sample assignment of flows to jammers is illustrated in Fig. 1. We denote the vector of jammer-to-flow assignment variables

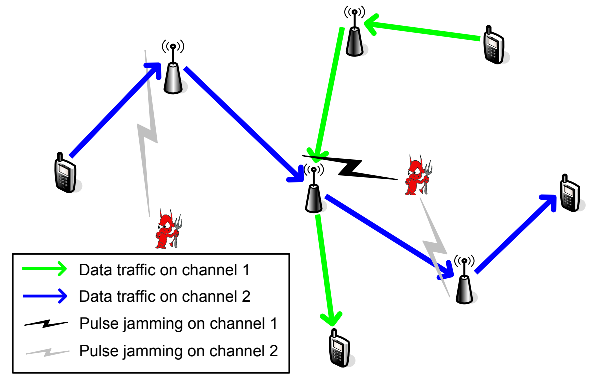



Fig. 1. A sample assignment of flow to jammers is illustrated for two network flows. In this example, jammer B completely jams the flow on channel 1, and jammers A and B collaboratively jam the flow on channel 2 by jamming packets at corresponding receivers requiring minimum resource expenditure.

$x_{jf}$ for all jammers $j$ and flows $f$ by $\mathbf{x}$, for a single jammer $j$ by $\mathbf{x}_j$ and for a single flow $f$ by $\mathbf{x}_f$. Letting $c_j$ denote the total resource availability for jammer $j$, we define the resource expenditure $\lambda_j(\mathbf{x}_j)$ as the fraction

$$\lambda_j(\mathbf{x}_j) = c_j^{-1} \sum_{f \in \mathcal{F}} c_{jf} r_f x_{jf} \tag{1}$$

of total resources exhausted in the flow-jamming attack. We let $\Lambda(\mathbf{x})$ denote the vector of resource expenditure variables for $j \in \mathcal{J}$. For a given set of costs $c_{jf}$ and rates $r_f$, the flow-jamming attack is uniquely specified by the vector $\mathbf{x}$ and the schedule of specific packets to be jammed by individual jammers. We claim that the packet jamming schedule has little effect on the impact of the flow-jamming attack and do not further address the scheduling issues which may arise. We note that the assumption of non-interfering flows essentially means that the schedule for each flow can be arranged independently.

We note that if the jammers in $\mathcal{J}$ are controlled by a centralized adversary, the adversary can leverage the costs $c_{jf}$ and rates $r_f$ for all jammers $j \in \mathcal{J}$ and flows $f \in \mathcal{F}$ to optimize the jammer-to-flow assignment $\mathbf{x}$ as a resource allocation problem [9]. However, if the set of jammers $\mathcal{J}$ operates with no centralized control, each jammer $j \in \mathcal{J}$ must compute the corresponding jammer-to-flow assignment $\mathbf{x}_j$ using a distributed protocol based on local information. These attack formulations are respectively addressed in Sections III and IV for various evaluation metrics.

### C. Evaluation Metrics

We define metrics to evaluate the effect of a flow-jamming attack on traffic flows and the resource expenditure of the jammers. We let $\|\cdot\|_1$ denote the $\ell_1$ vector sum norm [10].

*Definition 1:* The *jamming impact* $I(\mathbf{x})$ of a flow-jamming attack with jammer-to-flow assignment $\mathbf{x}$ is defined as the average fraction of jammed flow rate over all flows in $\mathcal{F}$, given by

$$I(\mathbf{x}) = |\mathcal{F}|^{-1}\|\mathbf{x}\|_1.$$

The metric of jamming impact reflects the overall effect of the flow-jamming attack on the network flows in $\mathcal{F}$ and can be used to determine the worst-case flow-jamming attack on the network flows.

*Definition 2:* The *jamming efficiency* $E(\mathbf{x})$ of a flow-jamming attack with jammer-to-flow assignment $\mathbf{x}$ is defined as the ratio of jamming impact to average resource expenditure, given by

$$E(\mathbf{x}) = \frac{|\mathcal{F}|^{-1}\|\mathbf{x}\|_1}{|\mathcal{J}|^{-1}\|\Lambda(\mathbf{x})\|_1}.$$

The metric of jamming efficiency relates the effect of the flow-jamming attack on the network flows to the resource expenditure of the jammers. We also note that the ratio $I(\mathbf{x})/E(\mathbf{x})$ is the average resouce expenditure, so this metric is expressable using those already defined. The normalization in Definition 2 thus allows for the following interpretation, independent of the number of flows $|\mathcal{F}|$ and jammers $|\mathcal{J}|$. If a jamming impact of $I(\mathbf{x}) = 1$ is achieved with $E(\mathbf{x}) \geq 1$, the jammers in $\mathcal{J}$ are able to completely jam the flows in $\mathcal{F}$ using a fraction $E(\mathbf{x})^{-1}$ of the available resources. If a jamming impact of $I(\mathbf{x}) < 1$ is achieved with $E(\mathbf{x}) = 1$, the jammers in $\mathcal{J}$ exhaust the maximum available resources in order to jam an average fraction $I(\mathbf{x})$ of each flow.

*Definition 3:* The *jamming resource variation* $V(\mathbf{x})$ of a flow-jamming attack with jammer-to-flow assignment $\mathbf{x}$ is defined as the relative difference between the maximum and minimum resource expenditure, given by

$$V(\mathbf{x}) = 1 - \frac{\min_j \Lambda(\mathbf{x})}{\max_j \Lambda(\mathbf{x})}.$$

The jamming resource variation measures the balance in resource expenditure over the set of jammers. If the resource variation is large, i.e. near 1, some jammers will fully exhaust their battery energy and be unable to participate in the flow-jamming attack, thus degrading the lifetime of the attack. In, on the other hand, the resource variation is small, i.e. near 0, then the minimum jammer lifetime will be maximized, thus prolonging the duration of the flow-jamming attack.

