Emergency connectivity in ad-hoc networks with selfish nodes

George Karakostas[1,2,⋆] and Euripides Markou[2,3,⋆⋆]

[1] Department of Computing & Software.
[2] School of Computational Engineering & Science.
McMaster University, 1280 Main Street West, Hamilton, Ontario L8S 4K1, Canada.
[3] Department of Computer Science, University of Ioannina, Greece.
E-mail: karakos@mcmaster.ca, emarkou@cs.uoi.gr

Abstract. Inspired by the CONFIDANT protocol [1], we define and study a basic reputation-based protocol in multihop wireless networks with selfish nodes. Its reputation mechanism is implemented through the ability of any node to define a threshold of tolerance for any of its neighbors, and to cut the connection to any of these neighbors that refuse to forward an amount of flow above that threshold. The main question we would like to address is whether one can set the initial conditions so that the system reaches an equilibrium state where a non-zero amount of *every* commodity is routed. This is important in emergency situations, where all nodes need to be able to communicate even with a small bandwidth. Following a standard approach, we model this protocol as a game, and we give necessary and sufficient conditions for the existence of non-trivial Nash equilibria. Then we enhance these conditions with extra conditions that give a set of necessary and sufficient conditions for the existence of connected Nash equilibria. We note that it is not always necessary for all the flow originating at a node to reach its destination at equilibrium. For example, a node may be using unsuccessful flow in order to effect changes in a distant part of the network that will prove quite beneficial to it. We show that we can decide in polynomial time whether there exists a (connected) equilibrium without unsuccessful flows. In that case we calculate (in polynomial time) initial values that impose such an equilibrium on the network. On the negative side, we prove that it is NP-hard to decide whether a connected equilibrium exists in general (i.e., with some nodes using unsuccessful flows at equilibrium).

1 Introduction

In recent years there has been a great effort in designing robust and efficient wireless networks of devices that take upon themselves certain network responsibilities that used to be the responsibilities of a central network designer in traditional network design. For example, in ad-hoc networks the topology of the

⋆ Research supported by an NSERC Discovery Grant and MITACS.
⋆⋆ Research supported by MITACS.

network is the result of cooperation amongst the nodes themselves: in a multi-hop wireless network, a successful transmission between a pair of nodes requires the cooperation of intermediate nodes in order for the transmitted packets to reach their destination. While this may be guaranteed in networks with a central authority forcing the nodes to cooperate, in the absence of such an authority cooperation may not be guaranteed. This is due to the *selfishness* of each node, i.e., the effort by the node to maximize its own utility without caring about the results of its actions on the overall network-wide outcome. For example, if battery life is a valuable resource for a node, forwarding packages between two other nodes consumes energy that doesn't result in any kind of pay-off for this node, and as a result it may decide to stop cooperating in forwarding packages for others. If this behavior prevails throughout the whole network, it may eventually result in zero throughput for everybody, a phenomenon better known as the "Tragedy of the Commons" [5]. To cope with this problem one can offer *incentives* to nodes such as rewards for their cooperation or punishment for non-cooperation.

The two most commonly proposed forms of incentives are micro-payments, and reputation-based mechanisms. One of the main motivation for developing them is the desire of the network designer to not permanently punish a misbehaving node, but 're-socialize' it if it changes its uncooperative behavior.

Micro-payment schemes are based on the concept of distribution of credit to nodes, so that nodes are compensated for their cooperation by (virtual) credit payments, that they can then use to pay intermediate nodes for forwarding their own traffic. Hence if a node is consistently uncooperative, it will run out of credit and will have to stop transmitting. Usually, the distribution and/or the expenditure of credit is controlled by a central authority. Examples of such protocols are [2, 10, 11, 3].

