Analysis of a forwarding game without payments

George Karakostas
McMaster University
karakos@mcmaster.edu

Anastasios Viglas
University of Sydney
taso.viglas@sydney.edu.au

***Abstract*—We consider a forwarding game on directed graphs where nodes need to send certain amount of flow (packets) to specific destinations, possibly through several relay nodes. All nodes in the network act selfishly and will forward packets only if it is to their benefit. The model assumes that each node receives some utility from sending it flow to the predetermined destinations and from receiving flow. However each node has to decide whether to relay flow as an intermediate node from other sources, as relaying has an associated cost. This model assumes that there is no payment scheme. Somewhat surprisingly, this game has possibly several strategies that allow a significant amount of the flow to be routed while all nodes have a positive outcome, which suggest that in this model the nodes have indeed incentives to relay flow even if payments are not explicitly allocated. Although previous theoretical work establishes the existence of these strategies (Nash equilibrium solutions), it is not known how often networks have such solutions, and what percentage of flow is actually relayed through the network. In this work we simplify the original network model, and provide the first experimental evaluation of these equilibria for various classes of graphs. We provide clear evidence that these equilibrium solutions are indeed significant and establish how these equilibria depend on various properties of the network such as average degrees and flow demand density.**

I. Introduction

We consider the scenario of a connected network, modelled as a directed unweighted graph. In this network we have a number of designated origin-destination pairs s, t, each associated with a positive parameter $d_{s,t}$. These origin-destination pairs describe the network flow demands in the network: source node s wants to send an amount of flow $d_{s,t}$ to its target node t. Each node might be a designated source (and therefore would like to send flow to specific destinations), or a destination (and therefore would like to receive flow from predetermined source nodes), or, in most cases, it is both a source and a destination. Nodes receive utility from all flow successfully delivered from a source to a destination. All nodes need to pay a cost which is proportional to the amount of traffic they need to transmit. If a source node can communicate directly with its target then of course it is to the benefit of both to have this communication. If however there is no direct connection, then an intermediate node, or several intermediate nodes must be used as relays, or forwarding nodes. These nodes can decide to relay traffic and therefore pay the cost of transmission themselves for someone else's traffic. Although this seems counterintuitive, it has been established very recently that there exist cases where it is the node's overall benefit to relay someone else's traffic., although not necessarily all of the traffic requests. Therefore assuming that each node plays strategically, there exist cases where it is the benefit of every node in the network to relay traffic for others, even though everyone is selfish (tries to maximize its own utility) and there are no payments allocated.

The recent work of Karakostas et al. [1] establishes theoretically that these solutions exist in instances of this network traffic problems, however those results do not give any insight whether such solutions exist often, and whether they are realistic. For example, it is possible that only an infinitesimal fraction of the network instances actually have such solutions. Or it might be the case, that in the only such solutions, only an infinitesimal amount of the original flow finds its way to its destination.

In this paper we provide a conclusive experimental investigation of this model. We show that these solutions are realistic, in the sense that random network instances have non-trivial solutions with high probability, and those flow solutions carry a significant fraction of the total flow. We also establish how these equilibrium solutions depend on various network parameters. Our work includes minor modifications and simplifications to the model used in previous work [1]. Our main contribution is establishing the fact that the equilibrium strategies exist very often, carry a significant amount of flow, and also showing how the strategies are affected by different parameters of the network.

The paper is organized as follows. We start with an overview of related work and background required in section II. We proceed in section III with a formal definition of the network flow problem. We present a simplified version of the model and describe the the theoretical results derives for this simplified model in section IV. Then we give an experimental evaluation of this model in section V, concluding in section VI.

II. Background and related work

In multi-hop networks, selfish behaviour is a frequent and reasonable assumption that captures the behaviour of self-interested entities that need to coexist and possibly cooperate in a common environment. A selfish node in a network will choose an action that maximizes its own utility (or payoff) without any concern about the result of its decisions to the rest of the nodes. Selfish behaviour has been studied using game theoretic techniques in many different areas and problem settings, including wireless ad-hoc multi-hop networks [2]. For a wireless sensor network for example, every node needs to preserve its battery life, as it is usually a scarce resource.

