Distributed Throughput Maximization in Wireless Networks Using the Stability Region

Gustavo Vejarano, Member, IEEE, Dexiang Wang, Member, IEEE, Ritwik Dubey, Member, IEEE, and Janise McNair, Member, IEEE

Abstract—In this paper, a game-theoretical framework for the design of distributed algorithms that control the transmission range (TR) of nodes in order to maximize throughput in Wireless Multihop Networks (WMNs) is proposed. It is based on the stability region of the link-scheduling policy adopted for the network. The stability region is defined as the set of input-packet rates under which the queues in the network are stable (i.e., positive recurrent). The goal of the TR-control algorithms is to adapt the stability region to a given set of end-to-end flows. In the algorithms, the flows control distributively the nodes’ TRs using the stability region in order to enable higher end-to-end packet rates while guaranteeing stability. In order to demonstrate how the algorithms can be designed using the proposed game-theoretical framework, a new TR-control algorithm for IEEE-802.16 WMNs is developed. Its convergence is demonstrated, and a performance bound is calculated. Finally, simulation results show that the algorithm is able to find the optimal TRs more effectively. The TRs achieve throughput levels that are at least 90 percent of the optimal throughput for 72 percent of the simulated scenarios, whereas the classic approach of spatial-reuse maximization does this for 62 percent of the scenarios.

Index Terms—Transmission-range control, transmission power, stability region, potential games

1 INTRODUCTION

T he stability region of wireless multihop networks (WMNs) is a key factor of their performance. The end-to-end throughput and delay experienced by flows established across the network are directly related to the stability region: the larger the region, the lower the delay and the higher the throughput flows can support under longer distances (i.e., number of hops).

Our motivation is the maximization of the total end-to-end throughput when a set of flows is given. The set of flows represents the traffic between end users that is generated by their applications such as end-to-end video/audio sessions. The maximization is done via stability-region and transmission-range (TR) control that the flows perform. Specifically, our approach consists of a distributed TR-control algorithm that is executed by the flows in the network. First, each flow identifies all other flows it interferes with when there are packet transmissions along its path, and then, the flows control collaboratively the TR of nodes in the network in order to maximize the total end-to-end throughput while guaranteeing stability. The distributed and collaborative aspects of our approach are formulated using game theory.

The results of our TR-control algorithm show that in order to maximize the total throughput for a given set of flows, the stability region of the network’s link-scheduling policy needs to be considered. Therefore, heuristics that ignore the stability region do not always perform well. For example, the classic approach of increasing the spatial reuse [1], [2], [3], [4], [5] in the network is outperformed by our TR-control approach. This is also the case of [6], [7], [8], where optimal topologies for the Greedy Maximal Scheduling (GMS) policy are identified, and the case of [9], where a randomized transmission-power control algorithm is proved to be throughput optimal for wireless networks that operate under the pick-and-compare scheduling policy [10], [11], [12], [13]. Therefore, there is not a general rule that can be applied to all networks for controlling their nodes’ transmission powers. It is necessary to consider its stability region, which can be adapted by means of TR control.

Based on this observation, we propose a game-theoretical framework that, different from the current approaches (see related work and contributions in Appendix 1, which is available in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.202), is valid for any scheduling policy whose stability region has been characterized. We apply our framework to the particular case of the policy proposed in [14] for IEEE-802.16 WMNs [15].

Our results also show that if routing is considered (i.e., determining the throughput-optimal paths for the flows’ source and destination nodes) along with TR control, the throughput-optimal paths are those that avoid congestion. This is in agreement with the results in [16], [17], where it is proven that an opportunistic routing policy that routes packets along the paths with an expected low overall congestion is throughput-optimal.

The paper is organized as follows. Section 2 describes the network model. In Section 3, the potential game for
maximizing the total end-to-end throughput is proposed and analyzed. In order to demonstrate how the potential game can be used for a specific network, a distributed TR-control algorithm for IEEE 802.16 WMNs is developed in Section 4. Also, a performance bound for the algorithm is calculated, and the proposed algorithm is compared with the centralized and heuristic algorithm of [18] and the classic approach of spatial-reuse maximization (i.e., nodes transmitting at minimum TRs that guarantee connectivity).

Finally, the paper is concluded in Section 5.

2 NETWORK MODEL

A WMN whose communication graph is denoted by \( \mathcal{G} = (\mathcal{N}, \mathcal{L}) \) is considered. \( \mathcal{N} \) is the set of nodes in the network, and \( \mathcal{L} \) is the set of links. Links are directional. The link directed from node \( i \) to node \( j \) is denoted by \((i, j)\). The nodes’ transmissions are omnidirectional, and a link belongs to \( \mathcal{L} \) if and only if the TR of the source node covers the destination node and the link belongs to at least one of the the data flows defined below, i.e., if a node’s TR covers another node, but it never sends packets to the covered node, then there is no link between these two nodes.

The vector of TRs of the nodes in \( \mathcal{N} \) is denoted by \( \mathbf{r} \), and the vector of their maximum TRs is denoted by \( \mathbf{r}_{\text{max}} \). The Euclidean distance between nodes \( i \) and \( j \) is denoted by \( ||i, j|| \). It is assumed that nodes that do not belong to any data flow have zero TR (i.e., they do not perform any type of transmissions).

The interference set of link \((i, j)\) is denoted by \( \mathcal{E}_{\text{int}}^{(i,j)} \). It contains all the links in \( \mathcal{L} \) that interfere with link \((i, j)\), and it is a function of \( \mathbf{r} \). The interference set of link \((i, j)\) under the TRs given by \( \mathbf{r}_{\text{max}} \) is denoted by \( \mathcal{E}_{\text{int}}^{(i,j)}(\mathbf{r}) \). The transmission packet rate of link \((i, j)\) is denoted by \( \lambda_{ij} \), and the vector of link packet rates is denoted by \( \mathbf{\lambda} = [\lambda_{ij}]_{(i,j) \in \mathcal{E}} \). The interference set of node \( j \) is denoted by \( \mathcal{E}_{\text{int}}^{(j)} \). It includes the links that interfere with at least one of \( j \)’s incoming links (i.e., \( \mathcal{E}_{\text{int}}^{(j)} = \{ \mathcal{E}_{\text{int}}^{(i,j)} : (i, j) \in \mathcal{L} \} \)).

The data traffic consists of a set of flows denoted by \( \mathcal{F} \). The flows in \( \mathcal{F} \) are enumerated. The \( n \)-th flow is denoted by \( f_n \). It consists of a path and a mean input-packet rate which are denoted by \( \mathcal{P}_{in} \) and \( \lambda_{in} \), respectively (i.e., \( f_n = (\mathcal{P}_{in}, \lambda_{in}) \)). Path \( \mathcal{P}_{in} \) is the set of nodes on which flow \( f_n \) is established. The data traffic on flow \( f_n \) is generated at its source node by a data-packet-arrival process that is Poisson distributed with mean \( \lambda_{in} \). The packets leave the network once they arrive at the flow’s destination node. The nodes in the path that forward the data packets from source to destination are the intermediate nodes. Every node that is an intermediate or destination node of at least one flow has a maximum packet rate that it can assign to its incoming traffic while guaranteeing the stability of its incoming links’ queues. Each of these nodes equally divides its maximum packet rate among all the flows for which it is an intermediate or destination node. The maximum packet rate that node \( j \) supports for each of these flows while guaranteeing stability is denoted by \( \lambda_{\text{max}} \). It is a function of \( \mathcal{E}_{\text{int}}^{(j)} \).

The degree of a link is defined as the number of flows it belongs to. It is denoted by \( d_{\text{in}} \) for link \((i, j)\).

Node \( j \) is a 1-hop neighbor of node \( i \) if node \( i \) is within node \( j \)’s TR. Node \( j \) is a 2-hop neighbor of node \( i \) if node \( j \) is a 1-hop neighbor of any of node \( i \)’s 1-hop neighbors. The active 1-hop neighborhood of a node is the set of 1-hop neighbors that are intermediate or destination nodes of at least one flow. It is denoted by \( \mathcal{S}_i^{(r)} \) for node \( i \), and it is a function of \( r \). The direct 1-hop neighborhood of a node is the set of 1-hop neighbors that send data packets to the node. Therefore, the direct 1-hop neighbors of a node always precede the node in at least one flow’s path. Node \( i \)’s direct 1-hop neighborhood is denoted by \( \mathcal{S}_i \).

Time is divided into frames, and each frame is divided into a control-subframe and a data-subframe. Control-subframes are divided into control-time-slots that are used for the transmission of scheduling packets, and data-subframes are divided into data-time-slots that are used for the transmission of data packets. A packet reception over link \((i, j)\) is successful if and only if no other link in \( \mathcal{E}_{\text{int}}^{(j)} \) is activated while the packet is being transmitted. If such link is activated, there is a collision at node \( j \) (i.e., the destination node of link \((i, j)\)) and the packet reception is unsuccessful.

A feasible schedule \( \mathcal{H} \) on \( \mathcal{H} \), where \( \mathcal{H} \) is some subset of \( \mathcal{L} \), is a link-activation vector [0, 1]_{\mathcal{H}} \) that when all the links in it are activated simultaneously, all the packet receptions are successful. A feasible schedule \( \mathcal{H} \) on \( \mathcal{H} \) is maximal if, when all the links in \( \mathcal{H} \) are activated, no more links can be activated without violating the interference constraints. The set of all possible maximal schedules on \( \mathcal{H} \) is denoted by \( \mathcal{M}_{\mathcal{H}} \), and its convex hull is denoted by \( \mathcal{C}(\mathcal{M}_{\mathcal{H}}) \).

Table 1 summarizes the previous notation. Also, in Appendix 2 available online, the guarantee of fairness among flows and the interference and packet-arrival models in this network model are discussed.

3 STABILITY-REGION ADAPTATION

We define the normalized transport capacity of a WMN based on the transport capacity defined in [20] and the queuing-system stability region defined in [21].

Definition 1. For a given set of flows, the normalized transport capacity (NTC) is the maximum total number of packets transported in the WMN from the flows’ sources to destinations per distance unit per time unit that guarantees the stability of all link queues.

For example, if there are 2 flows in the network, the distance between the flows’ source and destination is 5 meters, the links can transmit up to \( 10^9 \) bits per packet, and each flow can transport up to \( 10^3 \) bits per second while guaranteeing stability, the NTC is \( 2 \times 5 \times 10^4 = 10^2 \) packets-meter per second.

It should be noted that the NTC definition requires a prespecified set of flows. Therefore, the NTC is not an absolute performance metric of the WMN. It is a performance metric of the WMN for the given set of flows, so a WMN may have different NTC values for different sets of flows. Also, the NTC definition considers the geographical

1. Please see [14] for the derivation of this packet rate.

2. This is the frame structure for the WMNs in [14], [15], [19] and references therein.
distance from source to destination due to the important role it plays in the notion of fairness in WMNs [22].

The highest packet rate that a flow can transport while guaranteeing stability is determined by the packet rates that the intermediate and destination nodes can forward and receive respectively. Let $P_{int}$ be the set of intermediate and destination nodes of $f_n$ (i.e., $f_n$’s path without the source node), and let $\{\lambda_{max}(E^j_r) : j \in P_{int} \}$ for some $f_n \in F$ be the set of upper-bounds for the packet rates the nodes can forward/receive that guarantee stability. These upper-bounds are determined from the link-scheduling policy’s stability region as explained in Section 3.1. In order to guarantee stability of the WMN, the source node of every flow cannot generate data packets at rates higher than the minimum upper-bound among the upper-bounds of the intermediate and destination nodes of the flow, i.e., the WMN is stable if

$$\lambda^f < \min \{\lambda_{max}(E^j_r) : j \in P_{int} \} \quad \forall \lambda^f \in F. \quad (1)$$

The NTC is then determined by the distance between flows’ source and destination nodes and the upper-bounds for the flows’ packet rates given by (1). The source and destination locations are given and cannot be modified to increase the NTC. However, the stability region and the paths can be jointly controlled in order to increase the NTC. In this paper, we focus mainly on one of the two dimensions of the problem, i.e., we consider the problem of increasing the NTC by controlling the stability region distributively. For the second dimension, i.e., determining the paths of the flows, we show that the routing technique of avoiding nodes that experience high levels of contention maximizes the NTC.

### 3.1 Access Schemes and the Stability Region

The stability region is defined for WMNs in which time is slotted and interfering links take turns to access the slots such that collisions are avoided [21]. The link-scheduling policy coordinates the access of the links to the slots. The stability region of a link-scheduling is the set of packet-arrival rates for which the policy stabilizes the system. In terms of capacity, the optimal link-scheduling policy is the one whose stability region is a superset of the stability region of any other link-scheduling policy. It is given by $\Lambda = \{\lambda : \lambda \leq \phi \text{ for some } \phi \in C_0(M_L)\}$ [21]. In terms of complexity, it is usually the case that the link-scheduling policies with larger stability regions are more complex [23]. For example, the optimal scheduling policy [21] requires the solution of an NP-Hard problem (i.e., maximal weighted matching) [24] at every time-slot.

The stability region of different link-scheduling policies is characterized by a set of conditions that if satisfied, the WMN is guaranteed to be stable. The basic idea behind our NTC-adaptation approach is based on these sufficient conditions for stability. We claim that these can be controlled by manipulating the nodes’ TRs such that the highest packet rates the flows can support while guaranteeing stability are increased.

It should be noted that the conditions for stability of the different scheduling policies are sufficient but not necessary. Therefore, the conditions provide bounds that if met, the network is guaranteed to be stable. However, if they are not met, the network may still be stable. Therefore, in order to characterize the stability region of a policy, the tightness of the bounds calculated for the policy is important. When the bounds are tight, the stability region is characterized more accurately. This tightness plays an important role in our proposed TR-control approach because our approach adapts the stability region by means of the bounds. In this paper, we assume that the tightness of the bounds is sufficient to control the stability region successfully. We corroborate this by means of simulation in Appendix 10 available online. However, it is noted that calculating tight bounds is still an open problem as shown in [16], [26], [27].

In order to illustrate our TR-control approach, we first classify the sufficient conditions that guarantee stability into two main categories, and then, we show how these conditions can be used to control the stability region in order to increase the NTC.

#### 3.1.1 Greedy and Constant-Time Scheduling Policies

The link-scheduling policies proposed in [14], [19], [26], [28], [29], [30] belong to the first category. These policies are

3. See [11] for a review and comparison of the different link-scheduling policies, and see [14], [25], [26] for link-scheduling policies in IEEE 802.11, IEEE 802.16, and Aloha-based WMNs.