Reputation-based systems are based on lists that the nodes keep on the *reputation* of their neighbors, i.e., the fraction of packets forwarded by them. They use this information in order to decide how much traffic they should forward towards their neighbors. This may be decided in a *Tit-for-Tat* fashion, i.e., when a node has to relay a packet on behalf of a neighbor, it does so with the same probability with which this neighbor forwards its own packets (see [8, 9] for examples of such mechanisms). Or, the amount to forward can be decided according to (centralized or local) *ratings tables*, that give the nodes an indication of the behavior of other nodes; if a node's rating of another node falls below a certain threshold, then the latter cannot be trusted to forward traffic, and therefore nothing is forwarded to it by the former, i.e., the edge connecting the two nodes is *cut* by the first node. An example of such a mechanism that actually distributes the reputation information so that each node can form its own ratings table is the CONFIDANT protocol [1]. More recent protocols [6, 7] limit the distribution of reputation information only to one-hop neighbors.

Our results: In this work we address the connectivity issues arising in such reputation-based systems. More specifically, we would like to study whether it is possible in such a selfish environment to lead all nodes towards an equilibrium with good connectivity properties. In fact, we are very ambitious: we are looking for driving them towards an equilibrium that permits a non-zero quantity of *every* traffic demand to be satisfied. The reason for such a strict requirement is the fact that in an emergency situation police, firemen, emergency medical personnel, etc. should be able to communicate with each other even if the achieved bandwidth is very small (but still enough for emergency signals to be able to travel through the network). From the above, it is not at all obvious whether such a goal can be achieved, given the fact that each network node is autonomously playing a protocol game, after it's been set in its initial condition. Given the game-theoretic nature of such protocols, it is only natural to study them in terms of their (Nash) equilibrium states. Under this light, and given the rules of the game, i.e., the protocol, the most appropriate (indeed, in some cases the only) time a network designer can intervene in order to control the outcome is during the setting of the initial conditions, or, equivalently, by 'rebooting' the protocol with new initial values. This can be achieved by a separate broadcasting channel that all nodes are listening ('snooping') in, and whose packets are of the highest priority. Obviously, this is a very intruding method, and it would defy the purpose of selfishness if it were to be applied very frequently. But one does not (hopefully) expect catastrophic emergency situations to arise that frequently. Therefore broadcasting will not be used often.

Inspired by the CONFIDANT mechanism, we study a basic reputation-based system. The strategy of every node consists of the amount of traffic flow it sends to its various receivers, the routing of this flow, the amount of flow it forwards for every commodity in which it doesn't participate as a sender or a receiver, and a non-negative *threshold* value for each outgoing edge. The latter set of values is an abstraction of the reputation mechanism: if the amount of flow that is forwarded by node x to node y (including flow that originates at x), but is cut by y is more than the threshold value x has for y, then x disconnects edge (x, y). Later on, y may end up cutting flow that is less than the current threshold value of x for (x, y), in which case (x, y) reappears. The utility for every node increases with the flow originating at or destined for this node and reaches its destination, while decreases with the flow sent out or forwarded by this node (because, for example, the node has to spend battery energy to transmit).

The main drawback of this protocol is the assumption that every node has to make its strategy known to every other node. But at the same time, this complete knowledge of the game state gives great potential power to each node to affect parts of the network that are very far away, even in counter-intuitive ways, e.g., by sending flow whose sole purpose is to affect the current topology and discourage the flow of other nodes. Hence, this assumption may make our demand for complete connectivity even harder to achieve, and it may mean that things can be easier in a more restricted setting. As a first step towards achieving this goal, we are able to characterize the complexity of computing initial values that lead to a connected Nash equilibrium in our protocol. We do that, by giving necessary and sufficient conditions for the existence of *non-trivial* Nash equilibria. Then we enhance these conditions with extra conditions that give a set of necessary and sufficient conditions for the existence of *connected* Nash equilibria. Note that it is not always necessary for all the flow originating at a node to reach its destination at equilibrium. As mentioned above, a node may be using such *unsuccessful* flow in order to effect changes in a distant part of the network that will prove quite beneficial to it. We show that in case there is a connected Nash equilibrium *without unsuccessful flows*, we can calculate (in polynomial time) initial values that impose such an equilibrium on the network using linear programming. On the other hand, if the connected Nash equilibrium(-ia) exist, but nodes are allowed to use unsuccessful flows, then it is NP-hard even to decide whether an equilibrium exists.