However, if nodes choose to refuse to relay traffic, the network will cease to function. This will lead to no flow being delivered to its destination and all utilities being equal to zero for all nodes[1]. This creates a kind of worst case equilibrium (a standstill) in the network: the strategy of a node not relay for anyone results in their own flow not being relayed. Naturally the following question arises: does there exist a strategy, where nodes do relay flow for others (and therefore do pay the cost for someone else's flow) which which relieve the network from the trivial, no-flow standstill situation mentioned above? Several recent papers [1], [3], [4], [5] show that indeed these strategic solutions do exist for relatively natural network relay models. There are cases where it is to the benefit of *everyone* involved to relay traffic, because this will lead to a better utility outcome for themselves [1]. There are many ways to avoid the trivial solution of zero-relaying, which is a form of the well-known "tragedy of the commons". Payment schemes is a common way to provide incentives to intermediate nodes to relay packets. Reputation-based protocols are based on keeping records of the past actions of neighbors: each node keeps track of the amount of traffic its neighbors has forwarded in the past and follows a specific protocol to decide the amount of traffic it will route in each round. The decisions can be local [2], [6], [7] (each node decides according to its own private information about the past actions of its neighbors) or centralized (a central authority collects all information as a central repository, and decisions are based on the statistics from the entire network) [8], [4].

We focus on the work related to connectivity in such networks based on reputation systems, following the analysis of [1]. The main result we focus on, shows that equilibrium forwarding strategies do exist, without any payment or actual explicit reputation system. In fact the main theoretical results shows that such equilibrium forwarding strategies exist that route a non-zero fraction from *every* source-destination pair. This is a surprising and interesting result that raises many immediate questions about the practical properties of such equilibrium strategies. Note that the result does not guarantee that such equilibrium strategies exist for every network instance. Far from that, there are several simple networks that certainly do not have any such equilibrium strategies except trivial ones (the ones that connect only neighboring source-destination pairs and therefore has no relaying involved). The natural question to ask is how often do these networks have such equilibrium flows, and how significant these solutions are. The theoretical results guarantee that if an equilibrium exists, a non-zero amount of flow is routed for every source-destination pair, however it may be possible that the fraction of the flow routed is insignificant. In this work we answer both these questions, giving positive answers that show the practical importance of this model as well as the already established theoretical one.

Similar Nash equilibrium solutions for relay games are also explored experimentally in by Félegyhźi et al. [9]. The relay game in that work is not using payments or any other explicit incentive for relaying nodes, but it is based on reputation. The game is modelled as a repeated game and conditions for the existence of equilibrium solutions are established theoretically. The authors also present experiments to establish the probability that a random network will indeed allow an equilibrium relay solution. However the experimental results are used mainly to explore particular strategies for the repeated game, or check whether specific conditions hold.

[1]Here we ignore flow demands between source-destination pairs that can communicate directly. These flows will always be successful in our model, and form what we will refer to as a trivial solution, or trivial equilibrium.

