The COMMIT Protocol for Truthful and Cost-Efficient Routing in Ad Hoc Networks with Selfish Nodes*

Stephan Eidenbenz[†], Giovanni Resta[‡], Paolo Santi[‡]

Abstract

We consider the problem of establishing a route and sending packets between a source/destination pair in ad hoc networks composed of rational selfish nodes, whose purpose is to maximize their own utility. In order to motivate nodes to follow the protocol specification, we use side payments that are made to the forwarding nodes. Our goal is to design a fully distributed algorithm such that: (*i*) a node is always better off participating in the protocol execution (individual rationality), (*ii*) a node is always better off behaving according to the protocol specification (truthfulness), (*iii*) messages are routed along the most energy-efficient (least cost) path, and (*iv*) the message complexity is reasonably low. We introduce the COMMIT protocol for individually rational, truthful, and energy-efficient routing in ad hoc networks. To the best of our knowledge, this is the first ad hoc routing protocol with these features. COMMIT is based on the VCG payment scheme, in conjunction with a novel game-theoretic technique to achieve truthfulness for the sender node. By means of simulation, we show that the inevitable economic inefficiency is small. As an aside, our work demonstrates the advantage of using a cross-layer approach to solving problems: leveraging the existence of an underlying topology control protocol, we are able to simplify the design and analysis of our routing protocol, and to reduce its message complexity. On the other hand, our investigation of the routing problem in presence of selfish nodes disclosed a new metric under which topology control protocols can be evaluated: the *cost of cooperation*.

*A preliminary version of this paper appeared in Proceedings IEEE Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN), 2005.

[†]Los Alamos National Labs, Los Alamos, NM, USA; LA-UR-07-2474

[‡]IIT-CNR, Pisa, Italy.

Index terms: wireless ad hoc networks, cooperation in ad hoc networks, cooperative routing, energy-efficiency, topology control.

1 Introduction

Ad hoc networks are expected to revolutionize wireless communications in the next few years: by complementing more traditional networking paradigms (Internet, cellular networks, satellite communications), they can be considered as the technological counterpart of the concept of "ubiquitous computing". However, in order for this scenario to become reality, several issues raised by ad hoc networking must be adequately addressed. One of these issues, which may be one of the reasons for the lack of commercial applications based on ad hoc networks so far, is how to stimulate cooperation among the network nodes. In fact, the nodes of an ad hoc network are in general owned by different authorities (private users, professionals, companies, and so on), and a voluntary and "unselfish" participation of the nodes in the execution of a certain network-wide task cannot be taken for granted.

One of the fundamental tasks any ad hoc network must perform is routing: since the network is in general multi-hop, a routing protocol is needed in order to discover and maintain routes between far away nodes, allowing them to communicate along multi-hop paths. Unless carefully designed, routing protocols are doomed to perform poorly in presence of "selfish" node behavior: in general, a network node has no interest in forwarding a packet on behalf of another node, since this action would only have the effect of consuming its resources (energy, and available bandwidth). Thus, if many of the nodes act selfishly (as it might be the case when nodes are owned by different authorities), few multi-hop communications can take place, and the network functionality is compromised.

In order to circumvent this problem, several authors have recently proposed to stimulate cooperation using incentives. These incentives can take the form either of reputation systems (basically, "badly behaving" nodes are detected and isolated from the rest of the network) [5, 6], or of (sometimes virtual) monetary transfer (basically, the sender of a message pays a certain amount of money to the relay nodes to motivate them to forwarding its message) [2, 3, 7, 8, 9, 10, 29, 28].

Most of the approaches proposed in the literature, such as those presented in [10, 28], are focused on the packet forwarding phase of a routing protocol: the route to the destination is already known, and the goal is to identify strategies that motivate nodes to forward packets along this route. Relatively little attention has been devoted to the problem of stimulating cooperation in the *route discovery phase* of a routing protocol. Clearly, this is a prerequisite for the actual implementation of any of the packet-forwarding schemes introduced in the literature.

To the best of our knowledge, Anderegg and Eidenbenz were the addressing this problem [2], where the authors present the Ad Hoc-VCG routing protocol. This protocol is based on monetary transfer, and has several nice features: it discovers the most energy-efficient path between the source and the destination, and it is *truthful*, i.e., it stimulates the nodes to behave according to the protocol specification[1]. However, Ad Hoc-VCG suffers from three major problems: (*i*) it assumes that the source cannot act strategically (i.e., the source node follows the protocol specification by assumption), and (*ii*) the number of messages that must be exchanged in order to find the route to the destination is quite high – in the order of $O(n^3)$, where n is the number of network nodes. Differently from Ad Hoc-VCG, which focused more on the process of building the routes, the more recent CORSAC protocol proposed by Zhong et al. in [29] considers both route discovery and packet forwarding on the computed routes. The authors introduce a novel solution concept called cooperation-optimal protocol, and prove that it is optimal (i.e., utility maximizing) for a selfish user to fulfill the routing decision in the packet forwarding phase.[2] However, CORSAC suffers the same above limitations (*i*) and (*ii*). As discussed in Section 2, these turn out to be major drawbacks of existing proposals of truthful routing protocols for ad hoc networks, that could prevent their utilization in many application scenarios.

In this paper, we present COMMIT, a protocol for route discovery and packet forwarding in ad hoc networks that enjoys the same nice features as Ad Hoc-VCG (energy-efficiency and truthfulness). Contrary to [2, 29], in our model we allow the sender to act strategically, and we prove that the protocol remains truthful also in this scenario. Further, COMMIT satisfies individual rationality.

A major difference between existing approaches [2, 29] and COMMIT is the network model, in which it is assumed that the costs used to compute routes are associated to *nodes*, and not

[1]This is a very informal definition of truthfulness. A more formal definition of this notion will be given in Section 3.2.

[2]As an aside, we note that [29] claims that Ad Hoc-VCG is not truthful in CORSAC's network scenario.

Protocol	Truth. Routes	Truth. Forward	Strategic sender	Message Compl.
Ad Hoc-VCG	Yes	Partially	No	$O(n^3)$
CORSAC	Yes	Yes	No	$O(n^3)$
COMMIT	Yes	Partially	Yes	$O(\|M\|^2 d)$

Table 1: Main features of incentive compatible routing protocols.

to *links* as in [2, 29]. This assumption is coherent with a scenario in which routing is executed on top of a *periodic* topology control protocol. In periodic topology control, every node v in the network is assigned with a transmit power level $l(v)$, which will be used to send and forward packets (independently of the actual receiver) until the next topology check[3]. As we shall see, using node instead of link costs simplifies the game theoretical analysis of the protocol, and it reduces the message complexity to $O(|M|^2 d)$, where $|M| \leq n-2$ and d is the maximum node degree in the communication graph. Considering that most topology control algorithms build communication graphs with small degree ($d = O(\log n)$, or even $d = O(1)$ in some cases [4, 26]), this is a significant improvement over the $O(n^3)$ message complexity of Ad Hoc-VCG and CORSAC.

Relying on an underlying topology control protocol can be seen as decomposing the routing task into two sub-tasks: (i) building a desirable network topology and (ii) performing route discovery and packet forwarding on the resulting topology. Thus, the findings of our paper (namely, that the design of a truthful routing protocol is simplified if an underlying topology control protocol is assumed) are in line with recent research indicating that decomposing complex tasks into simpler sub-tasks eases the design and analysis of truthful distributed protocols [24, 29].

The features of Ad Hoc-VCG, CORSAC, and COMMIT are summarized in Table 1.

Before presenting COMMIT, in the next section we describe an application scenario in which the utilization of existing truthful routing protocols seems unrealistic. This scenario motivated our research.

[3] Note that the other typical approach to topology control, called *per-packet* (where nodes select the transmit power on a per-packet basis, depending on the intended receiver), could be used in combination with link-based incentive compatible routing protocols such as Ad Hoc-VCG and CORSAC.

2 Application scenario and motivation

We consider a wireless network used to access a certain service (e.g., Internet access). In principle, ad hoc networking could be used to increase the service coverage: instead of requiring each customer to be directly connected to the base station (which is inside the coffee shop), customers could be allowed to reach the base station along multi-hop paths, using the wireless devices (laptop, PDA, and so on) of other customers as intermediate nodes. This way, the area in which the service is available could be much larger than the radio coverage area of the base station.

We remark that the mechanisms described in this paper can be used to establish any type of connection between a service provider and a customer along wireless, multi-hop paths, where the relay nodes are in general other customers. In the following, we will conventionally call the customer who wants to establish a connection to the service the 'sender', the intermediate wireless nodes the 'relays', and the service provider the 'destination' of the communication, regardless of the actual data flow between the sender and the destination. For instance, in case the provided service is Internet access, most of the traffic is likely to be downlink (i.e., from the destination to the sender, according to our terminology). Nevertheless, the data session is initiated by the customer with a route discovery (or service discovery) phase, and the customer will pay for both the ingoing and outgoing traffic. For this reason, we have adopted the terminology introduced above.