Our results are derived using game-theoretic concepts, which is the standard approach for analyzing such protocols, modeled as games. But we emphasize that, other than the assumptions mentioned above, we don't impose any restrictions on the network topology, or any statistical distribution on the nodes' decisions.[1]

2 Model and Terminology

In this section we describe our model for the network and the protocol the nodes follow. The set of connections that can be realized is given by a directed graph $G(V, E)$. We emphasize that, depending on the current state of the game, not all these edges may be present. For every origin-destination pair (commodity) $(u, v),\ u, v \in V$ there is a demand $d_{(u,v)}$ that u wants to send to v. The flow is splittable, and u decides how to split and route this flow. Again, the current state of the game may not allow u to send all of $d_{(u,v)}$, so the latter serves more as an upper bound on the flow actually sent. We denote by $\mathcal{P}_i$ the set of paths connecting the i-th origin-destination pair in G, and let $\mathcal{P} := \cup_i \mathcal{P}_i$.

The current state of the network, together with the nodes' strategies are described by the following set of variables:

- $F^y_{(u,e,e',v)}$ **with** $e, e' \in E, e = (x, y), e' = (y, z), u, v, x, y, z \in V$ **and** $y \neq u, v$: This is the flow of commodity (u, v) that y receives through e, and forwards further through e'.
- $f^y_{(u,e,e',v)}$ **with** $e, e' \in E, e = (x, y), e' = (y, z), u, v, x, y, z \in V$ **and** $y \neq u, v$: This is the *decision* variable of y that sets an upper bound on the amount of flow $\sum_{g=(w,x)} F^x_{(u,g,e,v)}$ *routed through* e' that y actually forwards through e', i.e., $F^y_{(u,e,e',v)} = \min\{f^y_{(u,e,e',v)}, \sum_{g=(w,x)} F^x_{(u,g,e,v)}$ *routed through* $e'\}$ (notice that edge e' can be disconnected; in that case, what is being forwarded by y through e' is simply lost). We emphasize that $f^y_{(u,e,e',v)}$ is just the y's decision

[1] We don't assume any kind of synchronization amongst the nodes, but we do assume that the decision variables changes are instantaneous. Note that the game modeling the protocol is *not* a repeated game, and there isn't any notion of *rounds*.

variable that determines what y will do *if* there is flow from u to v which has been forwarded from x to y and needs to be forwarded through $e' = (y, z)$, while $\sum_{g=(w,x)} F^x_{(u,g,e,v)}$ is the actual flow that comes to y from x through e. So y maintains such a variable $f^y_{(u,e,e',v)}$, for every incoming edge $e = (x, y)$ and every outgoing edge $e' = (y, z)$, and every commodity (u, v).

- $O^y_{(u,e,v)}$ **with** $e \in E, e = (x, y), u, v, x, y \in V$ **and** $y \neq u, v$: This is an auxiliary variable, defined as $O^y_{(u,e,v)} = \sum_{e'=(y,z)} F^y_{(u,e,e',v)}$. It is simply the total flow of commodity (u, v) coming to y through edge e, and being forwarded by y through all its outgoing edges $e' = (y, z)$.
- $I^y_{(u,e',v)}$ **with** $e' \in E, e' = (y, z), u, v, y, z \in V$ **and** $y \neq v$: This is also an auxiliary variable, defined as $I^y_{(u,e',v)} = \sum_{e=(x,y)} F^y_{(u,e,e',v)}$. It is simply the total flow of commodity (u, v) coming to y through all its incoming edges $e = (x, y)$, and being forwarded by y through edge e'. Note that $I^y_{(y,e',v)}$ is the flow originated at y and routed through e' with destination v.
- ϵ^y_x: This auxiliary variable is defined as $\epsilon^y_x = \sum_{com.(u,v), v\neq y} (I^x_{(u,e,v)} - O^y_{(u,e,v)})$, i.e., as the part of the total flow that comes to y through e and is being blocked by y.
- $s^u_{(u,P,v)}$: This is the *decision* variable of u that determines how much flow of commodity (u, v) node u routes through path P (whether this flow amount eventually reaches v or not).
- $THR_x(y)$: This is the *decision* variable of node x that defines an upper bound on the flow forwarded by x and cut by y that x can tolerate before it cuts edge (x, y). We consider edge (x, y) disconnected when $\epsilon^y_x > 0$ AND $THR_x(y) \leq \epsilon^y_x$. Hence edge (x, y) exists in the network provided $\epsilon^y_x = 0$ OR $THR_x(y) > \epsilon^y_x$.