III. Definitions

We describe a model here that is significantly simplified compares to that presented previously [1] but still captures an important sub-class of the forwarding game, where only successful flows are routed in the network. We will explain this distinction in more detail further on. Let $G = (V, E)$ be a directed graph, representing a connected network that consists of nodes that are elements of the set V. If nodes nodes $u \in V$ can communicate directly by sending data to node $v \in V$, then there is a directed edge $(u, v) \in E$. The special case were G is undirected is the a reasonable model for wireless ad-hoc networks, when communication links are undirected. We are also given a set of source-destination pairs $(s_i, t_i) \in V$ for $i = 1..k$, and flow demands $d_i \geq 0$ for each pair. We will call each such pair a commodity. The i-th commodity, would like to send its flow demand d_i from the source s_i to the target t_i. Each commodity can choose to split the flow along any number of paths from s_i to t_i. The set of all paths[2] between s_i and t_i is denoted by P_i. The amount of flow that commodity i assigns on edge $e = (u, v)$ is denoted by f_e^i or f_{uv}^i. The general model in [1] allows intermediate nodes to decide to drop a certain amount of flow (decide not to relay). Therefore on a connection $e = (u, v)$ the node v might transmit a certain amount of flow f_e^i, but node u may decide to only retransmit, for example half of it on the next edge towards it destination. In this case there an amount of flow is not successful: it is transmitted by a source node, but it never reaches its destination. One of the theoretical results in [1] is that some networks have equilibrium flow solutions that use only *successful flows*. That is a node never transmits more flow that is actually relayed to its destination. We focus on this particular case for our experiments[3]. Note that the model we define here is different and significantly simplified compared to the one in [1]. However there is no loss in generality of the model for the particular case we are interested in (equilibrium solutions with successful flows only). For every edge $e = (u, v)$ there are two "strategic" parameters associated with the decision of how much flow the

[2]Note that we describe the model using these sets of paths as it is more intuitive and easier to formulate our results. This formulations would be exponential in size and would not be useful in practice, and it is often used in related literature. Later on in this work, we show how to formulate his model in terms of the graph edges so that all formulations are polynomial in size.

[3]Note that there are also equilibrium solutions with unsuccessful flows according to [1] but those are more complex and less practical to work with as it NP-hard to compute them, or to check if they exist.

edge should carry. The receiving node v needs to decide how much flow from this edge, it will be willing to relay further. Note that an edge $e = (u, v)$ will carry "through flow" (flow that needs to be relayed by v to some destination) and "arriving flow" (flow with destination v). The maximum amount of flow of the edge $e = (u, v)$ that v is going to tolerate is denoted by β_e. This means that v is simply not going to relay anything more than this limit. The limit β_e needs to be decided by node v and is one of the main strategic variables in the model. On the same edge, u also has a certain threshold that shows its own tolerance of dropped flow from v. If v is dropping a lot of flow (β_e is too low) then u might decide not to send any flow, by cutting off the edge e. Note that this is an important decision that can hurt v because the edge e also carries flow with destination v. In other words, u will send two kinds of flow to v, flow to relay further, and flow with destination v. If v does not relay enough then u can cut off the edge, and v will lose the flow with destination v it received through that edge. For every edge $e = (u, v)$, the node u has a strategic variable α_e that denotes the minimum amount of through-flow that v is expected to relay. If v relays less then the edge e is automatically cut- off. So, whenever $\beta_e < \alpha_e$ the edge e will be cut off. Cutting off an edge this way is part of the model definition. An edge $e = (u, v)$ can be cut off either by u, by increasing its expectation α_e above v's limit, or it can be cut off by v lowering the flow it relays (reducing β_e). This completes the description of the main parameters and variables of the network.

Now we need to define the utility function for the players (network nodes). The utility function we introduce here is a simple generalization of the one in [1]. A natural way to measure the utility of a node in this model is roughly the following. A node u gets utility from receiving flow (that has destination the node u), and by the fact that flow with origin u actually arrives to its destination. On the other hand, a node u will incur cost (negative utility) whenever it needs to transmit[4] flow (its own or relayed traffic). We define the utility of a node y in the following equation.

$$\begin{aligned} U(v) = w_s \sum_{\substack{e \in out(v) \\ x \in V}} f_e^{vx} + \\ w_r \sum_{\substack{e \in in(v) \\ x \in V}} f_e^{xv} - \\ \sum_{\substack{e \in out(v) \\ x,y \in V}} f_e^{xy} \end{aligned} \tag{1}$$