4. The link-scheduling policies in [10], [11], [12], [13] (i.e., pick-and-compare policies), are not discussed here since they already reach the optimal stability region, but at the cost of high end-to-end delays [9].
characterized by the sets of interfering links \( \{ E^{(i,j)}_r : (i, j) \in \mathcal{L} \} \) and an increasing function \( f^{(i,j)} : \lambda^{(i,j)}_r \to R^+ \), where \( \lambda^{(i,j)}_r \) is the set of transmission packet rates of the links in \( E^{(i,j)}_r \) (i.e., \( \lambda^{(i,j)}_r \triangleq \{ \lambda(k,l) : (k,l) \in E^{(i,j)}_r \} \)). Under these policies, the WMNs are stable if \( f^{(i,j)}(\lambda^{(i,j)}_r) < 0 \) for every \( (i, j) \in \mathcal{L} \). The intuition behind this is that the combination (i.e., \( f^{(i,j)} \)) of the transmission packet rates of interfering links (i.e., \( \lambda^{(i,j)}_r \)) cannot exceed a certain limit in order to guarantee the stability of the WMN. Otherwise, the queues of the interfering links may build up and never return to the empty state, making the network unstable. For example, greedy scheduling policies [19] require that the conditions in (2) be met, where \( \lambda^{(i,j)}_r \) is the capacity of link \((i, j)\) in units of packets-per-slot, and \( k \) is the maximum number of non-interfering links in the interference set of any link in the network

\[
f^{(i,j)} = \sum_{(k,l) \in E^{(i,j)}_r} \lambda^{(k,l)}_r - \kappa < 0 \quad \forall \quad (i, j) \in \mathcal{L}.
\] (2)

The sets of interfering links \( \{ E^{(i,j)}_r : (i, j) \in \mathcal{L} \} \) can be modified by means of TR control in order to minimize the rate at which the functions \( f^{(i,j)} \) increase with the transmission packet rates in \( \lambda^{(i,j)}_r \). In this way, the NTC can be increased by means of TR control.

### 3.1.2 Maximal Scheduling Policies

The link-scheduling policies discussed in [27], [31], [32] belong to the second category. These policies are characterized by the concept of local-pooling factor, which is defined as follows [31]. Let \( \mathcal{H} \) be some subset of links (i.e., \( \mathcal{H} \subseteq \mathcal{L} \)). The local-pooling factor \( \sigma^* \) of a WMN is

\[
\sigma^* = \sup \{ \sigma : \mu \neq \nu \quad \forall \quad \mu, \nu \in Co(\lambda_\mathcal{H}) \quad \forall \quad \mathcal{H} \in \mathcal{L} \}.
\]

Therefore, \( \sigma^* \) is the largest factor \( \sigma \) such that no feasible link schedule, say \( \mu \), weighted by \( \sigma \) (i.e., \( \mu \sigma \)) dominates any other feasible schedule \( \nu \). This factor is an indicator of how different the effectiveness of the different maximal schedules are from each other [27]. If a schedule dominates another schedule, the dominant schedule is more effective, where effectiveness makes reference to the schedule’s ability to reduce queue lengths. Therefore, the larger the local-pooling factor, the closer the effectiveness of the different maximal schedules. When the different maximal schedules are similarly effective, GMS policies are able to support packet rates that are closer to the boundaries of the optimal stability region. Given that \( \sigma^* \) depends on the sets of maximal schedules on \( \mathcal{H} \) and the sets of maximal schedules are determined by the interference sets \( \{ E^{(i,j)}_r : (i, j) \in \mathcal{L} \} \), \( \sigma^* \) can be modified by considering the interference sets of links. These sets can be controlled with the nodes’ TRs. Therefore, the NTC of the WMN can be increased by means of TR control.

### 3.1.3 Stability-Region Adaptation

All the previous link-scheduling policies depend on the links in the interfering sets \( \{ E^{(i,j)}_r : (i, j) \in \mathcal{L} \} \), and these sets depend on the nodes’ TRs (i.e., \( r \)). Therefore, the stability regions can be controlled by means of \( r \) in order to adapt them to the given set of flows. The goal of this adaptation can be formulated as the improvement of the NTC [18] by solving the NTC-Adaptation Problem, which is defined next.

The vector of nodes’ TRs \( r \) is feasible if the nodes do not exceed their maximum TR (i.e., \( r \leq r_{\text{max}} \)) and none of the flows are broken (i.e., \( r \geq r_{\text{min}} \)). Then, \( r \) is feasible if \( r_{\text{min}} \leq r \leq r_{\text{max}} \).

Let \( \lambda^{(i,j)}_{\text{max}}(r) \) be the highest packet rate that flow \( f_n \) supports while guaranteeing stability. According to (1), \( \lambda^{(i,j)}_{\text{max}}(r) \) is given by the minimum of the highest packet rates supported by the nodes in \( P_{\text{max}}^{f_n} (\text{i.e., } \min \{ \lambda^{(i,j)}_{\text{max}}(r) : j \in P_{\text{max}}^{f_n} \}) \), and according to the previous discussion on stability regions, these packet rates are a function of the node-interference set \( E^{(i,j)}_r \), which in turn, is a function of \( r \). Therefore, \( \lambda^{(i,j)}_{\text{max}}(r) \) depends on \( r \) as shown in

\[
\lambda^{(i,j)}_{\text{max}}(r) \triangleq \min \{ \lambda^{(i,j)}_{\text{max}}(r) : j \in P_{\text{max}}^{f_n} \}.
\] (3)

**Definition 2.** The NTC-Adaptation Problem (NTC-AP) is the problem of finding a Pareto optimal vector of TRs \( r^* \), i.e., \( r^* \) is such that there is no feasible \( r \) that meets the following condition: \( \lambda^{(i,j)}_{\text{max}}(r^*) > \lambda^{(i,j)}_{\text{max}}(r) \) for at least one \( f_n \) in \( F \) and \( \lambda^{(i,j)}_{\text{max}}(r^*) \geq \lambda^{(i,j)}_{\text{max}}(r) \) for all other \( f_n \) in \( F \).

According to (3), the NTC-AP can be solved by controlling the values of the highest packet rates supported by the nodes \( \{ \lambda^{(i,j)}_{\text{max}}(E_r^{(i,j)} : j \in P_{\text{max}}^{f_n} \} \) which are defined by the particular link-scheduling policy adopted in the network.

In the following section, the framework for solving the NTC-AP is proposed. It is based on potential-game theory [33]. In this framework, the flows are the players of the potential game, and they adapt the nodes’ TRs that affect the highest packet rates that guarantee stability. The framework is based on the finite-improvement-path property (see Definition 2) of potential games in which players take turns iteratively to make moves in order to increase their own utilities until equilibrium is reached, i.e., until no player can individually increase its utility. Using this framework, a distributed algorithm for solving the NTC-AP in IEEE 802.16 WMNs is proposed and analyzed in Section 4. In this algorithm, the players determine their moves to increase their utilities iteratively based on the greedy policy of [14].

### 3.2 Distributed TR-Control Algorithms Using Potential Games

The following definitions, summarized in Table 2, are used to formulate the NTC-AP as a normal-form game.

Individual flows form the player set \( \mathcal{F} \triangleq \{ f_1, f_2, \ldots, f_n, \ldots, f_N \} \). Each flow \( f_n \) can autonomously set the TR of the nodes in the set \( S^f_n \) which is defined as follows. \( S^f_n \) is the set of nodes that are able to affect \( f_n \)’s highest packet rate that guarantees stability when all the allowed TR levels are considered. Therefore, these are the nodes that are within some maximum distance from the intermediate and destination nodes of the flow (i.e., \( P_{\text{max}}^{f_n} \)). This distance depends on the maximum TR of the nodes and the interference model. Specifically, \( S^f_n \) is defined in terms of the sets \( S^f \) as follows. \( S^f \) is the set of nodes that are able to affect the highest packet rate that node \( j \) supports for any of its incoming flows (i.e., \( \lambda^{\max}_j(E_r^{(i,j)}) \)) when all the allowed TR
TABLE 2
Notation: Potential Game

| $S^i$ | The set of nodes that are able to affect $\lambda^{i}_{\text{max}}(E^i)$ with their TRs (see (4)) |
| $S/S_n$ | The set of nodes that are able to affect $\lambda^{i}_{\text{max}}(r)$ with their TRs (see (5)) |
| $F^n$ | The set of flows whose highest supported packet rates are affected by any of the moves that $f_n$ can make (see (6)) |
| $\mathcal{R}$ | The game’s action space: the set of feasible TRs of the nodes controlled by the flows, i.e., the nodes in $\bigcup_{f_n \in F} S^{f_n}$ |
| $\mathcal{R}_n$ | The action space of $f_n$: the set of feasible TRs of the nodes controlled by $f_n$, i.e., the nodes in $S^{f_n}$ |
| $\mathcal{R}_{-n}$ | The set of feasible TRs of the nodes not controlled by $f_n$, i.e., the nodes in $(\bigcup_{f_n \in F} S^{f_n}) \setminus S^{f_n}$ |
| $r_{n}$ | The vector of TRs of the nodes not controlled by $f_n$, i.e., the nodes in $(\bigcup_{f_n \in F} S^{f_n}) \setminus S^{f_n}$ |
| $\mu_n(r)$ | The utility function of flow $f_n$ as a function of $r$ in the NTC-AP game (see (7)) |
| $\mu(r)$ | The vector of utility functions $[\mu_1(r), \mu_2(r), \ldots, \mu_N(r)]$ in the NTC-AP game |
| $c^{f_n}(r)$ | The cost function of flow $f_n$ as a function of $r$ in the Lin-NTC-AP game (see (16)) |
| $\lambda_1(r)$ | An OPF of the NTC-AP game (see Theorem 1), and also the total throughput (see (8)) |
| $d_{x,y}^{(i,j)}$ | The number of active hidden nodes of link $(i, j)$, i.e., the number of nodes in $S_2^{x}(r) \setminus S_2^{y}(r)$, as a function of $r$ |
| $a$ | The vector $[a^{f_1}_{(i,j)}, a^{f_2}_{(i,j)}]_{i,j \in \mathcal{L}}$ |
| $c^t(r)$ | The contention level of node $j$ (see (13)) |
| $c^{f_n}(r)$ | The total contention experienced by $f_n$ (see (15)) |
| $c_v(r)$ | The contention variation experienced by $f_n$ (see (15)) |
| $c_{\text{lin}}(r)$ | An OPF of the Lin-NTC-AP game, and also the total contention and contention variation experienced by all the flows (see (17)) |
| $\text{c}_{\text{NTC}}(a)$ | The objective function of the NTC-AP as formulated in (14) |

Table 2
Notation: Potential Game

levels are considered. Therefore, $S^j$ is determined from the sets of interfering links $E^i_{\text{max}}(i,j) \in \mathcal{L}$, i.e., node $j$’s interfering nodes when all the nodes transmit at the maximum TR. $S^j$ is given by (4). $^5$ According to (4), in $S^j$, only the incoming links of node $j$ established with its direct 1-hop neighbors (i.e., $S^j_1$) are considered

$$S^j = \{ k : k \in \bigcup_{i \in S^j_1} E^i_{\text{max}}(i,j) \}. \quad (4)$$

$S^{f_n}$ is defined by (5). It is the union of the $S^j$ of the nodes along $f_n$’s path with the exception of the source node (i.e., $j \in \mathcal{P}^{f_n}$). The source node is excluded because this node does not limit the flow’s highest packet rate that guarantees stability. It is the ability of the intermediate and destination nodes to forward and receive the data packets generated by the source node what limits the highest packet rate supported by the flow

$$S^{f_n} = \bigcup_{j \in \mathcal{P}^{f_n}} S^j. \quad (5)$$

The action space of flow $f_n$ is the set of feasible TRs of the nodes in $S^{f_n}$. This set is denoted by $\mathcal{R}_n$. An action of $f_n$

is denoted by the vector $r_n$. Therefore, action $r_n$ belongs to $\mathcal{R}_n$, and it specifies the TRs controlled by $f_n$ (i.e., the TRs of the nodes in $S^{f_n}$). The game’s action space $\mathcal{R}$ is the set of feasible TRs of the nodes controlled by the flows (i.e., the nodes in $\bigcup_{f_n \in F} S^{f_n}$). Action $r_{-n}$ specifies the TRs not controlled by $f_n$ (i.e., the TRs of the nodes in $(\bigcup_{f_n \in F} S^{f_n}) \setminus S^{f_n}$). Therefore, $r_{-n}$ belongs to $\mathcal{R}_{-n} = \mathcal{R} \setminus \mathcal{R}_n$.

When flow $f_n$ makes a move (i.e., updates its action vector $r_n$), the highest packet rates that it and other flows support may be affected. This relation between $f_n$ and other flows is reflected on its utility function $\mu_n : \mathcal{R} \rightarrow \mathbb{R}$. Let $F^n$ be the set of flows whose highest supported packet rates are affected by any of the moves of $f_n$ (see (6)). Flow $f_n$’s utility function $\mu_n$ is defined from the highest packet rates supported by the flows in $F^n$ as given by (7). Therefore, according to (7), flow $f_n$’s utility increases not only when its highest supported packet rate $\lambda^{r_n}_{\text{max}}(r)$ increases, but also when the highest supported packet rates of the flows in $F^n$ increase as well. In this way, $f_n$ is encouraged to collaborate with the flows that are affected or potentially affected by its moves, i.e., when any flow makes a move, the flow tends to benefit other flows as well because in this way its utility function can be increased more effectively. This type of collaboration is known as a social context, and its Nash-equilibria properties are studied in [34, 35]. However, a flow’s move may affect other flows negatively as well. This is shown by means of an example in Appendix 6 available online

$$F^n \triangleq \{ f_m : S^{f_m} \cap S^{f_n} \neq \emptyset, f_m \in F \} \quad (6)$$

$$\mu_n(r) \triangleq \sum_{f_n \in F^n} \lambda^{r_n}_{\text{max}}(r). \quad (7)$$

The vector of utility functions is $\mu = [\mu_1, \mu_2, \ldots, \mu_N] : \mathcal{R} \rightarrow \mathbb{R}^N$.

Definition 3. The game NTC-AP = $(F, R, \mu(r))$ is an ordinal potential game (OPG) if there exists a function $V : \mathcal{R} \rightarrow \mathbb{R}$ such that $\forall f_n \in F, \forall r_{-n} \in \mathcal{R}_{-n}$, and $x_n, y_n \in \mathcal{R}_n$

$$V(x_n, r_{-n}) - V(y_n, r_{-n}) > 0$$

$$\Leftrightarrow \mu_n(x_n, r_{-n}) - \mu_n(y_n, r_{-n}) > 0.$$

$V$ is called the ordinal potential function (OPF) of the NTC-AP game.

Theorem 1. The game NTC-AP = $(F, R, \mu(r))$ is an OPG. An OPF is given by $\lambda_T(r)$, which is the total highest packet rate supported by the network for the set of flows $F$ (see (8))

$$\lambda_T(r) \triangleq \sum_{f_n \in F} \min \left\{ \lambda^{f_n}_{\text{max}}(r) : j \in \mathcal{P}^{f_n} \right\}. \quad (8)$$

See Appendix 3 available online for the proof of Theorem 1.

Given that potential games posses (pure-strategy) equilibrium, the following result is immediate.

Corollary 1. The game NTC-AP = $(F, R, \mu(r))$ possesses a pure-strategy equilibrium.

6. Please see [33] for the theory of potential games.

7. The direct dependence of $\lambda^{f_n}_{\text{max}}(r)$ on $E_{\text{ij}}^i$, i.e., $\lambda^{f_n}_{\text{max}}(E_{\text{ij}}^i)$ (see (3)), has been omitted in (8), i.e., $\lambda^{f_n}_{\text{max}}(r)$, to simplify the notation.

