Consiglio Nazionale delle Ricerche

A Framework for Incentive Compatible Topology Control in Non-Cooperative Wireless Multi-Hop Networks

S. Eidenbenz, P. Santi

IIT TR-12/2005

Technical report

Luglio 2005

Istituto di Informatica e Telematica

A Framework for Incentive Compatible Topology Control in Non-Cooperative Wireless Multi-Hop Networks

Stephan Eidenbenz Paolo Santi

***Abstract*— In this paper we consider the problem of building and maintaining a network topology with certain desirable features in a wireless multi-hop network where nodes behave like selfish agents. We first provide examples showing that existing topology control approaches are not resilient to strategic node behavior, indicating the need of considering possible selfish node behavior at the design stage. Given this observation, we propose a general framework that can be used as a guideline in the design of incentive compatible topology control protocols. As examples of application of our framework to specific topology control protocols, we present incentive compatible distributed algorithms for building the minimum spanning tree (MST) and the k-closest neighbors graph, which are very well-known topology control approaches. To the best of our knowledge, the ones presented in this paper are the first incentive compatible realizations of topology control presented in the literature.**

I. Introduction

Topology formation and maintenance are key tasks for any wireless multi-hop network. In fact, nodes in a wireless multi-hop network typically have the capability of adjusting their transmit power below a maximum value. Since decreasing the transmit power has the positive effect of reducing the interference level in the network and the power consumption at the node, but an excessive reduction of the node transmit power levels could lead to network disconnection, the goal of the topology formation task is to appropriately set the node transmit power levels in such a way that the resulting network topology has the desired features while reducing the interference level and the node power consumption as much as possible. In addition, as nodes can move and dynamically join/leave the network, the task of topology maintenance comes into play, with the goal of reconfiguring the node transmit power levels so that to maintain the desired network topology.

Several protocols for distributed topology formation and maintenance in wireless multi-hop networks, called *topology control protocols*, have been recently introduced in the literature. Formally, the topology control task (TC for short) can be described as follows. We are given a communication graph $G = (N, E)$, where N is the set of network nodes and E is the set of all possible links in the network, i.e. the set of all wireless links (u, v) that are sustainable when node u transmits at maximum power and the other network nodes (including node v) are silent. For this reason, G is also called the *maxpower graph*. The topology control task consists in selecting a subset of the edges in E such that the selected link set satisfies a number of desirable properties such as connectivity, sparseness, planarity, etc. To account for node mobility and dynamic join/leave of network nodes, it is typically assumed that the topology control task is executed periodically, or on demand when excessive link failures occur that point to topology changes. Once the desired network topology has been determined, it is assumed that packets are sent through the network along the selected subset of links only.

Although a great variety of topology control concepts and protocols have been proposed in the literature (see, e.g., [2], [5], [7], [14], [15], [16], [20], [21], [23]), all of them are based on the assumption that individual network nodes act in an altruistic, non-selfish way for the common good of having a well-functioning network. Unfortunately, this assumption does not hold in all application scenarios in which nodes are owned by different, independent, profit-maximizing entities, such as an ad hoc phone user, certain types of wireless mesh networks, and so on. As it will be discussed in detail in Section II, selfish node behavior has a disruptive effect on topology control protocols, since the individual goals of a network node often conflict with the goals of the network designer. As a consequence of this, current approaches to the topology control problem are doomed to perform poorly in a non-cooperative wireless multi-hop network, unless adequate countermeasures are taken.

While the problem of stimulating cooperation in wireless networks has been addressed at various levels of the network architecture (e.g., at the MAC [9], [18], [22], routing [1], [3], [8], [11], [24], [25], and application layer [4], [12]), to the best of our knowledge the problem of building and maintaining the network topology in a non-cooperative wireless multi-hop network has not been addressed so far. The only paper that considers the issue of cooperation in topology formation is [10], where the authors study the Nash Equilibria[1] of some topology control games. However, the analysis reported in [10] is based on a centralized approach to the topology control problem, and does not give any clue on how nodes could be motivated to cooperate in the construction of a certain desired network topology. Another related branch of research focused on studying cooperation issues in the power control problem

S. Eidenbenz is with Los Alamos National Labs, Los Alamos NM, USA. Paolo Santi is with IIT-CNR, Pisa, Italy.

[1] Quite informally, a set $S = \{S_1, \ldots, S_n\}$ of strategies for nodes $u_1, \ldots, u_n$ is a Nash Equilibrium if no node u_i has an incentive to deviate from its strategy S_i, provided the other nodes do not change their strategy.

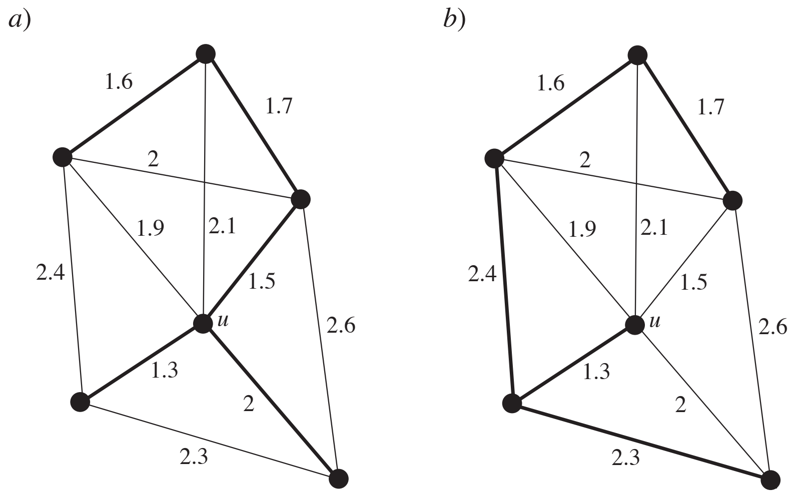

Fig. 1. Disruptive effect of selfish node behavior in constructing the MST: the globally optimal solution (the MST on the left) is sub-optimal from node u's selfish point of view. Hence, node u has an incentive to build a different topology (the spanning tree on the right), which is optimal from u's viewpoint, but globally sub-optimal. Edges are labeled with their weight, which is proportional to the link length. Links in the constructed topology are in bold.

arising in cellular networks, which consists in varying the transmit power level at the base station and/or at the mobile phone so that to optimize the link quality [17], [26]. However, the focus in power control is different than in the case of topology control: namely, optimizing a single transmission between a mobile user and the base station, instead of forming and maintaining a communication graph with certain desired features.

We believe solving the problem of stimulating cooperation in the task of building and maintaining the network topology is fundamental to a successful realization of the wireless multi-hop paradigm in a non-cooperative environment, as this task is a necessary building block on which protocols at higher and lower layers rely. In other words, it is quite unrealistic to assume (as, for instance, has been done in [11]) that network nodes act selfishly in performing the routing task, while they are willing to cooperate in the task of topology control.

In this paper, we introduce a framework to stimulate selfish network nodes to cooperate in the formation and maintenance of the network topology. Our framework is based on game-theoretic concepts from mechanism design, and it can be used as a guideline in the design of incentive compatible topology control protocols for wireless multi-hop networks. As examples of application of our framework, we present an incentive compatible implementation of two popular topology control approaches, and we formally prove that it is in the best interest of the network nodes to behave according to the specifications of our proposed incentive compatible topology control protocols.

The rest of this paper is organized as follows. In Section II, we present examples of the disruptive effect of selfish node behavior on topology control, thus motivating the need for considering selfish node behavior at the design stage. In Section III, we introduce our framework for an incentive compatible realization of topology control. Then, we present the application of our framework to two popular topology control approaches: MST-based topology control (Section IV) and closest neighbor-based topology control (Section V). Finally, Section VI concludes the paper.