In order to successfully implement such wireless multi-hop access service, intermediate nodes should be motivated to act "unselfishly", relaying packets on behalf of other nodes. Typically, intermediate nodes receive a compensation in the form of a payment of money for their "unselfish" behavior, which covers the cost that a node incurs by forwarding.

Since in this scenario the newcomer does not know the route to the access point, incentives must be given also to perform route discovery. So, routing according to the Ad Hoc-VCG protocol seems a reasonable choice[4]. Ad Hoc-VCG is based on the following idea [2]: The sender starts a route discovery process, declaring the destination of its packets. As a result of

[4]As outlined in the Introduction, Ad Hoc-VCG and CORSAC share many relevant features. For this reason, in the following and in the remainder of the paper we focus the attention on one of the protocols, namely Ad Hoc-VCG. Unless otherwise stated, all the considerations about Ad Hoc-VCG made in this paper apply to CORSAC as well.

the route discovery phase, the sender receives a message indicating the path P to the destination (if any), and the cost of sending (or receiving) the packet along P. The amount that the sender pays is divided among the nodes on P, in such a way that every node receives an amount of money that is at least equal to (actually, it is usually greater than) its real cost for forwarding the packet. In other words, the sender pays an amount of money which must at least cover the cost of sending a packet along P. In one of the two payment models presented in [2], the sender also pays the premiums (i.e., the amount of money exceeding the actual cost of sending a packet) to the intermediate nodes. In the other model, the premiums are paid by a central authority, which accumulates all the benefits in the network and divides them equally amongst all the nodes.

Unfortunately, Ad hoc-VCG is of little help in the application scenario described above. In fact, in Ad hoc-VCG it is assumed that both the sender and the destination of the communication act truthfully. In other words, *Ad hoc-VCG works only if both endpoints of the communication behave well by hypothesis*. This assumption, and in particular the assumption on the sender's behavior, is quite unrealistic in the application scenario considered. In fact, in this scenario many nodes act as sender and relay node at the same time, and the assumption above implies that a node would behave strategically when forwarding packets on behalf of someone else, but it would become a "good guy" (no strategic behavior) when it sends its own packets.

Another unrealistic aspect of Ad hoc-VCG is the fact that *it is assumed that, after the route discovery phase, the sender actually sends out/receive data packets and pays the amount of money due for sending/receiving the packets*. In other words, once the sender has started the route discovery phase, it cannot withdraw the connection request. This mechanism is fundamental for the correct execution of the routing protocol: if intermediate nodes in the winning path P would not be sure that the payment will actually take place, they would loose their incentive to participate in the route discovery phase. In Ad hoc-VCG, when the sender issues the route discovery message, it has no idea of the amount of money that it will pay. In fact, the sender does not know the actual cost of communicating to the destination. Furthermore, in one of the payment models proposed in [2] the sender also has to pay premiums exceeding the costs to the intermediate nodes, and these premiums could be quite high. Considering our application scenario, the above assumption would imply that a customer, after issuing the request for the service (e.g., Internet access), would be forced to pay an amount of money that she does not know in advance. Clearly, nobody would use such a service.

In this paper, we propose a sender-centric approach to the design of incentive compatible routing protocols for ad hoc networks, which results in a protocol called COMMIT. The basic idea is inspired by the business model of the *priceline.com* website [21]. On this website, customers declare the maximum amount of money they are willing to pay for a certain service (e.g., a hotel of a certain category in a certain city). When a customer presents the request, she is required to provide to the system all details for payment (e.g. credit card data) before her request is processed. If the system finds a "provider" matching the request (e.g., a hotel with the correct features and a price not exceeding the offered one), then the request is automatically accepted, and the transaction takes place.

We believe a similar approach is suitable to the application scenario described in this section: when a new customer wants to access the service, she issues a "connection request", stating the maximum amount of money she is willing to pay for it. The connection request represents a full commitment[5] of the new customer: if the connection can actually take place at a cost less than the declared price, the newcomer must pay the corresponding amount of money. This way, *the customer has always full control of the maximum amount of money she will spend for sending/receiving the packets.*

In the following we design the COMMIT routing protocol based on this idea, and we show that *it is resilient to strategic sender behavior.* Thus, COMMIT overcomes one of the main limitations of Ad hoc-VCG (assuming that the sender always behaves well). On the other hand, we retain the assumption that the destination acts truthfully. However, as discussed in the following, this assumption is less critical, as the service provider's interest is that the mechanism used to access the service works properly. The application scenario given can of course be viewed as a mesh network scenario, where nodes communicate with a base station in an ad-hoc fashion. See [18, 19, 27] for key papers in this crucial area. Truthful destination behavior is a reasonable assumption in mesh network scenarios; however, we believe truthful destination behavior can be reasonably assumed in all scenarios where the destination has an interest in receiving packets, which is typically the case even for general-purpose ad hoc networks. We thus do not want to limit COMMIT to only mesh network-type scenarios.

[5]This is why we called our protocol COMMIT.

Further, we prove that COMMIT always chooses the most energy-efficient path between the source and the destination, that is truthful, and that it satisfies individual rationality. Energy efficiency is the key design criterion for any routing protocol as transceiver devices always have very limited battery power. Indeed, COMMIT selects the *least-cost* path between sender and destination, for any specific cost metric that is chosen. To simplify discussion, in this paper we chose energy as the reference metric, but COMMIT can be used as it is also in combination with other metrics. With truthful, we mean that the best selfish strategy for every node (excluding the destination) is to follow the protocol specification. With individual rationality, we mean that it is rational for the selfish node to participate in the protocol execution. Note that, given the discussion above, *executing Ad hoc-VCG is not individually rational for the sender*. Finally, COMMIT relies on a network model that is much more realistic than the model defined for Ad hoc-VCG, in particular with respect to how it addresses topology control. However, straightforward modifications allow COMMIT to work also in the network model proposed for Ad hoc-VCG.

3 The system model

3.1 Network model

We consider an ad hoc network composed of n nodes. The wireless links between nodes are represented in the *communication graph* G. In this paper, we consider only *symmetric* wireless links; i.e., an edge between nodes v and w appears in G if and only if v is within w's transmitting range, and w is within v's transmitting range. Further, we assume that the (symmetric) communication graph G that describes the network topology is 2-connected (with respect to the destination): i.e., there exist at least two node-disjoint paths from any node to the destination node in G.[6]

To establish the communication graph, the nodes execute a topology control protocol. At the end of the protocol execution, every node v determines its transmitting range r_v, which

[6] 2-connectedness is a not a strict requirement in the sense that an occasional occurrence of a non-2-connected communication graph will cause the protocol to fail, but communication simply cannot take place without 2-connectedness since a node that happens to lie on all paths between a sender and the destination could demand an unlimited amount of money for its forwarding service. We will see later that we need an even stronger assumption, which we show to hold in a vast majority of simulation cases. Thus, 2-connectedness is not a strong assumption.

will be used to send packets to neighbor nodes. The power required to achieve a transmitting range r_v is generally believed to be proportional to r_v^α, where α is a constant between one and six. We remark that v will transmit with range r_v independently of the actual 1-hop neighbor to which the packet is directed. These transmitting ranges imply a directed connection graph (possibly) with non-symmetric links. Since we only consider symmetric links, data will never be transmitted along links that only work in a single direction. Using symmetric links only is a standard assumption in the topology control community [4, 26], since it offers a variety of conveniences such as the fact that sending ACKs is always possible.

The topology control protocol is executed periodically: r_v is periodically updated but, in the period of time between consecutive topology checks, the same transmitting range r_v is used for any transmission. For the sake of clarity of illustration, we assume that no link failures (due to node mobility, where a node moves out of range) occur during the route discovery phase and the subsequent data session before the topology control protocol executes its next round. This assumption is reasonable for real-life mobility, and by reducing the period length between topology control updates, we can make link failures a very rare event. Alternatively, we could introduce a standard broken-link mechanism that interrupts a data session and enforces an early execution of the next topology control round. Thus, after each round of the topology control protocol, routes of data sessions have to be recomputed from scratch. Our model of periodic topology updates is realistic for real-life hardware (such as the CISCO Aironet wireless cards [11]) and significantly reduces the message complexity when compared with Ad hoc-VCG [2] and CORSAC [29].

Any topology control strategy can be used in combination with our routing protocol. In this paper, we present experimental results we have obtained by simulating the following strategies:

a) KNeigh: the node's transmitting range is computed using the KNeigh protocol [4]: every node considers the k closest neighbors, and sets the transmitting range to the value needed to reach the farthest *symmetric* neighbor amongst the k closest nodes.

b) CBTC: the node's transmitting range is computed using the CBTC protocol [26]: every node sets its transmit power to the minimum value such that at least one neighbor is present in any cone of degree ρ centered at the node. The communication graph is then restricted to the symmetric links.

c) CTR: all the nodes have the same transmitting range, which is set to the critical value for connectivity, i.e., to the minimum value r such that the communication graph generated when every node transmits with range r is connected with high probability [16, 22]. This scenario is a degenerated topology control mechanism, in which all the nodes have the same range, but the value of the common range is carefully chosen. Since all the nodes in the network have the same energy cost, the minimum-energy path coincides with the path of minimum hop-count.