8. Please see Corollary 2.2 and Lemma 4.5 in [35].
Definition 4. A path in \( R \) is a sequence \( \gamma = (r^0, r^1, \ldots) \) such that for every \( k \geq 1 \) there exists a unique player, say flow \( f_{a_k} \), such that \( r^k = (r^k_n, r^k_{n+1}) \) for some \( r^k_n \neq r^k_{n+1} \) in \( R_n \). \( r^0 \) is called the initial point of \( \gamma \), and if \( \gamma \) is finite, then its last element is called the terminal point of \( \gamma \). \( \gamma = (r^0, r^1, \ldots) \) is an improvement path if for all \( k \geq 1 \), \( \mu_k(r) > \mu_k(r^{k-1}) \), where \( f_n \) is the unique deviator at step \( k \). A game has the finite improvement path property if every improvement path is finite.

In [33], lemmas 2.3 and 4.2, it is shown that OPGs have this property. Therefore, in the NTC-AP game, the flows are guaranteed to converge to a Nash equilibrium point by following an improvement path. An example of this path is provided in Appendix 6 available online. An improvement path can be achieved with an algorithm that determines the move of a flow for every of its turns such that the flow’s utility is increased whenever possible. Therefore, the convergence and performance of the algorithm can be characterized by means of an analysis of the equilibrium of the game.

It needs to be noted that in the formulation of the NTC-AP game, a TR may be controlled by more than one flow. Therefore, the formulation is not a typical description of an NTC-AP game, a TR may be controlled by more than one flow.

Finally, the convergence and performance of the algorithm are characterized.

4 STABILITY-REGION ADAPTATION IN IEEE 802.16 WMNs

The size of the stability region of IEEE 802.16 WMNs that operate under the GM-RBDS policy is given in Theorem 1. In the GM-RBDS policy, links perform a handshake in order to reserve future data-time-slots for data-packet transmissions that avoid collisions. This handshake is performed with scheduling-packet transmissions on control-time-slots. A detailed description of the policy is in Appendix 4 available online.

Theorem 2. Network \( G \) under the GM-RBDS policy and the 2-hop interference model is stable if the packet rate of every incoming flow of node \( j \) is not greater than \( \lambda^j_{\text{max}}(r) \) for every \( j \in N \), where \( \lambda^j_{\text{max}}(r) \) is given by (9)

\[
\lambda^j_{\text{max}}(r) = \frac{1}{5 \sum_{i \in S_j^a} d^{(i,j)}_a |S^a_i(r) \setminus S^a_j(r)|}.
\]

Please see [36] for the proof of Theorem 1.

Corollary 2. Network \( G \) under the GM-RBDS and the 2-hop interference model is stable if the packet rate \( \lambda^j \) of every flow \( f_n \) in \( F \) satisfies

\[
\lambda^j < \min \left\{ \frac{1}{5 \sum_{i \in S_j} d^{(i,j)}_a |S^a_i(r) \setminus S^a_j(r)|} : j \in P^a_{\text{int}} \right\}. \tag{10}
\]

The only factor in (10) that depends on \( r \) is \( |S^a_i(r) \setminus S^a_j(r)| \), which will be denoted by \( \alpha^a_{i,j}(r) \) (i.e., \( \alpha^a_{i,j}(r) \) is the number of hidden active nodes of link \( (i, j) \) when the nodes’ TRs are set at \( r \)). The expression for \( \lambda^j_{\text{max}}(r) \) that defines \( \mu(r) \) (see (7)) and \( \lambda_T \) (see (8)) for the NTC-AP game in IEEE 802.16 WMNs under the GM-RBDS policy is then given as follows:

\[
\lambda^j_{\text{max}}(r) = \min \left\{ \frac{1}{5 \sum_{i \in S_j} d^{(i,j)}_a \alpha^a_{i,j} : j \in P^a_{\text{int}} \right\}. \tag{11}
\]

4.1 Linear Approximation of the NTC-AP Game

In the game, the goal of the flows is to increase the value of their utility functions (i.e., \( \mu(r) \)) by adjusting the values of the factors \( \{\alpha^a_{i,j} : (i, j) \in \mathcal{L}\} \). However, given that \( \lambda^j_{\text{max}}(r) \) is a nonlinear function of \( \alpha^a_{i,j} \), the utilities \( \mu(r) \) and the OPF \( \lambda_T(r) \) are also nonlinear functions of \( \alpha^a_{i,j} \). Therefore, in order to guarantee the convergence of the game to an equilibrium with a bounded performance, the game is approximated with a linear version of it. The difference between the performance of this approximation and the original formulation of the game is bounded, and in the worst case scenario, the bound increases linearly with the number of flows (see Theorem 5).

The NTC-AP game in IEEE 802.16 WMNs is approximated with a game whose utilities and OPF are linear functions of \( \alpha^a_{i,j} \). This linear version of the NTC-AP game will be called Lin-NTC-AP. Based on the linearity, the Nash equilibria of the Lin-NTC-AP game is analyzed in Section 4.3, and the difference between the solutions reached by the NTC-AP and Lin-NTC-AP games is upper-bounded in Section 4.4.

Proposition 1. The solution set of the optimization problem given by (12) is a Pareto-optimal solution to the NTC-AP (see Definition 2)

\[
\text{minimize } \sum_{f_{\text{in}}} \frac{1}{5 \lambda^j_{\text{max}}(r)} \text{ subject to } r_{\text{min}} \leq r \leq r_{\text{max}}. \tag{12}
\]

See Appendix 5 available online for the proof of Proposition 1.

In the following, the concept of contention level is introduced in order to define the Lin-NTC-AP game given by (15).

Definition 5. The contention level of node \( j \) is defined by (13). It is the summation of the number of hidden active nodes
\( d^{i,j} \) in every incoming link (i.e., \( \{(i,j): i \in S^j_d\} \)) weighted by the link degree (i.e., \( d^{i,j} \))

\[
e^i(r) \triangleq \sum_{i \in S^i_d} d^{i,j} a^{i,j}_f. \tag{13}
\]

Therefore, when the contention experienced by a node is high, the amount of traffic it can forward and/or receive is low because its grants may overlap with higher probability any of the grants transmitted by the hidden active nodes.\(^9\)

**Definition 6.** The bottleneck node of flow \( f_n \) is the node among the flow’s intermediate and destination nodes that experiences the highest level of contention, i.e., let \( i \) be the bottleneck node of \( f_n \), then \( i = \text{argmax}_{j \in P_{\text{int}}} e^j(r) \).

Therefore, the bottleneck node of a flow is the node among the flow’s intermediate and destination nodes that has the minimum highest packet rate these nodes support, which, according to (11), is the highest packet rate supported by the flow.

From (11) and (13), the problem given by (12) (i.e., NTC-AP) can be rewritten in terms of the contention level as given by (14). This result is achieved by direct substitution of (13) into (11) and then, (11) into (12)

\[
\min_{f_n \in F} \sum_{j \in P_{\text{int}}^h} \max \{ e^j(i) : j \in P_{\text{int}}^h \}
\]

subject to \( r_{\text{min}} \leq r \leq r_{\text{max}} \). \tag{14}

Therefore, the NTC-AP reduces to finding a set of feasible TRs that minimizes the highest contention level experienced by every flow.

**Remark.** According to (14), the problem of maximizing the NTC of a WMN is equivalent to minimizing the level of contention, as defined in Definition 5, experienced by the data flows in the network. This minimization can be achieved by means of TR control and routing. In this paper, we focus on the TR control. However, if routing was considered, the optimal paths would be those that allow the minimization of contention, which is in agreement with the results in [16], [17]. Intuitively, this could be achieved with a routing algorithm that evenly distributes the traffic across all network resources. Mathematically, this problem can be approximated as stated in (18).

The solution to (12) (i.e., NTC-AP) is approximated with the solution to the Lin-NTC-AP. The Lin-NTC-AP is formulated as given by (15). The terms \( c^r_f(r) \) and \( c^V_f(r) \) in (15) are defined as follows. The total contention experienced by \( f_n \) is \( c^r_f(r) \triangleq \sum_{j \in P_{\text{int}}^h} c^j(r) \). The mean contention experienced by \( f_n \) is \( c^k_f(r) \triangleq |P_{\text{int}}^h|^{-1} \sum_{j \in P_{\text{int}}^h} c^j(r) \). The contention variation experienced by \( f_n \) is \( c^r_f(r) \triangleq \sum_{j \in P_{\text{int}}^h} |c^j(r) - c^k_f(r)| \)

\[
\min_{f_n \in F} \left( c^r_f(r) + c^V_f(r) \right)
\]

subject to \( r_{\text{min}} \leq r \leq r_{\text{max}} \). \tag{15}

Therefore, in the Lin-NTC-AP, the goal is to minimize the total contention and the contention variation experienced by the flows. Intuitively, the Lin-NTC-AP approximates the NTC-AP based on the following two observations. First, by making sure that the maximum contention experienced by a flow at a certain node is not too different from the contention at the other nodes of the flow, the contention along the flow’s path is more uniform. This is achieved when the contention variation is reduced. Second, given that the contention is more uniform, reducing the total contention experienced by the flow also reduces the maximum contention along the flow’s path, which is the goal in the NTC-AP (see (14)).

In the Lin-NTC-AP game, which is defined next, each flow competes for minimizing the total contention and contention variation of itself and of any other flows it affects with its actions. Therefore, the player set \( F \), the players’ action sets \( \{R_n : f_n \in F\} \), and the game’s action space \( R \) are the same for both the NTC-AP and the Lin-NTC-AP games.\(^10\) The only difference in the formulation of these two games is the utility functions. In the Lin-NTC-AP game, the flows do not have an utility function but a cost function \( c^r_f(r) : R \rightarrow R \), which is defined as follows

\[
c^r_f(r) \triangleq \sum_{f_n \in F} \left( c^r_f(r) + c^V_f(r) \right). \tag{16}
\]

Therefore, in the Lin-NTC-AP game, the flows decrease their cost functions, i.e., they decrease the contention they experience, instead of increasing the utilities as in the NTC-AP game. In the NTC-AP game, the flows increase their utility functions in order to increase the throughput. In the Lin-NTC-AP game, the flows increase the throughput by decreasing the contention.

Note that the Lin-NTC-AP game is also an OPG.\(^11\) An OPF for this game is given by (17). This OPF represents the total contention and contention variation experienced by all the flows.

\[
c_{\text{Lin}}(r) \triangleq \sum_{f_n \in F} \left( c^r_f(r) + c^V_f(r) \right). \tag{17}
\]

### 4.2 Players’ Moves: The WiMAX-Mesh-NTC Algorithm

The algorithm in Fig. 1, called WiMAX-Mesh-NTC, is proposed for the flows to decrease the value of their cost functions.\(^12\) The algorithm requires that the TRs be initialized at \( r_{\text{min}} \). The algorithm also requires the following information which is constant for the whole duration of the game: the set \( P^a \) of paths of the flows in \( F^a \) (i.e., \( P^a \triangleq \{P^a : f_n \in F^a\} \)), the degrees \( D^a \) of the incoming links of the nodes that belong to the paths in \( P^a \) (i.e., \( D^a \triangleq \{d^{i,j} : j \in P^a, i \in S^j_d\} \)). Finally, the algorithm requires the following information that changes with the actions taken by the flows: the set \( H^a_r \) of hidden active nodes of the links whose degrees were included in \( D^a \) (i.e., \( H^a_r \triangleq \{S^j_d(r) \backslash S^j_d(r) : j \in P^a, i \in S^j_d\} \)). It is assumed that for

10. Please see Section 3.2 for the definitions of the player set \( F \), action sets \( \{R_n : f_n \in F\} \), and action space \( R \).

11. The proof that the Lin-NTC-AP game is an OPG follows the same argument of the proof that the NTC-AP game is an OPG (i.e., proof of Theorem 1). It has been omitted for the sake of brevity.

12. The WiMAX-Mesh-NTC algorithm is based on the HSRA algorithm proposed in [18].
each flow there is a flow controller that is in charge of the execution of the WiMAX-Mesh-NTC. Also, an example of the execution of the algorithm in a network of 7 flows and 16 nodes is given in Appendix 6 available online.

The WiMAX-Mesh-NTC algorithm works as follows. In the game, the flows take turns to make moves (e.g., by passing a token from flow to flow). For every move, a flow calculates its action according to WiMAX-Mesh-NTC as follows. The flow first calculates its cost function $c_j(r)$ for the current set of TRs $r$ (see line 2 in Fig. 1). Then, it finds its bottleneck node (see line 3 in Fig. 1) and the active nodes hidden from the bottleneck node is able to decrease the cost function, none of them is possible, the flow calculates its cost function $c_j(r)$ necessary to cover the bottleneck node. If such a TR was selected, and if it was, it sets the TR of the selected node to the minimum TR that covers the bottleneck node was selected, and if it was, it sets the TR of a node so that the flow’s bottleneck node is able to listen to it. This node is the active node hidden from the bottleneck node that decreases the flow’s cost function the most.

4.3 Nash Equilibria and Linear Integer Programming

The reason for approximating the NTC-AP game with the Lin-NTC-AP game is that due to the linear-integer-programming nature of the NTC-AP (see Theorem 3), its set of optimal solutions (i.e., arg min$_{r_{min} \leq r \leq r_{max}}$ $\sum_{j \in J} (c_j^0(r) + c_j^1(r))$) can be characterized with the Nash-equilibria (see Theorem 4). Also, given that the Lin-NTC-AP game is potential, it is guaranteed to converge to the optimal solution of the Lin-NTC-AP (see Corollary 3).

**Theorem 3.** The Lin-NTC-AP can be formulated in terms of the vector $a \in \mathbb{A}$ (i.e., the variables are the elements of vector $a$ and not the elements of vector $r$) as the linear integer program given by (18), where vector $f$ is determined by the paths of the flows only (see (20)), and $A$ is the feasible region determined by Constraints 1, 2, and 3 in Appendix 7 available online

$$\min f \cdot a$$
subject to $a \in A$. (18)

See Appendix 7 available online for the proof of Theorem 3.

**Theorem 4.** The optimal solution set of the Lin-NTC-AP and the Nash equilibria set of the Lin-NTC-AP game are equivalent.

See Appendix 8 available online for the proof of Theorem 4.

From Theorems 3 and 4 and the finite-improvement-path property of potential games, the following result is immediate.

**Corollary 3.** The WiMAX-Mesh-NTC algorithm always converges to an optimal solution of the Lin-NTC-AP.

Remark. According to (18), the Lin-NTC-AP consists of finding a vector $a$ whose scalar projection onto vector $f$ is minimum, and according to (15), this is equivalent to minimizing the total contention and contention variation. Therefore, with the WiMAX-Mesh-NTC algorithm, the flows adjust iteratively vector $a$ until they minimize the scalar projection. According to Corollary 3, this minimization is always achieved.

Remark. Vector $f$ is determined by the paths of the flows only (see (20)), and vector $a$ is determined by the hidden nodes of the links of the network only. Therefore, if the problem of joint routing and TR control was considered, both $f$ and $a$ would be controlled in order to minimize their scalar projections onto each other.