## III. Centralized Flow-Jamming Attacks

In this section, we formulate flow-jamming attacks which are optimal with respect to the evaluation metrics proposed in Section II-C. We first present the *maximum impact flow-jamming attack*, using the jamming impact $I(\mathbf{x})$ as the primary optimization metric and the jamming efficiency $E(\mathbf{x})$ as a secondary optimization metric. We next present the *efficient flow-jamming attack*, using the jamming efficiency as the primary optimization metric. We then present the *balanced flow-jamming attack*, using the jamming resource variation $V(\mathbf{x})$ as the primary optimization metric and the jamming impact $I(\mathbf{x})$ as the secondary optimization metric. Each flow-jamming attack is formulated as an optimization problem, and each is solved using linear programming techniques.

The jammer-to-flow assignment $\mathbf{x}$ corresponding to any flow-jamming attack must satisfy the following constraints. The resource expenditure $\lambda_j(\mathbf{x}_j)$ for each jammer $j \in \mathcal{J}$, as defined in (1), must satisfy the *supply constraint*

$$\lambda_j(\mathbf{x}_j) \leq 1, \tag{2}$$

as each jammer cannot exhaust more than the available resources. The assignment of jammers to each flow $f \in \mathcal{F}$ must additionally satisfy the *flow constraint*

$$\|\mathbf{x}_f\|_1 \leq 1, \tag{3}$$

as the jammers cannot jam more flow than is present.

### A. Maximum Impact Flow-Jamming Attacks

The maximum impact flow-jamming attack primarily maximizes the jamming impact $I(\mathbf{x})$ and secondarily maximizes the jamming efficiency $E(\mathbf{x})$. We develop an algorithm for maximum impact flow-jamming attacks by deriving a linear program corresponding to each of two cases: $I(\mathbf{x}) = 1$ and $I(\mathbf{x}) < 1$.

The case of $I(\mathbf{x}) = 1$ corresponds to the ability to achieve equality in the flow constraint in (3) for all $f \in \mathcal{F}$. If this condition can be achieved for the given resource supply variables $c_j$ for $j \in \mathcal{J}$ and network and jammer topology, the jamming efficiency is maximized by minimizing the total resource expenditure $\|\Lambda(\mathbf{x})\|_1$ subject to the supply constraint in (2). The formulation of this flow-jamming attack is stated in Fig. 2(a) as a linear program. We note that the flow-jamming attack in this case is representative of the Hitchcock problem [9] for minimum cost resource allocation, where the flow constraint is interpreted as a resource demand.

The case of $I(\mathbf{x}) < 1$ corresponds to the inability to achieve equality in the flow constraint in (3) for all $f \in \mathcal{F}$. In this case, we note that each jammer $j \in \mathcal{J}$ will contribute as much of the available resource supply as possible to the subset of $\mathcal{F}$ of flows $f$ with cost $c_{jf} < \infty$. Hence, the jamming impact and efficiency are simultaneously maximized by maximizing the total fraction of jammed flow rate $\|\mathbf{x}\|_1$ subject to the supply constraint in (2) and the flow constraint in (3). The formulation of this flow-jamming attack is stated in Fig. 2(b) as a linear program.

The combination of the linear programs in Fig. 2 yields the desired centralized algorithm for maximum impact flow-jamming attacks. The first step of the algorithm is to attempt to solve the linear program in Fig. 2(a), yielding the jammer-to-flow assignment vector $\mathbf{x}$ which maximizes $E(\mathbf{x})$ for $I(\mathbf{x}) = 1$. If a feasible solution to the first linear program does not exist, a solution to the linear program in Fig. 2(b) is computed. Since the algorithm involves computing solutions to linear programs, it runs in polynomial time [9]. Furthermore, a feasible solution is guaranteed, as the trivial solution $\mathbf{x} = 0$ is feasible for the linear program in Fig. 2(b).

### B. Efficient Flow-Jamming Attacks

The efficient flow-jamming attack aims to maximize the jamming efficiency $E(\mathbf{x})$ subject to the supply constraint in (2) and the flow constraint in (3). The optimization problem is formulated in Fig. 3(a). However, since $E(\mathbf{x})$ is not a linear

$$\begin{array}{ll} \min & \|\Lambda(\mathbf{x})\|_1 \\ \text{s.t.} & \lambda_j(\mathbf{x}_j) \le 1 \text{ for all } j \in \mathcal{J}, \\ & \|\mathbf{x}_f\|_1 = 1 \text{ for all } f \in \mathcal{F}, \\ & 0 \le x_{jf} \le 1 \text{ for all } j \in \mathcal{J}, f \in \mathcal{F}. \end{array}$$

(a)

$$\begin{array}{ll} \max & \|\mathbf{x}\|_1 \\ \text{s.t.} & \lambda_j(\mathbf{x}_j) \le 1 \text{ for all } j \in \mathcal{J}, \\ & \|\mathbf{x}_f\|_1 \le 1 \text{ for all } f \in \mathcal{F}, \\ & 0 \le x_{jf} \le 1 \text{ for all } j \in \mathcal{J}, f \in \mathcal{F}. \end{array}$$

(b)

Fig. 2. The maximum impact flow-jamming attack algorithm first attempts to solve the linear program in (a) with equality in the flow constraint (3). If no solution is feasible, the linear program in (b) is solved.

