MAINTAINING THROUGHPUT NETWORK CONNECTIVITY IN AD HOC NETWORKS

Ying Liu, Andrey Garnaev, Wade Trappe

WINLAB, Rutgers University, North Brunswick, NJ 08902, USA

ABSTRACT

This paper focuses on the challenge of maintaining reliable connectivity in an ad hoc network, where interference is possible. To cope with such interference, the paper introduces throughput connectivity and weighted throughput connectivity. Throughput connectivity reflects the possibility of establishing communication between nodes for a given power level. A weighted throughput protocol is more efficient in power allocation due to employing a continuous scale in Laplacian matrix. To illustrate these notions, two approaches to maximize connectivity were considered: (a) an adaptive transmission protocol that re-allocates transmission power between nodes, and (b) detecting and eliminating a malicious threat to maintain accumulated connectivity over time slots. The first problem was modeled by a maximin problem, and solved by Semi-Definite Programming. The second problem was modeled by a stochastic game and solved explicitly. 

Index Terms— Connectivity, Throughput Connectivity, Fiedler value, Jamming, Stochastic game

1. INTRODUCTION

In order for networks to be reliable, they must maintain their underlying connectivity, and resist to adversarial attack. An important characteristic of network connectivity is algebraic connectivity, as characterized by the network’s Fiedler value, which is the second smallest eigenvalue of the graph’s Laplacian. This measures how well-connected the graph is, and has been used to optimize a network’s design, and we now survey a few such works. A greedy heuristic algorithm was presented in [1], which adds edges (from a set of candidate edges) to a graph to maximize its algebraic connectivity. A distributed algorithm for the estimation and control of the connectivity of ad hoc networks for random topologies was suggested in [2], while a steepest-descent algorithm was proposed for control of the algebraic connectivity in [3]. The problem of improving network connectivity by adding a set of relays to increase number of links between network’s nodes was considered in [4]. Its simplified version was reduced to a semi-definite programming optimization problem. In [5] a genetic algorithm and swarm algorithm were applied for finding the best positions of adding nodes to a network to meet trade off between deployment cost and network’s connectivity. A decentralized algorithm to increase the connectivity of a multi-agent system was suggested in [6]. In [7], a problem of finding the best vertex positional configuration to maximize Fiedler value of a weighted graph was studied. Finally note that besides algebraic connectivity the other type of connectivity (such as global message connectivity, worst-case connectivity, network bisec tion connectivity, and k-connectivity, see [8]) are used in networks depending on characteristics to be maintained and methods used.

We note that in all of these papers the possibility of establishing a new communication link in a network did not depend on signal interference. Interference, however, can lead to a significant impact since signals sent to establish new communication links also serve as a noise for all the other links and their signals, thereby reducing the network’s capacity for maintaining existing communication links. To deal with this problem, in this paper, two types of connectivity are introduced. First is throughput connectivity, which reflects the possibility of establishing communication between nodes for a given power level. Second is weighted throughput connectivity, which associates with each link a weight corresponding to that link’s throughput. To illustrate these notions, two approaches to maximizing connectivity were considered: (a) an adaptive transmission protocol that re-allocates transmission power between nodes, and (b) detecting and eliminating a malicious threat to maintain accumulated connectivity over time slots. 

The first problem is modeled by a maximin problem, and is solved by a generic method. The second problem is modeled by a stochastic game and solved explicitly. Example applications of stochastic games in modeling network security can be found in [9, 10, 11, 12] and [13]. We also note that there is quite an extensive literature on detecting an intruder’s signal or its source (see, for example, books [14, 15, 16], papers on the detection of unknown signals [17, 18, 19, 20] and on game-theoretic modeling of spectrum scanning [21, 22]).

The paper is organized as follows. In Section 2, the new notions for a network’s connectivity are defined. In Section 3, the problem of designing an optimal transmission protocol to maximize a network’s connectivity is considered. In Section 4, an optimal scanning protocol to maintain a network’s connectivity is explored.