4.4 Performance Bound

The optimal solutions of the Lin-NTC-AP are not guaranteed to be optimal for the NTC-AP. However, the difference between the values of the NTC-AP objective function evaluated at its optimal solution and evaluated at the optimal Lin-NTC-AP solution is upper-bounded (see Theorem 5). Therefore, the optimal Lin-NTC-AP solutions are able to reach the maximum NTC within a bounded difference.

Let $a_{opt}^{Lin}$ and $a_{opt}^{NTC}$ be optimal solutions to the Lin-NTC-AP and NTC-AP respectively. Let $e_{NTC}(a)$ be the objective function of the NTC-AP as formulated by (14) (i.e., $e_{NTC}(a) = \sum_{j \in J} (c_j^0(r) + c_j^1(r))$). Let $d_{max}$ be the maximum link degree (i.e., $d_{max} = \max(d^{(i,j)} : (i, j) \in \mathcal{L})$).

13. Please see [37] and the references therein for solutions to the implementation of flow controllers.

14. Any set of TRs $r$ that satisfies the solution $a$ achieves the same objective-function value.

15. Please see lemmas 2.3 and 4.2 in [33].
Theorem 5. The difference between the values of the NTC-AP objective function evaluated at $a_{Opt}^{Lin}$ and $a_{Opt}^{NTC}$ is upper-bounded as follows:

$$c_{NTC}(a_{Opt}^{Lin}) - c_{NTC}(a_{Opt}^{NTC}) \leq d_{max} \frac{|F| - 2}{2}.$$ 

See Appendix 9 available online for the proof of Theorem 5. This proof is based on identifying the possible scenarios in which $a_{Opt}^{Lin}$ and $a_{Opt}^{NTC}$ differ the most. These scenarios, which are based on the one explained in Fig. 4, require that conditions on the location of the nodes and direction of the flows be met. For example, the worst-case scenario shown in Fig. 7 requires a central flow surrounded by other flows that are distributed in a star-like manner and directed outwards while another flow crosses the flows pointing outwards. If these conditions are not met, the difference between $a_{Opt}^{Lin}$ and $a_{Opt}^{NTC}$ is small. Given that such restrictive conditions are not likely to occur in random networks and can also be avoided by means of routing, it is expected that the TRs found by the WiMAX-Mesh-NTC algorithm be close to the optimal TRs. This is studied by means of simulation.

The performance evaluation of the algorithm was performed using the WiMAX-RBDS-Sim framework [38] for the OPNET simulator [39]. This evaluation is given in terms of the percent difference between the throughput achieved by the algorithm and the optimal throughput. The percent difference is defined as follows. Let $\lambda_T(r)$ be the total throughput supported by the flows in $F$ under TR setting $r$ (i.e., $\lambda_T(r) = \frac{1}{\sum_{f \in F} \lambda^f_{max}(r)}$). Let $r_{opt}$ be the feasible set of TRs that maximizes $\lambda_T$. The throughput percent difference $\delta_T$ at $r$ is given by

$$\delta_T(r) = \frac{\lambda_T(r_{opt}) - \lambda_T(r)}{\lambda_T(r_{opt})}.$$  

(19)

The WiMAX-Mesh-NTC algorithm was compared with the optimal, HSRA, MinPower, and MaxPower algorithms. The HSRA algorithm [18] is a heuristic and centralized algorithm that aims to find a set of TRs that maximizes $\lambda_T$. The MinPower algorithm aims to maximize the spatial reuse by setting the nodes’ TRs at the minimum values that do not disconnect any of the flows in $F$. The MaxPower algorithm sets all the nodes’ TRs at their maximum values.

The simulation results in Appendix 10 available online show that the percent difference of the WiMAX-Mesh-NTC, HSRA, MinPower, and MaxPower algorithms is not greater than 0.1 for 72 percent, 72 percent, 62 percent, and 64 percent of the simulated networks respectively, where each network consisted of 10 flows and 20 randomly located nodes. The simulation results also show that as the number of flows increases and the number of nodes remains the same, WiMAX-Mesh-NTC still outperforms MinPower and MaxPower, and it is outperformed by HSRA. However, while HSRA is centralized, WiMAX-Mesh-NTC is distributed, so WiMAX-Mesh-NTC does not require global information of the network while HSRA does.

Finally, the results show that WiMAX-Mesh-NTC converges in no more than three rounds of moves among flows for networks of 20 nodes and 20 flows.

5 CONCLUSION

A new framework for the development of distributed TR algorithms that maximize the total end-to-end throughput in WMNs was proposed. It can be used on any network whose link-scheduling policy’s stability region has been characterized. The framework consists of a potential game in which a given set of flows act as players that collaborate to control nodes’ TRs in order to maximize the packet rates they can support while guaranteeing stability. Based on the proposed framework, the WiMAX-Mesh-NTC algorithm was developed for WMNs that implement the IEEE-802.16 link-scheduling policy for mesh networks. The convergence of WiMAX-Mesh-NTC was characterized by means of the Nash equilibrium, and a performance bound was calculated by considering all the possible worst-case scenarios. Finally, the WiMAX-Mesh-NTC performance was compared by means of simulation with the performance of other TR-control algorithms (i.e., optimal, HSRA, MinPower, MaxPower). It was shown that WiMAX-Mesh-NTC outperforms MinPower and MaxPower, it performs as HSRA when the flow density is low, and it is outperformed by HSRA when the flow density increases.

REFERENCES

Gustavo Vejarano received the BS degree in electrical engineering from Universidad del Valle, Cali, Colombia, in 2005 and the MS and PhD degrees in electrical and computer engineering from University of Florida, Gainesville, FL, USA, in 2009 and 2011, respectively. He is an Assistant Professor in the Department of Electrical Engineering and Computer Science and Director of the Internet Laboratory at Loyola Marymount University. In 2006, he worked for ZTE Corporation as engineer for the deployment of a city-wide cellular network. During his PhD study, he conducted research on wireless communication networks at the Wireless and Mobile Systems Laboratory. His research was also applied in the development of multimedia communication technologies at the U.S. Army’s Communications-Electronics Research, Development, and Engineering Center, where he participated in the design, implementation, and evaluation of video and audio communication systems. He has been a mentor of the NASA-MUST and NSF-SEAGEP programs, which provide career guidance to undergraduate students in the science, technology, engineering, and math (STEM) disciplines. He is a member of the IEEE.

Dexiang Wang received the PhD degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA in 2011, where he was a member of the Wireless and Mobile Systems Laboratory. He conducted his PhD research in the areas of fault-tolerant all-optical WDM communication systems, robust multimedia communication over heterogeneous networks, green internet, wireless network throughput optimization via transmission power control, and energy-efficient cognitive radio sensor networks. Before starting his PhD research, he worked as a software engineer for the R&D team of Huawei Technologies on SDH optical network switching equipments. After that, he worked for the R&D team of UTStarcom Telecom on WCDMA-RNC products. Currently he is with Juniper Networks as a member of technical staff working on development of next-generation feature-enriched packet switching products. He is a member of the IEEE.

Ritwik Dubey received the BS degree in electronics and communications engineering from MNNIT, Allahabad, India, in 2007 and the MS degree in electrical and computer engineering from University of Florida, in 2011. He is a Software Engineer at Qualcomm Inc in Boulder, Colorado. After his post-graduate study, he worked for a brief period at Motorola Mobility before moving to Qualcomm. He has been working on 3GPP technologies for radio interface for both UE and nodeB side. His research interests are radio resource management and wireless sensor networks. He is a member of the IEEE.

Janise McNair received the BS and MS degrees in electrical engineering from the University of Texas at Austin, in 1991 and 1993, respectively, and the PhD degree in electrical and computer engineering from the Georgia Institute of Technology, in 2000. She is an Associate Professor in the Department of Electrical and Computer Engineering at the University of Florida, where she leads the Wireless and Mobile Systems Laboratory. Currently, she serves on the Editorial Board of the Elsevier Ad Hoc Networks Journal and the Springer Wireless Networks Journal, and she is a co-chair of publications for IEEE INFOCOM 2012. Formerly, she served as an Associate Editor of the IEEE Transactions on Mobile Computing, as workshop co-chair for IEEE INFOCOM 2010, as vice general chair of the IEEE TridentCom 2007 conference and as TPC co-chair of IFIP Networking 2007. Dr. McNair is also a participant in the 2008 DARPA Computer Science Study Group. Her current research interests are next generation wireless networks, including small satellite networks, sensor networks, and cognitive networks. She is a member of the IEEE.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.
APPENDIX A
RELATED WORK

Our stability-based TR-control approach is primarily based on the ideas presented in [6], [7], [8], [18]. In [6], the network is partitioned based on the notion of local pooling\(^{17}\), and each partition is assigned to a channel of the network. In this way, GMS is guaranteed to achieve the optimal stability region in each channel. In [7], [8], network topologies are identified for which GMS policies achieve the optimal stability region. Although [7], [8] provide insightful results for the understanding of GMS policies, the network topologies they identify are not suitable for real scenarios [31]. This is due to the conditions that guarantee optimality for such network topologies. These conditions include 1-hop interference, 1-hop traffic, and a topology that corresponds to an \(F\)-free graph\(^{18}\). In practical scenarios, WMNs hardly meet these conditions. Our goal is to address this limitation by providing a game-theoretical framework for the design of distributed TR-control algorithms. In this framework, the flows modify distributively the network topology using TR control to adapt the stability region.

Intuitively, our TR control can be described as follows. The maximum throughput that a given flow supports depends on the throughput that each of the nodes along the flow’s path supports, and the maximum throughput that a node supports depends on its scheduling policy and the conflicts\(^{19}\) with surrounding nodes which also need to schedule packet transmissions [41]. Therefore, by means of TR control, the nodes are able to reduce the number of conflicts either by decreasing the interference (i.e., reducing TR) and/or coordinating future packet-transmission times such that no conflicting transmissions are performed simultaneously. The latter approach is used in this paper, i.e., the flows control the TRs so that nodes are able to listen to each other’s schedules in order to coordinate future packet transmissions more successfully.

Mathematically, our TR control can be characterized using the stability region of the WMN as follows. The maximum throughput that nodes support is characterized by the physical-link capacity and the stability region of the link scheduling policy. This region is the set of input-packet rates supported by the links of the network that guarantee that their queues are stable (i.e., the link queues are positive recurrent) [21]. The physical-link capacity determines the maximum number of bits that the packets can carry. In our TR control approach, the nodes’ TRs are modified such that the stability region is adapted to the given set of flows. The goal of this adaptation is to maximize the highest input-packet rate supported by the flows that guarantee stability. This approach was originally proposed in [18], in which the TR-control algorithm was heuristic and centralized. In this paper, we propose and analyze a distributed TR-control algorithm by means of potential games [33]. Specifically, we formulate the TR-control optimization problem as a potential game in which the flows (i.e., players) maximize the packet rates they can support by adapting the stability region.

In [42], the Dynamic Routing and Power Control (DRPC) algorithm was proposed. DRPC finds throughput-optimal transmission powers as follows. At every time slot, DRPC is given the set of end-to-end nodes with their corresponding end-to-end data rates, the channel states, and the queue lengths, and finds the optimal set of routes and transmission powers that maximize throughput. However, this is a constrained optimization problem that requires global network state information (i.e., channel states and queue lengths) at every time slot, which makes it not practical for implementation. For example, in [43], it is proven that delayed network state information causes a reduction of the size of the stability region. Therefore, also in [42], a suboptimal distributed algorithm was proposed for random-access scheduling policies. Another distributed and heuristic DRPC-based algorithm without proved performance was proposed in [44]. In [45], a random transmission-power-selection algorithm for random-access scheduling policies was proposed. It is shown that it achieves maximal throughput in the following sense: the throughput achieved by any fixed transmission-power selection is at most equal to the throughput achieved by the random transmission-power-selection algorithm. In [46], a throughput-optimal randomized transmission-power control algorithm is developed for WMNs that operate under the pick-and-compare scheduling policy [10], [11], [12], [13].

Other transmission-power-control algorithms for throughput maximization were proposed in [1], [2], [3], [4], [5], [46], [47], [48], [49]. In [1], [2], [3], [4], [5], the total throughput is increased by increasing the spatial reuse. This is achieved by reducing the nodes’ transmission powers. In [46], [47], the total throughput is increased further by not only considering the spatial reuse but also the exposed and hidden nodes. In [48], a transmission-power-control algorithm for RTS/CTS-based protocols is proposed that decreases the area occupied by links during their transmissions, and it is shown that with this scheme, routing algorithms that favor short hops achieve higher levels of throughput. In [49], the problem of integrated link scheduling and transmission-power control for throughput optimization is shown to be NP-Complete. Therefore, a heuristic algorithm is developed. Its goal is to minimize the schedule length necessary to satisfy all the link loads determined by a given routing algorithm. In this way, the total throughput of the network is increased because more scheduling cycles can be performed per time unit.

17. Please see Section 3.1 for the definition of local pooling.
18. An \(F\)-free is a graph that does not have cycles of 6, 8, 9, ... edges nor cycles of 5 or 7 edges joined by a path of length greater than or equal to zero. Please see [8] for the formal definition of \(F\)-free graphs.
19. Scheduling conflicts arise between nodes when they attempt to transmit packets simultaneously and the interference they cause on each other is high enough to cause packet collisions.
Our approach differs from the previous algorithms in that it is based on the stability region of the link-scheduling policy. Therefore, our approach is not limited to a particular scheduling policy but can be used with any policy whose stability region has been characterized. Also, our approach differs in that it is based on potential games [33] in which the flows act as players who increase their corresponding throughput. It should be noted that game-theoretical approaches have already been used for their corresponding throughput. It should be noted that games [33] in which the flows act as players who increase any policy whose stability region has been characterized.

The contributions of this paper are as follows.

• A new framework for the development of distributed algorithms that maximize the total end-to-end throughput in WMNs is proposed. This framework is based on the stability region of the WMN's link-scheduling policy. It consists of a potential game in which a given set of flows act as players that collaborate to maximize end-to-end throughput while guaranteeing stability.

• Based on the proposed framework, a new distributed TR-control algorithm is developed for IEEE 802.16 WMNs. The Nash equilibrium of this game is characterized by means of integer-linear-programming techniques.

• A performance bound for the new TR-control algorithm is found and compared with simulation results. It is shown that the performance is superior when compared with the one achieved by the classic TR-control approach of spatial-reuse maximization.

• It is shown that if routing is considered along with TR control, the throughput-optimal paths are those that minimize the maximum levels of congestion experienced by the flows. This is in agreement with the results in [16], [17].

APPENDIX B

NETWORK MODEL: FAIRNESS AND INTERFERENCE AND PACKET-ARRIVAL MODELS

Remark. The reason for the definition of \( \lambda_{\text{max}}(E^j_r) \) as a maximum packet rate equally divided among all the flows that traverse node \( j \) (i.e., a flow that traverses \( j \) is not allowed to exceed \( \lambda_{\text{max}}(E^j_r) \)) is to guarantee fairness at node \( j \). This definition along with the definition of node degree enables our TR-control approach to calculate nodes’ TRs for flows that require prespecified levels of fairness relative to each other. For example, let \( f_1 \) and \( f_2 \) be the only two flows that traverse \( j \). Also, assume that it is required that \( f_1 \)'s packet rate be twice \( f_2 \)'s packet rate. Then in our framework (see Section 3.2), two flows would be specified on \( f_1 \)'s path while one flow on \( f_2 \)'s path. In this way, node \( j \) assigns twice the packet rate to the traffic on \( f_1 \)'s path when compared to the packet rate it assigns to \( f_2 \)'s path. This shaping of throughput profiles and fairness framework are studied in detail in [22]. Also, in this paper, it is assumed that the network controller that enforces that \( \lambda_{\text{max}}(E^j_r) \) not be exceeded is given. The implementation details of such a controller are out of the scope of the paper. Please see [37] for a solution to this control problem.