$$\begin{array}{ll} \max & \frac{|\mathcal{F}|^{-1}\|\mathbf{x}\|_1}{|\mathcal{J}|^{-1}\|\Lambda(\mathbf{x})\|_1} \\ \text{s.t.} & \lambda_j(\mathbf{x}_j) \le 1 \text{ for all } j \in \mathcal{J}, \\ & \|\mathbf{x}_f\|_1 \le 1 \text{ for all } f \in \mathcal{F}, \\ & 0 \le x_{jf} \le 1 \text{ for all } j \in \mathcal{J}, f \in \mathcal{F}. \end{array}$$

(a)

$$\begin{array}{ll} \min & |\mathcal{J}|^{-1}\|\Lambda(\mathbf{x})\|_1 - \epsilon^{-1}|\mathcal{F}|^{-1}\|\mathbf{x}\|_1 \\ \text{s.t.} & \lambda_j(\mathbf{x}_j) \le 1 \text{ for all } j \in \mathcal{J}, \\ & \|\mathbf{x}_f\|_1 \le 1 \text{ for all } f \in \mathcal{F}, \\ & 0 \le x_{jf} \le 1 \text{ for all } j \in \mathcal{J}, f \in \mathcal{F}. \end{array}$$

(b)

Fig. 3. The optimal efficient flow-jamming attack can be obtained by solving the non-linear optimization problem in (a). The linear approximation in (b) yields a solution within an additive constant $\epsilon$ of the optimal solution.

function in the jammer-to-flow assignment variables, we provide a linear approximation in Fig. 3(b) which approximates the optimal solution of the problem in Fig. 3(a) using a linear objective function. The following result proves that the linear approximation is tight.

*Theorem 1:* Let $\mathbf{x}^*$ denote the non-zero jammer-to-flow assignment which maximizes $E(\mathbf{x})$ and $\hat{\mathbf{x}}$ denote the non-zero jammer-to-flow assignment which minimizes $|\mathcal{J}|^{-1}\|\Lambda(\mathbf{x})\|_1 - \epsilon^{-1}|\mathcal{F}|^{-1}\|\mathbf{x}\|_1$ for a given $\epsilon > 0$. Then $0 \le E(\mathbf{x}^*) - E(\hat{\mathbf{x}}) \le \epsilon$.

*Proof:* Let $g(\mathbf{x}) = |\mathcal{J}|^{-1}\|\Lambda(\mathbf{x})\|_1$ and $\ell(\mathbf{x}) = |\mathcal{F}|^{-1}\|\mathbf{x}\|_1$ such that $E(\mathbf{x}) = \ell(\mathbf{x})/g(\mathbf{x})$ and $\mathbf{x}^*$ is given by

$$\begin{aligned} \mathbf{x}^* &= \arg\max_{\mathbf{x}>0} E(\mathbf{x}) = \arg\max_{\mathbf{x}>0} \frac{\ell(\mathbf{x})}{g(\mathbf{x})} \\ &= \arg\min_{\mathbf{x}>0} \frac{g(\mathbf{x})}{\ell(\mathbf{x})} = \arg\min_{\mathbf{x}>0} \epsilon\frac{g(\mathbf{x})}{\ell(\mathbf{x})} - 1, \end{aligned} \tag{4}$$

where (4) holds because the optimal solution is not changed by an affine transformation of the objective function. The solution $\hat{\mathbf{x}}$ is similarly given by

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}>0} g(\mathbf{x}) - \epsilon^{-1}\ell(\mathbf{x}). \tag{5}$$

Optimality of the corresponding solutions $\mathbf{x}^*$ and $\hat{\mathbf{x}}$ implies the inequalities

$$\frac{\ell(\mathbf{x}^*)}{g(\mathbf{x}^*)} \ge \frac{\ell(\hat{\mathbf{x}})}{g(\hat{\mathbf{x}})}, \tag{6}$$

$$\frac{g(\mathbf{x}^*) - \epsilon^{-1}\ell(\mathbf{x}^*)}{\epsilon^{-1}\ell(\mathbf{x}^*)} \le \frac{g(\hat{\mathbf{x}}) - \epsilon^{-1}\ell(\hat{\mathbf{x}})}{\epsilon^{-1}\ell(\hat{\mathbf{x}})}, \tag{7}$$

$$g(\hat{\mathbf{x}}) - \epsilon^{-1}\ell(\hat{\mathbf{x}}) \le g(\mathbf{x}^*) - \epsilon^{-1}\ell(\mathbf{x}^*). \tag{8}$$

The combination of (7) and (8) yields the inequality $\ell(\mathbf{x}^*) \ge \ell(\hat{\mathbf{x}})$. This result and (8) yield the inequality

$$g(\mathbf{x}^*) - g(\hat{\mathbf{x}}) \ge \epsilon^{-1}\left(\ell(\mathbf{x}^*) - \ell(\hat{\mathbf{x}})\right) \ge 0. \tag{9}$$

If $g(\mathbf{x}^*) = g(\hat{\mathbf{x}})$ then $E(\mathbf{x}^*) = E(\hat{\mathbf{x}})$, so the proof holds. Hence, for the remainder of the proof, assume $g(\mathbf{x}^*) > g(\hat{\mathbf{x}})$.