II. The case for incentive compatible TC

In this section, we present examples of the disruptive effect of selfish node behavior on some of the topology control protocols introduced in the literature. Before presenting the examples, we have to define a model of selfish node in a multi-hop wireless network.

A selfish node aims at increasing the benefit it gets from executing the protocol, while reducing as much as possible the incurred costs in the constructed topology. In the context of topology control, it is reasonable to assume that a node maximizes its benefit when it is connected to as many other nodes as possible (i.e., when the network is connected), and that a node incurs a cost which is proportional to the number and/or power cost of the links incident into it in the selected network topology. A formal definition of utility of a node in the context of topology control is deferred to the next section.

Note that network connectivity is a global property which cannot be verified locally, so a node u might not be able to check whether a certain power level setting at u results in a globally connected topology. To account for this, we consider two different models of selfish node: $i)$ the *global* model, in which a node u somehow knows whether a certain power level setting at u results in a globally connected topology, and (ii) the *local* model, in which a node u can only verify local properties of the generated network topology (e.g. number and positions of neighbors in the constructed topology). Note that the global model, although possibly unrealistic in many application scenarios, is worth of investigation since the selfish node is assumed to be 'more powerful' than in the local model. In other words, if a certain topology control protocol is shown to be resilient to selfish node behavior in the global model, it retains the same property in the weaker local model.

Let us now turn our attention to some of the topology control approaches introduced in the literature. One of the

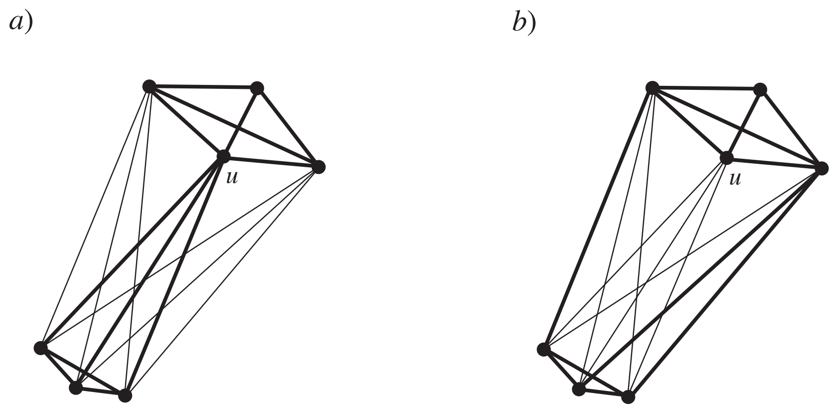

Fig. 2. Disruptive effect of selfish node behavior in constructing the KCN topology: the globally optimal solution (on the left) is sub-optimal from node u's selfish point of view. Hence, node u has an incentive to build a different topology (on the right), which is optimal from u's viewpoint, but globally sub-optimal. To make the drawing more readable, we omitted the edge weights, which are assumed to be proportional to the link length. Links in the constructed topology are in bold.

most studied approaches to the topology control problem is based on the computation of the MST on the maxpower graph, where links are assigned weights according to some criteria (e.g., link length, power cost, expected interference on the link, and so on). The interest in the MST is motivated by the fact that it is the topology of least total cost that maintains the network connected. Example of protocols based on this concept can be found in [7], [15], [20], [21].

Consider the situation depicted in Figure 1-a), which represents the MST computed on a certain maxpower graph under the assumption that all the nodes act unselfishly. In this situation, the goal (which is altruistically pursued by all network nodes) is to build a connected topology of minimum cost. Note that this globally optimal solution might be suboptimal from a certain node's point of view. For instance, consider node u in Figure 1-a). The cost incurred by u in the MST is 4.8, which equals the cost of the MST edges incident into u. However, if the goal of node u were to maximize its utility, the topology represented in Figure 1-b) would be the optimal choice. In fact, with this network topology node u would still be connected to all the other network nodes (i.e., its benefit would be maximized), but the incurred cost would be minimized (1.3, instead of 4.8).[2] Note that the topology of Figure 1-b) is suboptimal under a network-wide perspective, since its total cost is 9.3 instead of 8.1.

Thus, we are in a situation in which the desired designer goal (building a connected topology of minimum cost) is in contrast with the selfish goal of a network node (building a topology which maximizes its own utility). The consequence of selfish node behavior on MST-based topology control is disruptive: a selfish node has no interest in cooperating to the MST construction, while it is instead motivated to build a different, suboptimal network topology. The situation is even worse if several nodes act selfishly, since in this case the constructed network topology in general can be very different from the optimal topology, and it might even not satisfy fundamental properties such as connectivity.

It is interesting to note that the MST corresponds to a Nash Equilibrium of the Strong Connectivity Game analyzed in [10], where the goal of each node is to connect to every other node while reducing its transmit power level as much as possible. This fact indicates that, if node u belongs to the MST, it has no incentive to unilaterally changing its links, thus apparently contradicting the above observation about the disruptive effective of selfish node behavior on topology control. Indeed, the fact that the MST is a Nash Equilibrium is not in contrast with our observation, since in this paper we are considering the problem of *establishing* a topology with the desired features: the initial topology is the maxpower graph G (which is *not* a Nash Equilibrium), and the nodes have to agree on a better topology for sending packets. In this context, it is important to provide incentive to the nodes so that the composition of their local selfish behaviors results in the desired network topology. The fact that the final network topology is a Nash Equilibrium reinforces the argument that, once the MST has been established (using the IC-MST algorithm presented in Section IV), nodes have no incentive in changing the network topology.

Let us now consider another popular approach to the TC problem, which is based on the simple idea of building a topology in which each node is directly connected (using only bi-directional links) to a fixed number k of nodes. This idea has been exploited, for instance, in [5], [16], [21]. From a theoretical viewpoint, it has been proven in [27] that if the selected neighbors are the k closest neighbors and $k = c \ln n$, where n is the number of network nodes and c is a constant greater than 5.1774, then the resulting network topology is connected with high probability (under the assumption that nodes are distributed uniformly at random in the deployment area). In a more practical setting, Blough et al. have proven in [6] that connecting to 4-5 closest neighbors is sufficient to obtain a connected network (w.h.p.) for values of n up to 500. In the following, we denote the network topology in which every node is connected to the k closest neighbors by KCN. Note that, since we require links to be bi-directional, some of

[2]Here, we are implicitly assuming the global selfish node model, according to which a node u can somehow verify that the topology represented in Figure 1-b) is connected.

the nodes in KCN might have a degree higher than k.

Let us consider the example shown in Figure 2-a), where k is set to 3. In Figure 2-a), every node is connected to at least the 3 closest neighbors. Since links must be bi-directional, some of the nodes in the constructed topology can have more than 3 neighbors, as it is the case of node u in the example reported in Figure 2-a). What happens if node u act selfishly, instead of altruistically participating in the construction of the KCN topology? To answer this question, assume u behaves according to the local selfish node model, i.e. it can only verify local properties of the network topology. In this context, since node u cannot directly verify its ultimate goal (i.e., being connected to all the other nodes), it must rely on a local property which, under certain hypotheses, guarantees a high likelihood of generating a connected network. This is the case of the k closest neighbors property, which can be verified locally. Summarizing, we can assume that node u maximizes its benefit when it has a bi-directional direct link to each of its k closest neighbors. As in the previous example, the incurred cost of node u equals the sum of the weights of the edges incident into it in the constructed topology. Given this model, it is immediate to see that the network topology reported in Figure 2-b) is optimal under node u's selfish viewpoint, since the benefit is still maximized (u has bi-directional links to its $k = 3$ closest neighbors), and the incurred cost is minimized (exactly three edges are incident into u in the constructed topology, and these are shortest possible edges incident into u). As in the previous example, the optimal topology under node u's selfish perspective is suboptimal under a network-wide perspective, since the total cost of the constructed topology is higher than that of KCN. Hence, also in this case the effect of selfish node behavior on the topology control protocol is disruptive: if one or more nodes behave selfishly, the constructed topology can be very different from the desired topology, possibly impairing a fundamental property such as network connectivity.