The following definition will be used repeatedly throughout this paper:

Definition 1. *An edge (x, y) is* connected *if $\epsilon^y_x = 0$ OR $THR_x(y) > \epsilon^y_x$, and* disconnected *otherwise.*

Therefore, the *strategy* of a node x is determined by the vector $(\boldsymbol{s^x}, \boldsymbol{THR_x}, \boldsymbol{f^x})$. Note that the routing of the flow x sends out is incorporated in the values for $\boldsymbol{s^x}$. Therefore x decides the following:

- Threshold $THR_x(y) \geq 0$, and hence decides whether edge (x, y) is connected or not.
- Variables ϵ^x_w, by deciding $f^x_{(u,e,e',v)}$ which, in turn, change the flows $F^x_{(u,e,e',v)}$. As a result, x decides whether edge $e = (w, x)$ is connected or not.
- The routing of the flow originating at x and its quantity, by deciding $s^x_{(x,P,y)}$ for any path P connecting x to y. But always $\sum_P s^x_{(x,P,y)} \leq d_{(x,y)}$.

We repeat that every node sees all decision variables of all other nodes, we don't assume any kind of synchronization amongst the nodes, but we do assume that the decision variables changes are instantaneous.

Definition of the utility function: Every node plays in a selfish way, i.e., so that its utility (defined below) is maximized. At any time t, we denote by $C_y^-, C_y^+, D_y^-, D_y^+$ the sets of connected incoming, connected outgoing, disconnected incoming and disconnected outgoing edges respectively, adjacent to node y. Then, for every node y its utility function is defined as follows:

$$util_t(y) = \begin{array}{c}\text{flow sent by } y \\ \text{and reached its} \\ \text{destination}\end{array} + \begin{array}{c}\text{flow received} \\ \text{by } y\end{array} - \begin{array}{c}\text{flow forwarded} \\ \text{by } y\end{array} - \begin{array}{c}\text{flow sent by } y \\ \text{and didn't reach} \\ \text{its destination}\end{array}.$$

More specifically,

$$util_t(y) = \sum_{e \in C_y^+} S_e^y + \sum_{e \in C_y^-} R_e^y - \sum_{e' \in C_y^+ \cup D_y^+} \sum_{u \neq y, v \in G} I^y_{(u,e',v)} - \left(\sum_{e' \in C_y^+ \cup D_y^+} \sum_{v \in G} I^y_{(y,e',v)} - \sum_{e \in C_y^+} S_e^y \right) \quad (1)$$

where

- S_e^y is the flow which has been sent by y (i.e. originated at y) through edge e and has reached its destination,
- R_e^y is the flow which has been received by y through edge e,
- $I^y_{(u,e',v)}$ is the flow of commodity (u, v) with $y \neq v$, and node y attempts to forward (or sent, if $u = y$) through edge e' (note that e' may be disconnected).

The intuition behind this definition of utility (which is very similar to the definition used in [1]), is that a node exchanges resource units (e.g., battery energy) for information units (i.e., packets received or sent successfully). Our assumption is that the correspondence is one for one. Different weighting of resources and information is a generalization left for future work.

Throughout this work, we use the standard definition of Nash equilibria, i.e., at equilibrium, no node gains an increase of its utility by changing its decision variables (strategy), while the other nodes maintain their own strategies. We will focus on *non-trivial* equilibria.