Multiplying (6) through by $g(\mathbf{x}^*)g(\hat{\mathbf{x}})$, subtracting the term $\ell(\mathbf{x}^*)g(\mathbf{x}^*)$, and rearranging non-zero terms yields the inequality

$$\frac{\ell(\mathbf{x}^*)}{g(\mathbf{x}^*)} \le \frac{\ell(\mathbf{x}^*) - \ell(\hat{\mathbf{x}})}{g(\mathbf{x}^*) - g(\hat{\mathbf{x}})}. \tag{10}$$

Combining the inequalities in (6), (9), and (10) yields

$$\frac{\ell(\hat{\mathbf{x}})}{g(\hat{\mathbf{x}})} \le \frac{\ell(\mathbf{x}^*)}{g(\mathbf{x}^*)} \le \epsilon. \tag{11}$$

Since both $\frac{\ell(\hat{\mathbf{x}})}{g(\hat{\mathbf{x}})}$ and $\frac{\ell(\mathbf{x}^*)}{g(\mathbf{x}^*)}$ are positive and bounded by $\epsilon$, their difference is also bounded by $\epsilon$. ■

### C. Balanced Flow-Jamming Attacks

The balanced flow-jamming attack primarily minimizes the jamming resource variation $V(\mathbf{x})$ and secondarily maximizes the jamming impact $I(\mathbf{x})$ and jamming efficiency $E(\mathbf{x})$. We develop an algorithm for balanced flow-jamming attacks by deriving a linear program corresponding to each of two cases: $I(\mathbf{x}) = 1$ and $I(\mathbf{x}) < 1$.

The case of $I(\mathbf{x}) = 1$ corresponds to the ability to achieve equality in the flow constraint in (3) for all $f \in \mathcal{F}$. If this condition can be achieved for the given resource supply variables $c_j$ for $j \in \mathcal{J}$ and network and jammer topology, the jamming resource variation is minimized with maximum jamming impact by minimizing the variable $\lambda = \max_j \Lambda(\mathbf{x})$ subject to the supply constraing in (2). In this case, minimizing $\lambda$ corresponds to maximizing the jamming efficiency $E(\mathbf{x})$. This flow-jamming attack can be formulated as a linear program by aiming to maximize $\lambda$ with the additional constraint $\lambda_j(\mathbf{x}_j) \le \lambda$ for all $j \in \mathcal{J}$, a stronger constraint than the supply

$$\begin{array}{ll} \min & \lambda \\ \text{s.t.} & \lambda_j(\mathbf{x}_j) \le \lambda \text{ for all } j \in \mathcal{J}, \\ & \|\mathbf{x}_f\|_1 = 1 \text{ for all } f \in \mathcal{F}, \\ & 0 \le x_{jf} \le 1 \text{ for all } j \in \mathcal{J}, f \in \mathcal{F}, \\ & 0 \le \lambda \le 1. \end{array}$$

(a)

$$\begin{array}{ll} \max & \lambda \\ \text{s.t.} & \lambda \le \lambda_j(\mathbf{x}_j) \le 1 \text{ for all } j \in \mathcal{J}, \\ & \|\mathbf{x}_f\|_1 \le 1 \text{ for all } f \in \mathcal{F}, \\ & 0 \le x_{jf} \le 1 \text{ for all } j \in \mathcal{J}, f \in \mathcal{F}, \\ & 0 \le \lambda \le 1. \end{array}$$

(b)

Fig. 4. The balanced flow-jamming attack algorithm first attempts to solve the linear program in (a) with equality in the flow constraint (3). If no solution is feasible, the linear program in (b) is solved.

constraint in (2). The formulation of this flow-jamming attack is stated in Fig. 4(a) as a linear program.

The case of $I(\mathbf{x}) < 1$ corresponds to the inability to achieve equality in the flow constraint in (3) for all $f \in \mathcal{F}$. In this case, the resource variation $V(\mathbf{x})$ is minimized with maximum jamming impact $I(\mathbf{x})$ and efficiency $E(\mathbf{x})$ by maximizing $\lambda = \min_j \Lambda(\mathbf{x})$ subject to the supply constraint in (2) for all $j \in \mathcal{J}$ and the flow constraint in (3) for all $f \in \mathcal{F}$. This flow-jamming attack can be formulated as a linear program by maximizing $\lambda$ subject to the additional constraint that $\lambda \le \lambda_j(\mathbf{x}_j)$, introducing a lower bound into the supply constraint in (2) for each $j \in \mathcal{J}$. The formulation of this flow-jamming attack is stated in Fig. 4(b) as a linear program.

The combination of the linear programs in Fig. 4 yields the desired centralized algorithm for balanced flow-jamming attacks. The centralized algorithm is obtained using a similar technique to that in Section III-A, first attempting to solve the linear program in Fig. 4(a), solving that in Fig. 4(b) if no solution is feasible. As previously discussed, the algorithm runs in polynomial time and is guaranteed to have a feasible solution.

## IV. Distributed Flow-Jamming Attacks

In this section, we develop a distributed flow-jamming attack algorithm in which each jammer $j$ uses local information to compute the jammer-to-flow assignment $\mathbf{x}_j$. We provide an algorithm which minimizes resource expenditure and maximizes jamming impact $I(\mathbf{x})$. We note that this approach may lead to a higher jamming efficiency $E(\mathbf{x})$ than the efficient flow-jamming attack in Section III-B and discuss this phenomenon in Section IV-B. We assume that each jammer $j$ exchanges information with a subset $\mathcal{J}_j \subset \mathcal{J}$ of neighboring jammers and has knowledge of the subset $\mathcal{F}_j \subseteq \mathcal{F}$ of flows for which $c_{jf} < \infty$. We assume the variables $c_{jf}/c_j$ are distinct for all $j \in \mathcal{J}$ and $f \in \mathcal{F}$, noting that this assertion holds with probability 1 if there is any source of randomness in $c_{jf}$. We address the distributed attack in general and discuss the impact of the neighborhood size $|\mathcal{J}_j|$ on the algorithm performance. A description of the algorithm and the heuristic used to develop it are as follows, and the algorithm is given in Fig. 5.