The parameters $w_s \geq 2$ and $w_r \geq 1$ can be used to model the utility of successful flow, and also the trade-off between successful flow utility and cost of transmission. The parameter w_s is used for the flow that is sent from a node, and w_r is used for the utility of arriving flow. We assume that $w_s \geq 2$ since we need to have some utility from sending flow to a destination after subtracting the cost of transmission (if $w_s = 1$ then it does not make any sense to transmit any flow). The paramters w_s, w_r depend on the application and also other details of the model. Determining values for this parameters can be a difficult task. Our results consider the general case

We have now defined all the ingredients of the this game theoretic model. There is a player for each node, with the utility function defined in equation (1). Each player (node v) needs to decide on its own strategy, which incudes the following variables:

- α_e: The tolerance values for each out going edge $e = (v, x)$. If the target neighbor x on the edge e is not relaying at least α_e then the edge is cut off
- β_e: The drop thresholds for each incoming edge $e = (x, v)$. v will relay at most β_e of flow coming from e.
- f_e^{vx}: The flow assignment that v will route towards all of its assigned targets x.

The strategy profile of a node σ_v contains all of the above parameters: all values α_e for out-going edges, all β_e for incoming edges and all flow assignments f_e^{vx} for all edges in the graph and all destination nodes x.

Each node will choose to relay traffic in a way that maximizes it own utility as defined by equation (1). In order to maximize its utility it needs to pick a strategy profile σ_v wisely. A Nash equilibrium in this network game, is a complete strategy profile for all nodes $\sigma = (\sigma_{v_1}, \sigma_{v_2}, \ldots, \sigma_{v_n})$ such that no node has a unilateral incentive to change its own strategy. In other words, assuming that the nodes are using the strategies in σ, no node v can increase its own utility by changing only its own parameters σ_v in the strategy profile σ_v. This is the standard definition of Nash equilibrium, adapted for the network game we have defined here. We will refer to such a strategy profile as the *Nash equilibrium* or the equilibrium solution.

Now recall that the network instance includes many source-destination pairs, and for each pair there is some amount of demand (maximum amount of flow available to be sent to the destination). In fact every node will be part of some such pair otherwise there is no reason for being part of the network in the first place. It is easy to see that if the source node u of a source-destination pair u, v is connected directly to the target v (there is a directed edge (u, v) in the graph) then it is always beneficial for both nodes for u to send all of its d_{uv} demand to v. Therefore there is a trivial equilibrium solution, where only neighbor demands are routed in the network and no other flow is sent. We will call this the *trivial Nash equilibrium* or simply the trivial solution. Obviously we are interested in non-trivial equilibria and in what follows. We say that a flow assignment is *connected* if it routes a non-zero amount of flow for each commodity. A connected non-trivial Nash equilibrium solution is what we are interested in. In what follows, whenever we refer to an equilibrium solution we mean a connected non-trivial Nash equilibrium solution, unless we explicitly want to make a reference to trivial solutions or commodities with zero flow routed.

[4]We do not explicitly associate cost with receiving flow because this can be easily modelled in the utility function by choosing the weights appropriately.

IV. EXISTENCE OF EQUILIBRIUM SOLUTIONS

Looking at the definition of the model we see that the equilibrium solution depends heavily on the choice of the α and β parameters. A node would rather relay as little through-traffic as possible: through traffic only incurs cost for really node. However if the node starts reducing the amount of flow it is supposed to relay, then incoming edges will eventually be cut off and this will stop the node from receiving flow destined for it and therefore lose utility by that fact. Therefore the decision to relay less traffic needs to be balance with the traffic we expect to receive from each edge. This is precisely the point that is used to characterize the equilibrium solutions in this network game. Following the analysis of [1] we can extend the main theorem regarding successful flows to our network model that has slightly generalized utility function. The main difference in the utility function we introduce here, is the use of the scaling parameters w_s and w_r.

So far we have described the flow assignments in terms of edges. We now switch to path flows as this formulation makes the description of the theorem and flow splits more intuitive. Everything can be described in terms of edges and in fact we do use the edge based formulation in our experiments.