2. NETWORKS’ CONNECTIVITY

We model a wireless network consisting of n nodes in radio range of each other. We denote a node by \( v_i = (x_{1i}, x_{2i}) \), \( i \in [1, n] \), where \((x_{1i}, x_{2i}) \) is the coordinate for node \( v_i \). Let
$V = \{v_i, i \in [1, n]\}$ be the set of all nodes. We assume that, when each node communicates, it emits the same power in all directions. Of course, due to fading gains, pathloss and mutual interference of the signals, not every signal can reach each receiver. Let $P = (P_1, \ldots, P_n)$ be the transmission power allocation where signal $P_i$ is the signal power sent by node $i$ to every other node. Interference between signals could take place, and its effect depends on the distance between the receiver and the sender. Namely, the throughput of received signal by node $j$ is

$$T_{ij}(P) = \begin{cases} 0, & \text{SINR}_{ij}(P) < \epsilon, \\ \ln(1 + \text{SINR}_{ij}(P)), & \text{SINR}_{ij}(P) \geq \epsilon, \end{cases}$$

where $\epsilon \geq 0$ is a threshold value for SINR, and $\text{SINR}_{ij}(P) = \left(\frac{h_i P_i}{d_{ij}^\theta}ight) (\sigma^2 + \sum_{k \neq i, k \neq j} h_k P_k / d_{kj}^\theta)$ with $\sigma^2$ is the background noise, $h_i$ is the fading channel gain, and $d_{ij}$ is the distance between node $v_i$ and node $v_j$.

To define a communication network’s topology beyond the nodes, $\textit{links}$ (edges) between nodes have to be established. Note that due to its communication background this topology has to depend on communication type maintained by the network. In this paper we consider symmetric communication, i.e., two nodes (say, node $i$ and node $j$) are considered to be linked if and only if $T_{ij}(P)$ and $T_{ji}(P)$ are positive. A link means a possibility to maintain communication. Since communication is symmetric, link is undirected. Denote the link between node $v_i$ and node $v_j$ by $e_{ij}$. Let $E(P)$ be the set of all links. It is clear that the graph $\Gamma(P) = (V, E(P))$ is simple, i.e., there is no self loop for each node and there are not multiple links connecting two nodes.

The graph $\Gamma(P)$, associated with a network, can be represented by the Laplacian matrix as

$$L_{ij}(\Gamma(P)) = \begin{cases} -1, & i \neq j \text{ and } v_i \text{ and } v_j \text{ are linked}, \\ 0, & i \neq j \text{ and } v_i \text{ and } v_j \text{ are not linked}, \\ -\sum_{k=1, k \neq i}^n L_{ik}, & i = j, \end{cases}$$

where $L_{ii}(\Gamma(P))$ equals the number of nodes connected with node $v_i$. Also, it is possible to consider a weighted network by assigning throughput as weight for each link, in which case the weighted network can be represented by a Laplacian matrix as

$$L_{ij}(\Gamma(P)) = \begin{cases} -w_{ij}, & i \neq j \text{ and } v_i \text{ and } v_j \text{ are linked}, \\ 0, & i \neq j \text{ and } v_i \text{ and } v_j \text{ are not linked}, \\ -\sum_{k=1, k \neq i}^n L_{ik}, & i = j, \end{cases}$$

where $w_{ij} = T_{ij}(P) + T_{ji}(P)$ is total throughput of symmetric communication between node $v_i$ and node $v_j$, and $L_{ii}(\Gamma(P))$ is the total throughput of symmetric communication between node $v_i$ and others nodes.