Examples similar to the ones reported above can be easily found for virtually all of the topology control approaches proposed in the literature. Although not exhaustive of the many opportunities for selfish node behavior in topology control protocols, we believe the examples described in this section provide sufficient evidences of the need for considering selfish node behavior *at the design stage* of a topology control approach. In the following, we propose an incentive-based framework for designing topology control protocols that can be applied to most of the TC approaches introduced in the literature.

III. A GENERAL FRAMEWORK FOR INCENTIVE COMPATIBLE TC

Before introducing our framework, we need to model the topology control task as a game. We recall that the topology control task can be concisely defined as follows. We are given the maxpower graph $G = (N, E)$, where N is the set of network nodes and E is the set of all possible wireless links that can be sustained in the network. Each link $e = (u, v)$ in the maxpower graph is assigned a weight w_e, which is defined according to some criteria (e.g., link length, link power cost, expected interference on the link, and so on). The goal of TC is to select a subset E' of the edge set, such that the induced communication graph satisfies some desirable properties (e.g., connectivity, sparseness, low interference level, and so on), while the cost of E' is reduced as much as possible. The cost of E' can be defined as the sum of the weights of the edges in E', or as the maximum of the weights of the edges in E', or in other ways.

In the remainder of this paper we make the following assumptions:

A1. the edge weight is defined as follows:

$$w_e = n_e \cdot P_{uv} \ , \tag{1}$$

where P_{uv} represents the minimum transmit power needed to sustain the link $e = (u, v)$ with the desired properties (e.g., at most a given BER at a certain data rate), and n_e denotes the number of packets that will be sent across link e until the next topology control protocol execution.

A2. the cost function on the selected edge set E' is defined as the sum of the weights of the edges in E'. Formally,

$$c(E') = \sum_{e \in E'} w_e \ .$$

We remark that assumption A1. above is made for the sake of presentation only, and that the framework proposed in this paper remains valid for arbitrarily defined edge weights.

Parameter n_e in (1) depends on the set of sessions which take place in the time interval between two successive executions of the topology control protocol, and on how these sessions are routed through the network. Unless accurate information about the expected data traffic and the routing algorithm are known, it is difficult to predict the value of n_e for each link in the network. For this reason, in the following we assume all n_e's in the graph to be equal, and, consequently, we remove n_e from the definition of edge weight. Again, we remark that this assumption is made for the sake of presentation only, and that our framework can be applied with no modification in case of different n_e values on the links.

A final assumption made in our model concerns the wireless medium, which is assumed to be symmetric: P_{uv} corresponds to the minimum power required to sustain both the link from u to v and the reversed link from v to u. Note that this is the only assumption we make about the properties of the wireless channel. In particular, we *do not* assume that the radio coverage area is a perfect circle, nor that nodes have the same maximum transmit power.

In order to come up with a game-theoretic model for topology control, we need to determine what the utility functions of our players (i.e., network nodes) are, and what strategies they could follow in order to maximize their gain. Our goal as protocol designers is then to create a mechanism based on monetary transfers that makes behaving in accordance with the prescribed topology control protocol a *dominant strategy*[3]

[3]A strategy S for a player p is said to be *dominant* if, no matter what the strategies the other players play, S is the utility-maximizing strategy for p.

for the nodes. A topology control protocol (including the rules for the monetary transfer) that satisfies this property is said to be *incentive compatible*[4].

When defining a utility function, we have to distinguish a case in which a node is content with the outcome of the game (in our setting, the outcome of the game is the selected edge set E'), as opposed to an uncontent case. As discussed in Section II, the status of being content or not depends on the type of information the node has access to: in the global selfish node model, the player is assumed to be able to verify global network properties, while in the weaker local selfish node model the player can only verify local properties of the network topology. Example of situations in which a node is content can be 'I have a connection to all the other network nodes' (global model), and 'I have a direct, bi-directional link to at least my k closest neighbors' (local model). However, we outline that the framework described here in principle can be applied independently of the property which makes a node content.

To define the utility function of the player, we need also to determine the incurred costs of the player for a given outcome of the game. According to our global cost model, we assume that the incurred cost of node u for the outcome E' equals the sum of the cost of the links of E' incident into u. This cost model is coherent with a per-packet approach to topology control, in which nodes can change the transmit power level on a per-packet basis. In this context, the sum of the links of E' incident into u is an estimation of the cost node u incurs for sustaining the links in E'. How to generalize our framework to different definitions of incurred cost of the player is subject of ongoing research.

We are now ready to define the utility function used in our topology control game:

Definition 1 (Utility function): Let u be a node in $G = (N, E)$, and let $E' \subseteq E$ be the set of links in the topology built (possibly on a subset of the network nodes) at the end of the protocol execution. Let $\mathcal{G}$ be the (global or local) goal that node u is pursuing, and let $T^u_{\mathcal{G}}$ denote the set of topologies that make node u content according to goal $\mathcal{G}$. The *utility* of node u for a given constructed topology E' is defined as follows:

$$U(u,E') = \begin{cases} -\sum_{e=(u,v)\in E'} w_e + \\ \quad -pay(u,E') + pr(u,E') & \text{if } (N,E') \in T^u_{\mathcal{G}} \\ -M & \text{otherwise} \end{cases} \tag{2}$$

If node u is content with the network topology (N, E'), it incurs a cost of w_e for each adjacent edge $e = (u, v) \in E'$, representing the fact that it will have to transmit packets along these links until the topology is updated. Our framework define monetary transfers that we summarize in the utility function as a payment $pay(u, E')$ that node u will have to make in order to be allowed to participate in the network, as well as a premium payment $pr(u, E')$ that u will receive from other nodes, which represents the 'added value' that u brings to the network. Finally, if node u is not content with network topology (N, E'), it gets a negative utility of $-M$, where M is a large constant representing the fact that the node was not able to achieve its goal. Since we are considering a scenario in which nodes are selfish, but at the same time they are willing to connect to each other, we assume that $M \gg pay(u, E'), pr(u, E')$ and of all the edge weights.

[4]The terms *truthful* and *strategy-proof* are also used in the game theory literature.

The final step in the definition of our topology control game is determining the set of possible strategies for a player. To this purpose, we observe that every topology control protocol relies on nodes determining the weights w_e of the links to their neighbors. Such weights are typically calculated either through successively growing the emission energy in a test phase and then determining minimum energy emission levels by having receiver nodes reply to such a test packet, or simply by announcing GPS-based coordinates to all neighbors by emitting at maximum power. In a coordinate-based solution, it is usually assumed that a link weight is a known function of the Euclidean distance between the two nodes.

In the most general setting, we can thus assume that a node has at least the option of falsely declaring the weights w_e of its adjacent links. Many other possible cheating behaviors of selfish nodes are possible, but they depend on the specific topology control protocol at hand. For this reason, we do not consider them in our general framework, but we carefully discuss them in the following sections, which are devoted to protocol-specific incentive compatible realizations of topology control.

The field of mechanism design offers a standard solution, called a VCG-based mechanism [19], to incentivize all nodes to participate without lying about the weights of their adjacent links. We present this general scheme here and show in subsequent sections how it can be implemented efficiently for a few specific topology control approaches.