For every source destination pair (u_i, v_i) we denote the set of all possible paths connecting u_i to v_i by P_i. We also denote by $\mathcal{P}$ the set of all relevant paths in the network, $\mathcal{P} = \cup_i P_i$

Recall that each node is essentially trying to optimize the amount of flow it will get to its destination, but it will also need to make sure the assignment it proposes makes sense for the nodes it needs to use as relays. We can indeed write this joint optimization problem as a linear program, and for $w_s = 2$ we get the following.

$$
\begin{aligned}
&maximize && \sum_{P\in\mathcal{P}} f_P && (2)\\
&subject\ to: && && \\
& \sum_{\substack{P\ni e\\ through\\ edge}} f_P - w_r \cdot \sum_{\substack{P\ni e\\ final\\ edge}} f_P \leq 0 && && \forall e \in E\\
& && \sum_{P\in P_i} f_P \leq d_{u_i v_i} && \forall i\\
& && f_P \geq 0 && \forall P \in \mathcal{P}
\end{aligned}
$$

Note that the size of this linear program is exponential in the number of nodes since it is formulated using paths. However we can easily convert this to a linear program written on the network edges that has size polynomial in the size of the network.

We define D to be the total demand in the network between neighboring nodes. A trivial equilibrium solution will route a total flow equal to D. The following theorem describes the non-trivial equilibrium solutions [1].

Theorem 1: A network game has non-trivial equilibrium solutions with only successful flows if and only if the linear program described in (2) has a solution f^*P with objective value $\sum_{P\in\mathcal{P}} f^*_P > D$.

The complete proof of this theorem is analogous to the proof of theorem 2 in [1].

V. EVALUATION

In this section, we present an extensive set of experiments to evaluate the forwarding model presented above. Our main goals are the following:

1) The theoretical results state that equilibria solutions may exist for some networks. How often do these network games actually have such equilibrium solutions? Are these solutions practically significant?
2) These equilibrium solutions route non-zero fraction of the available flow potentially from every commodity. But is this routed fraction significant or very close to zero?

We answer both these basic questions by analysing random families of graphs with randomly chosen source-destination pairs and demands. For these random graphs we solve the linear program (2) and find the equilibrium flow assignments.

Graphs are generated according to the Erdös-Renýi model (G_{nm}), and the Barabasi-Albert powerlaw model. Various edge densities are used. self loops and multi-edges are not allowed. Commodity pairs are chosen uniformly at random. Flow demands are chosen uniformly at random from 1 to a predetermined maximum value. All demands are integers. This ensures that all flows are integral as well. We choose $w_r = 1$ for our experiments. As discussed above, the parameter should be greater or equal to 1 in general. The boundary case of $w_r \geq 1$ means that arriving flow is worth as much as the cost to transmit it.

Given a randomly generated instance (graph plus commodities) we solve the linear program (2) and compare the total routed flow at equilibrium with the best possible solution. The optimal solution is the case that all flow is routed (complete cooperation). Note that trivial flow, which is the flow between source-destination pairs that happen to be neighbors, will always be routed so when we compute the ratio of equilibrium flow versus optimal, we subtract the trivial flow. In order to understand the efficiency of the equilibrium solution, we define the *equilibrium flow ratio*, to be (equilibrium flow - trivial flow) divided by (optimal flow - trivial flow). This ratio will be always between 0 and 1. Our experiments explore the equilibrium flow ratio for varying values of edge density, commodity density, and graph type. The experiments are done on a 16-processor Intel Xeon 2.27GHz machine with 24GB of memory. The linear program is solved by the CPLEX solver, using IBM's OPL Studio. Results are also presented on the running time of the LP solver.