Definition 2. *A* trivial *equilibrium is any equilibrium with* $\boldsymbol{f^x} = \mathbf{0}, \forall x$, *and with* $s^u_{(u,(u,v),v)} = d_{(u,v)}$, $\forall$ *commodities* (u, v) *s.t.* $(u, v) \in E$ *and* $s^u_{(u,P,v)} = 0$ *otherwise.*

So from now on, whenever we write 'equilibrium' we mean 'non-trivial equlibrium', unless otherwise stated. We also assume that there is always at least one demand between non-adjacent nodes in G, since otherwise a trivial equilibrium is a connected one, and this case is not very interesting.

Definition 3. *An amount of flow with origin a node u and destination a node v routed through a path P is* successful *if it reaches node v, otherwise it is* unsuccessful.

3 Characterization of Nash equilibria

In this section we give necessary and sufficient conditions for the existence of an equilibrium. Our hope will be that these conditions (probably together with additional ones) will simplify the study of connected equilibria.

Definition 4. *An unsuccessful flow Φ which has been routed through edge e is* responsible *for disconnecting edge e if e would be connected without Φ.*

We group the (non-disconnected) incoming and outgoing edges for a node x as follows:

- group 1: these edges transfer only successful flows,
- group 2: these edges transfer successful and unsuccessful flows,
- group 3: these edges transfer only unsuccessful flows.

The proof of the following Theorem appears in the full version:

Theorem 1. *The game is at an equilibrium if and only if for any node x the following conditions hold:*

1. *$\epsilon_x^y = 0$, where $g = (x, y) \in C_x^+$ (i.e., node y does not cut any flow forwarded by x through the connected edge g),*
2. *if there is a successful flow between nodes u,$v \neq y$ routed through edge $g = (x, y)$, then $THR_x(y) = 0$,*
3. *if there is no unsuccessful flow going through an edge $e = (t, x)$, then $R_e^x \geq \sum_{u \neq x,v} \sum_{g=(x,y)} F^x_{(u,e,g,v)}$ (i.e., the flow that node x receives through edge $e = (t, x)$ is not less than the total flow which is coming through e and x has to forward, if all this latter flow is successful),*
4. *for any disconnected edge $g' = (x, y') \in D_x^+$ it holds that $THR_x(y') = \epsilon_x^{y'} > 0$, node x does not send any flow through g', and the (unsuccessful) flows which are responsible for disconnecting g' are being sent by at least two nodes, other than x,*
5. *let $e = (t, x)$ be an incoming connected edge to x such that all unsuccessful flows which pass through e, have been routed through outgoing disconnected edges $g' = (x, y_i') \in D_x^+$ of x; then:*
   - *$THR_t(x) = 0$,*
   - *$R_e^x \geq \sum_{u \neq x,v} \sum_{g' \in D_x^+} F^x_{(u,e,g',v)} + \sum_{u \neq x,v} \sum_{g \in C_x^+} F^x_{(u,e,g,v)}$,*
6. *the flow that node x sends successfully through all of its (connected) outgoing edges $\Phi(x)$ is maximized over all possible routings $\boldsymbol{s^x}$,*

7. *any combination of the following possible actions taken by x cannot increase its utility:*
   (a) *disconnecting a number of edges of group* 2*,*
   (b) *decreasing the unsuccessful flow that x lets go through edges of group* 3*,*
   (c) *connecting edges $e' = (t', x) \in D_x^-$,*
   (d) *sending successful and unsuccessful flow through the outgoing edges of x,*
   (e) *increasing thresholds*

Theorem 1 is essentially a codification of all the conditions that happen simultaneously at equilibrium. But showing that such a (non-trivial) equilibrium exists (or, even more, compute it) is non-trivial. In fact, we will show that deciding the existence of an equilibrium is NP-hard. But it turns out it is much easier to check whether there is a non-trivial equilibrium with only successful flows; this can be reduced to the solution of a simple LP.