Given a communication graph $G = (V, E)$ and a topology control protocol $\mathcal{A}$, let $E_{\mathcal{A}} = E'$ denote the set of edges that are in the topology constructed by protocol $\mathcal{A}$, and let $E^{-u}_{\mathcal{A}}$ denote the set of edges that are in the topology constructed by protocol $\mathcal{A}$ when executed on the maxpower graph that does not contain node u (i.e., the graph with node set $N - \{u\}$). Finally, for any link set E'' let $|E''|$ denote the sum of the weights of all edges in E'', i.e., $|E''| = \sum_{e\in E''} w_e$. Then, we define the premium for node u as follows:

$$pr(u,E') = |E^{-u}_{\mathcal{A}}| - |E'| + \sum_{e=(u,v)\in E'} w^u_e \ , \tag{3}$$

where w^u_e is the cost of edge e as declared by node u.

Note that these premiums need to be funded in some way. While there are several ways of doing this, a standard solution is to equally subdivide these costs among all the $|N| = n$ nodes participating in the network (see e.g. Chapter 9 in [13]). Thus, we define the payment function as follow:

$$pay(u,E') = \frac{\sum_{u\in N} pr(u,E')}{n} \ .$$

This completes the definition of our general framework for an incentive-compatible implementation of topology control: a node u in the network receives a premium which equals the 'added value' that node u brings to the network (i.e., the difference between the cost of the computed topology without u and the cost of the topology including u), plus the declared cost incurred by u for joining the network. The overall amount of money which must be paid to the nodes is equally subdivided among the network nodes.

Note that the premium that a node receives might exceed, be equal to, or be less than the payment due by the node. However, even if a node ends up the game paying some money to be part of the network, this situation is still preferable to not joining the network, as this would drive down the utility function of the node to $-M$ (we recall that M is much larger than the due payment and of the edge weights).

Observe that the premium that a node receives depends on the costs incurred by node u in the computed topology *as declared by node* u itself. So, a selfish node might be tempted to falsely declaring these costs, so that to increase its utility. While a formal proof that this cheating behavior does not increase the utility of the node is deferred to the analysis of the protocol-specific incentive compatible TC implementations, we give here the intuition behind this proof.

The premium received by a node as defined in (3) is composed of three terms: the first term is not influenced by node u's declaration, since it is the cost of the topology computed on the graph which does not contain u; the third term can actually be increased by overdeclaring the cost of the edges incident into u. However, if these costs are overdeclared by, say, $\Delta > 0$ overall, also the cost of the computed topology $|E_{\mathcal{A}}|$ is increased of the same amount, and the premium due to node u is decreased by Δ. So, node u has no way of increasing its utility (we remark that the two other terms in the utility function of u do not depend on the declared edge weights) by falsely reporting its edge weights.

The framework described in this section is based on the well-known VCG mechanism, which has the nice feature of motivating nodes not to lie when reporting their weights. However, this comes at the price of paying nodes in excess to their real cost for being part of the network (in fact, $|E_{\mathcal{A}}^{-u}| - |E_{\mathcal{A}}|$ is in general a positive quantity), disclosing an economic inefficiency of the mechanism: the sum of the payments due to the nodes (which equals the sum of the premiums paid to the nodes) is in general higher than the cost of the generated topology E'. Unfortunately, it has been shown that, under realistic assumptions, there is no way of removing this inefficiency if the goal is to design an incentive compatible mechanism [19].

IV. Incentive compatible MST topology control

A. Problem definition and model

In this section we consider the problem of building and maintaining the MST of the maxpower graph G in a scenario in which the network nodes are selfish agents. To simplify the presentation of our protocol, we assume that no two edges e_1, e_2 in G exist such that $w_{e_1} = w_{e_2}$. Note that this assumption can be accomplished by ordering node IDs in lexicographical order, and by breaking ties according to the IDs of the endpoints of the edge. Under this assumption, the MST of G is uniquely defined.

To model selfish agents, we assume that each node is assigned with a *utility function*, which represents the benefit that the node gets in participating in the MST construction. The utility function is the same for all the network nodes, and it is defined as follows:

Definition 2 (Utility function for MST construction): Let u be a node in $G = (N, E)$, and let $T = (N_T, E_T)$ be the topology built (possibly on a subset of the network nodes) at the end of the MST protocol execution. The utility of node u, which is inspired by the general framework of Section III, is defined as follows:

$$U(u,T) = \begin{cases} -\sum_{e=(u,v)\in E_T} \frac{w_e}{2} - & \\ \quad -pay(u,T) + pr(u,T) & \text{if } u \in N_T \\ -M & \text{otherwise} \end{cases} \quad (4)$$

In the above definition we are assuming the global selfish node model. This is consistent with the nature of the MST, whose computation requires global knowledge. Differently from the general framework, we are also assuming that the cost of an edge is equally subdivided between its endpoints.

The goal of each node participating in the protocol execution is to maximize its utility function (selfish agent). To achieve its goal, a node can deviate from the behavior prescribed by the protocol, for instance by not sending a message, or by reporting false information, or by sending a message more than once. However, nodes are not allowed to coordinate their cheating behaviors in order to form a coalition (*no collusion*). Our goal, as the protocol designer, is to devise a mechanism (more specifically, a pricing rule) such that *every* node participating in the protocol maximizes its utility when a certain *social optimum* is achieved. This property is known as incentive compatibility in the game theory literature. In simple words, incentive compatibility ensures that the combination of the agents' selfish behaviors results in a desirable "social" behavior.

Returning to the problem at hand, our designer goal is to device a protocol such that the function

$$Soc(T) = \sum_{u \in N} U(u,T) \ ,$$

which represents the *social utility function*, is maximized when the constructed topology T is the MST built on G.

B. *The* IC-MST *protocol*

Our incentive compatible protocol for building the MST, called IC-MST, is essentially an incentive compatible implementation of the Prim's algorithm for building the MST, and it is defined as follows.

In order to simplify the presentation, we assume that there exists a network node (the *initiator*) that initiates the protocol at a certain time, and that the initiator node is unique (i.e., no other node in the network can initiate IC-MST execution). This assumption is reasonable, for instance, in wireless Mesh networks, where the initiator is a wireless Access Point, and the goal is to establishing (possibly multi-hop) wireless connections with mobile nodes in the vicinity of the AP. In this scenario, building a MST rooted at the AP is a reasonable choice.

IC-MST proceeds in rounds, adding a new node (with a corresponding edge) to the constructed topology at each round, until all the network nodes are connected ($n-1$ rounds in total). At each round, nodes are requested to pay an amount of money, which is delivered in a secure way to the initiator. The initiator also collects its own money. After round $n-1$, there is a last round called the *premium round*, during which the initiator delivers (in a secure way) the premiums to all the network nodes (including itself).

At round 1, the initiator starts the protocol execution, finding the edge e_1 of minimum cost incident into it, and the second best edge e_1^2. The node at the other endpoint of edge e_1 joins the network topology, forming a network with 2 nodes (denoted T_1). Edge e_1 is stored in the first element of the array $Edges[\ ]$, which keeps track of the network edges (this array and the arrays for payment tracking are stored at the initiator node). The initiator then collects the payments, which amount to $w_{e_1^2}$. The money due is equally shared between the two nodes in T_1. Furthermore, $w_{e_1^2}$ is stored in the first element of the array $Payments[\ ]$, which keeps track of the payments performed in the various rounds.