Figure 1 shows two series of histograms. On the left we see a series of histograms that plot the resulting equilibrium flow ratio distribution for 100 different experiments for increasing edge density. On the right we have a similar series of histograms for increasing commodity density. Each histogram also shows the average and standard deviation of the plotted values. We see that for a relatively sparse network with average degree 4 (top left histogram), the equilibrium solution will route just below 14% of the available flow. For a network

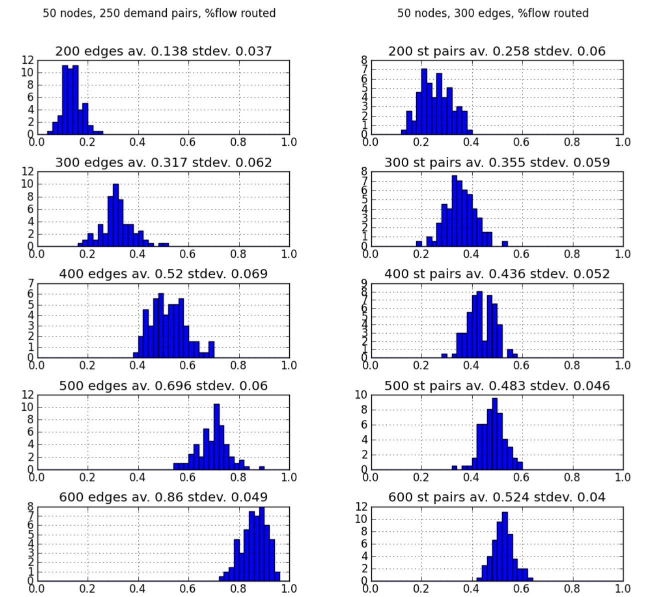

Fig. 1. Percent of available flow routed at equilibrium for Erdös-Renýi graphs as a function of edge density, and commodity density

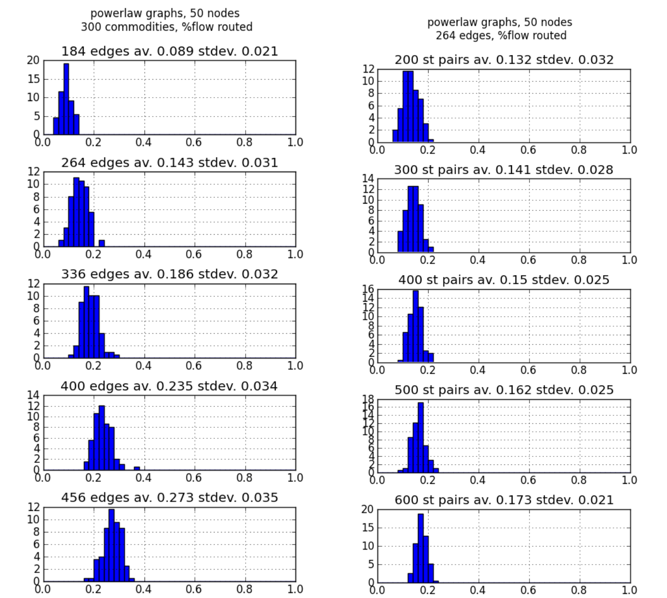

Fig. 2. Percent of available flow routed at equilibrium for powerlaw graphs as a function of edge density (left) and demand density (right).

with average degree 10 however, the equilibrium solution is expected to route close to 70% of the available flow. For average degree 12 the equilibrium flow is over 85% of the total available (bottom left histogram). We see a similar but slightly more modest increase as the commodity density increases. Figure 1 shows that equilibrium flows carry non-negligible flow in the network and they increase significantly when the network becomes more dense, or when the flow demands in the network become more dense. This is an interesting, typically game theoretic result: the more overloaded the network becomes, the more efficient are the equilibrium flows. In other words, the incentives become stronger for nodes to relay when the network is more dense with flow demands, or with possible flow paths. Going back to the description of the equilibrium constraints of this game, we see that essentially an equilibrium flow solution is looking for cases where a node has an incentive to relay flow in order to keep edges open, that carry flow towards that node. In some sense, these open edges are a kind of deal between nodes across an edge $e = (u, v)$: u will only continue to send v flow with destination v (and therefore flow that v wants to receive) provided that v relays enough flow further in the network. Increasing the edge density by adding more edges in the network, increases the probability that there are such deals to be made across edges for each flow demand. If the network is sparse then the existing paths may not bring together nodes whose interests are aligned. The same holds true for increasing density of commodities. The more source-destination pairs we add to the network the more likely it becomes that edges will remain open as the interests of the two nodes on those edges are aligned. Figure 1 also shows that edge density has a stronger impact on the equilibrium flow routed. A dense network is likely to reach a very good level of efficiency, routing almost all available flow.