### A. Heuristic for Efficient Flow-Jamming

We develop the distributed algorithm using a greedy heuristic based on the following observations. First, given two flows $f_1$ and $f_2$ and a single jammer $j$, the jamming efficiency is maximized by assigning resources to the flow with lower normalized cost $c_{jf}/c_j$ before assigning resources to the other flow. This allows the jammer $j$ to maximize the jammed flow rate at minimum resource expenditure. Second, given two jammers $j_1$ and $j_2$ and a single flow $f$, the jamming efficiency is maximized by assigning resources to the jammer with lower normalized cost $c_{jf}/c_j$ before assigning flow to the other jammer.

We next apply the greedy heuristic to the case of an arbitrary number of flows and jammers, computing a strict partial ordering [11] on the set $\mathcal{J} \times \mathcal{F}$ of ordered pairs $(j, f)$ and computing jammer-to-flow assignments according to the ordering. In a local neighborhood, each jammer $j \in \mathcal{J}$ must construct the sub-ordering on $\mathcal{J}_j \times \mathcal{F}_j$ and assign resources to flows according to the sub-ordering. At a given instant during the execution of the attack, each jammer $j$ considers only the single flow $f^*$ with minimum normalized cost $c_{jf^*}/c_j$, only assigning resources to $f^*$ if there is no neighboring jammer $j'$ for which $c_{j'f^*}/c_{j'} < c_{jf^*}/c_j$.

To perform the heuristic algorithm, each jammer $j$ must know the normalized costs $c_{j'f}/c_{j'}$ for each neighboring jammer $j' \in \mathcal{J}_j$ and flow $f \in \mathcal{F}_j$. The normalized costs are constant, but can be updated to $\infty$ for any neighboring jammer that will not contribute further to a flow. Furthermore, in assigning resources to a flow $f$, jammer $j$ must know the fraction $y_f$ of flow which is already assigned to neighboring jammers $j' \in \mathcal{J}_j$. The variables $y_f$ must be updated during the attack to reflect to progressive assignment of flow to neighboring jammers. Similarly, jammer $j$ must notify the neighboring jammers when each jammer-to-flow assignment variable $x_{jf}$ is determined and when its resources have been exhausted. The distributed flow-jamming attack algorithm is presented in its entirety in Fig. 5.

If the flow $f^*$ with minimum normalized cost for jammer $j$ can be assigned to a neighboring jammer with lower normalized cost, jammer $j$ is required to wait for a notification. If every jammer in $\mathcal{J}$ is simultaneously waiting for notifications, however, the algorithm will stall indefinitely. Termination of

**Distributed Flow-Jamming Attack for $j \in \mathcal{J}$**

$x_{jf} \leftarrow 0$ for $f \in \mathcal{F}_j$

$y_f \leftarrow 0$ for $f \in \mathcal{F}_j$

**while** $\lambda_j(\mathbf{x}_j) < 1$ **and** $\{f \in \mathcal{F}_j : x_{jf} = 0, y_f < 1\} \neq \varnothing$

  $f^* \leftarrow \underset{f \in \mathcal{F}_j : x_{jf}=0, y_f<1}{\arg\min} c_{jf}/c_j$

  **if** $c_{jf^*}/c_j < c_{j'f^*}/c_{j'}$ for all $j' \in \mathcal{J}_j$

    $x_{jf^*} \leftarrow \min\left(1 - y_f, \frac{1-\lambda_j(\mathbf{x}_j)}{c_j^{-1} c_{jf^*} r_{f^*}}\right)$

    transmit $(j, f^*, x_{jf^*})$

  **else**

    wait for notification

    **if** $(j', f, y)$ received

      $y_f \leftarrow y_f + y$

      $c_{j'f} \leftarrow \infty$

    **else if** $(j', \infty)$ received

      $c_{j'f} \leftarrow \infty$ for each $f \in \mathcal{F}_j$

    **end if**

  **end if**

**end while**

transmit $(j, \infty)$

Fig. 5. This distributed algorithm approximates the efficient flow-jamming attack algorithm given in Section III-B.

the distributed algorithm in finite time is guaranteed by the following result.

*Theorem 2:* The distributed flow-jamming attack algorithm in Fig. 5 terminates in finite time for all jammers $j \in \mathcal{J}$.

*Proof:* The algorithm stalls indefiniately if every jammer is simultaneously waiting for notifications. If a single jammer is not waiting, the resulting notification will allow neighboring jammers to progress. We thus show that there is always at least one jammer that is not waiting. We prove the desired result by constructing a directed graph $G = (V, E)$ corresponding to the partial ordering discussed above. The vertex set $V$ of $G$ is given by the set $\mathcal{J} \times \mathcal{F}$. A directed edge $((j_1, f_1), (j_2, f_2))$ is in the edge set $E$ of $G$ if and only if $c_{j_1 f_1}/c_{j_1} < c_{j_2 f_2}/c_{j_2}$ and either $f_1 = f_2$ or $j_1 = j_2$. Since it is constructed from the partial ordering, $G$ is an acyclic graph [11], as otherwise a cycle traversing the vertex $(j_1, f_1)$ in $G$ would represent a sequence of strict inequalities beginning and ending with $c_{j_1 f_1}/c_{j_1}$. Since $G$ is a directed acyclic graph, there must be at least one vertex $(j^*, f^*) \in V$ with no incoming edge [11]. By construction of the partial ordering, jammer $j^*$ will not wait for any other jammer to compute the jammer-to-flow assignment variable $x_{j^* f^*}$ for flow $f^*$. As the algorithm progresses and variables $x_{jf}$ are computed, the corresponding vertices $(j, f)$ are removed from $G$. Since any subgraph of a directed acyclic graph is a directed acyclic graph, there always exists such a vertex $(j^*, f^*)$. ∎