Remark. The interference model (i.e., \( \zeta^{(i,j)}_r \)) and the packet-arrival model (i.e., Poisson distributed) we consider are based on the models adopted for the stability regions of the link scheduling policies considered in this paper, which are described in Section 3.1. The use of more accurate interference and packet-arrival models, such as interference models based on the signal to interference and noise ratio (SINR) and packet-arrival models based on heavy-tailed traffic, is still an open research problem for several of the policies. For example, the scheduling policies in [14], [19], [26], [27], [28], [29], [30], [31], [32] consider models that are in accordance with the models of this paper, the policies in [12], [13] are based on the SINR, and in [54], the problem of network stability under heavy-tailed traffic is studied. In this paper, we do not focus on the characterization of stability regions using more accurate interference and packet-arrival models. Our focus is on the use of the stability region as a means for controlling the nodes’ TRs. We leave the problem of introducing accurate interference models into our framework for future research as the stability analysis of scheduling policies is developed using accurate interference and traffic models.

APPENDIX C

PROOF OF THEOREM 1

From (3), (8), and (7), \( \lambda_T(r) \) can be rewritten as follows.

\[
\lambda_T(r) = \sum_{f_m \in F} \lambda^m_{\text{max}}(r) - \sum_{f_m \in F \setminus F^n} \lambda^m_{\text{max}}(r) = \mu_n(r) + \sum_{f_m \in F \setminus F^n} \lambda^m_{\text{max}}(r)
\]

Therefore,

\[
\lambda_T(x_n, r_n) - \lambda_T(y_n, r_n) = \mu_n(x_n, r_n) - \mu_n(y_n, r_n) + \sum_{f_m \in F \setminus F^n} \lambda^m_{\text{max}}(x_n, r_n) - \sum_{f_m \in F \setminus F^n} \lambda^m_{\text{max}}(y_n, r_n)
\]

The highest packet rates supported by the flows in \( F \setminus F^n \) (i.e., \( \{\lambda^m_{\text{max}}(r) : f_m \in F \setminus F^n\} \)) are independent of the actions of \( f_n \) (i.e., \( \lambda^m_{\text{max}}(x_n, r_n) = \lambda^m_{\text{max}}(y_n, r_n) \) \( \forall f_m \in F \setminus F^n, \forall r_n \in R_n, \forall x_n, y_n \in R_n \) because, by definition (see (6)), \( F^n \) is the set of flows whose highest packet rates are affected by \( f_n \)'s actions \( r_n \in R_n \). Therefore,

\[
\sum_{f_m \in F \setminus F^n} \lambda^m_{\text{max}}(x_n, r_n) = \sum_{f_m \in F \setminus F^n} \lambda^m_{\text{max}}(y_n, r_n)
\]
APPENDIX D
LINK SCHEDULING AND THE STABILITY REGION

In IEEE 802.16 WMNs [15], time is divided into frames according to the description given in Section 2. During control-time-slots, the nodes are selected by an election algorithm [15], [55], [56] such that when a node is selected, none of its 1-hop and 2-hop neighbors are selected. The election algorithm selects nodes in every control-time-slot, and a node transmits a scheduling packet every time it is selected. A scheduling packet carries three types of messages. These are request, grant, and confirmation. There can be more than one message per type in a scheduling packet (e.g., a scheduling packet may carry 3 requests, 1 grant, and 2 confirmations). The nodes make reservations of future data-time-slots for transmitting data packets stored in their link queues. The reservations are done on a per-link basis by means of a three-way handshake. First, when the link’s source node is selected by the election algorithm, it sends a request to the link’s destination node and waits for a reply. Then, when the link’s destination node is selected, it replies by sending a grant to the link’s source node. Finally, when the link’s source node is selected again, it sends a confirmation, which is a copy of the grant. The nodes that have either the link’s source or destination node as 1-hop neighbors (i.e., the nodes within the TR of either the link’s source or destination node) listen to the transmitted grant and confirmation. Every node in the WMN keeps track of the reserved future data-time-slots. Based on this information, the nodes determine the future data-time-slots they include in their requests and grants such that collisions are avoided. The nodes are able to use any link-scheduling policy [20] that adapts to the IEEE-802.16 standard [15] for determining such sets of data-time-slots (i.e., requested and granted data-time-slots). In the following, the 2-hop interference model and the link-scheduling policy proposed in [14] will be adopted. We give a brief description of this policy and its stability region in order to illustrate how the NTC-AP game can be used in IEEE 802.16 WMNs.

The adopted policy is the Greedy-Maximal Reservation-Based-Distributed-Scheduling (GM-RBDS) policy [14], which is summarized as follows.

Whenever a node is selected,

• for every outgoing link, request as many data-time-slots as unscheduled data packets stored in the link’s queue
• for every incoming link, grant as many data-time-slots as requested at the future data-time-slots that have not been reserved yet and that are the closest in time

The size of the stability region of the GM-RBDS policy depends on the ability of the links to perform the three-way handshakes successfully. If the probability that a link finishes successfully a three-way handshake is low, the link’s queue will decrease at a lower rate. Therefore, the link’s ability to forward data packets within some time range is going to be lower (i.e., the highest packet rate supported by the link is lowered), and this reduces the size of the stability region. The probability that a three-way handshake is successful depends on the grants received by the link’s source node from the time instant it sends the request until the time instant it receives the grant from the link’s destination node. If any of these grants is not heard by the link’s destination node and the link’s destination node’s grant overlaps them, the three-way handshake is unsuccessful. That is, the link’s source node will not be able to confirm the grant sent by the link’s destination node because other grants, previously received, already reserved the future data-time-slots granted by the link’s destination node. Therefore, the highest packet rate that a node supports for the flows it forwards or serves as a sink node depends on the following nodes: the nodes that transmit grants that are not received by it and are received by its 1-hop neighbors (i.e., hidden nodes). For example, consider some link (i, j). The nodes that transmit the grants received by i and j are the active 1-hop neighbors $S_1^i(r)$ and $S_2^i(r)$ of i and j respectively. The nodes in $S_1^i(r) \backslash S_2^i(r)$ transmit grants that are received by i but not by j. They are hidden from j. Therefore, if the grants transmitted by the nodes in $S_1^i(r) \backslash S_2^i(r)$ overlap the grants for link (i, j), which are transmitted by j, link (i, j)’s three-way handshakes are unsuccessful. When all the incoming links of node j are considered, the maximum packet rate $\lambda_{\text{max}}(r)$ that node j is able to grant for each of its incoming flows is given by (9).

APPENDIX E
PROOF OF PROPOSITION 1

Let $r_1$ be a solution to (12), and assume that $r_1$ is not a Pareto-optimal solution to the NTC-AP (see Definition 2). Let $r_2$ be a Pareto-optimal solution to the NTC-AP. Therefore, $\lambda_{\text{max}}(r_2) > \lambda_{\text{max}}(r_1)$ for at least one $f_n$ in $F$, and $\lambda_{\text{max}}(r_2) \geq \lambda_{\text{max}}(r_1)$ for all other $f_n$ in $F$, i.e., vector $[\lambda_{\text{max}}(r_2)]_{f_n \in F}$ dominates vector $[\lambda_{\text{max}}(r_1)]_{f_n \in F}$, so

$$\sum_{f_n \in F} \frac{1}{5\lambda_{\text{max}}(r_2)} < \sum_{f_n \in F} \frac{1}{5\lambda_{\text{max}}(r_1)}.$$

However, this contradicts the assumption that $r_1$ is a solution to (12).

20. The IEEE 802.16 standard [15] does not specify any particular link scheduling policy. It only provides the framework for implementing them (i.e., election algorithm, control and data subframes, and control messages).

to broadcast $S_i^{f_n}$. Therefore, each flow determines individually $S_i^{f_n}$ and broadcasts this information to all other flows. From the received $S_i^{f_n}$ sets, each flow determines $F^n$ according to (6), and then, it determines $P^n$ and $D^n$ by contacting the flows in its $F^n$. In order to determine $S_i^{f_n}$, a flow contacts the nodes along its path and requests from each of them the links that interfere according to (4) and (5). All of these operations need to be performed at $r_{\text{max}}$ (see (4)), which is shown in Fig. 2b. Second, the TRs of the nodes need to be set at $r_{\text{min}}$. This is shown in Fig. 2c. For the network of Fig. 2, the sets $S_i^{f_n}$ and $F^n$ are as follows.

$$S_1^{f_1} = \{1, 2, 3, 5, 6\}$$
$$S_2^{f_2} = \{1, 2, 3, 5, 6\}$$
$$S_3^{f_3} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$
$$S_4^{f_4} = \{4, 5, 6, 7, 8, 9, 10, 11, 12\}$$
$$S_5^{f_5} = \{7, 8, 9, 10, 11, 12, 15, 16\}$$
$$S_6^{f_6} = \{10, 11, 12, 13, 14, 15, 16\}$$
$$S_7^{f_7} = \{13, 14, 15, 16\}$$

$$F^1 = \{f_1, f_2, f_3, f_4\}$$
$$F^2 = \{f_1, f_2, f_3, f_4\}$$
$$F^3 = \{f_1, f_2, f_3, f_4\}$$
$$F^4 = \{f_1, f_2, f_3, f_4\}$$
$$F^5 = \{f_3, f_4, f_5, f_6, f_7\}$$
$$F^6 = \{f_4, f_5, f_6, f_7\}$$
$$F^7 = \{f_5, f_6, f_7\}$$

The sets $F^n$ show the distributed nature of the WiMAX-Mesh-NTC algorithm. Flows that are not close to each other do not need to exchange information among them during the execution of the algorithm. This is the case of $f_1$ and $f_2$ for which $F^1 \cap F^2 = \emptyset$. On the other hand, flows that are close to each other do need to exchange information among them. For example, $f_4$ is located near the center of the network, so it is closer to all other flows than any other flow. Therefore, this flow includes the highest number of flows in its $F^n$ (i.e., $|F^n| \geq |F^n| \forall n \neq 4$). Also, due to the distributed nature of the algorithm, none of the flows require global knowledge of the network (i.e., $F^n \neq F \forall n$).

After the network initialization, the flows take turns to make moves (e.g., by passing a token from flow to flow). At every turn, a flow makes its move in two steps. First, the flow needs to determine the set of hidden active nodes. The flow controller can do this by requesting from each node along the flow’s path the hidden nodes of the links directed to the node. For example, if $(i, j)$ is an incoming link of node $j$, $j$ can determine $S^i_1(v) \cup S^i_2(v)$ in order to reply to the controller’s request. Second, once

---

22. Please see [37] and the references therein for solutions to the implementation of flow controllers.
23. Please see Section 4.2 for the definition of $P^n$ and $D^n$.
24. Please see Section 4.2 for the definition of $H^n$. 

APPENDIX F

**WiMAX-Mesh-NTC Example**

An example of the operation of the WiMAX-Mesh-NTC algorithm is shown in Fig. 2. The network in Fig. 2 consists of 16 nodes (i.e., $N = \{1, 2, \ldots, 16\}$) and 7 flows (i.e., $F = \{f_1, f_2, \ldots, f_7\}$) which are shown in Fig. 2a. The maximum TRs of the nodes (i.e., $r_{\text{max}}$) are shown in Fig. 2b. The minimum TRs of the nodes that do not break any of the flows (i.e., $r_{\text{min}}$) are shown in Fig. 2c.

It is assumed that there is a flow controller for every flow in the network and that the controllers execute all of the following actions for their corresponding flow.

Before the execution of the WiMAX-Mesh-NTC algorithm, the network needs to be initialized in two steps as follows. First, the flows need to determine $P^n$ and $D^n$ from $F^n$. In order to determine $F^n$, the flows need
TABLE 3
Evolution of the Cost Functions of the Flows

<table>
<thead>
<tr>
<th>( i )</th>
<th>0</th>
<th>1, 2, 3</th>
<th>4, 5, ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_f^{i}(r^i) )</td>
<td>16</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>( c_{T}^{i}(r^i) )</td>
<td>22</td>
<td>16</td>
<td>14</td>
</tr>
<tr>
<td>( c_{f}^{i}(r^i) )</td>
<td>26</td>
<td>20</td>
<td>18</td>
</tr>
<tr>
<td>( c_{f}^{i}(r^i) )</td>
<td>30</td>
<td>24</td>
<td>22</td>
</tr>
<tr>
<td>( c_{f}^{i}(r^i) )</td>
<td>16</td>
<td>18</td>
<td>16</td>
</tr>
<tr>
<td>( c_{T}^{i}(r^i) )</td>
<td>14</td>
<td>16</td>
<td>14</td>
</tr>
<tr>
<td>( c_{T}^{i}(r^i) )</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

The flow has determined \( \mathcal{H}_n^r \), it needs to execute the WiMAX-Mesh-NTC algorithm (see Fig. 1).

In order to execute the algorithm, a flow just needs to evaluate the corresponding equations since it already has all the required information for the execution (i.e., \( \mathcal{P}^n \), \( \mathcal{D}^n \), and \( \mathcal{H}_n^r \)). Specifically, the first step in the algorithm is to calculate the flow’s cost function \( c_f^{i}(r) \), and this is done by evaluating (16). The second step is to find the flow’s bottleneck node \( b \), and this is done by evaluating Definition 6. The third step is to determine the set of active nodes hidden from \( b \) (i.e., \( \{S_b^i(r) \setminus S_b^i(r) : i \in S_b^i \} \) ), and this set can be obtained directly from \( \mathcal{H}_n^r \). Finally, from this set, the flow needs to select the node that decreases the flow’s cost function the most when the node’s TR is increased enough to cover \( b \).

Fig. 2c, Fig. 2d, and Fig. 2e show how the TRs are modified as the game evolves until it reaches equilibrium. The game evolves as the players (i.e., flows) make moves. The players make moves by taking turns in the following order after network initialization: \( f_1, f_2, ..., f_7, f_1, f_2, ..., f_7, ... \) The sets of TRs after each of these moves are denoted by \( r^1, r^2, ..., r^7, r^8, r^9, ..., r^{14}, \) respectively. Also, the initial set of TRs is denoted by \( r^0 = r_{\text{min}} \). When a flow makes a move, it can change the TR of one node only or not change any TR at all. In the example, only the TRs of nodes 6 and 12 are changed at \( r^1 \) and \( r^4 \) as shown in Fig. 2d and Fig. 2e respectively. The game reaches equilibrium at \( r^4 \), i.e., \( r^i = r^4 \) \( \forall i \geq 4 \). Given that the flows pass a token to take turns, they can verify the equilibrium one round after another. Therefore, it takes two rounds for the algorithm to finish executing. The evolution of the cost functions is shown in Table 3.