In order to simplify the presentation, in the following we assume that nodes can transmit using different power levels (e.g., 1mW, 5mW, 20mW, 30mW, 50mW and 100mW as in the CISCO Aironet 350 wireless card [11]). At the end of the topology control phase, every node chooses one of the power levels as its transmit power, which is retained until the next topology check. Choosing the power levels from a discrete set of values is not a requirement for COMMIT, but it is much more realistic to do so.

An important issue concerning the use of topology control in combination with COMMIT is cooperation between selfish nodes. In other words, the designer should avoid to add opportunities for the nodes to manipulate the topology control protocol in order to increase their utility in the routing task. In the following, we simply assume that nodes behave truthfully during the execution of the topology control protocol.

3.2 Modeling routing as a game

In this paper, we model the process of establishing a route between a source and a destination node as a game. The players of the game are the network nodes. With respect to a given data session, any node can play only one of the following roles: *source*, *relay (or intermediate) node*, or *destination*. We denote by S the sender, with v (or sometimes v_i) an arbitrary relay node, and with D the destination.

Although in principle our approach can be used for establishing a generic connection between arbitrary source/destination pairs, in the remainder of this paper we specialize our protocol to deal with the case in which the destination node is fixed, and provides some service (e.g., Internet access) to the other network nodes. In this scenario, it is reasonable to assume that the service provider is a trustworthy third party. Thus, the destination node in our model is not actually part of the game, but it is rather a "neutral referee", whose goal is to correctly compute the minimum energy (S, D) path, and the payment/premiums for S and the intermediate nodes.

The assumption that the service provider is trustworthy is quite common in the literature on incentive compatibility in ad hoc networks [2, 29], and it is also commonly used in the literature on game theory. For instance, when analyzing an auction protocol, it is usually assumed that the auctioneer acts honestly when determining the winners of the auction, and the amount of money they must pay [20]. Further motivation for our assumption of trustworthy destination can be found in Section 5.2.

We recall that in our model the goal is to establish a path between the sender and the destination, along which traffic packets *in both directions* will be routed (this is always possible since we are assuming that wireless links are bi-directional). The sender will pay for both the packets sent and received during the data session.

The sender S has a private information (its *type*), i.e., its willingness to pay for establishing a connection to the destination. In other words, we assume that the sender can quantify its desire to communicate with D in monetary terms. Assuming that m is the maximum per-packet price that S is willing to pay for the connection, we can model the *utility* of player S if the communication takes place as $u_S = m - c_S(D)$, where $c_S(D)$ represents the actual per-packet amount of money that S will pay. In case the connection cannot be established, we have $u_S = 0$.[7]

Let us now consider an arbitrary relay node v. In this case, the private type of the node is its power level $l(v)$ which, as described in the previous section, is assumed to be constant during the route discovery and data session phase, but is not known to the other nodes. In general, the cost c_v incurred by node v to relay a packet sent by S is determined by $l(v)$ and by other factors (e.g., the remaining energy in the battery, the bandwidth currently used by the node for its own connections, or any other type of consideration influencing the v's willingness to relay S's packet). For the sake of simplicity, in this paper we assume that $c_v = l(v)$. However, our approach remains valid if c_v is an arbitrary function of $l(v)$ and, more in general, and arbitrary cost function. In reality, the cost of transmitting a packet may be hard to predict

[7] In general, the utility of S if there is no connection is $0 - \bar{c}_S(D)$, where $\bar{c}_S(D)$ is the price paid by S when the connection is not possible. As we shall see, our protocol sets $\bar{c}_S(D) = 0$, so the overall utility of S in case of no connection is 0.

because of the notoriously poor reliability of wireless links. Retransmissions may be necessary, which significantly increases the cost incurred by the node. We make the assumption that a node has at least a good estimate of its expected transmission costs; various approaches exist to determine these costs including keeping statistics on retransmissions and taking long-term averages. Also in reality, a node expenses power not only for transmitting but also for listening and receiving, which we ignore in our model.[8]

The utility of node v if it takes part in the data session is $u_v = pay(v) - l(v)$, where $pay(v)$ is the per-packet payment that v receives for relaying S's packets. In case v does not take part in the data session, it gets 0 utility. In accordance with standard game-theoretic settings (see [20]), we assume that nodes act selfishly and are rational. In other words, we assume that each player in the game plays the strategy that maximizes her utility. Formally, we consider the following strategy base space: (i) a node can declare any value for its type, (ii) a node can drop control messages that it should forward, (iii) a node can modify messages before forwarding, and (iv) a node can create bogus messages containing wrong information. A strategy is a combination of strategies from the base space. Of course, one of the possible strategies for the nodes is to follow the protocol specification, i.e., declaring the true type and sending/relaying messages as prescribed. Using the game theory terminology, we call this strategy *truth-telling*[9].

The goal of a protocol designer is to devise a mechanism such that a globally desirable goal (called the social choice function in game theory) is achieved or optimized. In our case, the goal is to route messages along the most energy-efficient paths (as defined by the topology control protocol). All known mechanisms that achieve such goals define payments to players in such a way that truth-telling becomes a dominant strategy (i.e., a strategy that maximizes the utility for the player no matter what other players do) for every player. A protocol with this feature is called *truthful*, or *incentive compatible*, or *strategy proof*. Truthfulness is a very strong property, since it ensures that, even if a player has complete knowledge of the other players' types, and regardless of the strategy the other nodes play, truth-telling is always the dominant

[8]A thorough treatment of the effect of receive power consumption is a challenge for future research as it should involve various cross-layer effects, such as 802.11 RTS/CTS.

[9]Indeed, in standard (non distributed) game theory, the strategy of a player is simply her declared type. For this reason, the strategy in which the player behaves honestly is called truth-telling. In the distributed context, the player must also participate in the protocol by exchanging messages. By analogy, we call the honest node behavior truth-telling also in this case.

strategy. Thus, truthfulness is a much stronger property than, for instance, the existence of a Nash equilibrium (see [20] for an excellent introduction to game theory and mechanism design). Further discussion on this point is postponed to Section 6.

To complete the description of our game-theoretic model, we remark that we do not consider cross-layer effects. It is obvious that the holy grail of the selfish networking field is an incentive-compatible protocol stack. Combining protocols on different layers that are each individually incentive-compatible does not necessarily result in an incentive-compatible protocol stack. Similar arguments have been made for protocol efficiency: efficient protocols can be combined into a highly inefficient protocol stack. In analogy to the efficiency world, we believe incentive-compatible protocols on individual layers are a pre-requisite to any solution for a full, incentive-compatible protocol stack. We thus focus on a single layer for now. Note that at least two functions on the same layer (forwarding and routing) have been successfully combined in [29], which is a first step towards such a protocol stack. The first concrete implication is that we do not assume that a node will try to leverage the topology or MAC layer protocol to its advantage on the network layer. A second implication is that similarly a node will not optimize over sessions: for example, it will never refuse to participate in a session because it believes that a much more profitable session will start in the near future. In our model, a node only optimizes essentially on a per packet basis. The pragmatic exception to this rule is that we do assume that nodes are willing to forward control packets because of the potentially large pay-off. This standard assumption is discussed in more detail in Section 5.2.

Finally, we outline that in this paper we are not concerned with malicious node behavior, nor with coalition formation. In case of malicious nodes, players are allowed to choose irrational strategies (e.g., strategies leading to negative utility), as long as this is detrimental for the system. In case of coalitional games, players are allowed to coordinate their cheating behavior in order to fool the system. If this coordinated behavior increases the overall utility of the coalition, the surplus can be shared among its participants, which will then have an incentive to deviate from truth-telling. The current version of COMMIT is not resilient to malicious node behavior nor to coalitions. How to extend/modify our protocol in order to take malicious nodes and collusion into account is matter of ongoing research.

4 The COMMIT protocol

In this section, we describe the COMMIT protocol for incentive compatible and energy-efficient routing in ad hoc networks. We first describe the design guidelines of the protocol, and then present a detailed specification.

4.1 Design guidelines

The design goals of our protocol are: a) individual rationality; b) truthfulness; c) energy efficiency; d) limited message overhead.

A mechanism satisfies the individual rationality property if a node that executes the protocol never gets a negative utility. This property ensures that nodes are motivated to take part in the protocol, since this will never expose them to the risk of decreasing their utility (we recall that a node that does not participate in the protocol execution has 0 utility). *This fundamental property is not satisfied by Ad Hoc-VCG [2] and CORSAC [29], which are the only truthful routing mechanisms for ad hoc networks introduced so far.* The motivations for goal b) are clearly described in the previous sections. With energy efficiency, we mean that the path along which the communication between S and D (if feasible) will take place must be the path of minimum energy (least) cost. The energy cost of a path P is defined as $\sum_{v \in P, v \notin \{S,D\}} l(v)$. Energy efficiency is a key property in ad hoc networks. In fact, truthfulness would not be of much importance if we did not achieve the goal of energy efficiency. Finally, the protocol should minimize the overall number of messages exchanged in the session setup phase.