We note that the algorithm presented in this section assumes that each jammer $j$ knows the flow rate $r_f$ of each flow $f \in \mathcal{F}_j$. However, these parameters may not be available to the jammer $j$, especially if jammers are not exchanging information between neighborhoods. Hence, in this case, each jammer $j$ must compute an estimate $r'_{jf}$ of the residual throughput that has not been jammed by upstream jammers. Moreover, the algorithm does not account for over-provisioning of jamming resources between distant neighborhoods in the jammer network, so basing the attack on the residual flow $r'_{jf}$ and updating this quantity over time may reduce over-provisioning and reduce jamming resource expenditure. However, such an approach assigns a higher fraction of flow rate to jammers near the flow sources.

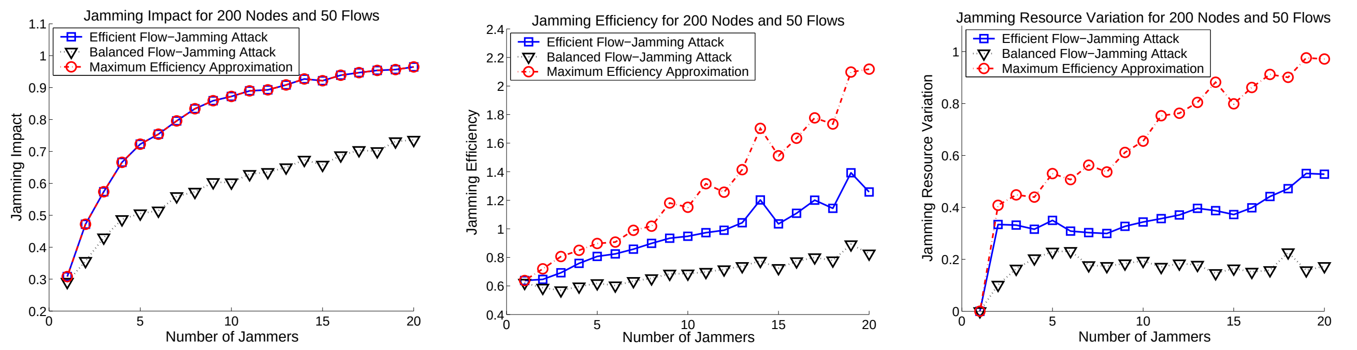
### *B. Impact of Neighborhood Size*

We expect the number of neighboring jammers $|\mathcal{J}_j|$ that exchange information with a jammer $j \in \mathcal{J}$ to significantly influence the performance of the distributed attack algorithm.

If the neighborhood size $|\mathcal{J}_j|$ is very large, the local sub-ordering of the strict partial ordering constructed by the heuristic algorithm consists of nearly the entire set $\mathcal{J} \times \mathcal{F}$. Hence, jammers may have to wait for a long period of time before computing jammer-to-flow variables. In addition, we note that the heuristic with global information can achieve a greater jamming efficiency $E(\mathbf{x})$ than the centralized efficient flow-jamming attack in Section III-B. However, this out-performance is achieved by decreasing both the jamming impact $I(\mathbf{x})$ and the resource expenditure $\|\Lambda(\mathbf{x})\|_1$, so the effect of the attack on the network traffic flows is reduced. The heuristic is able to achieve this greater jamming efficiency by effectively ignoring jammer-to-flow assignment variables $x_{jf}$ for which the corresponding normalized cost $c_{jf}$ is large enough to significantly increase resource expenditure for a negligible increase in jamming impact. We finally note that the communication overhead required to exchange with a large neighborhood size may be quite large, especially since every jammer may be required to wait for an extended period of time.

Alternatively, if the neighborhood size is small, there may be insufficient exchange of information such that the computed jammer-to-flow assignment $\mathbf{x}_f$ with respect to a single flow $f$ satisfies $\|\mathbf{x}_f\|_1 > 1$. In this case, multiple jammers are assigning resources to jam the same packets, leading to an increase in resource expenditure with no associated increase in impact. However, a smaller neighborhood size may allow the algorithm to conclude more rapidly and require significantly lower communication overhead to exchange information with neighboring jammers.

In the extreme case of $\mathcal{J}_j = \varnothing$, jammers do not exchange information, either to avoid revealing information to the network or to conserve resource expenditure required for communication. We note that the distributed algorithm in Fig. 5 can still be used, though many of the statements are




(a) (b) (c)

Fig. 6. We compare the maximum impact, efficient, and balanced flow-jamming attacks in Section III in terms of the (a) jamming impact, (b) jamming efficiency, and (c) jamming resource variation for the low-energy case and the specified network and jamming parameters.

vacuous. The distributed algorithm is equivalent in this case to the assignment of resources to the flows $\mathcal{F}_j$ increasing order of $c_{jf}$ until either $\lambda_j(\mathbf{x}_j) = 1$ or $\mathbf{x}_j = 1$.