Since $L(\Gamma(P))$ is positive semi-definite and symmetric, its eigenvalues are all nonnegative. By ordering the eigenvalues in an increasing way, we have: $0 = \lambda_1(\Gamma(P)) \leq \lambda_2(\Gamma(P)) \leq \ldots \leq \lambda_n(\Gamma(P))$. The eigenvector corresponding to the first eigenvalue is always $e^T = (1, \ldots, 1)$. The second eigenvalue $\lambda_2(\Gamma(P))$ is the algebraic connectivity of the system, and is an indicator of how connected the graph is, and is also called the Fiedler value. To emphasize that we consider connectivity based on the fact that there is bi-directional throughput (above a threshold value) for a link, we will use the term throughput connectivity and throughput Fiedler value. For a fixed transmission protocol involving a power assignment $P$, the throughput Fiedler value can be found as solution of the following optimization problem

$$\lambda_2(\Gamma(P)) = \min_{y^T \succ -e, y^T y = 0} \left\{ y^T L(\Gamma(P)) y \right\}.$$

Let us illustrate the behavior of throughput connectivity by the following example. Let the network consist of five nodes $(0, 0), (1, 0), (0, 1), (1, 1)$ and $(2, 0, 5)$ (Figure 1(a)), and $h = 1, \sigma^2 = 2, \epsilon = 0.1, 0.25$ and $P = (10, 15, 20, 10)$ and $P_5$ varies from 0.2 to 40. Of course, increasing $\epsilon$ yields a decrease in total throughput (Figure 1(b)). Throughput connectivity is piece-wise constant versus varying of the power (in our case, $P_5$, see, Figure 1(c)), while weighted throughput connectivity is piece-wise continuous on $P_5$ (Figure 1(d)). Thus, weighted throughput connectivity is more sensitive than throughput connectivity to a variation of the power. In this example, we can observe that there is a continuum where throughput connectivity obtains its maximum, and the value of this maximum is not too sensitive to the threshold $\epsilon$ (in the considered example they coincide for $\epsilon = 0.1$ and $\epsilon = 0.25$, and are equal to 3). Also, we can observe that there is a reduction of the set where the throughput connectivity obtains its maximum on reducing the threshold $\epsilon$, but there is no simple monotonic dependence between throughput connectivity and threshold $\epsilon$.

For weighted throughput connectivity, such dependence could be observed, as well as the fact that it obtains its maximum for a unique $P_5$.

### 3. OPTIMAL TRANSMISSION PROTOCOL

The network provider might improve the network’s connectivity by varying transmission power vector. Let $\Pi$ be the set of feasible transmission protocols. For example, it could be $\Pi(\mathcal{P}) = \{P \geq 0 : \sum_{i=1}^n P_i = \mathcal{P}\}$, where $\mathcal{P}$ is the total power allowed by the network’s provider among the nodes. Then, the problem of optimal transmission power assignment is given as the following maximin problem:

$$\lambda_2(\Gamma(P)) = \max_{P \in \Pi(\mathcal{P})} \min_{y^T \succ -e, y^T y = 0} \left\{ y^T L(\Gamma(P)) y \right\}. \quad (1)$$

This maximization problem of the second smallest eigenvalue of the Laplacian matrix on its inner parameters is equivalent to the following optimization problem (see, [23]):

$$\max_z, \quad \mathcal{P}_z,$$

subject to

$$L(\Gamma(P)) - zI \succ 0, \quad P \in \Pi(\mathcal{P}) \quad \text{and} \quad z > 0,$$