In order to ensure properties $a) - c)$, our mechanism will use side payments to some of the relay nodes (those in the winning (S, D) path). The mechanism we design must perform the following tasks:

- *winner determination*: determine the winning path (if any) along which the communication will take place.
- *payment computation*: in case the winning path exists, determine the price that S must pay for transmitting/receiving the packets, and the payments for the nodes in the winning path;

– *billing*: if the communication takes place, charge S and pay the nodes in the winning path according to the prices previously determined.

In our protocol, winner determination and payment computation are performed by the destination node D, based on the information provided by the network nodes; billing is done when the actual data session begins. Similarly to [2], in this paper we focus on the problem of winner determination and payment computation, leaving the details on how the payments are actually delivered to the nodes unspecified. Indeed, the problem of implementing electronic payments in ad hoc networks is a research thread in itself, which is addressed, for instance, in [9, 28]. In principle, any of the electronic payments methods presented in the literature can be used in combination with our routing protocol.

4.2 The pricing scheme

Before presenting the protocol specification, we describe the pricing scheme used by COMMIT, since the choice of the pricing scheme determines the minimum amount of information which must be communicated to the destination node (which is in charge of computing the payments).

In [14], it is shown that any pricing scheme that achieves individual rationality, truthfulness, energy efficiency, and pays only the nodes in the winning path must be based on the VCG mechanism[10]. When adapted to our setting, the VCG mechanism [20], which optimizes the socially desirable goal of energy efficiency, defines the following rules to determine the winning path and the relative payments. Let $c(P)$ denote the energy cost of an arbitrary (S, D)-path P (i.e., a path from S to D),where $c(P) = \sum_{v \in P, v \notin \{S,D\}} l(v)$. The winning path is the path of minimum energy cost, denoted by MP. For any node v in the winning path, let us denote with $c(P^{-v})$ the cost of the minimum energy (S, D)-path P^{-v} that does not include v. Thus, P^{-v} would have been the minimum cost path, if node v did not exist. Since we are assuming that the communication graph is 2-connected, this alternative path, which we call the *replacement path*, always exists. The payment for a node v in the winning path MP is defined as follows:

$$pay(v) = c(P^{-v}) - c(MP) + l(v) \ .$$

The payments for the nodes which are not on the winning path are set to 0.

[10]Although this result is proved with reference to a routing problem on the Internet, it can be easily adapted to our scenario.

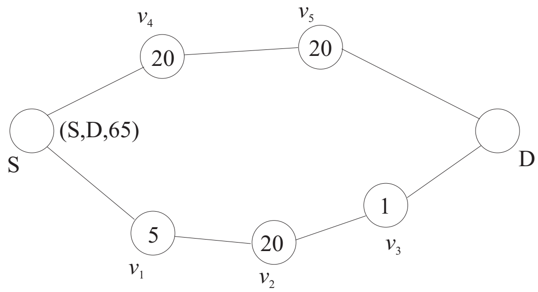

Figure 1: Example of cheating node behavior if $c_S(D)$ would be defined as $c_S(D) = \sum_{v \in MP, v \neq \{S,D\}} pay(v)$. The nodes are labeled with their true types.

A key novel feature of COMMIT (novel even in the broader context of distributed mechanism design and game theory) lies in the definition of the price $c_S(D)$ and in the subsequent definition of who makes which payments: The final step is to decide the price $c_S(D)$ that S must pay for sending the packets along MP. This price defines the *decision rule*, which determines whether the communication takes place or not. A trivial choice would be to set $c_S(D) = \sum_{v \in MP, v \notin \{S,D\}} pay(v)$. However, due to the presence of the reserve price m, this choice would leave space for a strategic behavior of the nodes in MP, that could declare a false type in order to drive $c_S(D)$ below m.[11]

This subtle example of strategic node behavior is depicted in Figure 1. The sender wants to establish a connection with the destination paying at most 65 for each packet. If all the nodes behaved truthfully, the communication would not take place. In fact, we have $MP = \{v_1, v_2, v_3\}$, $c(MP) = 26$, and $c(P^{-v_1}) = c(P^{-v_2}) = c(P^{-v_3}) = 40$, which implies the following payments for the nodes in MP:

$$pay(v_1) = 40 - 26 + 5 = 19,\ pay(v_2) = 40 - 26 + 20 = 34,$$

$$pay(v_3) = 40 - 26 + 1 = 15\ .$$

It follows that the total payment is $68 > 65$, and the communication does not take place,

[11] The reader could question whether an *explicit* reserve price (an implicit reserve mechanism is needed to ensure individual rationality of the sender) is needed at all. An implicit reserve mechanism could be implemented, for instance, by having the sender aborting the connection if the requested price is too high. However, this solution would require exchanging several (useless) control messages, resulting in a waste of resources. Our solution of having an explicit reserve price ensures that a minimal amount of control messages is exchanged to establish the connection (see also Section 4.3).

yielding a 0 utility for all the players. Let us now assume that node v_2 falsely declares power level 30. The winning path MP would remain the same, as well as the replacement path for all the nodes in MP. However, the payments would change as follows:

$$pay(v_1) = 40 - 36 + 5 = 9,\ pay(v_2) = 40 - 36 + 30 = 34$$

$$pay(v_3) = 40 - 36 + 1 = 5\ .$$

Thus, the total payment is now $48 < 65$, and the communication would take place, yielding an utility of $34 - 20 = 14$ for node v_2. Since v_2 would increase its utility by reporting a false type, it follows that defining $c_S(D)$ as $\sum_{v \in MP, v \notin \{S,D\}} pay(v)$ would result in a non-truthful mechanism.

In order to circumvent this problem, we set $c_S(D) = c(P^{-MP})$, where $c(P^{-MP})$ denotes the cost of the minimum energy path that does not contain *any* of the nodes in MP. We call this path the *global replacement path*. It is immediate to see that with this definition of $c_S(D)$ any false declaration of the nodes in MP would have no effect on $c_S(D)$. Thus, the truthfulness of the mechanism is not impaired.

Observe that the assumption of 2-connected communication graph does not imply that a global replacement path always exists. Indeed, this is a stronger property, since we require that one of the at least two node-disjoint paths that exist between S and D (because of 2-connectivity) is the minimum-energy path MP. We call this property *minimum-energy 2-connectivity*. To make the distinction between 2-connectivity and minimum-energy 2 connectivity clearer, consider the graph in Figure 1, and suppose there exists an extra edge between units v_3 and v_4. From the point of view of nodes S and D, the graph is 2-connected; however, if it happens that $MP = \{S, v_4, v_3, D\}$ is the minimum-energy path, then the graph is not minimum-energy 2-connected, since removing v_3 and v_4 from the graph would make it disconnected.

We have conducted simulation experiments to determine whether the communication graphs produced by the topology control protocols listed in Section 3.2 satisfy minimum-energy 2-connectivity on average. To this end, we distributed uniformly at random n nodes in a square region with a side length of one kilometer; for each value of n, we generated 5000 random placements. Given a random node placement, we generated the communication graph according to one of the topology control strategies described above (KNeigh, CBTC and CTR). In case of KNeigh, we set parameter k (number of 1-hop neighbors) to 10, since this value is the minimum

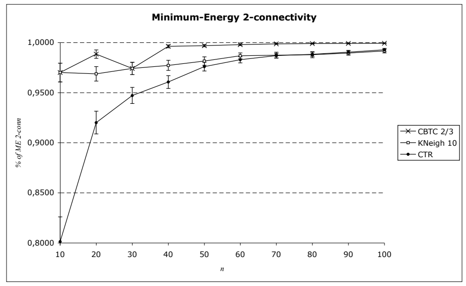

Figure 2: Percentage of (S, D) pairs for which there exists a global replacement path. The parameter k in KNeigh is set to 10, and the parameter ρ in CBTC is set to $2/3\pi$. The graphic reports also the 95% confidence interval.

one providing network connectivity with high probability [4]. Parameter ρ in CBTC is set to $2/3\pi$, which guarantees network connectivity [26]. Once the topology was formed, we selected up to 100 source/destination pairs at random and computed the minimum path and the global replacement path, if possible. The experimental results summarized in Figure 2 clearly show that global replacement paths exist with high probability: for CBTC, this probability always exceeded 97 percent, for other topology control protocols it was always above 80 percent and quickly increases to more than 98 percent as we increase the number of nodes.

In the remainder of this paper, we thus assume that the communication graph produced by the topology control protocol is minimum-energy 2-connected. For a discussion on the impact of this requirement on the underlying topology control layer, see Section 6.

Given the pricing scheme, we can define the winning path MP as *feasible* if $c_S(D) < m$. If this condition does not hold, the communication cannot take place, since the sender would be forced to pay an amount of money that exceeds m, violating the condition of individual rationality.

Note that in general we have $c(P^{-MP}) \neq \sum_{v \in MP, v \notin \{S,D\}} pay(v)$, i.e., the amount that the sender pays and the total amount that the intermediate nodes receive are not equal. In this case we say that budget is imbalanced.[12] In our protocol, we assume that the destination node D is in charge of balancing the budget, getting the additional money if $c(P^{-MP}) > \sum_{v \in MP, v \neq S} pay(v)$,

[12] The VCG mechanism is known to have imbalanced budgets [20], and in fact under reasonable assumptions no mechanism can achieve budget balance, energy-efficiency, and truthfulness simultaneously.