For powerlaw graphs we see a different picture. We generate powerlaw graphs using the Barabasi-Albert preferential attachment model. Figure 2 shows how the routed equilibrium flow changes when edge density increases and when demand density increases. For similar values of edge density and commodity density, we see that the expected equilibrium flow is significantly lower than what we see in Erdös-Renýi graphs. However we still see that increased density has a positive result (expected amount of flow routed is still increasing) but at a more modest rate. We also observe that increasing edge density has a marginally more positive effect on the flow routed at equilibrium. The equilibrium flows are calculated by solving the linear program from equation (2). The generated linear program has a size that grows fast with the graph size. For a graph with n nodes, m edges, c commodities, the number of constraints is bounded by $O(n \cdot m + m^2 \cdot c)$. This is a generous over-estimate as the expected node degree is smaller than what is used for the calculation of the upper bound (maximum possible degree is $O(m)$, the number of edges in the graph). The expected size of the linear program is $O(n \cdot d + d^2 \cdot c)$ where d is the maximum of the average out-degree and average in-degree. In practice the solve-time can vary greatly with different LP instances. The running time is shown in figure 3 for powerlaw graphs. The results show that the variance increases for as the number of edges (edge density) and the number of commodities increase.

Figure 4 compares the percent of available flow routed at equilibrium as a function of increasing commodity density. The results for Erdös-Renýi (G_{nm}) graphs are on the left side

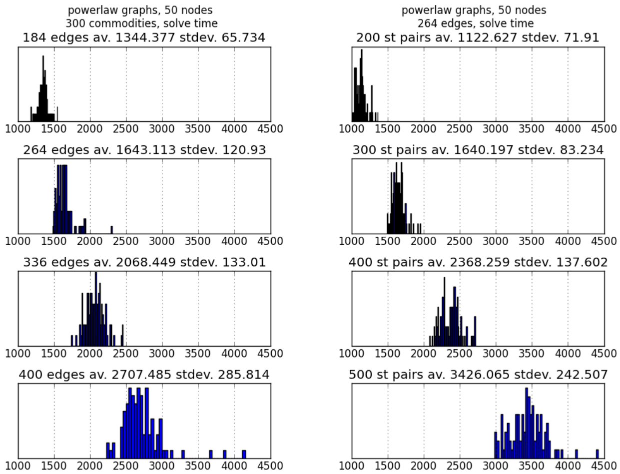

Fig. 3. Solve time for finding the flow routed at equilibrium for powerlaw graphs as a function of edge density (left) and demand density (right).

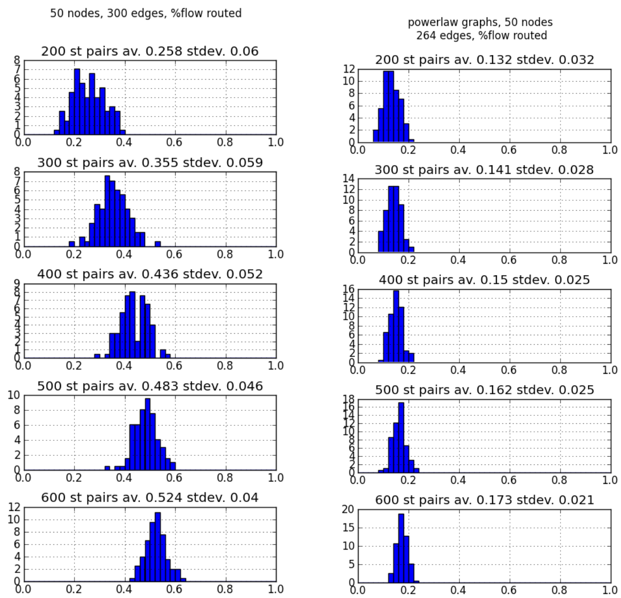

Fig. 4. Percent of available flow routed at equilibrium for Erdös-Renýi graphs and for Barabasi-Albert graphs as a function of commodity density