## V. Performance Evaluation

In this section, we compare the jamming impact, efficiency, and resource variation for various flow-jamming attacks. We first compare the performance of the maximum impact, efficient, and balanced centralized flow-jamming attacks using the linear program formulations in Section III. We then compare the performance of the centralized efficient flow-jamming attack in Section III-B to that of the distributed flow-jamming attack using the algorithm in Section IV.

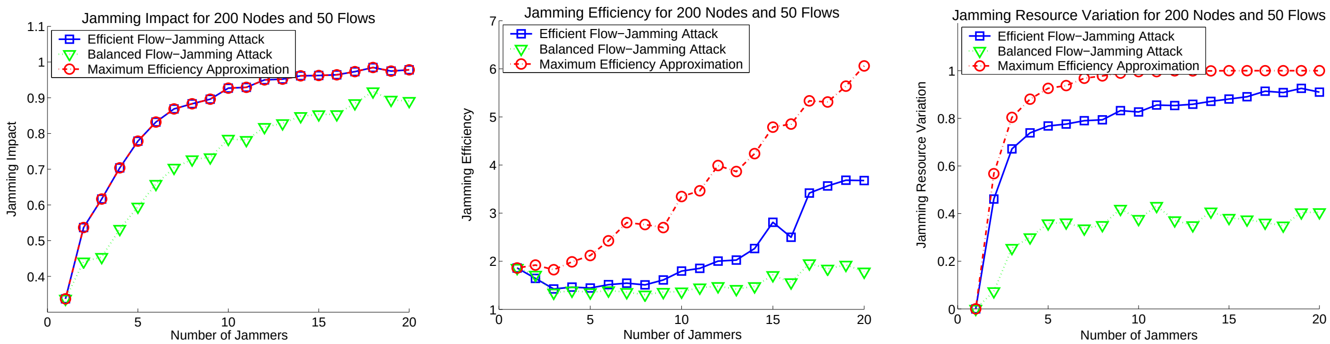
The simulated network in both cases consists of $|\mathcal{N}| = 200$ randomly deployed nodes with $|\mathcal{F}| = 50$ shortest-path flows between randomly selected source and destination nodes. The jammers in $\mathcal{J}$ are randomly deployed, and each cost $c_{jf}$ is proportional to the minimum distance from jammer $j$ to any non-source node participating in the flow $f$. The cost $c_{jf}$ is infinite if the minimum distance is beyond a fixed threshold equal to twice the communication range of the nodes in $\mathcal{N}$. The total jamming resources $\sum_{j\in\mathcal{J}} c_j$ are held constant and equally distributed to the jammers, effectively distributing the resources more evenly over the network as $|\mathcal{J}|$ increases. We perform two sets of simulations for each case to illustrate the effect of the total jamming resource, referring to these simulation sets as the *low-energy* and *high-energy* cases. In the low-energy case, the total resources are computed as that required to completely jam 10 flows of rate $r_f = 1$ using the maximum finite cost $c_{jf}$. In the high-energy case, the total resources are twice as much as in the low-energy case. Finally, each plotted curve illustrates an average over 100 simulated network instances to capture the average performance of the flow-jamming attacks.

### A. Comparison of Centralized Attacks

We simulate the three centralized flow-jamming attacks to compare the metrics of jamming impact, efficiency, and resource variation. Intuitively, we expect the jamming impact to be greatest for the maximum impact attack, followed by that of the efficient attack and the balanced attack. This is due to the fact that jamming impact is a secondary metric for the efficient and balanced attacks, and the balanced attack effectively imposes an additional constraint on the efficient attack. Similarly, we expect the jamming efficiency to be greatest for the efficient attack, followed by that of the maximum impact attack and the balanced attack. Finally, we expect the jamming resource variation to be least for the balanced attack, followed by the two remaining attacks, leading to the added benefit of a longer jammer lifetime under the balanced flow-jamming attack. Figs. 6 and 7 compare the jamming impact, efficiency, and resource variation for the centralized flow-jamming attack formulations in the low-energy and high-energy cases, respectively.

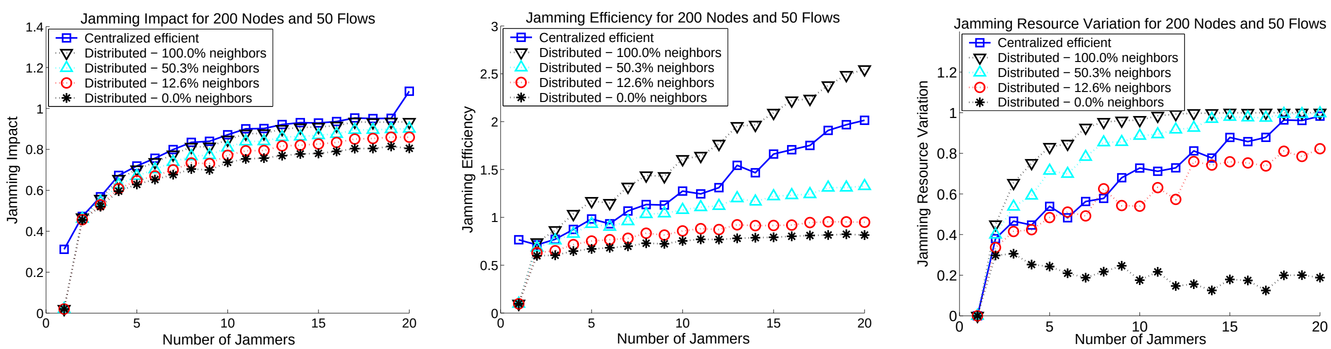
### B. Comparison of Centralized and Distributed Attacks