The following equations show how the cost function \( c_f^{i}(r) \) is evaluated in terms of the total contention \( c_{T}^{i}(r) \) and the contention variation \( c_{f}^{i}(r) \). For the sake of brevity, only the case of \( r^0 \) (see Fig. 2c) is shown.

\[
\begin{align*}
\quad & c_{f}^{i}(r^0) = c_f^{i}(r^0) + c_f^{i}(r^0) = d^{(1,2)}(a_f^{1,2}) + d^{(2,3)}(a_f^{2,3}) \\
\quad & = 2 \times 1 + 2 \times 2 = 6 \\
\quad & c_{T}^{i}(r^0) = c_{f}^{i}(r^0) = 6 \\
\quad & c_{f}^{i}(r^0) = c_f^{i}(r^0) + c_f^{i}(r^0) = d^{(4,5)}(a_f^{4,5}) + d^{(5,6)}(a_f^{5,6}) \\
\quad & = 1 \times 1 + 1 \times 1 = 2 \\
\quad & c_{T}^{i}(r^0) = c_{f}^{i}(r^0) + c_f^{i}(r^0) = d^{(7,8)}(a_f^{7,8}) + d^{(8,9)}(a_f^{8,9}) \\
\quad & = 1 \times 2 + 1 \times 1 = 3 \\
\end{align*}
\]

\[
\begin{align*}
\quad & c_{f}^{i}(r^0) = c_f^{i}(r^0) + c_f^{i}(r^0) = d^{(10,11)}(a_f^{10,11}) + d^{(11,12)}(a_f^{11,12}) \\
\quad & = 1 \times 1 + 1 \times 2 = 3 \\
\quad & c_{T}^{i}(r^0) = c_{f}^{i}(r^0) + c_f^{i}(r^0) = d^{(13,15)}(a_f^{13,15}) + d^{(14,15)}(a_f^{14,15}) + d^{(15,16)}(a_f^{15,16}) \\
\quad & = 1 \times 1 + 1 \times 1 + 1 \times 1 = 3 \\
\quad & c_{f}^{i}(r^0) = c_f^{i}(r^0) = d^{(13,15)}(a_f^{13,15}) + d^{(14,15)}(a_f^{14,15}) \\
\quad & = 1 \times 1 + 1 \times 1 = 2 \\
\end{align*}
\]
TABLE 4
OPF of the Lin-NTC-AP game (i.e., $c_{\text{Lin}}(r)$), objective function of the NTC-AP (i.e., $c_{\text{NTC}}(r)$), and OPF of the NTC-AP game (i.e., $\lambda_T(r)$)

<table>
<thead>
<tr>
<th>$i = 0$</th>
<th>$i = 1, 2, 3$</th>
<th>$i = 4, 5, \ldots$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_{\text{Lin}}(r^i)$</td>
<td>32</td>
<td>26</td>
</tr>
<tr>
<td>$c_{\text{NTC}}(r^i)$</td>
<td>17</td>
<td>14</td>
</tr>
<tr>
<td>$\lambda_T(r^i)$</td>
<td>$\frac{3}{7}$</td>
<td>$\frac{4}{7}$</td>
</tr>
</tbody>
</table>

TABLE 5
Highest-Packet Rates (packets per data-time slot) that Flows Support

<table>
<thead>
<tr>
<th>$i = 0$</th>
<th>$i = 1, 2, 3$</th>
<th>$i = 4, 5, \ldots$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_{\text{max}}^{j}(r^i)$</td>
<td>$\frac{3}{7}$</td>
<td>$\frac{4}{7}$</td>
</tr>
<tr>
<td>$\lambda_{\text{max}}^{j}(r^i)$</td>
<td>$\frac{3}{7}$</td>
<td>$\frac{4}{7}$</td>
</tr>
<tr>
<td>$\lambda_{\text{max}}^{j}(r^i)$</td>
<td>$\frac{3}{7}$</td>
<td>$\frac{4}{7}$</td>
</tr>
<tr>
<td>$\lambda_{\text{max}}^{j}(r^i)$</td>
<td>$\frac{3}{7}$</td>
<td>$\frac{4}{7}$</td>
</tr>
<tr>
<td>$\lambda_{\text{max}}^{j}(r^i)$</td>
<td>$\frac{3}{7}$</td>
<td>$\frac{4}{7}$</td>
</tr>
</tbody>
</table>

The competition among flows can be seen in Table 3. When $f_1$ makes its move at $r^1$ (Fig. 2d) to decrease its cost function from 22 to 16, the cost functions of $f_5$ and $f_6$ increase from 16 to 18 and from 14 to 16 respectively. On the other hand, when $f_4$ makes its move at $r^4$ (Fig. 2e) to decrease its cost function from 22 to 24, all other cost functions decrease as well with the exception of $f_7$ whose cost function stays the same.

The evolution of the global performance of the network in terms of contention and packet rate is shown in Table 4. It includes the OPF of the Lin-NTC-AP game $c_{\text{Lin}}(r)$ (see (17)), the objective function of the NTC-AP $c_{\text{NTC}}(r)$ (see Section 4.4), and the total packet rate $\lambda_T(r)$ (see (8)). The results show that as the game evolves, the OPF of the Lin-NTC-AP always decreases. This confirms that when flows decrease their cost functions individually with their moves, they decrease a global measure of the network (i.e., $c_{\text{Lin}}(r)$). Given that $c_{\text{Lin}}(r)$ approximates $c_{\text{NTC}}(r)$ (see Theorem 5) and that decreasing $c_{\text{NTC}}(r)$ increases $\lambda_T(r)$ (see (14) and Proposition 1), the individual actions of the flows increase the total packet rate $\lambda_T(r)$. In this particular example, the flows are able to maximize $\lambda_T(r)$ globally. However, this is not guaranteed to be always the case as demonstrated in Theorem 5 and studied in Appendix G.

The evolution of the individual flow packet rates is shown in Table 5. The results show how the moves of an individual flow affect the packet rates of other flows. These moves take place at $r^1$ and $r^4$. When $f_1$ makes its move at $r^1$ (see Fig. 2d), the packet rates of both $f_1$ and $f_2$ (i.e., $\lambda^{f_1}_{\text{max}}(r)$ and $\lambda^{f_2}_{\text{max}}(r)$) increase from $\frac{1}{20}$ to $\frac{1}{17}$, and the packet rate of $f_4$ (i.e., $\lambda^{f_4}_{\text{max}}(r)$) decreases from $\frac{1}{15}$ to $\frac{1}{17}$. When $f_4$ makes its move at $r^4$ (see Fig. 2e), only the packet rate of $f_4$ changes; it increases from $\frac{1}{15}$ to $\frac{1}{17}$.

APPENDIX G
PROOF OF THEOREM 3

The objective function $c_{\text{Lin}}$ is formulated in terms of $a$ as follows.

$$c_{\text{Lin}}(a) = \sum_{f_n \in \mathcal{F}} \left( c_{f_n}^{\text{Lin}}(a) + c_{f_n}^{V}(a) \right)$$

$$= \sum_{f_n \in \mathcal{F}} \left( \sum_{j \in \mathcal{P}^{\text{Lin}}_{\text{int}}}_m c_j^f(a) + \sum_{j \in \mathcal{P}^{\text{Lin}}_{\text{int}}}_m |c_j^f(a) - c_{f_n}^{\text{Lin}}(a)| \right)$$

$$= \sum_{f_n \in \mathcal{F}} \sum_{j \in \mathcal{P}^{\text{Lin}}_{\text{int}}}_m c_j^f(a) + \sum_{f_n \in \mathcal{F}} \sum_{j \in \mathcal{P}^{\text{Lin}}_{\text{int}}}_m \left( c_j^f(a) - \frac{1}{|\mathcal{P}^{\text{Lin}}_{\text{int}}|} \sum_{k \in \mathcal{P}^{\text{Lin}}_{\text{int}}} c_j^k(a) \right)$$

$$= \sum_{f_n \in \mathcal{F}} \sum_{j \in \mathcal{P}^{\text{Lin}}_{\text{int}}}_m \left( \sum_{i \in \mathcal{S}_d^k} d^{i,j} a^{i,j} + \frac{1}{\mathcal{P}^{\text{Lin}}_{\text{int}}} \sum_{i \in \mathcal{S}_d^k} \sum_{m \in \mathcal{S}_f^{i,k}} \left( m^{i,j}_m d^{i,j} a^{i,j} - \sum_{i \in \mathcal{L}} \sum_{m \in \mathcal{S}_f^{i,k}} m^{i,j}_m d^{i,j} a^{i,j} \right) \right)$$

Let $d^{i,j}$ be the number of flows for which node $j$ is an intermediate or destination node. Let $m^{i,j}_k$ be an indicator that link $(i, j)$ points to node $k$ (i.e., $m^{i,j}_k = 1$ if $j = k$, otherwise $m^{i,j}_k = 0$). Let $m^{i,j}_k$ be an indicator that link $(i, j)$ points to an intermediate or destination node of flow $f_n$ (i.e., $m^{i,j}_n = 1$ if $j \in \mathcal{P}^{\text{Lin}}_{\text{int}} f_n$, otherwise $m^{i,j}_n = 0$). The objective function $c_{\text{Lin}}$ can be formulated in terms of $d^{i,j}, m^{i,j}_k,$ and $m^{i,j}_n,$ as follows.

$$c_{\text{Lin}}(a) = \sum_{(i, j) \in \mathcal{L}} d^{i,j} a^{i,j} + \sum_{f_n \in \mathcal{F}} \sum_{k \in \mathcal{P}^{\text{Lin}}_{\text{int}}}_m \left( \sum_{(i, j) \in \mathcal{L}} m^{i,j}_k d^{i,j} a^{i,j} - \frac{1}{|\mathcal{P}^{\text{Lin}}_{\text{int}}|} \sum_{(i, j) \in \mathcal{L}} m^{i,j}_k d^{i,j} a^{i,j} \right)$$

The objective function $c_{\text{Lin}}$ can be formulated as a linear function of a using the vectors $d \triangleq \left\{ d^{i,j} a^{i,j} \right\}_{(i, j) \in \mathcal{L}}, m_k \triangleq \left\{ m^{i,j}_k d^{i,j} \right\}_{(i, j) \in \mathcal{L}}, m_f \triangleq \left\{ m^{i,j}_n d^{i,j} \right\}_{(i, j) \in \mathcal{L}},$ and $f \triangleq d + \sum_{f_n \in \mathcal{F}} \sum_{k \in \mathcal{P}^{\text{Lin}}_{\text{int}}} \left( m_k - m_f \right) a$ as follows.

$$c_{\text{Lin}}(a) = d \cdot a + \sum_{f_n \in \mathcal{F}} \sum_{k \in \mathcal{P}^{\text{Lin}}_{\text{int}}} \left( m_k - m_f \right) a = f \cdot a$$
In the following, the feasible region of \( a \) is characterized based on the feasible region of \( r \), i.e., \( r_{\min} \leq r \leq r_{\max} \).

Let \( S_a^{i,j}(r) \) be the set of nodes in \( S_a^i(r) \) that are closer to node \( i \) than to node \( j \). Note that the value of \( S_a^{i,j}(r) \) varies with the nodes’ TRs. The nodes in \( S_a^{i,j}(r_{\max}) \) are the nodes that for some feasible \( r \) (i.e., \( r_{\min} \leq r \leq r_{\max} \)) are hidden nodes of link \((i,j)\), i.e., the nodes that do not belong to \( S_a^{i,j}(r_{\max}) \) are not hidden nodes of link \((i,j)\) for any feasible value of \( r \). We refer to the nodes in \( S_a^{i,j}(r_{\max}) \) as the potential-hidden nodes of link \((i,j)\).

The feasible region of the Lin-NTC-AP can be formulated in terms of the following three constraints, which are defined in terms of \( |S_a^{i,j}|_r \triangleq |S_a^{i,j}(r)| \).

**Constraint 1.**

\[
a^{(i,j)} \geq |S_a^{i,j}|_{r_{\min}} \quad \forall \ (i,j) \in \mathcal{L}
\]

**Constraint 2.**

\[
a^{(i,j)} \leq |S_a^{i,j}|_{r_{\max}} \quad \forall \ (i,j) \in \mathcal{L}
\]

**Constraint 3.**

\[
a^{(i,j)} > a^{(h,k)} - |S_a^{h,k}\setminus S_a^{i,j}|_{r_{\max}} \quad \text{if} \quad |S_a^{h,k}\cap S_a^{i,j}|_{r_{\max}} > |S_a^{h,k}\cap S_a^{i,j}\cap S_a^{h,k}|_{r_{\max}} \forall \ (i,j),(h,k) \in \mathcal{L}, (i,j) \neq (h,k)
\]

Constraints 1 and 2 guarantee that \( a^{(i,j)} \) is not lower and greater than its minimum and maximum possible values respectively. These constraints can be explained as follows. The value of \( a^{(i,j)} \) represents the number of active nodes that cover \( i \) and do not cover \( j \) (i.e., active nodes that \( i \) is able to listen to and that are hidden from \( j \)). This number cannot be lower/greater than the number of active nodes that are closer to \( i \) than to \( j \), and that cover \( i \) at minimum/maximum TR (i.e., \( |S_a^{i,j}|_{r_{\min}} \) and \( |S_a^{i,j}|_{r_{\max}} \)) respectively.

Constraint 3 guarantees that when two different links (e.g., \((i,j)\) and \((h,k)\)) share potential-hidden nodes (i.e., \( S_a^{h,k} \cap S_a^{i,j} \neq \emptyset \)), the link that has, at maximum TR, the highest number of shared potential-hidden nodes closest to its source node always has a higher or equal number of hidden nodes. When the TRs of the shared potential-hidden nodes are being increased, they cover first the link’s source node that is closest to them. Therefore, the link with the highest number of shared potential-hidden nodes closest to its source node always has higher or equal number of hidden nodes. This is shown in the example of Fig. 3. In Fig. 3a, two links (i.e., \((i,j)\) and \((h,k)\)) and their potential-hidden nodes are shown with their corresponding maximum TRs. The potential-hidden nodes of link \((i,j)\) are nodes 1, 2, ... , 5 (i.e., \( S_a^{i,j}(r_{\max}) = \{1,2,\ldots,5\} \)). The potential-hidden nodes of \((h,k)\) are nodes 1, 2, ... , 7 (i.e., \( S_a^{h,k}(r_{\max}) = \{1,2,\ldots,7\} \)). The potential-hidden nodes shared by \((i,j)\) and \((h,k)\) are 1, 2, ... , 5 (i.e., \( S_a^{i,j}(r_{\max}) \cap S_a^{h,k}(r_{\max}) = \{1,2,\ldots,5\} \)). Whenever the TR of any of these nodes is being increased, the source node of link \((h,k)\) (i.e., node \( h \)) is never covered after the source node of link \((i,j)\) (i.e., node \( i \)) has been covered. Therefore, link \((i,j)\) always has at least as many hidden nodes as link \((h,k)\). The potential-hidden nodes of link \((h,k)\) which are not potential-hidden nodes of link \((i,j)\) (i.e., the nodes in \( S_a^{h,k}(r_{\max}) \setminus S_a^{i,j}(r_{\max}) = \{6,7\} \)) are able to increase the number of hidden nodes of \((h,k)\) (i.e., \( |S_a^{h,k}(r) \setminus S_a^{i,j}(r)| \)) without increasing the number of hidden nodes of \((i,j)\) (i.e., \( |S_a^{i,j}(r) \setminus S_a^{h,k}(r)| \)). Therefore, in order to account for these nodes, the factor \( |S_a^{h,k}(r) \setminus S_a^{i,j}(r)|_{r_{\max}} \) is introduced in Constraint 3 by subtracting it from the number of hidden nodes of \((h,k)\). For example, nodes 6 and 7 in Fig. 3 are potential-hidden nodes of \((h,k)\) but not of \((i,j)\) (i.e., \( S_a^{h,k}(r_{\max}) \setminus S_a^{i,j}(r_{\max}) = \{6,7\} \)), so they are able to increase the number of hidden nodes of \((h,k)\) without increasing the number of hidden nodes of \((i,j)\).