and the results for powerlaw (Barabasi-Albert) graphs are on the right. We see clearly that the equilibrium solution has a stronger dependence on the available commodity flow pairs for G_{nm} graphs rather than for powerlaw graphs. Powerlaw graphs exhibit a small strongly connected nodes with a long tail of less connected nodes. This makes it easier for the highly connected nodes to route their flow, if both source and destination nodes are in the highly-connected components. For the rest of the nodes (in the loosely connected tail) relaying does not happen as easily. This is consistent with the results for sparse graphs that we have shown above. As the commodity density increases, the size of the highly connected component does not change. Therefore most of the added commodity pairs will naturally fall in the loosely connected tail and therefore will have less chances of finding willing relaying nodes. This results in the effect that we see, where the increasing commodity density does not have a strong impact in the routed traffic at equilibrium.

VI. Conclusion

We considered a natural relaying problem modelled as a game theoretic problem. We focused on a recent game theoretic model that established the existence of equilibrium flows that are based on natural incentives for relaying as opposed to payments. We experimentally established that these equilibrium solutions can carry a significant fraction of the available flow. Our experiments show that the efficiency of the equilibrium solution increases with edge density and demand density in the network. In addition, we give strong evidence that powerlaw networks are less efficient in the sense that the equilibrium solution is expected to route a smaller fraction of the available flow compared to Erdös-Renýi type of graphs with similar characteristics.

References

[1] G. Karakostas and E. Markou, "Emergency Connectivity in Ad-Hoc Networks with Selfish Nodes," *Algorithmica*, no. DOI 10.1007/s00453-012-9675-x, August 2012.

[2] S. Buchegger and J.-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes : Fairness In Dynamic Ad-hoc NeTworks)," in *Components*, ser. Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), vol. 48, no. 3. Citeseer, 2002, pp. 226–236. [Online]. Available: http://portal.acm.org/citation.cfm?id=513828

[3] A. Urpi, M. Bonuccelli, and S. Giordano, "Modelling cooperation in mobile ad hoc networks : a formal description of selfishness," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003. [Online]. Available: http://hal.archives-ouvertes.fr/inria-00466742/

[4] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao, "Cooperation in Wireless Ad Hoc Wireless Networks," in *IEEE Infocom*, 2003, p. 2009.

[5] P. Michiardi and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Conference on Communications and Multimedia Security*, ser. Advanced Communications and Multimedia Security. IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security, B. Jerman-Blazic and T. Klobucar, Eds., vol. IFIP 228, no. December. Kluwer Academic Pub, 2002, pp. 107–121. [Online]. Available: http://portal.acm.org/citation.cfm?id=737297

[6] Q. He, D. Wu, and P. Khosla, "SORI: a secure and objective reputation-based incentive scheme for ad-hoc networks," pp. 825–830 Vol.2, 2004. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1311293

[7] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Sustaining Cooperation in Multi-Hop Wireless Networks," *Symposium A Quarterly Journal In Modern Foreign Literatures*, vol. pages, pp. 231–244, 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1251220

[8] F. Milan, J. J. Jaramillo, and R. Srikant, "Achieving cooperation in multihop wireless networks of selfish nodes," *Proceeding from the 2006 workshop on Game theory for communications and networks GameNets 06*, p. 3, 2006. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1190195.1190197

[9] M. Felegyhazi, J. P. Hubaux, and L. Buttyan, "Nash equilibria of packet forwarding strategies in wireless ad

hoc networks," pp. 463–476, 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1610590