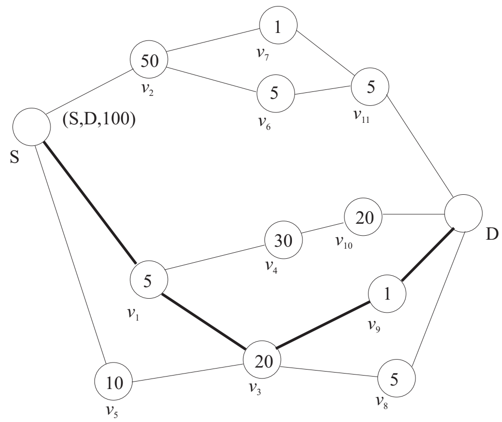

Figure 3: Example of network topology. Intermediate nodes are labeled with the corresponding power level (type). The sender offers a price of 100 for establishing a connection to the destination. The communication will take place along the minimum-energy path (bold edges).

or contributing to the payments if $c(P^{-MP}) < \sum_{v \in MP, v \notin \{S,D\}} pay(v)$. This assumption is coherent with our reference scenario, in which D is the service provider. Since the service provider is involved in many sessions, it is possible that its overall balance is close to 0. Even if this is not the case (for instance, because $c(P^{-MP}) < \sum_{v \in MP, v \notin \{S,D\}} pay(v)$ most of the time), the service provider can modify the price of the fixed (e.g., per-connection, or monthly) fee that the customers must pay to access the service in order to not reduce its revenue.

Let us clarify our pricing scheme with the example in Figure 3. The sender wants to establish a connection with the destination, and is willing to pay at most 100 for it. For the moment, let us assume that the information regarding the network topology and nodes' types is known to the destination (we see how to implement this phase of the protocol in the next sub-section). D computes the winning (minimum-energy) path MP, the replacement paths for all nodes on MP and the global replacement path P^{-MP}:

$$MP = \{v_1, v_3, v_9\} \quad c(MP) = 26 \ , \ P^{-v_1} = \{v_5, v_3, v_9\} \quad c(P^{-v_1}) = 31 \ ,$$

$$P^{-v_3} = \{v_1, v_4, v_{10}\} \quad c(P^{-v_3}) = 55 \ , \ P^{-v_9} = \{v_1, v_3, v_8\} \quad c(P^{-v_9}) = 30 \ ,$$

$$P^{-MP} = \{v_2, v_7, v_1 1\} \quad c(P^{-MP}) = 56 \ .$$

The price that S should pay is $c(P^{-MP}) = 56 < 100$, so MP is feasible.

The payments for the nodes in the winning path are computed as follows:

$$pay(v_1) = c(P^{-v_1}) - c(MP) + l(v_1) = 31 - 26 + 5 = 10$$

$$pay(v_3) = 55 - 26 + 20 = 49\ ,\quad pay(v_9) = 30 - 26 + 1 = 5\ .$$

The total payments amount to 64. Since S will pay only 56, the remaining 8 units of money are paid by the destination. Note that, if the type of node v_{11} is 20 instead of 5, we have $c(P^{-MP}) = 71$, with all the other costs unchanged. In this situation, the sender would pay 71 for the communication (which is still below 100), and the 7 units of money remaining after paying all the intermediate nodes would be retained by the destination.

In order to evaluate the impact of the different topology control strategies on budged balance, we performed a set of simulations, with the same experimental setting used to obtain Figure 2. Figure 4 reports the average budget balance of the communication, i.e., the average difference between the cost of the global replacement path $c(P^{-MP})$ and the total communication cost $\sum_{v\in MP, v\notin\{S,D\}} pay(v)$. The values reported in the graphic are normalized with respect to the total communication cost (i.e., they are $\frac{c(P^{-MP})-\sum_{v\in MP,v\notin\{S,D\}} pay(v)}{\sum_{v\in MP,v\notin\{S,D\}} pay(v)}$). Figure 4 shows that on the average the destination node must contribute some money to support the communication. The relative amount of this contribution, which strongly depends on the number of nodes and on the topology of the communication graph, can be confined to less than 5% of the total cost with the CTR topology control protocol[13]. Thus, using the global replacement path to define the payment that the sender needs to make is a novel idea in distributed game theory that, combined with a suitable topology control protocol, turns out almost balance the budget on the average.

4.3 Protocol specification

In this section, we describe in details the distributed implementation of our approach. COMMIT consists of two phases:

- *Route discovery*: The communication graph is computed by a limited flooding process, the winning path MP and payments are computed by D and communicated to S.

[13] In our provider model, this imbalance implies that the provider should take the imbalance into account when determining the monthly fee that users are charged for access (through traffic estimates) in such a way that the provider is able to make a profit without having to raise access fees on a regular basis.

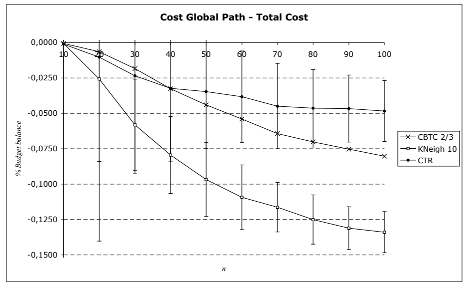

Figure 4: Budget balancing in the different scenarios. The plot shows the difference between the cost of the global replacement path and the total communication cost, normalized with respect to the total communication cost. To improve clarity, the 95% confidence intervals are shown only for KNeigh and CTR.

– *Data transmission* (only if MP is feasible): Data packets and payments are sent along the winning path MP from source S to destination D (or vice-versa) until the sender terminates the connection or until the topology control protocol updates the topology.

In the route discovery phase, sender S sends (using power $l(S)$) a route discovery message $RD(S, D, m)$, indicating that it wants to start a data transmission session with node D, and that it wishes to pay at most m for this service for each data packet that is sent in this session. The route discovery request is committing for node S, subject to the price constraint: if a path to the destination is found such that the total payment $c_S(D)$ of S is at most m, then S must send the packets and pay the correct amount of money. Otherwise, node S will eventually detect that communicating with D at the given price is not possible.

In the route discovery phase, an intermediate node v_k receives messages of the form

$$RD(S, D, m, v_1, l(v_1), \ldots, v_{k-1}, l(v_{k-1}))$$

where path $v_1, \ldots v_{k-1}$ indicates a path from sender S to node v_{k-1}. The amount of money that is left once v_k receives the message is the original offer by S minus all costs along the path, i.e., $m - \sum_{i=1}^{k-1} l(v_i)$. Node v_k builds up its own local view of the communication graph by receiving messages: whenever it receives a path containing information about the existence of an edge that it does not yet know, it adds this information to its local view. Node v_k then appends to the message that contains new information the fields $v_k, l(v_k)$, and forwards it with power $l(v_k)$. In order to prevent other nodes from altering the fields $v_k, l(v_k)$, these fields are cryptographically signed by node v_k. Moreover, v_k signs the field v_{k-1} to acknowledge that an edge between v_{k-1} and v_k exists.

This flooding process is repeated until the route discovery message arrives at the destination D. The destination does not forward messages, but other than that it acts just like a regular intermediate node: it collects the RD messages arriving from different nodes, and builds up a complete view of the communication graph. Once the destination has received all information, it computes the minimum energy path $MP = \{S, v_1, \ldots, v_k, D\}$ from sender S to the destination, the replacement paths P^{-v_i} for each intermediate node v_i on the minimum energy path MP, and the global replacement path P^{-MP}. Given this, D determines whether MP is feasible (i.e., if $c_S(D) = c(P^{-MP}) < m$) and, in case the answer is positive, it computes the payment/premiums for S and the nodes in MP. It then sends back this information (winning path, payments, and the global replacement path) to sender S using the reverse of path MP. In order to avoid that intermediate nodes manipulate the payments, we assume that this message is cryptographically signed by D. To this purpose, a secure routing protocol such as the one proposed in [17] can be used[14]. The sender S then sends a test packet along the global replacement path in order to verify that this path actually exists, asking each node v in P^{-MP} to sign that the two neighbors of v on P^{-MP} are actually neighbors. The destination receives the signed test packet, checks all signatures, and then sends a packet along the reverse MP path to the sender to indicate that it can start the data transmission phase.

After the route discovery phase, the *data transmission* phase takes place, in which the sender sends/receives its data packets to/from the destination via the computed minimum energy path. With each packet, it includes an electronic payment that is due to the intermediate nodes. The nodes on MP forward the data packet and collect the payments. Several methods for payment distribution and collection have been proposed in the literature [9, 28], and any of those could be applied here. The data transmission phase ends when the sender has transmitted its last packet or when the topology control protocol changes the network topology in order to account for node mobility. The latter case forces the sender to initiate a new route discovery phase.

[14] Actually, the ARIADNE protocol of [17] has been recently shown to be not necessarily secure [1] (depending on the notion of security), and the issue of provably secure routing in ad hoc networks is still open.