We simulate the centralized efficient flow-jamming attack presented in Section III-B and the distributed flow-jamming attack presented in Section IV to compare the metrics of jamming impact, efficiency, and resource variation. We expect the jamming impact resulting from the centralized attack to be greatest with that of the distributed attack increasing with the neighborhood size. We expect the jamming efficiency resulting from the distributed attack to similarly increase with the neighborhood size, and in some cases to out-perform the jamming efficiency of the centralized attack. However, we note that the distributed attack requires a significantly lower resource expenditure to achieve a lower jamming impact, achieved by ignoring high-cost flows as discussed in Section IV-B. We further expect the jamming resource variation to increase with neighborhood size. When the neighborhood size is large, the larger amount of information distributed throughout the network increases the chance that a distant jammer can jam a flow at a lower cost, so the well-positioned jammers in the network are likely to do a majority of the jamming. When the neighborhood size is small, each jammer exhaust a higher fraction of the available resources, though resources may be wasted. Figs. 8 and 9 compare the jamming impact, efficiency, and resource variation for the attacks of interest in the low-energy and high-energy cases, respectively.



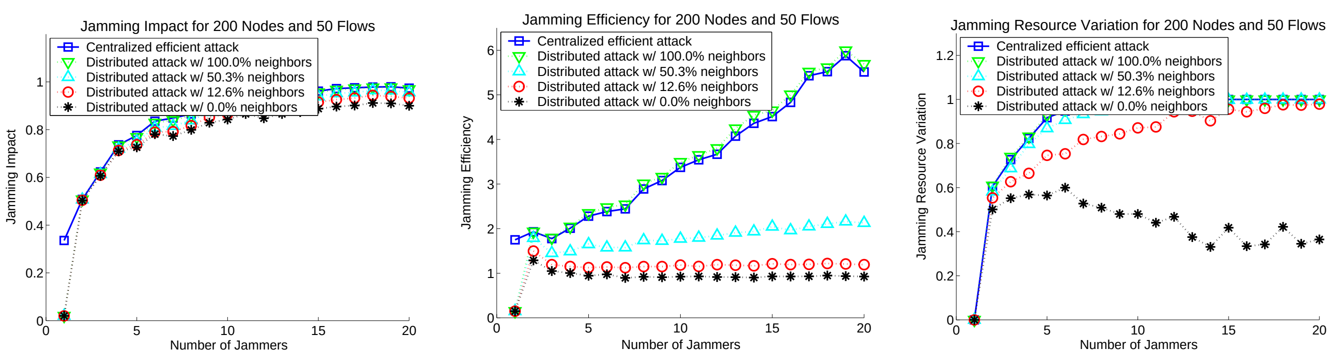

(a) (b) (c)

Fig. 7. We compare the maximum impact, efficient, and balanced flow-jamming attacks in Section III in terms of the (a) jamming impact, (b) jamming efficiency, and (c) jamming resource variation for the high-energy case and the specified network and jamming parameters.




(a) (b) (c)

Fig. 8. We compare the centralized efficient flow-jamming attack in Section III-B to the distributed attack in Section IV in terms of the (a) jamming impact, (b) jamming efficiency, and (c) jamming resource variation for the low-energy case and the specified network and jamming parameters. The jamming neighborhood size is specified as the average percentage of $|\mathcal{J}| - 1$.




(a) (b) (c)

Fig. 9. We compare the centralized efficient flow-jamming attack in Section III-B to the distributed attack in Section IV in terms of the (a) jamming impact, (b) jamming efficiency, and (c) jamming resource variation for the high-energy case and the specified network and jamming parameters. The jamming neighborhood size is specified as the average percentage of $|\mathcal{J}| - 1$.

We emphasize the fact that the distributed attack with sufficiently large neighborhood size out-performs the efficient flow-jamming attack in terms of jamming efficiency, as seen in Figs. 8(b) and 9(b). This is due to the reduction in jamming impact, as seen in Figs. 8(a) and 9(a) and the corresponding reduction in resource expenditure.

## VI. Conclusion

We presented and modeled the efficient *flow-jamming attack* in which an adversary selectively jams packets in network traffic flows. We proposed the evaluation metrics of jamming impact, efficiency, and resource variation and formulated optimal flow-jamming attacks with respect to these metrics using linear programming. We demonstrated the ability for a resource-constrained adversary to perform flow-jamming attacks efficiently with a significant impact on the network traffic flows. Furthermore, we showed that efficient flow-jamming attacks can be performed using a distributed algorithm in the absence of centralized control of the jammers.

## References


[1] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 2001.

[2] R. A. Poisel, *Modern Communication Jamming Principles and Techniques*. Artech House, 2004.

[3] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Prentice Hall, 2001.

[4] J. Bellardo and S. Savage, “802.11 denial-of-service attacks: Real vulnerabilities and practical solutions,” in *Proc. USENIX Security Symposium*, Washington, DC, Aug. 2003.

[5] D. J. Thuente and M. Acharya, “Intelligent jamming in wireless networks with applications to 802.11b and other networks,” in *Proc. 25th IEEE Communications Society Military Communications Conference (MILCOM’06)*, Washington, DC, Oct. 2006.

[6] A. D. Wood and J. A. Stankovic, “Denial of service in sensor networks,” *IEEE Computer*, vol. 35, no. 10, pp. 54–62, Oct. 2002.

[7] G. Lin and G. Noubir, “On link layer denial of service in data wireless LANs,” *Wireless Communications and Mobile Computing*, vol. 5, no. 3, pp. 273–284, May 2005.

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

[9] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Dover, 1998.

[10] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[11] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. MIT Press, McGraw-Hill, 2000.