At the generic round i, the initiator asks each node in the current tree T_{i-1} to report the weights of the edges incident into it whose other endpoint is not in T_{i-1}. If no such edge exists, we are done (all the n nodes are included in the topology), and the initiator starts the premium round (see below). If a node u is adjacent to $h > 2$ edges with nodes not in T_{i-1}, it is sufficient that it reports to the initiator the cost of the two edges of mininum cost. If these costs were already reported by u to the initiator in a previous round, node u simply does not report any cost. After all the costs have been collected, the initiator selects the minimum cost edge $e_i = (u, v)$ as the new edge to be added to T_{i-1}. Edge e_i is stored in $Edge[i]$, and the node at the endpoint of e_i which was not in T_{i-1} (say, node v) is added to the newly formed topology T_i. Before computing the payments, the initiator sends a message to node v, asking him to send the list of its neighbors, along with the corresponding link costs. This message, which is necessary to prevent some types of cheating behavior (see the proof of Theorem 1), is encrypted and digitally signed by node v, so that the initiator can rely on this information. After topology T_i has been built, the initiator computes the payments due by the nodes in T_i (including itself), according to the current rule:

$$\begin{aligned} Payments[i] &= Payments[i-1] + w_{e_i^2} \\ pay(v) &= \frac{Payments[i]}{i+1} \\ pay(u) &= \frac{Payments[i]}{i+1} - \frac{Payments[i-1]}{i} \\ &\quad \text{for each } u \in T_{i-1}\ , \end{aligned}$$

where $Payments[i-1]$ denotes the total payments collected up to round $i-1$, $Payments[i]$ denotes the total payments collected up to round i, and $w_{e_i^2}$ is the cost of the second best edge connecting a node in T_{i-1} with a node outside T_{i-1}.

The above defined payment rule is inspired by the general framework introduced in Section III. In principle, we want to ensure that all the nodes joining the network equally share the cost of setting up the network topology. This rule should apply independently of the round in which a node joins the network, since otherwise nodes would be motivated to either anticipate or delay the moment at which they join the network. This explains why we keep track of the total payments collected up to round $i-1$: since these have been equally shared amongst the nodes in T_{i-1}, when a new node joins the network the additional payment due by a node in T_{i-1} equals $\frac{Payments[i]}{i+1}$ (the total amount of money that it must pay), decreased by $\frac{Payments[i-1]}{i}$ (the amount of money already paid by the node). On the contrary, the new node joining the network is charged the entire due payment $\frac{Payments[i]}{i+1}$.

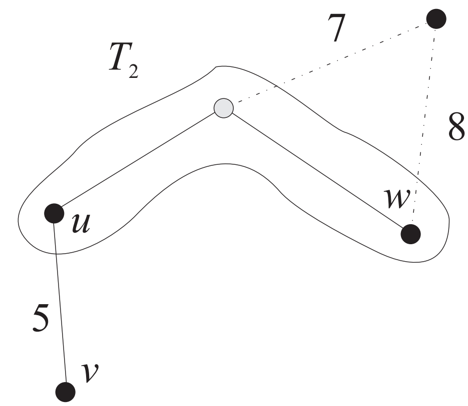

Fig. 3. Example of IC-MST's execution. The initiator node is light gray.

Note that at each round the total cost is increased by the cost of the second best edge joining a node in T_{i-1} with a node outside T_{i-1}. Again, this rule is inspired by the general framework for incentive compatible topology control described in Section III.

Finally, we have to define the premium round, which is again dictated by our general framework. The initiator scan the array $Edges[\ ]$ starting from the first element. For each edge $e_i = (u, v)$, it computes the premiums due to nodes as follows:

$$pr(u) = pr(v) = \frac{Payments[i] - Payments[i-1]}{2}\ .$$

Note that $Payments[i] - Payments[i-1]$ corresponds to the cost of the second best edge joining a node in T_{i-1} with a node outside T_{i-1}. This premium is equally divided between the nodes at the endpoints of the edge. The premiums due to the nodes are summed up as new edges are considered, until all the edges in the network topology have been considered. Then, the initiator delivers to each node (including itself) the corresponding premium in a secure way.

Figure 3 reports an example of IC-MST's execution. At round 2, the network topology is composed of the initiator (light gray node), and of nodes u and w. Let us assume that the total payments paid by the nodes up to round 2 amount to 10, i.e. $Payments[2] = 10$. The next edge to be included in the topology is (u, v), since it is the edge of minimum cost joining a node in T_2 with a node outside T_2. However, the total of the payments is increased by 7, i.e. the cost of the second best edge. Hence, we have $Payments[3] = 17$, and the payments to the nodes are computed as follows:

$$\begin{aligned} pay(v) &= \frac{Payments[3]}{4} = \frac{17}{4} \\ pay(u) &= pay(w) = pay(initiator) = \\ &= \frac{Payments[3]}{4} - \frac{Payments[2]}{3} = \\ &= \frac{17}{4} - \frac{10}{3} = \frac{11}{12} \end{aligned}$$

An important point to discuss is how the edge weights are computed in IC-MST, since implementing this task improperly might impair the incentive compatibility property of our protocol.

The idea is to force each node to either overdeclare, or underdeclare, or correctly declare the cost of all the edges incident into it. In other words, we want to avoid that a node u can, say, correctly declare the cost of a certain edge (u, v), while at the same time overdeclaring the cost of another edge (u, w). To this purpose, edge weights are computed by exchanging *hello* messages between the nodes: when the initiator starts the protocol execution, it sends a hello message at maximum transmit power, indicating that it is starting the construction of the network topology. The message includes the initiator's ID and the transmit power used to send the message. All the nodes within its radio coverage area[5], after waiting for a random time (this is to avoid collisions), reply with another hello message sent at maximum power, reporting the ID and transmit power of the sender. When the initiator receives a hello message from a neighbor node, say v, it compares the transmit power P_v included in the message with the received power P_v^R. The difference between these power levels, namely $P_v - P_v^R$, corresponds to the path loss experienced by the wireless link connecting the initiator with v, and the minimum power needed to sustain a link with the desired features to v can be computed accordingly. If the required power to sustain the link to v exceeds the initiator's maximum transmit power[6], then node v is not included in the initiator's neighbor list, as the link to v is unidirectional and only bi-directional links are considered in our approach.

Note that receiving a hello message for the first time triggers the recipient node to send a hello message in turn (with a random delay). After all the nodes in the network have sent their hello message, all the edge weights in the communication graph can be correctly computed.

At any time during IC-MST execution, a node which does not behave according to the protocol specifications can be identified as a cheater and excluded from the network topology. For instance, a node which tries to send several hello messages to induce false edge weights at its neighbors can easily be identified as a cheater by the nodes in its vicinity, and excluded from the network topology. This repudiation mechanism is fundamental to ensure the incentive compatibility property of IC-MST (see proof of Theorem 1 below).

[5]We recall that we are not assuming that the coverage area is a perfect circle, nor that all the nodes have the same maximum transmitting range.

[6]In general this is possible, since nodes can have different maximum transmit powers.

C. Protocol Analysis

Theorem 1: Assume the maxpower G is strongly connected and has minimum node degree $\delta \geq 2$. Then IC-MST is incentive compatible, i.e. a network node maximizes its own utility when it behaves according to IC-MST's specifications.

Proof: First, we observe that it is in the interest of a selfish node to join the network topology (otherwise it gets utility $-M$). Given this fact, how can a node maximize its utility function?

Observe that the last term in the definition of utility function (4) can be rewritten as follows:

$$pr(u, T) = \sum_{e=(u,v)\in T} pr(e) \ .$$

The pricing rules ensures that, for any given edge e in the final topology T, both endpoints of e receive a premium exceeding the cost of the edge. In other words, for each edge $e = (u, v) \in T$, we have $pr(e) - \frac{w_e}{2} > 0$. It follows that node u will to include as many edges incidents into it as possible in the final network topology.

A node participating in IC-MST can cheat in several different ways. We show that none of this cheating possibilities leads to an increase of the node utility.