where $I$ is the $n \times n$ identity matrix, and “$\succ$” represents positive definiteness. By definition, Laplacian matrix $L(\Gamma(P))$ is symmetric. Thus, $L(\Gamma(P)) - zI$ is also symmetric. Therefore, (2) belongs to Semi-Definite Programming (SDP) problems [24]. It can be solved by SDP optimization tools, such as SDPT3 [25, 26], SDPA-M [27, 28] and CSDP [29].
In this section, we consider a problem where an adversary wants to damage connectivity of a network $\Gamma$ by attacking its nodes, while an IDS (Intrusion Detection System), scanning nodes, intends to detect the adversary to stop his malicious activity. We assume that all the actions (scanning by the IDS and attacking by the adversary) are performed in discrete time slots $1, 2, \ldots, \infty$. At each time slot, the adversary can choose a node to attack, and the IDS can choose a node to scan. If node $i$ is attacked, then connectivity of the undamaged network $\Gamma_i = \Gamma \setminus \{v_i\}$ is $C_i$. If the rivals choose different nodes then the IDS gets connectivity for an un-jammed network as an instantaneous payoff, and the game moves to the next time slot and is played recursively with discount factor $\delta$. If the rivals choose the same node, then with probability $1 - \gamma$ the adversary is detected and eliminated from the network. Then, the network keeps on working, and the IDS gets as instantaneous payoff the discounted connectivity $C_0$ of the whole network. With probability $\gamma$ the adversary is not detected, the game moves to the next time slot and is played recursively with discount factor $\delta$. This game can be considered as a two-state (1 and 2) stochastic game $G = (G^1, G^2)$. State 1 represents the malicious state in which the network is vulnerable to an attack by the adversary, while state 2 represents the state in which the adversary is detected and is not a threat to the network anymore. Stochastic game $G = (G^1, G^2)$ can be described in matrix form as follows:

$$
G^1 = \begin{pmatrix}
C_1(1, 0) & C_2(1, 0) & \ldots & C_n(1, 0) \\
C_1(1 - \gamma) & C_2(1 - \gamma) & \ldots & C_n(1 - \gamma) \\
\vdots & \vdots & \ddots & \vdots \\
C_1(1 - \gamma) & C_2(1 - \gamma) & \ldots & C_n(1 - \gamma)
\end{pmatrix},
$$

$$
G^2 = \begin{pmatrix} 1 \end{pmatrix} C_0(0, 1).
$$

In state 1, matrix notation is used such that each entry corresponds to a pair of nodes $(i, j)$ chosen by the IDS and the adversary. The value in the left part of each entry is the instantaneous payoff (un-jammed connectivity) to the IDS in this zero-sum stochastic game, while the right part gives the probability distribution over the future states. Thus, if $i \neq j$ then the instantaneous payoff to the IDS is $C_j$, and the next state is state 1. If $i = j$ then the instantaneous payoff to the IDS is $C_j$, and the next state is state 1 with probability $\gamma$, and it is state 2 with probability $1 - \gamma$. Note that the payoff at the next epoch is discounted with discount rate $\delta$.

In state 2, the rivals are passive, since the adversary is detected and cannot attack the network anymore. The game cannot leave this safe state. At each time slot the IDS obtains the discounted payoff $C_0$, which is the connectivity of un-jammed network. Thus, the total accumulated discounted payoff in state 2 is equal to $(1 + \delta + \delta^2 + \ldots)C_0 = C_0/(1 - \delta)$. Thus, the game $G$ is equivalent just to the game $G^1$ with a single state. The game $G^1$ has a solution in (mixed) stationary strategies, i.e., the strategies that are independent of history and current time slot. A (mixed) stationary strategy to the IDS is a probability vector $p^T = (p_1, p_2, \ldots, p_n)$, where $p_i$ is the probability to scan node $i$ and $e^T p = 1$. A (mixed) stationary strategy to the jammer is a probability vector $q^T = (q_1, q_2, \ldots, q_n)$, where $q_i$ is the probability to jam node $i$, and $e^T q = 1$. Solution of the game $G^1$ is given as a solution to the Shapley (-Bellman) equation game [30]:
In this paper, the concept of throughput connectivity and weighted throughput connectivity was introduced to to describe the reliability of a network’s communication in the presence of signal interference due to an adversary. In particular, we have shown a difference between selfish and cooperative power allocation, namely, selfishly increasing power by a node might reduce network connectivity, while cooperative allocation improves the connectivity. Also, we have shown how a repeated jamming attack could impact the accumulated network connectivity, and how to reduce this impact by designing a scanning protocol. Our future work is focused on applying the new connectivity metrics we introduced to other scenarios, including moving nodes in a theater for coordinating network connectivity, and to apply the throughput connectivity to Massive MIMO 5G networks.

5. CONCLUSIONS
6. REFERENCES