For every edge $e = (u, v)$, we set $d(e)$ equal to $d_{(u,v)}$ if commodity (u, v) exists, and 0 otherwise. Let $D := \sum_{e \in E} d(e)$. We will use the following notation:

- $e \in^* P$, when edge $e \in P$ is *not* the last edge of P,
- $e \in^0 P$, when edge $e \in P$ is the last edge of P.

In the following LP, variables $x(P)$ represent the amount of flow sent along path P:

$$\max \quad \sum_{P \in \mathcal{P}} x(P) \quad \text{s.t.} \tag{LP-S}$$

$$\begin{aligned}
\sum_{P: e \in^* P} x(P) - \sum_{P: e \in^0 P} x(P) &\leq 0 && \forall e \in E \\
\sum_{P \in \mathcal{P}_i} x(P) &\leq d_{(u_i, v_i)} && \forall i \\
x(P) &\geq 0 && \forall P \in \mathcal{P}
\end{aligned}$$

Theorem 2. *A non-trivial equilibrium with only successful flows exists if and only if* (LP-S) *has a solution $x(P)$ with $\sum_{P \in \mathcal{P}} x(P) > D$.*

Proof: The proof appears in the full version. □

The solution of (LP-S) by standard techniques [4] implies the following

Corollary 1. *We can compute in polynomial time user strategies that are at equilibrium with only successful flows, if such an equilibrium exists.*

4 Connected equilibria

In this section we study the following question: given an underlying network topology along with a set of demands between nodes, is it possible to assign values to the decision variables, so that the game converges to a connected equilibrium, when such an equilibrium exists?

Recall that we call the network *connected* iff a non-zero amount of every commodity reaches its destination. Therefore, if, in addition to being at equilibrium, we want the network to be connected, we have to add to Theorem 1 the condition that for every commodity (u, v), there is a successful non zero flow sent from u to v through a path P in the network. This translates to the following condition for every edge $e = (x, y)$ in path P: $THR_x(y) \geq \epsilon_x^y = 0$ AND $I^x_{(u,e,v)} > 0$ (especially when $y \neq v$, it must hold $THR_x(y) = \epsilon_x^y = 0$, as follows from condition 2 of Theorem 1).

Theorem 3. *A network is at a* connected *equilibrium if and only if in addition to the Theorem 1 conditions, for every commodity* (u, v)*, either edge* (u, v) *is connected or there is a path connecting* u, v*, so that for every edge* $e = (x, y)$ *in the path it holds that* $I^x_{(u,e,v)} > 0$ AND $\epsilon_x^y = 0$.

It is easy to see that there are cases in which it is impossible for a game to converge to a connected equilibrium. For example, suppose that there is an edge $e = (x, y)$ in the network such that node x is neither a source nor a sink, and there is a commodity (u, v) such that all paths between u and v pass through e. Then it is easy to see that, in any equilibrium, there will be no flow from u to v. Indeed, suppose that there is a connected equilibrium. Hence there should be an edge $e = (t, x)$ in the network which carries some successful flow. If e carries *only* successful flow then the condition 3 of Theorem 1 would be violated. On the other hand if e carries successful *and* unsuccessful flow condition $7(a)$ would be violated since x would have a profit to disconnect edge e and gain in its utility.

As mentioned in the Introduction, the proof of existence, and the computation of strategies that lead to connected equilibria is, in general, very difficult, since we will prove in the next section that it is an NP-hard problem. But, building on the results of the previous section, we can prove the existence (or not) of a connected equilibrium with only successful flows in polynomial time, and compute strategies that achieve it. Using the characterization of such equilibria by Theorem 3, we can reduce this computation to the solution of the following extension of (LP-S):

$$\begin{aligned} & \max \quad w \qquad\qquad \text{s.t.} && \text{(LP-C)} \\ \sum_{P: e \in^* P} x(P) - \sum_{P: e \in^0 P} x(P) &\leq 0 && \forall e \in E \\ \sum_{P \in \mathcal{P}_i} x(P) &\leq d_{(u_i, v_i)} && \forall i \\ \sum_{P \in \mathcal{P}_i} x(P) &\geq w && \forall i \\ x(P) &\geq 0 && \forall P \in \mathcal{P} \\ w &\geq 0 \end{aligned}$$

Similarly to Theorem 2, we can prove the following

Theorem 4. *A* connected *equilibrium with only successful flows exists if and only if* (LP-C) *has a solution* $x(P), w$ *with* $w > 0$.