At the beginning of the protocol execution, a node (say, u) is requested to send a hello message at maximum power, including in the message the transmit power used. Node u might not send the help message at all, but in this case it would not be included in the network topology, driving its utility down to $-M$. If node u sends the message at a power less than the maximum, it exposes itself to the risk of not reaching nodes that it would have otherwise reached if sending the message at maximum power, possibly failing to connect to the network even if this would be possible. On the other hand, the weight of an edge is computed based on the received power at the receiver end, and on the transmit power as included in the hello message by the sender (not the actual transmit power). Sending the hello message with a decreased power would result in a lower received power at the receiver end, driving up the cost of all the edges incident into u. In turn, this lowers the likelihood of having many of u's edges in the MST, driving down its utility function. If u overdeclares the transmit power in its hello message it increases the weight of all the edges incident into u, with a negative effect on its utility. On the other hand, if u underdeclares the transmit power, all its incoming links would be compromised, as its neighbor nodes would use an incorrect (too low) transmit power to send the messages to u. Hence, u would be disconnected from the network, driving its utility down to $-M$. We also observe that u has no interest in both anticipating nor delaying the transmission of the help message, as the moment at which it joins the network topology has no effect on the payments due to the initiator, nor on the premiums received. Finally, u cannot send the hello message more than once, as, in case it would send several hello messages, its neighbors would immediately identify node u as a cheater, excluding it from the network topology (we are assuming no collusion). Thus, we have proved that a selfish node has no interests in deviating from

IC-MST's specifications during the hello message exchange phase.

Let us now consider the subsequent stages of IC-MST's execution. Let us consider the moment in which a certain node u is first inserted into the network topology. The fact that it is inserted in the network topology depends on the weights of the edges incident into it, which are computed during the hello message exchange phase. Since we have shown above that this phase is performed correctly by a selfish node, these weights are computed correctly. Note that in principle node u might increase its utility by overdeclaring *only* the cost of the second best edge connecting it to a node in the current topology, but this is not possible, because the mechanism that we use to compute the edge weights ensures that either all the weights of the edges incident into a node are overdeclared, or all of them are underdeclared, or all of them are declared correctly. After node u is included in the network topology for the first time, it is requested to send an encrypted message to the initiator, declaring its neighbor list and corresponding edge costs. If u does not send this message, the initiator can identify u as a cheater, excluding it from the network topology. Furthermore, sending bogus information exposes node u to the risk of being identified as a cheater, as the initiator keeps track of the neighbor lists obtained from the other nodes and can easily perform a cross check (here, the assumption of no collusion between nodes is necessary). So, we have proved that it is in node u's best interest to behave according to IC-MST's specifications when it is first included in the network topology.

The final case to consider is when node u is already part of the network topology, and a new node joins the network. In principle, node u might increase its utility by not reporting the cost of the second best edge, say e, incident into it (we recall that reporting a bogus cost would expose u to the risk of being identified as a cheater and excluded from the network topology). However, also in this case the initiator will eventually receive from the node at the other end of edge e the list of neighbors, including node u and the corresponding edge cost. By cross checking this information, the initiator can easily identify u as a cheater and excluding it from the network (this is true under the assumption of no collusion). This proves that a selfish node has no interest in cheating also when it is already part of the computed topology.

To prove the theorem, it is sufficient to observe that in the last round the premiums are delivered in a secure way, as well as the payments due by the nodes in the various rounds of IC-MST's execution.

■

Note that the incentive compatibility property of IC-MST relies on the fact that payments/premiums can be gathered/delivered to the nodes in a secure way. Indeed, how to implement secure crediting in wireless multi-hop networks is a research field in itself (see, for instance, [25]). While a detailed discussion of this issue is beyond the scope of this paper, we sketch here a repudiation-based mechanism that can be used to motivate nodes to propagate the payment info to/from other nodes in the tree. We have two critical points in IC-MST: $i)$ when a new edge $e = (u, v)$ is added at a certain step, nodes u and v must deliver payments to the initiator; and $ii)$ the delivery of premiums during the premium round. Regarding $i)$, we first notice that, since for every edge in the MST incident into it a node always receives a premium that exceeds the payment for the edge, node u and v will to send the payments to the initiator. The payment message is encrypted, so that intermediate nodes in the path P to the initiator can only forward or drop the message, but they cannot modify it. If one of the nodes in P drops the message, the initiator realizes that something is wrong (it is in fact expecting the payments for edge e), and it can exclude all the nodes in P from the network topology. Hence, an intermediate node is motivated to forward the payment to avoid being excluded from the network topology. The argument for $ii)$ is similar: At the end of the premium round, the initiator individually asks to each node (using digitally signed messages) if it has received the premium: if a node u responds 'no' (or does not respond at all), all the nodes in the path P from the initiator to node u are excluded from the network. The use of digital signatures for the reply message prevent nodes in P from forging u's reply.

Theorem 2: The social optimum is achieved when the topology T computed by IC-MST is the MST of the maxpower graph G.

Proof: We recall that the social utility function is defined as follows:

$$Soc(T) = \sum_{u \in N} U(u, T) \ .$$

Assuming that the generated topology T is connected (otherwise at least one of the nodes' utility would be $-M$, driving down the social utility), the social utility function can be rewritten as follows:

$$Soc(T) = \sum_{u \in N} \left(- \sum_{e=(u,v) \in E_T} \frac{w_e}{2} - pay(u, T) + pr(u, T) \right) \ .$$

Observe that, given our definition of the pricing scheme, the sum of the payments over all the network nodes equal the sum of the premiums paid to the nodes, i.e.

$$\sum_{u \in N} (-pay(u, T) + pr(u, T)) = 0 \ .$$

It follows that

$$Soc(T) = \sum_{u \in N} \left(- \sum_{e=(u,v) \in E_T} \frac{w_e}{2} \right) = - \sum_{e \in E_T} w_e \ ,$$

implying that the function $Soc(T)$ is maximized (i.e., the social optimum is achieved) when T is the MST. ■

Theorem 3: The IC-MST protocol has $O(n^2)$ message complexity.

Proof: All the edge weights can be computed after all the network nodes have sent their hello message, i.e. with n messages overall.

During round i, a node in T_{i-1}, say u, is requested to report the weights of its best edges to nodes outside the current network topology, which can be combined into one message which is passed to the parent of u in the current topology T_{i-1} (we

recall that all the intermediate topologies built by IC-MST are trees). The parent of u waits for all its children to report their weights, then it computes the best two weights out the ones received by its children and its own, and forward a unique message up in the tree. This process is repeated until all the information has been conveyed to the initiator node. Hence, at round i at most i messages are exchanged to propagate the information to the initiator, with a total of $O(n^2)$ messages during the $n-1$ rounds needed to build the network topology. It is easy to see that a similar approach can be used to collect the payments from the nodes in T_i, sending $O(n^2)$ messages overall during the $n-1$ rounds.
In the final round, the premiums are delivered to the nodes along the tree in a top-down fashion, sending $O(n)$ messages overall. It follows that the message complexity of IC-MST is $O(n^2)$. ■

V. Incentive-compatible KCN topology control

In this section we consider the problem of computing and maintaining the KCN topology in a scenario in which the network nodes are selfish agents. We recall that the KCN topology of parameter k is a graph in which every node has a direct, bi-directional link to at least its k closest neighbors. More in particular, KCN is obtained from the maxpower graph $G=(N,E)$ as follows: for each edge $e=(u,v)\in E$, include e in KCN if and only if u is one of the k closest neighbors of node v, or v is one of the k closest neighbors of node u.

Although in the final KCN topology all links are bi-directional, in the process of building the KCN graph it is important to distinguish between incoming and outgoing edges of a node. Hence, in the remainder of this section (u,v) denotes a directed edge from node u to node v. Furthermore, we use the following notation: for any given node u, N_k^u denotes the set of the k closest neighbors of node u, and N_T^u denotes the set of u's bi-directional neighbors in the topology T computed by the topology control algorithm. Note that $N_k^u \subseteq N_{KCN}^u$.