According to (20) and Constraints 1, 2, and 3, the Lin-NTC-AP can be formulated as the linear integer program in (18).

\[\begin{align*}
25. \text{It is being assumed that } |S_a^{h,k}\setminus S_a^{i,j}|_{r_{\max}} = 0. \text{ The general case of } |S_a^{h,k}\setminus S_a^{i,j}|_{r_{\max}} \geq 0 \text{ is considered next.}
26. \text{It is assumed that the nodes } 1, 2, \ldots, 7 \text{ in Fig. 3 are active.}
\]

\[\begin{align*}
27. \text{Nodes } 6 \text{ and } 7 \text{ have not been considered yet, i.e., it is being assumed that } |S_a^{h,k}\setminus S_a^{i,j}|_{r_{\max}} = 0.
\end{align*}\]
APPENDIX H
PROOF OF THEOREM 4

Let \( a^{opt} \) be an optimal solution. Assume that \( a^{opt} \) is not a Nash equilibrium. Therefore, there exists some player \( f_n \) and an \( a_n \) such that \( c_{f_n}(a_n, a^{opt}_n) < c_{f_n}(a^{opt}_n, a^{opt}_n) \). Given that the Lin-NTC-AP game is an OPG, and \( c_{Lin}(a) \) is an OPF, the previous result implies that \( c_{Lin}(a_n, a^{opt}_n) < c_{Lin}(a^{opt}_n, a^{opt}_n) \). This contradicts the assumption that \( a^{opt} \) is optimal.

Let \( a^eq \) be a Nash equilibrium. Assume that \( a^eq \) is not optimal. Therefore, there exists some player \( f_n \) and an \( a_n \) such that \( c_{Lin}(a_n, a^{eq}_n) < c_{Lin}(a^{eq}_n, a^{eq}_n) \). Given that the Lin-NTC-AP game is an OPG, and \( c_{Lin}(a) \) is an OPF, the previous result implies that \( c_{Lin}(a_n, a^{eq}_n) < c_{Lin}(a^{eq}_n, a^{eq}_n) \). Also, given that the Lin-NTC-AP can be formulated as the linear integer program in (18) (see Theorem 3), an improvement path from \((a^{eq}_n, a^{eq}_n)\) to \((a_n, a^{eq}_n)\) is guaranteed to exist. This contradicts the assumption that \( a^{eq} \) is a Nash equilibrium.

\[ \square \]

APPENDIX I
PROOF OF THEOREM 5

The WiMAX-Mesh-NTC performance bound can be proved based on the following observation. An active node that is a hidden node of one or more links of a flow contributes to the contention experienced by the flow. When the TR of an active node partially covers a flow, the node becomes an active hidden node of the flow’s links that originate within the TR and terminate outside the TR. This is shown in Fig. 4 in which link \((i, j)\) originates within \( h \)’s TR (i.e., \( i \) is covered by \( h \)’s TR) and terminates outside \( h \)’s TR (i.e., \( j \) is not covered by \( h \)’s TR). Given that \( h \) is a hidden node of one of \( f_n \)’s links (i.e., \((i,j)\)), \( h \) contributes to the contention experienced by the link, and as a consequence, it contributes to the contention experienced by \( f_n \). According to (13) and (14), \( h \) contributes an additive factor of \( d^{(i,j)} \) and of \( \frac{d^{(i,j)}}{|P_{int}^{f_n}|} \) to \( f_n \)’s total contention and mean contention respectively. The contribution of \( h \) to the contention variation in \( f_n \) can be positive or negative depending on the nodes’ TRs. For example, in Fig. 4, \( h \) increases \( f_n \)’s contention variation because without \( h \), all the nodes in \( P_{int}^{f_n} \) experience the same contention, while with \( h \), node \( j \)’s contention is increased by \( d^{(i,j)} \) while the other nodes’ contentions remain the same. Therefore, without \( h \), \( f_n \)’s contention variation is zero, while with \( h \), \( f_n \)’s contention is greater than zero.

**Proof:** In the NTC-AP, the goal is to minimize the maximum contention experienced by each of the flows (see (14)). Fig. 5 shows the cases in which TRs of active nodes determined from \( a^{opt}_{Lin} \) do not minimize the maximum contention of a flow. The notation in Fig. 5 is as follows. The flow whose maximum contention is not minimized by \( a^{opt}_{Lin} \) is \( f_1 \). The only TRs that are shown are the TRs of the active nodes that can be modified in order to minimize the maximum contention of \( f_1 \). The TRs shown with dashed lines are the optimal TRs of the Lin-NTC-AP (i.e., TRs determined from \( a^{opt}_{Lin} \)), and the TRs shown with solid lines are the optimal TRs of the NTC-AP (i.e., TRs determined from \( a^{opt}_{NTC} \)). In the case of one single flow whose maximum contention is not minimized by \( a^{opt}_{Lin} \), there are three possible TR-configurations in which the optimal Lin-NTC-AP TRs differ from the optimal NTC-AP TRs. These are shown in Fig. 5. In the general case, i.e., when the maximum contention experienced by two or more flows are not minimized by \( a^{opt}_{Lin} \), the TR-configuration for each of these flows corresponds to one of the three possible TR-configurations shown in Fig. 5. Therefore, the three possible TR-configurations describe all the possible ways in which optimal Lin-NTC-AP TRs do not minimize the maximum contention experienced by one or more flows in the network. The first of these TR-configurations is shown in Fig. 5a, the second is shown in Fig. 5b and Fig. 5c, and the third is shown in Fig. 5d and Fig. 5e.

The following analysis is based on the following hidden nodes in Fig. 5. In Fig. 5a, Fig. 5b, Fig. 5c, only one hidden node is considered per figure. This node is the sink of \( f_2 \). In Fig. 5d and Fig. 5e, several hidden nodes are considered per figure. These nodes are the sinks of \( f_5, f_6, ..., f_5+|P_{int}^{f_4}| \).

The contributions that the hidden nodes in Fig. 5a, Fig. 5b, Fig. 5c, Fig. 5d, and Fig. 5e make to the contention variation of \( f_1 \) and \( f_3 \) are denoted by \( \Delta_{V}^{(f_1)} \) and \( \Delta_{V}^{(f_3)} \) respectively. The contribution that the hidden node in Fig. 5c makes to the contention variation of \( f_4 \) is denoted by \( \Delta_{V}^{(f_4)} \). \( \Delta_{V}^{(f_4)} \) is different from zero only when the TRs are determined from \( a^{opt}_{Lin} \) (i.e., dashed-line TRs). When the TRs are determined from \( a^{opt}_{NTC} \) (i.e., solid-line TRs), the nodes are no longer hidden nodes of any of \( f_1 \)’s links, so they do not contribute to \( f_1 \)’s contention variation (i.e., \( \Delta_{V}^{(f_1)} = 0 \)). On the other hand, \( \Delta_{V}^{(f_3)} \) and \( \Delta_{V}^{(f_4)} \) are different from zero only when the TRs are determined from \( a^{opt}_{NTC} \) (i.e., solid-line TRs). When the TRs are determined from \( a^{opt}_{NTC} \) (i.e., dashed-line TRs), the nodes are no longer hidden nodes of any of the links of \( f_3 \) and \( f_4 \), so they do not contribute to the contention variation of \( f_3 \) and \( f_4 \) (i.e., \( \Delta_{V}^{(f_3)} = \Delta_{V}^{(f_4)} = 0 \)).

It is assumed that the values taken by \( \Delta_{V}^{(f_1)} \), \( \Delta_{V}^{(f_3)} \), and \( \Delta_{V}^{(f_4)} \) that are greater than zero meet the following inequalities. Otherwise, if this assumption is not made,
and in TR-configuration 2-multiple-flows (Fig. 5c), and in TR-configuration 3 (Lin-NTC-AP TRs): multiple active nodes are hidden nodes of multiple links.

Fig. 5. Suboptimal TR-configurations of the Lin-NTC-AP

The solution \( a_{\text{Lin}}^{\text{opt}} \) becomes optimal. In TR-configuration 1 (Fig. 5a), TR-configuration 2-single-flow (Fig. 5b), and TR-configuration 3 (Fig. 5d and Fig. 5e), \( \Delta f_1^i > \Delta V^i \), and in TR-configuration 2-multiple-flows (Fig. 5c), \( \Delta f_1^i + \Delta f_3^i > \Delta V^i \).

Tables 6 to 8 show the objective-function values of the Lin-NTC-AP and the NTC-AP evaluated at \( a_{\text{Lin}}^{\text{opt}} \) and \( a_{\text{NTC}}^{\text{opt}} \) for the three TR-configurations. The values of the objective functions evaluated at \( a_{\text{opt}}^{\text{Lin}} \) (i.e., \( c_{\text{Lin}}(a_{\text{opt}}^{\text{Lin}}) \) and \( c_{\text{NTC}}(a_{\text{opt}}^{\text{NTC}}) \)) are given in terms of the values of the objective functions evaluated at \( a_{\text{Lin}}^{\text{opt}} \) (i.e., \( c_{\text{Lin}}(a_{\text{opt}}^{\text{Lin}}) \) and \( c_{\text{NTC}}(a_{\text{opt}}^{\text{NTC}}) \)). In this way, the factors that cause the difference between the objective-function values can be identified. For example, Table 6 states that

\[
\begin{align*}
c_{\text{Lin}}(a_{\text{opt}}^{\text{Lin}}) - c_{\text{Lin}}(a_{\text{opt}}^{\text{NTC}}) &= \Delta V^i - \Delta f_1^i, \\
c_{\text{NTC}}(a_{\text{opt}}^{\text{Lin}}) - c_{\text{NTC}}(a_{\text{opt}}^{\text{NTC}}) &= d_{\text{max}}.
\end{align*}
\]

The following analysis is divided into two parts. In the first part, the reason that the Lin-NTC-AP game reaches suboptimal solutions of the NTC-AP in the three TR configurations in Fig. 5 is proved. In the second part, the maximum difference between the optimal and suboptimal solutions is calculated. Both parts are based on the effects of switching from the TRs given by \( a_{\text{opt}}^{\text{Lin}} \) (i.e., dashed-line TRs) to the ones given by \( a_{\text{opt}}^{\text{NTC}} \) (i.e., solid-line TRs).

**The Lin-NTC-AP game reaches suboptimal solutions of the NTC-AP in the three TR configurations in Fig. 5**

In TR-configuration 1 (see Fig. 5a and Table 6), the dashed-line TR of the destination node of flow \( f_2 \) partially covers flow \( f_1 \) and does not cover \( f_3 \), and the solid-line TR covers \( f_3 \) completely and partially covers \( f_3 \). According to (17), the value of \( c_{\text{Lin}} \) changes due to changes on the total contention (i.e., \( \sum_{f_n \in F} c_{f_n}^V(r) \)) and contention variation (i.e., \( \sum_{f_n \in F} c_{f_n}^r(r) \)). The total contention is decreased by the value of the degree of the link of \( f_1 \) that is partially covered by the dashed-line TR, and it is increased by the value of the degree of the link of \( f_2 \) that is partially covered by the solid-line TR28. In the worst case scenario (i.e., when \( c_{\text{NTC}}(a_{\text{opt}}^{\text{Lin}}) - c_{\text{NTC}}(a_{\text{opt}}^{\text{opt}}) \) is maximum), these two link degrees are equal to \( d_{\text{max}} \), so the total contention does not change. The contention variation is increased by \( \Delta V^i - \Delta f_1^i \). Therefore, the total difference between \( c_{\text{Lin}}(a_{\text{opt}}^{\text{opt}}) \) and \( c_{\text{Lin}}(a_{\text{opt}}^{\text{Lin}}) \) is \( \Delta f_1^i - \Delta V^i \). This result is the reason that, for TR-configuration 1, the Lin-NTC-AP game selects the dashed-line TR, so it reaches a suboptimal equilibrium that does not minimize \( c_{\text{NTC}} \).

In TR-configuration 2-single-flow (see Fig. 5b and Table 6), the total contention is decreased by the values of the degrees of the two links of \( f_1 \) that are partially covered by the dashed-line TR, and it is increased by the values of the degrees of the two links of \( f_3 \) that are

---

**TABLE 6**
Lin-NTC-AP and NTC-AP objective function values: TR-configuration 1 (see Fig. 5a), and TR-configuration 2-single-flow (see Fig. 5b)

<table>
<thead>
<tr>
<th></th>
<th>( c_{\text{Lin}}(a_{\text{opt}}^{\text{Lin}}) )</th>
<th>( c_{\text{Lin}}(a_{\text{opt}}^{\text{Lin}}) + \Delta V^i - \Delta f_1^i )</th>
<th>( c_{\text{NTC}}(a_{\text{opt}}^{\text{Lin}}) )</th>
<th>( c_{\text{NTC}}(a_{\text{opt}}^{\text{Lin}}) - d_{\text{max}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lin-NTC-AP</strong></td>
<td>( a_{\text{Lin}}^{\text{opt}} )</td>
<td>( a_{\text{NTC}}^{\text{opt}} )</td>
<td>( a_{\text{NTC}}^{\text{opt}} )</td>
<td>( a_{\text{NTC}}^{\text{opt}} )</td>
</tr>
<tr>
<td><strong>NTC-AP</strong></td>
<td>( c_{\text{Lin}}(a_{\text{opt}}^{\text{Lin}}) )</td>
<td>( c_{\text{Lin}}(a_{\text{opt}}^{\text{opt}}) + \Delta V^i - \Delta f_1^i )</td>
<td>( c_{\text{NTC}}(a_{\text{opt}}^{\text{Lin}}) )</td>
<td>( c_{\text{NTC}}(a_{\text{opt}}^{\text{Lin}}) - d_{\text{max}} )</td>
</tr>
</tbody>
</table>

**TABLE 7**
Lin-NTC-AP and NTC-AP objective function values: TR-configuration 2-multiple-flows (see Fig. 5c)

<table>
<thead>
<tr>
<th></th>
<th>( c_{\text{Lin}}(a_{\text{opt}}^{\text{Lin}}) )</th>
<th>( c_{\text{Lin}}(a_{\text{opt}}^{\text{Lin}}) + \Delta V^i - \Delta f_1^i )</th>
<th>( c_{\text{NTC}}(a_{\text{opt}}^{\text{Lin}}) )</th>
<th>( c_{\text{NTC}}(a_{\text{opt}}^{\text{Lin}}) - d_{\text{max}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lin-NTC-AP</strong></td>
<td>( a_{\text{Lin}}^{\text{opt}} )</td>
<td>( a_{\text{NTC}}^{\text{opt}} )</td>
<td>( a_{\text{NTC}}^{\text{opt}} )</td>
<td>( a_{\text{NTC}}^{\text{opt}} )</td>
</tr>
<tr>
<td><strong>NTC-AP</strong></td>
<td>( c_{\text{Lin}}(a_{\text{opt}}^{\text{Lin}}) )</td>
<td>( c_{\text{Lin}}(a_{\text{opt}}^{\text{opt}}) + \Delta V^i - \Delta f_1^i )</td>
<td>( c_{\text{NTC}}(a_{\text{opt}}^{\text{Lin}}) )</td>
<td>( c_{\text{NTC}}(a_{\text{opt}}^{\text{Lin}}) - d_{\text{max}} )</td>
</tr>
</tbody>
</table>

---

28. See the example in Fig. 4 for the explanation.
partially covered by the solid-line TR. In the worst-case scenario, the link degrees are equal to \(d_{\text{max}}\), so the total contention does not change. The contention variation is increased by \(\Delta f_3^f - \Delta f_4^f\). Therefore, the total difference between \(c_{\text{Lin}}(a_{\text{opt}}^f)\) and \(c_{\text{Lin}}(a_{\text{opt}}^f) + \Delta f_3^f - \Delta f_4^f\). This is the reason that, for TR-configuration 2-single-flow, the Lin-NTC-AP game selects the dashed-line TR, so it reaches a suboptimal equilibrium that does not minimize \(c_{\text{NTC}}\).