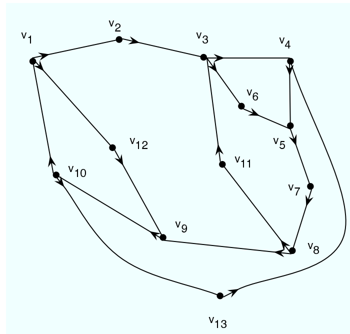

Fig. 1. A variable-subgraph.

Again, the solution of (LP-C) by standard techniques [4] implies the following

Corollary 2. *We can compute in polynomial time user strategies that induce a* connected *equilibrium with only successful flows, if such an equilibrium exists.*

5 NP-hardness of existence of a connected Nash equilibrium

Suppose a network is given together with a set of demands. In this section we prove that it is NP-hard to decide whether there exist values for the decision variables of the nodes so that the game converges to a connected equilibrium (that possibly uses successful and unsuccessful flows). We prove this by reduction from the satisfiability problem (SAT):

Sketch of the reduction: Given an instance I of the SAT problem we construct (in polynomial time on the number of the boolean variables) a network and a set of demands between nodes. The basic element of the construction is the *variable-subgraph* (Figure 1) which corresponds to a boolean variable of I and it is constructed in such a way, so that in any connected Nash equilibrium, in exactly one of its edges there is no successful flow at all. We then show that there is a truth assignment which satisfies an instance I of SAT if and only if there is a Nash equilibrium in the constructed network with the network being connected (i.e., for any demand there is a flow being delivered). We prove this by giving explicitly values to decision variables of the nodes so that the network is connected at a Nash equilibrium. We show that if a boolean variable A has value $FALSE$ in the truth assignment and appears as $\neg A$ in a literal of I (negative literal) then the corresponding subgraph (Figure 2a) is connected at a Nash equilibrium with only successful flows. If variable A has value $TRUE$ in the truth assignment and appears as A in a literal of I (positive literal) then the corresponding subgraph (Figure 2b) is connected at a Nash equilibrium with successful and unsuccessful flows.

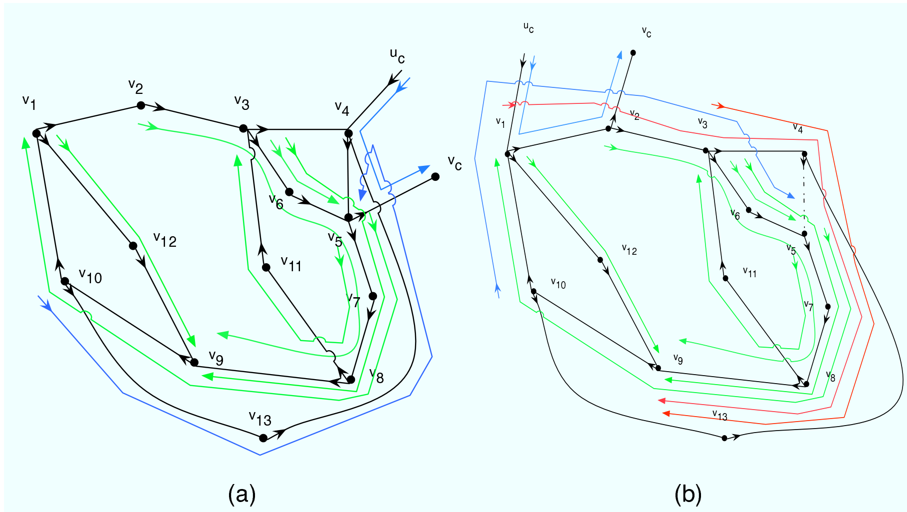

Fig. 2. (a) A negative-literal-subgraph which corresponds to a variable with value $FALSE$, with routed flow-paths. (b) A positive-literal-subgraph which corresponds to a variable with value $TRUE$, with routed flow-paths.