An important issue to address in the data transmission phase is whether selfish nodes have an interest in forwarding packets after the routing discovery phase. As it has been noted in [29], route discovery and data transmission are different phases of the protocol execution, and a truthful implementation of route discovery does not necessarily imply a truthful implementation of the subsequent data forwarding phase. To solve this problem, the authors of [29] introduce the concept of cooperation-optimal protocol, and they show that CORSAC is cooperation-optimal. An alternative concept introduced in distributed mechanism design is the concept of faithfulness [24], which refers to an equilibrium in which no selfish node has an incentive to deviate from the suggested behavior, provided the other nodes behaves rationally. Although a formal proof of this fact is beyond the scope of this paper, it is easy to see that COMMIT satisfy faithfulness: given the pricing rule defined in Section 4.2, and given the fact that payments to the nodes are delivered only after actual packet transmission, it would be irrational for a selfish node u to drop a packet (under the assumption that all the other nodes behave rationally), since the payment node u receives for forwarding a packet is not less than the actual cost of sending the packet.

Optimizations. The route discovery phase of COMMIT as described above leaves room for improvement.

A first optimization is the following. Instead of forwarding whole paths every time a new path is received, the nodes could forward only new edges that it has learned of and that give rise to new paths. This reduces the message complexity of the route discovery phase.

The second optimization is somewhat more involved. An intermediate node v_k can compute whether a newly received path is feasible in the sense that it has a non-negative amount of money left at v_k. If the path is not feasible, there is no point in forwarding it because communication will not take place even if this path is either the minimum-energy path, or a replacement path for a node on MP, or the global replacement path. Thus, node v_k has no economic incentive to propagate the route request, and will simply drop it. Note that this "selfish" behavior of v_k turns out to be beneficial for the whole network, since the dropped message was useless. In other words, with this optimization implemented only RD messages referring to paths that have some chance to win the auction, or that are needed to compute the payments, will circulate in the network, eventually reaching the destination node D.

If the first optimization measure is implemented, node v_k still adds the new information from the path into its local view of the communication graph and forwards this information as soon as it receives information regarding an edge that renders the path feasible.

5 Protocol analysis

5.1 Energy efficiency

Assuming that all nodes act truthfully (which we will prove in the next subsection), it is straightforward to see that COMMIT computes the most energy-efficient (least cost) path to route along. Since the destination knows the complete communication graph, it is simple to compute a minimum-energy path and the replacement paths in polynomial time using one of several algorithms (see [23]) for computing the shortest path. Thus, we have the following proposition:

Proposition 1. *If all nodes act truthfully, COMMIT computes the most energy-efficient (least cost) route from the sender S to the destination D.*

5.2 Truthfulness and individual rationality

In this section, we show that truth-telling is a dominant strategy, and that the protocol satisfies individual rationality. We consider each type of player (sender, relay node, and destination) separately. For every type of player, we show that truth-telling is the dominant strategy, and that participating in the protocol is individually rational. When analyzing the behavior of one player, we assume that all the other players act truthfully. This is only for the sake of presentation, as the argumentation below applies also when the other players play arbitrary strategies.

Sender. Individual rationality for the truthful sender follows immediately by observing that, given our pricing mechanism, S will never pay a price that exceeds m. Thus, participating in the protocol will never decrease the sender's utility.

Let us now prove that truth-telling is the dominant strategy for the sender. Let us denote with m_f the false type declared by S, and with m the true type. We have the following cases:

1) $m_f < m$. Let us denote with $c(P^{-MP})$ the cost of the global replacement path. We have the following sub-cases:

   1.a) $c(P^{-MP}) < m_f < m$. In this case, the communication takes place with both declarations, and the utility of the sender remains the same. This is implied by the fact that the price paid by S is $c(P^{-MP})$, which does not depend on the sender's declaration.

   1.b) $m_f \leq c(P^{-MP}) < m$. In this case, if the sender would declare m_f instead of m the communication would not take place. Lying about its type, S would decrease its utility from $m - c(P^{-MP}) > 0$ to 0.

   1.c) $m_f < m \leq c(P^{-MP})$. In this case, declaring m_f instead of m would leave the sender's utility unchanged at 0.

2) $m_f > m$. The proof is along the same lines of case 1) above.

Since the sender never increases its utility by declaring a false type, we can conclude that truth-telling is a dominant strategy for the sender.[15]

Relay nodes. The proof of thruthfulness and individual rationality for the relay nodes is reported in the Appendix.

Destination. In our protocol, we simply assume that the destination node D acts truthfully. This assumption, which is done also in [2, 29], is motivated by the observation that it is in the destination's interest to receive the data. If we consider the reference application scenario of Section 2, the destination is actually the service provider, whose interest is that the new connection is established, and the customers are happy. By computing the payments truthfully (as it is assumed here), the provider will satisfy both the sender (which pays at most

[15] A tempting alternative to our payment rule for the sender would be to simply require a fixed price b (rather than the complicated rule with the global replacement path) that the sender would have to pay for every connection. However, such a scheme would prevent many connections from being established that could have been established under the global replacement path rule (i.e., all connections with $b > m > c(P^{-MP})$). This is highly undesirable from a social point of view and in fact even from the point of view of the provider who charges a constant per connection fee in addition to the variable component that we are concerned with in this section.

the offered price) and the intermediate nodes (which receive payments that cover their cost, plus a premium), while achieving a network-wide goal (energy efficiency). Under our working assumption of no collusion, the service provider has no interest in letting the sender pay less than the correct price, or that the intermediate nodes get overpayments, since this would end up making the counterpart (the sender, or the intermediate nodes) somewhat unhappy. This argumentation further validates our assumption of truthful destination.

Observe that with respect to Ad hoc-VCG and CORSAC we have one additional assumption on the destination, namely that it balances the payments in case the winning path is feasible. As discussed above, we believe this assumption is economically meaningful: since a node is in general the destination of several data sessions, it is possible that the overall balance after a certain time is close to zero. This argumentation is confirmed by the simulation results reported in Section 4.2, especially if COMMIT is combined with the CTR topology control protocol. In case the destination is the service provider, there is an additional possibility to balance the cost: increase/decrease the fixed fee that the customers must pay in order to access the service. Thus, summing up, we have proved the following theorem:

Theorem 1. *If the COMMIT protocol is executed in an ad hoc network to route messages, behaving truthfully is a dominant strategy and individually rational for all nodes (except for the destination).*

5.3 Message complexity

Theorem 2. *COMMIT has $O(|M|^2 d)$ message complexity, where M is the subset of all relay nodes in the communication graph such that their minimum energy path to the sender has cost lower than m (the reserve price), and d is the maximum node degree in the communication graph.*

Proof. Assume that we implement COMMIT with both optimization options, i.e., only edges are forwarded and paths longer than m are thrown away. Clearly, $|M| \leq n-2$ and messages are only passed between the source, destination, and nodes in M. Since each node in M forwards edge information about at most $O(|M|d)$ edges, we have a total message complexity of $O(|M|^2 d)$. □

Considering that $|M| \leq n-2$ (actually, it might be $|M| \ll n-2$ depending on the value of m), and that most of the topology control protocols build communication graphs with small degree ($d = O(\log n)$, or even $d = O(1)$ is some cases) this is a significant improvement over the $O(n^3)$ message complexity of Ad hoc-VCG.

6 The cost of cooperation

In our protocol, the payment for establishing the communication exceeds the actual cost of the minimum-energy path. This is due to the fact that, in order to motivate the intermediate nodes to cooperate, they must be given some premiums. The difference between the overall amount of these premiums and the cost of the minimum-energy path can be interpreted as the *cost of cooperation*.

The cost of cooperation is a measure of the economic inefficiency induced by the need of stimulating selfish nodes to act unselfishly. This inefficiency occurs when the minimum-energy path has a cost below the offered price m (so, in principle, the communication should take place), but $c(P^{-MP}) > m$, causing the communication to be aborted.

From the protocol designer's point of view, the cost of cooperation should be as low as possible (note that, on the contrary, from the intermediate nodes' point of view this cost should be as high as possible). Unfortunately, unless some a-priori (probabilistic) information on the player's types is known to the destination, the VCG mechanism (which is the cause of the economic inefficiency) is essentially the only pricing scheme that achieves truthfulness, individual rationality, and routing along the minimum-energy path [14, 20].

In case of COMMIT, the cost of cooperation depends on the distribution of the energy cost of the paths connecting to D: if all these paths have approximately the same cost, then the cost of cooperation is relatively low; otherwise, it can be quite high. For example, in the scenario of Figure 3 the cost of cooperation is $64 - 26 = 38$, i.e., a very large percentage of the total amount of money that the sender and the destination will pay. It is not difficult to build worst-case scenarios in which the cost of cooperation is very high.