The utility function is the same as the one defined in the general framework of Section III, where a selfish node is content with a certain topology $T=(N,E')$ if and only if $N_k^u \subseteq N_T^u$. Note that here we are using the local selfish node model. In order to implement closest neighbor-based topology control, we also assume that the weight w_e of edge e equals its length raised to some positive path loss exponent α. Indeed, the protocol presented here can be used with arbitrary definitions of edge weight, then it can be used to implement a wider class of topology control approaches, where the goal is to have bi-directional connections to at least the k 'best' neighbors, where 'best' means with minimum edge weight.

In order to define the premium function, we introduce the concept of *replacement link* for a certain node u. Let $v_1,\ldots,v_k$ be the k closest nodes to u, and let v_{k+1} be the $k+1$-closest node to u. If any of the nodes v_i, with $i=1,\ldots,k$, would not be part of the network, node u should establish a link with node v_{k+1} in order to be content. For this reason, (u,v_{k+1}) is the replacement link of node u for all edges (u,v_i), with $i=1,\ldots,k$. In the following, the cost of

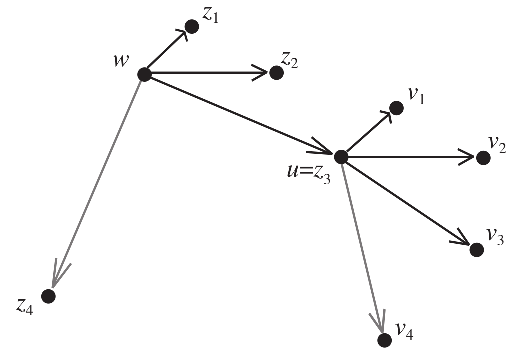

Fig. 4. Example of replacement links (gray edges). Parameter k is set to 3.

the replacement link of node u for edge (u,v) is denoted by $w_u^{-(u,v)}$.

An example clarifying our definitions of replacement link is reported in Figure 4. Edge (u,v_4) is u's replacement link for edges (u,v_1), (u,v_2) and (u,v_3): if any of these edges is deleted from the graph, node u must establish a link to node v_4 in order to be content. Edge (w,z_4) is w's replacement link for edge (w,u): if edge (w,u) is deleted from the graph, node w must establish a link to node z_4 in order to be content.

We recall that from the definition of the general framework we have:

$$pr(u,KCN) = |KCN^{-u}| - |KCN| + \sum_{(u,v):v\in N_{KCN}^u} w_{(u,v)}^u \ ,$$

where KCN^{-u} represent the KCN topology computed without node u, and $w_{(u,v)}^u$ represents the weight of edge (u,v) as declared by node u. Due to the localized nature of the KCN graph, the cost of the KCN^{-u} topology can be rewritten as follows:

$$|KCN^{-u}| = |KCN| - \sum_{(u,v):v\in N_{KCN}^u} w_{(u,v)}^u + \sum_{(v,u):u\in N_k^v} w_v^{-(v,u)} \ ,$$

from which we have:

$$pr(u,KCN) = \sum_{(v,u):u\in N_k^v} w_v^{-(v,u)} \ .$$

In words, node u receives a premium for each edge (v,u) such that u is one of the k closest neighbors of node v, and the amount of the premium for each such edge (v,u) equals the cost of the replacement link of node v for link (v,u).

Similarly to the general framework, premiums are funded by evenly subdividing all payments among the network nodes, i.e.

$$pay(u,KCN) = \frac{\sum_{u\in N} pr(u,KCN)}{n}$$

Note that the assumption of equally subdividing all payments among the nodes implies the existence of a centralized authority that collects payments and distributes premiums to the nodes. This task was performed by the initiator node in the IC-MST protocol. Since in KCN there is no initiator node, we simply assume the existence of such a Centralized Authority, and that nodes can send/receive payments from the CA in a secure way.

We can now present the IC-KCN distributed algorithm for an incentive compatible computation of the KCN graph. The algorithm is executed by each node $u \in N$.

ALGORITHM IC-KCN (for node u)

1) Compute distances to neighbors; exchange digitally signed distance information with neighbors; w.l.o.g. let the link-weight ordered set of $h > k$ neighbors be $v_1, \ldots, v_h$; let $w_i = w_{(u,v_i)}$; note that node u could cheat in this computation by either underdeclaring or overdeclaring the weights to its neighbors.
2) Include links $(u, v_1), \ldots, (u, v_k)$ in u's local view of the T_{KCN} topology;
3) For each neighbor $v_i \in \{v_1, \ldots, v_k\}$: request establishment of the reverse link (v_i, u) in T_{KCN}; include digitally signed (by neighbors) distance information for neighbors $v_1, \ldots, v_{k+1}$ in the message; compute the cost of the replacement link for (v_i, u) (which equals w_{k+1}), and communicate this cost to the CA in a secure way;
4) Upon receiving link establishment requests for link (u, v_i) for $0 < i \leq h$, including v_i's signed neighbor distance information: check digital signatures and issue a premium request for link (u, v_i) to the CA. If digital signature is incorrect, ignore neighbor v_i in the future; otherwise, establish link (u, v_i).
5) Wait for crediting from the CA: send the requested payments, and receive the premiums; after that, the node is ready for operating on the established network topology.

Distance to neighbors (and, consequently, link weights) can be computed using a similar technique to the one used in IC-MST: each node sends a hello message at maximum power, reporting its ID and the power used to send the message. Although a node can falsely report the value of the transmit power used to send the hello message, the advantage of using this technique is that we can ensure that a node either truthfully report, or overdeclare, or underdeclare the weights of all its incident edges.

Similarly to the case of IC-MST, we assume that nodes are selfish, but they cannot collude with each other.

Theorem 4: Assume the maxpower graph G is strongly connected and has minimum node degree equal to $h > k$. Then, the protocol IC-KCN executed with parameter k is incentive compatible, i.e., for any selfish node u it is in its best interest to behave according to the protocol specifications.

Proof: (Sketch) First, by applying similar arguments as in the proof of Theorem 1, we can show that a node cannot increase its utility by falsely reporting its transmit power level in the hello message. Thus, a node can correctly compute the distances to its neighbors, and correctly compute the list of its k closest neighbors.

Node u has no interest in trying to establish outgoing links to neighbors further than the k closest ones, since premiums are received only for incoming links to node u. So, by trying to establish more and/or longer outgoing links node u can only decrease its utility.

Node u cannot increase its utility by falsely reporting the weight of its replacement link, since this weight does not concur to the formation of its premium. Falsely reporting the weight of the replacement link might increase (or decrease) the premium of one of u's neighbors but, since we are assuming no collusion, u has no interest in doing that.

Node u has no interest in reject a link establishment request from an incoming neighbor v, since node u receives a premium for establishing the reverse link (u, v) that exceeds the cost of (u, v). Hence, accepting the link establishment request from node v increases the utility of node u.

Finally, the node is willing to receive payment/premium from the CA, since, even if the balance between payments and premiums negative, the node is content, and its utility is greater than if the node were excluded from the network. ■

Theorem 5: The IC-KCN protocol executed with parameter k has $O(n \cdot k)$ message complexity.

Proof: To prove the theorem it is sufficient to note that:

– distance between nodes is computed by having each node sending a hello message at maximum power, hence exchanging n messages overall (excluding collisions);
– every node in the network sends exactly k reverse link establishment messages;
– every node exchanges $O(k)$ messages with the CA for reporting cost of the replacement link, premium requests, and billing.

■

Note that $k < n-1$, so the message complexity of IC-KCN is $O(n^2)$. Indeed, in many situations k can be assumed to be $O(\log n)$ (see [27]), and the message complexity of IC-KCN becomes $O(n \log n)$.