Theorem 5. *Given a network and a set of demands between nodes, it is NP-hard to decide whether there exist values for the decision variables of the nodes so that the game converges to a connected Nash equilibrium.*

The details and proofs appear in the full version of the paper.

6 Conclusion

The question of inducing Nash equilibria with specific attributes is a very general one, and applies to any protocol. In this work we study the property of connectivity, but other natural goals are the maximization of total utility, the maximization of the minimum demand satisfied (similar to concurrent multicommodity flow problems), the maximization of total bandwidth etc. We focused on a basic reputation-based model for ad-hoc networks, but the achievement of most of these goals remains open for this model as well. On the other hand, we were able to characterize the Nash equilibria for it in a way that allowed us to study connectivity properties in a very general setting, i.e., for general topologies and multiple commodities. We would like to combine these properties with additional ones, e.g., maximization of the minimum demand. This would involve network design decisions at the level of setting-up the topology, since there are simple examples with throughput (i.e. the minimum (over all commodities) fraction of satisfied demand) equal to $\frac{d_{min}}{(k-1)d_{max}}$, where d_{min}, d_{max} are the minimum, maximum demands respectively, and k is the number of commodities. Hence, a natural extension of our results would be to study these extra network design decisions when the installation of every new edge incurs a cost. Another natural extension would be the study of a minimal subset of nodes whose setting of initial values induces an equilibrium with the desired properties. Note that in our results we set the initial values for all nodes, thus inducing an equilibrium 'in one shot'.

References

1. S. Buchegger and J.-Y. Le Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes Fairness In Dynamic Ad-hoc NeTworks. In *Proceedings of MOBIHOC02*, pp. 226–236, 2002.
2. L. Buttyan and J.-P. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. In *Proceedings of ACM/Kluwer Mobile Networks and Applications*, vol. 8(5), pp. 579–592, 2003.
3. S. Eidenbenz, G. Resta, and P. Santi. COMMIT: a sender-centric truthful and energy-efficient routing protocol for ad hoc networks with selfish nodes. In *Proceedings of IEEE Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, 2005.
4. M. Grötschel, L. Lovász, and A. Schrijver. Geometric Algorithms and Combinatorial Optimization. Springer Verlag, Berlin 1993.
5. G. Hardin. The Tragedy of the Commons. *Science,* Vol. 162, No. 3859, pp. 1243–1248, December 1968.
6. Q. He, D. Wu, and P. Khosla. SORI: A Secure and Objective Reputation-based Incentive Scheme for Ad-hoc Networks. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC2004)*, pp. 825–830, 2004.
7. R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Sustaining Cooperation in Multihop Wireless Networks. In *Proceedings of Second USENIX Symposium on Networked System Design and Implementation (NSDI'05)*, 2005.
8. F. Milan, J. J. Jaramillo, and R. Srikant. Achieving cooperation in multihop wireless networks of selfish nodes. In *Proceedings of the 2006 workshop on Game theory for communications and networks (GameNets)*, 2006.
9. V. Srinivasan, P. Nuggehalli, C.-F. Chiasserini, and R. Rao. Cooperation in wireless ad-hoc networks. In *Proceedings of IEEE INFOCOM 03*, pp. 808–817, 2003.
10. S. Zhong, J. Chen, and Y. R. Yang. Sprite: A simple, Cheat-proof, Credit-based System for Mobile Ad-hoc Networks. In *Proceedings of IEEE INFOCOM03*, pp. 1987–1997, 2003.
11. S. Zhong, L. E. Li, Y. G. Liu, and Y. R. Yang. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks: an integrated approach using game theoretic and cryptographic techniques. In *Wireless Networks*, Vol. 13(6), pp. 799–816, 2007.