However, in our approach we have a way to reduce (to a certain extent) the cost of co-operation: *changing the topology of the network*. In other words, the network designer could use the underlying topology control protocol to build communication graphs with the desired feature (many paths with approximately the same energy cost), thus reducing the average cost of cooperation. More specifically, the designer could determine which topology control protocol is more effective in reducing the cost of cooperation, then it could design an incentive compatible realization of the selected protocol along the guidelines described in [13]. The fact that topology control has a strong influence on the economic efficiency of COMMIT is supported by the simulation results concerning budget balancing: by changing the topology control protocol used in combination with COMMIT, the average budget imbalance can be reduced by approximately 15%. We believe this observation is quite interesting, since it discloses a new metric (besides traditional metrics such as connectivity, node degree, etc.) that can be used to evaluate the performance of topology control algorithms.

Observe that in this paper we rely on a relatively strong property of the communication graph, namely that it is minimum-energy 2-connected. To the best of our knowledge, none of the existing topology control protocols guarantee this property in the worst-case. However, it is our intuition that graphs generated by common protocols such as those presented in [4, 26], or some straightforward variation of these protocols, satisfy this property on the average. Extensive simulations, whose results we partly reported in Figure 2, strongly support this intuition.

Since the cost of cooperation might be quite high, a natural question to ask is the following: are side payments (or other forms of incentives) really necessary to stimulate cooperation in ad hoc networks? In order to answer this question, we use the notion of Nash Equilibrium (NE), which is well known in game theory [20]. NE can be intuitively described as follows: a set of strategies (one for each player) is a NE if every player has no incentive for changing her strategy, given that the other players do not change their strategies as well. The notion of NE is much weaker than the notion of truthfulness: in a NE, we can identify a best player strategy (e.g., truth-telling) *given the strategies of the other players*; on the other hand, if a protocol is truthful, *any* player is always better off behaving truthfully, *regardless of the strategy played by the other nodes.*

In practice, the difference between NE and truthfulness may be dramatic: if a system is in a NE (say, all nodes are behaving well), but a fraction of nodes start deviating from this strategy (e.g., dropping packets), then the other nodes will eventually change their strategies, possibly ending in a different NE (e.g., every node drops all the packets). Conversely, truthful protocols are resilient to any fraction of "badly behaving" nodes.

The NE of packet forwarding strategies for ad hoc networks has been investigated in two recent papers [15, 25]. In particular, in [15] Felegyhazi et al. show that the strategy in which every node drop all the packets is a NE. They also show that, under certain conditions that depends on the network topology, more cooperative strategies can be a NE as well. Unfortunately, these conditions are very unlikely to occur in real networks, and the authors of [15] conclude that, in practice, *an incentive mechanism is needed to stimulate cooperation.*

7 Conclusion and future work

In this paper, we have introduced the COMMIT protocol for individually rational, truthful, and energy efficient routing in ad hoc networks. Besides presenting and analyzing our protocol, we have discussed several issues related to cooperation in ad hoc networks. In particular, we have identified a quantity that can be considered the intrinsic cost of cooperation, and pointed out that topology control can be used to curb this cost.

This paper also discloses interesting avenues for further research. In particular, the interplay between topology control and routing in a selfish environment should be carefully investigated. Recently, we have proposed truthful implementations of some topology control protocols [13]. While in principle composing two individually truthful protocols (topology control and routing) does not necessarily imply that the composition of the protocols is truthful (see, e.g., the observations in [29]), we believe that truthful implementations of the individual tasks are a good starting point for designing a comprehensive truthful solution.

References

[1] G. Acs, L. Buttyan, I. Vajda, "Provably Secure On-Demand Source Routing in Mobile Ad Hoc Networks", *IEEE Transactions on Mobile Computing*, to appear.

[2] 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.

[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] 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.

[5] S. Buchegger, J. Le Boudec, "Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks", *in Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pp. 403–410, IEEE Computer Society, 2002.

[6] S. Buchegger, J. Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes – Fairness in Dynamic Ad-hoc NeTworks", *in Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Lausanne Switzerland, 2002.

[7] L. Buttyan and J. Hubaux, "Enforcing Service Availability in Mobile Ad-Hoc WANs", *in Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Boston, August 2000.

[8] L. Buttyan and J. Hubaux, "Nuglets: a virtual currency to stimulate cooperation in self-organized ad hoc networks", *in Technical Report EPFL*, DSC, 2001.

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

[10] K. Chen, K. Nahrstedt, "iPass: an Incentive Compatible Auction Scheme to Enable Packet Forwarding Service in MANETs", *Proc. IEEE ICDCS 2004*, 2004.

[11] *Cisco Aironet 350 data sheets*, available at `http://www.cisco.com/en/US/products/hw/wireless`.

[12] 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.

[13] S. Eidenbenz, P. Santi, G. Resta, "A Framework for Incentive Compatible Topology Control in Non-Cooperative Wireless Multi-Hop Networks", *Proc. 2nd ACM Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS)*, pp. 9–18, 2006.

[14] J. Feigenbaum, C. Papadimitriou, R. Sami, S. Shenker, "A BGP-based Mechanism for Lowest-Cost Routing", *Proc. ACM PODC*, pp. 173–182, 2002.

[15] M. Felegyhazi, L. Buttyan, J.P. Hubaux, "Nash Equilibria of Packet Forwarding Strategies in Wireless Ad Hoc Networks", *IEEE Transactions on Mobile Computing*, vol. 5, n. 5, pp. 463–476, 2006.

[16] P. Gupta, P.R. Kumar, "Critical Power for Asymptotic Connectivity in Wireless Networks", *Stochastic Analysis, Control,Optimization and Applications*, Birkhauser, Boston, pp.547-566,1998.

[17] Y. Hu, A. Perrig, D. Johnson, "Ariadne: a Secure On-Demand Routing Protocol for Ad Hoc Networks", *Proc. ACM Mobicom 02*, pp. 12–23, 2002.

[18] Y.-D. Lin, Y.-C. Hsu, "Multihop Cellular: A New Architecture for Wireless Communications", *Proc. IEEE INFOCOM 2000*, 2000.

[19] H. Luo, R. Ramjee, P. Sinha, L. Li, S. Lu, "UCAN: A Unified Cellular and Ad Hoc Network Architecture", *Proc. ACM Mobicom 03*, pp. 353–367, 2003.

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

[21] `http://www.priceline.com`

[22] P. Santi, D.M. Blough, "The Critical Transmitting Range for Connectivity in Sparse Wireless Ad Hoc Networks", *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, pp. 25–39, January-March 2003.

[23] R. Sedgewick, *Algorithms*, Addison-Wesley, 1992.

[24] J. Shneidman, D. Parkes, "Specification Faithfulness in Networks with Rational Nodes", *Proc. ACM PODC 2004*, pp. 88–97, 2004.

[25] V. Srinivasan, P. Nuggenhalli, C. Chiasserini, R. Rao, "Cooperation in Wireless Ad Hoc Networks", *Proc. IEEE Infocom 03*, pp. 808–817, 2003.

[26] 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.

[27] H. Wu, C. Qios, S. De, O. Tonguz, "Integrated Cellular and Ad Hoc Relaying Systems: iCAR", *IEEE Journal on Selected Areas in Communications, 19(10), October 2001*, 2001.

[28] 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.

[29] S. Zhong, L. Li, Y. Liu, Y. Yang, "On Designing Incentive Compatible Routing and Forwarding Protocols in Wireless Ad Hoc Networks - an Integrated Approach Using Game Theoretical and Cryptographic Techniques", *Proc. ACM Mobicom 05*, pp. 117–131, 2005.

A Individual rationality and truthfulness of relay nodes

Individual rationality for the truthful relay node follows immediately by observing that, given our pricing mechanism, in case the node is in the winning path its payment is at least as high as its cost. In other words, a relay node will never get a negative utility when acting truthfully.

We now show that it is in a relay node's best interest to follow the protocol specification. Similarly to Ad hoc-VCG [2], we assume that the nodes are willing to forward packets in the route discovery phase because of the potential payoff. This assumption is reasonable if the data session is relatively long as compared to the route setup phase (the application scenario of Section 2 is a good example of this situation). If this is the case, the cost of transmitting the few control packets exchanged in the route setup phase can be considered as negligible as compared to the potential payoff of being in the winning path.

In those situations in which the cost of the route setup phase cannot be neglected, our protocol can be extended along the guidelines described in [2], where a variation of Ad hoc-VCG that pays the nodes even for participating in the route discovery phase is described in the Appendix.

COMMIT requires that a test message is sent along the global replacement path before the data session starts. As we shall see, sending this message is needed in order to prevent one of the possible cheating behaviors of the relay nodes. However, in general the nodes in the global replacement path have no interest in forwarding the test packet to the destination, since they know that they are not part of the winning path. In order to deal with this situation, nodes in the global replacement path can be paid a unit amount of money, along the guidelines described in the Appendix of [2]. An alternative approach to deal with this problem in the reference scenario of Section 2 is the following. Since the destination knows the identity of the nodes in the global replacement path P^{-MP}, and knows that S will send a test packet along P^{-MP} before starting the data session, it can take some countermeasures in case the test packet is not received. An obvious countermeasure is to interrupt the service delivery to all the nodes in P^{-MP}. In this case, since the cost of sending a control packet can be considered as negligible, nodes in P^{-MP} would be motivated to forward the test packet on S's behalf, in order to preserve the "external utility" provided by accessing the service.