In TR-configuration 2-multiple-flows (see Fig. 5c and Table 7), the total contention is decreased by the values of the degrees of the two links of \(f_1\) that are partially covered by the dashed-line TR, and it is increased by the values of the degrees of the link of \(f_3\) and the link of \(f_4\) that are partially covered by the solid-line TR. In the worst-case scenario, the link degrees are equal to \(d_{\text{max}}\), so the total contention does not change. The contention variation is increased by \(\Delta f_3^f + \Delta f_4^f - \Delta f_1^f\). Therefore, the total difference between \(c_{\text{Lin}}(a_{\text{opt}}^f)\) and \(c_{\text{Lin}}(a_{\text{opt}}^f) + \Delta f_3^f - \Delta f_4^f\). This result is the reason that, for TR-configuration 2-multiple-flows, the Lin-NTC-AP game selects the dashed-line TR, so it reaches a suboptimal equilibrium that does not minimize \(c_{\text{NTC}}\).

In TR-configuration 3 (see Fig. 5d, Fig. 5e, and Table 7), the total contention is decreased by the values of the degrees of all the links of \(f_1\) that are partially covered by the dashed-line TRs, and it is increased by the values of the degrees of the links of \(f_3\) that are partially covered by the solid-line TRs. In the worst-case scenario, the link degrees are equal to \(d_{\text{max}}\). Given that there are a total of \(P_{\text{int}}^f + 1\) TRs that cover partially the links of \(f_1\) and \(f_5\), the total decrease and increase are each equal to \((P_{\text{int}}^f + 1)d_{\text{max}}\). Therefore, the contention does not change. The contention variation is increased by \(\Delta f_3^f - \Delta f_4^f\). Therefore, the total difference between \(c_{\text{Lin}}(a_{\text{opt}}^f)\) and \(c_{\text{Lin}}(a_{\text{opt}}^f) + \Delta f_3^f - \Delta f_4^f\). This result is the reason that, for TR-configuration 3, the Lin-NTC-AP game selects the dashed-line TRs, so it reaches a suboptimal equilibrium that does not minimize \(c_{\text{NTC}}\).

### Table 8

Lin-NTC-AP and NTC-AP objective function values:
TR-configuration 3 (see Fig. 5d and Fig. 5e)

<table>
<thead>
<tr>
<th></th>
<th>Lin-NTC-AP</th>
<th>NTC-AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_{\text{opt}}^f)</td>
<td>(c_{\text{Lin}}(a_{\text{opt}}^f))</td>
<td>(c_{\text{NTC}}(a_{\text{opt}}^f))</td>
</tr>
</tbody>
</table>

The maximum difference between the optimal solution of the NTC-AP and the solution calculated by the Lin-NTC-AP game is bounded

The suboptimal equilibrium of the Lin-NTC-AP game reaches a value for \(c_{\text{NTC}}\) that is greater than the optimal by a difference which is upper-bounded as follows.

In TR-configuration 1 (see Fig. 5a and Table 6), when the TR changes from the dashed-line TR to the solid-line TR, the maximum contention experienced by \(f_1\) is decreased by the value of the degree of the link of \(f_1\) that is partially covered by the dashed-line TR. The maximum contention experienced by \(f_3\) does not change because \(f_3\) is the cause of the maximum contention of \(f_3\) when the dashed-line TR is set, and when the solid-line TR is set, this maximum contention is not exceeded, i.e., when the solid-line TR is set, the nodes of \(f_3\) whose incoming links are partially covered by the TRs of \(f_4\) and \(f_2\) experience the same maximum contention, which is equal to \(2d_{\text{max}}\). This is shown in Fig. 6 in which the nodes of \(f_3\) with maximum contention are highlighted and only the TRs that cause their contention are included. Therefore, given that the maximum contention experienced by \(f_1\) is decreased by a value equal to one link degree and the maximum contention experienced by \(f_3\) is not varied, \(c_{\text{NTC}}(a_{\text{opt}}^f)\) is higher than \(c_{\text{NTC}}(a_{\text{opt}}^f)\) by at most \(d_{\text{max}}\).

In TR-configuration 2-single-flow (see Fig. 5b and Table 6), the maximum contention experienced by \(f_1\) is decreased by the value of the degree of one of the two links of \(f_1\) that are partially covered by the dashed-line TR. The maximum contention experienced by \(f_3\) does not change because \(f_3\) is the cause of the maximum contention of \(f_3\) when the dashed-line TR is set, and when the solid-line TR is set, this maximum contention is not exceeded, i.e., when the solid-line TR is set, the nodes of \(f_3\) whose incoming links are partially covered by the TRs of \(f_4\) and \(f_2\) experience the same maximum contention, which is equal to \(2d_{\text{max}}\). Therefore, given that the maximum contention experienced by \(f_1\) is decreased by a value equal to one link degree and the maximum contention experienced by \(f_3\) is not varied, \(c_{\text{NTC}}(a_{\text{opt}}^f)\) is higher than \(c_{\text{NTC}}(a_{\text{opt}}^f)\) by at most \(d_{\text{max}}\).

In TR-configuration 2-multiple-flow (see Fig. 5c and Table 7), the maximum contention experienced by \(f_1\) is decreased by the value of the degree of one of the two links of \(f_1\) that are partially covered by the dashed-line TR. The maximum contention experienced by \(f_3\) does not change because \(f_3\) is the cause of the maximum contention of \(f_3\) when the dashed-line TR is set, and when the solid-line TR is set, this maximum contention is not exceeded, i.e., when the solid-line TR is set, the nodes of \(f_3\) whose incoming links are partially covered by the TRs of \(f_4\) and \(f_2\) experience the same maximum contention, which is equal to \(2d_{\text{max}}\). Therefore, given that the maximum contention experienced by \(f_1\) is decreased by a value equal to one link degree and the maximum contention experienced by \(f_3\) is not varied, \(c_{\text{NTC}}(a_{\text{opt}}^f)\) is higher than \(c_{\text{NTC}}(a_{\text{opt}}^f)\) by at most \(d_{\text{max}}\).
equal to one link degree and the maximum contentions experienced by \( f_3 \) and \( f_4 \) are not varied, \( c_{\text{NTC}}(a_{\text{Lin}}^{\text{opt}}) \) is higher than \( c_{\text{NTC}}(a_{\text{Lin}}) \) by at most \( d_{\text{max}} \).

In TR-configuration 3 (see Fig. 5d, Fig. 5e, and Table 8), the maximum contention experienced by \( f_1 \) is decreased by twice the value of the degree of the link of \( f_1 \) that is partially covered by two dashed-line TRs. The maximum contention experienced by \( f_3 \) does not change because \( f_2 \) and \( f_4 \) are the cause of the maximum contention of \( f_3 \) when the dashed-line TRs are set, and when the solid-line TRs are set, this maximum contention is not exceeded, i.e., when the solid-line TRs are set, the two nodes of \( f_3 \) whose incoming links are partially covered by the TRs of \( f_2 \) and \( f_4 \) and by the TRs of \( f_5 \) and \( f_6 \) respectively experience the same maximum contention, which is equal to \( 3d_{\text{max}} \). Therefore, given that the maximum contention experienced by \( f_1 \) is decreased by a value equal to two link degrees and the maximum contention experienced by \( f_3 \) is not varied, \( c_{\text{NTC}}(a_{\text{Lin}}^{\text{opt}}) \) is higher than \( c_{\text{NTC}}(a_{\text{Lin}}) \) by at most \( 2d_{\text{max}} \).

In the general case, i.e., when the maximum contention experienced by two or more flows are not minimized by \( a_{\text{Lin}}^{\text{opt}} \), the TR-configuration for each of these flows corresponds to one of the three possible TR-configurations. The maximum possible difference between \( c_{\text{NTC}}(a_{\text{Lin}}^{\text{opt}}) \) and \( c_{\text{NTC}}(a_{\text{Lin}}) \) is achieved when TR-configurations 1 or 2 single-flow are repeated as many times as possible. The reason is that these configurations are the ones that use the lowest number of flows. Therefore, they are the ones that can be replicated the highest number of times. For each replication, the value is increased by \( d_{\text{max}} \). This is shown in Fig. 7 for TR-configuration 2-single-flow which is repeated the maximum number of times, i.e., \( \frac{r_{\text{max}} - r_{\text{opt}}^2}{2} \) times.

\section*{APPENDIX J}

\section*{SIMULATION RESULTS}

The simulation was configured as follows\textsuperscript{29}. The link scheduling policy was GM-RBDS [14]. The flow paths were obtained using the min-hop routing algorithm. The nodes were uniformly distributed in a square such that the node density was always kept at 15 nodes per area unit. There were a total of 20 nodes. The maximum TR for all the nodes was set at 0.3 (i.e., \( r_{\text{max}} = 0.3 \)). The connectivity of the flows with the nodes’ TRs set at \( r_{\text{max}} \) was verified before executing the min-hop routing, WiMAX-Mesh-NTC, HSRA, and MinPower algorithms. The source and destination of every flow were uniformly distributed across all the nodes in the network. The min-hop algorithm calculated the flow paths when the TRs were set at \( r_{\text{max}} \).

Each of the TR-control algorithms (i.e., WiMAX-Mesh-NTC, HSRA, MinPower, and MaxPower) calculated the set of TRs for 400 different networks configured as explained previously. These networks were divided into four groups of 100 networks each depending on the number of flows in the networks. From the first to the fourth group, the networks had 5, 10, 15, and 20 flows respectively. The algorithms calculated a set of TRs for each of the 400 networks, and the throughput percent difference \( \delta_T \) (see (19)) was calculated for each of the TR sets.

Fig. 8 shows the ratio of TR sets calculated by the algorithms that had a \( \delta_T \) less than or equal to 10%. Therefore, Fig. 8 shows the ability of the algorithms to reach the maximum possible total throughput (i.e., \( \lambda_T(r_{\text{opt}}^\text{opt}) \)). According to Fig. 8, the WiMAX-Mesh-NTC algorithm always outperforms the MinPower and MaxPower algorithms. For example, when there are 10 flows (see Fig. 8b), 72% of the TR sets calculated by WiMAX-Mesh-NTC have a \( \delta_T \) of at most 10% while 62% and 64% of the TR sets calculated by MinPower and MaxPower have such \( \delta_T \). Therefore, in all the cases the WiMAX-Mesh-NTC algorithm was able to find more effectively a set of TRs that maximizes the total throughput \( \lambda_T \). This result is also confirmed in Fig. 9, which shows the cumulative distribution of \( \delta_T \). According to Fig. 9, the cumulative distribution of \( \delta_T \) is always greater for WiMAX-Mesh-NTC than for the MinPower and MaxPower algorithms.

When the WiMAX-Mesh-NTC and HSRA algorithms are compared, the flow density needs to be considered. Let the flow density be the ratio among the number of flows and the total number of nodes. Given that in Fig. 8 and Fig. 9 the number of nodes does not change while the number of flows increases from 5 to 20, the flow density increases. Fig. 9a and Fig. 9b show that the algorithms have the same performance when the flow density is low. Fig. 9c and Fig. 9d show that WiMAX-Mesh-NTC has lower performance than HSRA when the flow density is medium or high. The intuition behind this behavior is that the probability that any of the worst-scenario cases shown in Fig. 5 takes place increases with the flow density, and such cases affect the performance of WiMAX-Mesh-NTC (see Theorem 5). This is not the case for HSRA because HSRA is a centralized algorithm, so it has a global view of the network. Therefore, it is able to avoid the worst-case scenarios more effectively. However, the advantage of WiMAX-Mesh-NTC over HSRA is that it is distributed while HSRA is not.

\textsuperscript{29} This is the same configuration used in [18].
Mesh-NTC requires knowledge about the flows in $F^n$ while HSRA requires knowledge about all the flows in $F$. Therefore, WiMAX-Mesh-NTC is more amenable for implementation and less vulnerable to the negative effects of network-state-information delay [43].

It should be noted that as the flow density increases, the performance of all the algorithms become closer to each other. This can be seen in Fig. 9 as the number of flows increases from 5 in Fig. 9a to 20 in Fig. 9d. Therefore, when the flow density is high, the classic approach of maximizing spatial reuse and WiMAX-Mesh-NTC are similarly effective.

The convergence of the WiMAX-Mesh-NTC algorithm, which is guaranteed according to Corollary 3, is shown in Fig. 10. Fig. 10 shows the distribution of the number of TR updates that took place on the 400 simulated scenarios. By TR update, it is meant that a flow changes the TR of a node. The results show that no more than 3 TR updates were necessary for any of the scenarios. If the network dynamics (e.g., creation and deletion of flows, new nodes’ locations) change at a rate lower than the time necessary for calculating all the updates, the algorithm is able to converge to a stable set of TRs. Therefore, in the simulated scenarios, if network dynamics were considered, it would be necessary that they did not change faster than the time required for calculating 3 TR updates. Fig. 10 also shows that as the flow density increases, more TR updates are necessary in average. This can be seen on the columns for 0 TR updates. The columns decrease with the flow density. Their values are 75%, 75%, 68%, and 66% for 5, 10, 15, and 20 flows respectively. Finally, these results also show that for at least 29% of the simulated scenarios, WiMAX-Mesh-NTC found TR sets different from the MinPower TR sets. This is the reason WiMAX-Mesh-NTC outperforms MinPower as discussed in Fig. 8 and Fig. 9.

REFERENCES

[50] S. Buzzi, G. Colavolpe, D. Saturnino, and A. Zappone, “Potential games for energy-efficient power control and subcarrier allocation