On a final note, we have tried several modifications to the above defined payment rules to enable a *local* billing mechanism (i.e., nodes exchange payment/premium only with neighbors), thus avoiding the need for a central authority. Unfortunately, accomplishing this task seems to be very challenging: all of our approaches ended up in situations in which a node can cheat and increase its utility. However, if we move to a more restricted setting in which only overdeclaration is possible, but not underdeclaration,[7] a local billing mechanism, where a node simply pays its replacement edge cost to its k nearest neighbors becomes strategy-proof. We leave the investigation of how to implement general incentive-compatible KCN topology control without a central authority as future work.

VI. CONCLUSIONS

In this paper, we have studied the problem of building and maintaining a network topology with certain desired features in a wireless multi-hop network with selfish nodes. We have shown that this problem can be tackled only if selfish node behavior is accounted for at the design stage, indicating the

[7] Such a setting may in fact be closer to reality if one imagines a level-based neighbor discovery protocol in conjunction with authenticated responses to challenges.

need of rethinking the current approaches to the topology control problem. To address this need, we have introduced a general framework for designing incentive compatible topology control protocols, and we have applied our framework to designing incentive compatible realizations of two popular topology control approaches.

Recent technology trends indicate that wireless multi-hop networks formed by nodes belonging to different authorities and governed by conflicting interests will become widespread. This is the case, for instance, of many application scenarios of wireless Mesh networks, and of the general vision of a future in which ubiquitous mobile computing will become reality. In view of these trends, we believe the study reported in this paper constitutes an important building block of next generation wireless multi-hop networks.

The implementations of incentive compatible TC presented in this paper are only examples of application of our framework, that can be applied to other topology control protocols such as, for instance, cone-based topology control (CBTC, [23]). Furthermore, the general framework for incentive compatible topology control presented in this paper leaves space for several generalizations. For instance, we can modify the definition of the utility function to account for different cost metrics (e.g., max instead of total edge cost). Furthermore, we can adopt a more general notion of benefit that a node gets from a certain network topology T: instead of being either 'content' or 'uncontent' with T, a node might display a certain degree d of satisfaction with T, where d is a constant such that $0 \leq d \leq 1$. Whether our proposed framework can be applied with this more general notion of node benefit and with different cost metrics is an open problem, which is matter of ongoing research.

References

[1] L. Anderegg, S. Eidenbenz, "Ad hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad hoc Networks with Selfish Agents", *Proc. ACM Mobicom*, pp. 245–259, 2003.

[2] M. Bahramgiri, M. Hajiaghayi, V.S. Mirrokni, "Fault-tolerant ad 3-Dimensional Distributed Topology Control Algorithms in Wireless Multi-hop Networks", *Proc. IEEE Int. Conference on Computer Communications and Networks*, pp. 392–397, 2002.

[3] N. Ben Salem, L. Buttyan, J.P. Hubaux, M. Jakobsson, "A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop Cellular Networks", *in Proc. ACM MobiHoc 03*, pp. 13–24, 2003.

[4] N. Ben Salem, J. Hubaux, M. Jakobsson, "Reputation-based Wi-Fi Deployment - Protocols and Security Analysis", *Proc. WMASH '04*, pp. 29–40, 2004.

[5] D. M. Blough, M. Leoncini, G. Resta, and P. Santi, "The k-NEIGH Protocol for Symmetric Topology Control in Ad Hoc Networks", *in Proc. ACM MobiHoc 03*, pp. 141–152, June 2003.

[6] D.M. Blough, M. Leoncini, G. Resta, P. Santi, "K-NeighLev: a Practical Realization of Neighborhood-Based Topology Control in Ad Hoc Networks", *Tech. Rep. IIT-TR-09/2003,*, Istituto di Informatica e Telematica, Pisa - Italy, Sept. 2003.

[7] M. Burkhart, P. Von Rickenbach, R. Wattenhofer, A. Zollinger, "Does Topology Control Reduce Interference?", *Proc. ACM MobiHoc*, pp. 9–19, 2004.

[8] L. Buttyan, J. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks", *ACM/Kluwer Mobile Networks and Applications*, Vol. 8, No. 5, October 2003.

[9] M. Cagalj, S. Ganeriwal, I. Aad, J. Hubaux, "On selfish behavior in CSMA/CA networks", *Proc. IEEE Infocom '05*, Miami - FL, USA, March 13-17, 2005.

[10] S. Eidenbenz, V.S. Kumar, S. Zust, "Equilibria in Topology Control Games for Ad Hoc Networks", *Proc. ACM DIALM-POMC 2003*, pp. 2–11, 2003.

[11] S. Eidenbenz, G. Resta, P. Santi,"COMMIT: A Sender-Centric Truthful and Energy-Efficient Routing Protocol for Ad Hoc Networks with Selfish Nodes, *Proc. IEEE Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, 2005.

[12] E. Efstathiou, G. Polyzos, "Fully Self-Organized Fair Peering of Wireless Hotspots", *Wiley European Trans. on Telecommunications*, to appear.

[13] J.R. Green, J.-J. Laffont, "Incentives in Public Decision Making", *North Holland Publishing Company*, 1979.

[14] Z. Huang, C. Shen, C. Srisathapornphat, C. Jaikaeo, "Topology Control for Ad Hoc Networks with Directional Antennas", *Proc. IEEE Int. Conference on Computer Communications and Networks*, pp. 16–21, 2002.

[15] N. Li, J. Hou, L. Sha, "Design and Analysis of an MST-based Topology Control Algorithm", *Proc. IEEE Infocom 03*, pp. 2003.

[16] J. Liu, B. Li, "MobileGrid: Capacity-aware Topology Control in Mobile Ad Hoc Networks", *Proc. IEEE Int. Conference on Computer Communications and Networks*, pp. 570–574, 2002.

[17] A.B. MacKenzie, S. Wicker, "Game Theory and the Design of Self-Configuring, Adaptive Wireless Networks", *IEEE Communications Magazine*, Vol. 39, n. 11, pp. 126–131, 2001.

[18] A.B. MacKenzie, S. Wicker, "Stability of Multipacket Slotted Aloha with Selfish Users and Perfect Information", *Proc. IEEE Infocom 03*, pp. 1583–1590, 2003.

[19] A. Mas-Colell, M. Whinston, J. Green, *Microeconomic Theory*, Oxford University Press, New York, 1995.

[20] K. Moaveni-Nejad, X.Y. Li, "Low-Interference Topology Control for Wireless Ad Hoc Networks", *Ad Hoc and Sensor Networks: an International Journal*, Vol. 1, n. 1–2, pp. 41–64, 2005.

[21] R. Ramanathan, R. Rosales-Hain, "Topology Control of Multihop Wireless Networks using Transmit Power Adjustment", *Proc. IEEE Infocom 2000*, pp. 404–413, 2000.

[22] M. Raya, J. Hubaux, I. Aad, "DOMINO: A System to Detect Greedy Behavior in IEEE 802.11 Hotspots", *Proc. ACM MobiSys 2004*, pp. 84–97, 2004.

[23] R. Wattenhofer, L. Li, P. Bahl, Y. Wang, "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks", *Proc. IEEE Infocom 2001*, pp. 1388-1397, 2001.

[24] W. Wang, X-Y. Li, Y. Wang, "Truthful Multicast in Selfish Wireless Networks", *Proc. ACM MobiCom*, pp. 2004.

[25] S. Zhong, Yang Richard Yang, J. Chen, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-hoc Networks", *Proc. IEEE Infocom 2003*, pp. 1987–1997, 2003.

[26] M. Xiao, N.B. Schroff, E.K.P. Chong, "A Utility-Based Power Control Scheme in Wireless Cellular Systems", *IEEE/ACM Trans. on Networking*, vol. 11, n. 10, pp. 210–221, 2003.

[27] F. Xue, P.R. Kumar, "The Number of Neighbors Needed for Connectivity of Wireless Networks", *Wireless Networks*, Vol 10, n. 2, pp. 169–181, 2004.