Let us now analyze the different cheating behaviors of the relay nodes. An intermediate node v could: a) lie about its type (power level $l(v)$); b) propagate a path with false information; c) intentionally fail to propagate a path with new information; d) combine above possibilities.

Cheating option *a*). Let $l(v)$ and $l_f(v)$ denote the true and the declared type of v, respectively. Let us first suppose $l(v) < l_f(v)$. In this case, if $v \notin MP$ with the true declaration, it would remain out of the winning path also declaring $l_f(v)$, and the utility would remain unchanged at 0. Assume then that $v \in MP$ in the truthful case. First, we observe that v's declaration has no effect on the decision rule. In other words, v has no way to turn MP into a feasible path (in case it is not feasible) by simply reporting a false type. As an effect of the overdeclaration, v might be kicked off the winning path, decreasing its utility from a positive value (we recall that when a node is on the winning path and reports truthfully, it always gets a positive utility) to 0. In case v would remain in the winning path overdeclaring its type, its utility would remain unchanged. In fact, denoting with $c(MP)$ and $c_f(MP)$ the cost of the winning path in the truthful and false scenario, respectively, we have $c_f(MP) = c(MP) - l(v) + l_f(v)$. Since the cost of P^{-v} does not depend on v's declaration, we have $pay_f(v) = c(P^{-v}) - c_f(MP) + l_f(v) = c(P^{-v}) - c(MP) + l(v) - l_f(v) + l_f(v) = pay(v)$. So, overdeclaring the type would not increase v's payment, leaving the utility unchanged.

Let us now suppose $l(v) > l_f(v)$. In this case, if v is in the winning path MP with the truthful declaration, it would remain in MP also underdeclaring its type. By applying the same argument as above, it is easy to show that v's utility would not be changed by the false declaration. Let us now assume that v is not in MP. If underdeclaring its type is not sufficient for v to join the winning path, then its utility remains unchanged at 0. However, it might be the case that v's underdeclaration would drive it in the winning path. We show that this cheating behavior results in a negative utility for v. Let $c(MP)$ denote the cost of the true winning path, and $c(MP_v)$ the true cost of the minimum energy path including v. Since v is not in MP, and assuming for simplicity that the minimum energy path is unique, we have $c(MP_v) > c(MP)$. Let $c_f(MP_v)$ denote the cost of MP_v as resulting from v's underdeclaration. By hypothesis, we have $c_f(MP_v) < c(MP)$. Let us now compute the payment $pay_f(v)$ for v in the false scenario. We have $pay_f(v) = c(P^{-v}) - c_f(MP_v) + l_f(v)$. Observing that $c(P^{-v}) = c(MP)$ and $c_f(MP_v) = c(MP_v) - l(v) + l_f(v)$, we can write $pay_f(v) = c(MP) - c(MP_v) + l(v) - l_f(v) + l_f(v) = c(MP) - c(MP_v) + l(v)$. Hence, the utility of v under the false scenario is $u_v = pay_f(v) - l(v) = c(MP) - c(MP_v) < 0$. Thus, by underdeclaring its type v would reduce its utility from 0 to a negative value. Finally, we observe that also in this case v's declaration has no effect on the decision rule.

Cheating option *b*). First, we observe that a node cannot alter the declared power levels of other nodes as they are signed by these nodes. Hence, v can propagate false information only by creating a false edge e' in one of the paths. However, the existence of e' must be authenticated by both endpoints of e'. It follows that v can create a false edge only between another node and v itself or between another node and one of v's neighbors. In particular, node v could report a false paths by falsely creating a neighbor as follows: node v could take a message $RD(S, D, m, v_1, l(v_1), \ldots, v_{i-1}, l(v_{i-1}))$ and then forward a message $RD(S, D, m, v_1, l(v_1), \ldots, v_{i-h}, l(v_{i-h}), v, l(v))$, with its signature verifying that v_{i-h} is one of its neighbors. We call this action "creating a false neighbor". Node v could also report a false path by simply forwarding a message $RD(S, D, m, v_1, l(v_1), \ldots, v_{i-h}, l(v_{i-h}))$ without appending its own information and again deleting some of the nodes in the original message. Thus, node v could create an edge (v_{i-h}, v_{i+1}), where v_{i+1} is a neighbor of v. We call this action "creating a false overhop path".

Let us first consider the situation in which v creates a false neighbor. Observe that the false edge $e' = (v_{i-h}, v)$ is incident in v. Further, we observe that reporting an additional edge in the graph can only *decrease* the cost of some of the paths in it.

Assume that v is in the winning path MP in the truthful scenario, and that MP is feasible.

In this situation, v's utility is $u_v = c(P^{-v}) - c(MP)$ (we recall that we are assuming that v declares truthfully). By reporting the false edge e', v could reduce the cost of the (false) winning path MP_f, thus increasing its utility. However, MP_f contains the false edge (v_{i-h}, v), which does not exist in the communication graph G. Since the payments are delivered during the data session and (v_{i-h}, v) is not in G, v would receive 0 payment instead of $pay(v) = c(P^{-v}) - c(MP) + l(v)$, thus reducing its utility. The only possibility to get some payment in the scenario with the false edge e' is that the intermediate nodes between v_{i-h} and v accept to cooperate with v_{i-h} and v, forming a "spontaneous coalition". However, collusion between selfish nodes is not allowed in our model.

Let us now assume that v is in MP, but MP is not feasible. In this case, v's utility is 0, and the only possibility for v to increase its utility would be to reduce the cost of the global replacement path P^{-MP}. Since the false edge e' is incident in v, it cannot belong to P^{-MP}, and the cost of P^{-MP} remains unchanged also in case of false edge reporting. Thus, the utility of v would remain unchanged at 0.

The third scenario to consider is when v is not in the true minimum-energy path MP, but it is in the (false) minimum energy path MP_f created by falsely reporting edge e'. Since edge e' is not in G and the payments are delivered only during the data session, node v would remain with 0 utility, unless a "spontaneous coalition" is formed to simulate edge e', but coalitions are not allowed in our model.

Let us now consider the case of a false overhop edge $e' = (v_{i-h}, v_{i+1})$, where v_{i+1} is one of v's neighbors. In this case, the false edge e' is not incident to v.

Assume that v is in the true minimum energy path MP, and that MP is feasible. In this case, v's utility is $u_v = c(P^{-v}) - c(MP)$. Since falsely reporting e' could only decrease $c(P^{-v})$, while leaving $c(MP)$ unchanged (actually, there is even the possibility that reporting e' kicks v out of the winning path), this action can only reduce v's utility.

Assume that v is in the true minimum energy path MP, but MP is not feasible. In order to increase its 0 utility, node v could try to reduce the cost of the global replacement path by falsely reporting edge e'. However, the protocol prescribes that, before starting the data session, a test message is sent along the global replacement path. Since e' does not exist in the communication graph G, the test message would not reach the destination, and the data session would be aborted. The only possibility to avoid this is that the nodes at the endpoints of edge e', the intermediate nodes that should simulate the existence of e', and some of the nodes in MP would form a "spontaneous coalition" C, which is not allowed in our model.

Finally, let us assume that v is not on the winning path MP. Since v is not one of the endpoints of edge e', falsely reporting e' would leave v out of the minimum energy path anyway, leaving its utility unchanged at 0.

Cheating option *c*). This cheating option can be equivalently restated as "v fails to propagate the information about an edge e". We start by observing that if the information about e reaches the destination through a path non involving v, then v's bad behavior will have no effect on the payments and on the decision rule; consequently v's utility would be unchanged.

Let us assume that v is in the winning path MP in the truthful scenario, and that the winning path is feasible. In this case, v's utility is $u_v = c(P^{-v}) - c(MP)$. How can node v increase its utility by failing to report some edge e? If e is on MP, then not reporting it to the destination can only increase the cost of MP (possibly even kicking v out of the winning path), reducing v's utility. On the other hand, if v would not report the information about an edge in P^{-v}, then this information would reach D anyway by means of the nodes in P^{-v}. Thus, node v has no incentive in not reporting edge information in this case.

Assume now that v is in the winning path MP, but that MP is not feasible because $c(P^{-MP})$ exceeds m. Also in this case, v has no way to increase its utility by not reporting some edge e, since not reporting an edge could only result in increasing the cost of some path.

Let us now assume that v is not in the winning path MP in the truthful scenario, and that it tries to join the winning path by not reporting one of the edges e. Let us denote with MP_v the minimum energy (S, D) path that includes v in the truthful scenario. Clearly, we have $c(MP_v) > c(MP)$ (for simplicity, we are assuming that the minimum energy path is unique). Since all the nodes in MP report truthfully, and by not reporting an edge v can only increase the cost of $c(MP_v)$, there is no way for v to turn MP_v into the winning path.

Cheating option *d*). Cheating opportunity d) combines options a), b), and c), but even combinations do not increase v's utility: many of such combinations could result in additional utility for a "spontaneous" coalition of nodes, but collusion is not allowed in our model.