Pervasive and Mobile Computing 7 (2011) 467–478

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Fast track article

Peer-to-peer overlay topology control for mobile ad hoc networks

Afzal Mawji [a], Hossam Hassanein [a,b,*], Xiangyang Zhang [a]

[a] *Telecommunications Research Lab, School of Computing, Queen's University, Kingston, Ontario, Canada, K7L 3N6*
[b] *Department of Computer Science, King Saud University, Saudi Arabia*

A R T I C L E I N F O

Article history:
Available online 28 April 2011

Keywords:
Peer-to-peer networks
Mobile ad hoc networks
Overlay topology control

A B S T R A C T

The enormous popularity of peer-to-peer (P2P) applications and the increased use of mobile devices make running P2P applications on mobile ad hoc networks (MANETs) natural. However, simply applying existing P2P overlay techniques, which are designed for wired networks, to MANETs is undesirable due to the differences between the two types of networks. In P2P systems, peers are often selfish rather than cooperative (which is expected of MANET nodes). In this paper, we study the construction of P2P overlays by selfish peers in the context of MANETs, and propose a computationally feasible game-theoretic heuristic algorithm. In our P2P-MANET creation game, peers seek to maintain as few neighbors as possible while minimizing their distances to all destinations, in an effort to reduce energy consumption and to improve response time. We find that Nash equilibria are difficult to find at best, and may not even exist in most cases. Our heuristic is fairly stable relative to the minimum cost algorithm, and when the degree–constraint is relaxed, it approximates the minimum cost. The lack of global knowledge of the overlay and underlying network at individual peers does not allow each peer to fully exploit such information for its own best interest, and actually may reduce the total cost network-wide.

1. Introduction

P2P applications are enormously popular on the Internet and their uses vary from file sharing to Voice-over-IP to gaming and more. Increasingly, users are moving from wired networks toward mobile networks. Mobile ad hoc networks (MANETs) are essential in some environments, such as in disaster recovery, or military battlefields, where the communications infrastructure has been damaged or is non-existent. In other environments, such as at a company meeting where the participants would like to share documents, such networks may prove more convenient than using a wireless LAN. Recently, the Wi–Fi Alliance has announced a new specification called "Wi–Fi Direct" that allows devices to connect to one another directly, without the use of a base station [1]. The code-name for the specification was "Wi–Fi peer-to-peer". A natural evolution is for P2P applications to run on a cooperative MANET (we use the term P2P–MANET to refer to such networks).

Simply applying existing P2P overlay techniques to MANETs is undesirable due to the lack of infrastructure, node mobility, and energy issues. It is important that the overlay topology reflects the underlying network, both to reduce energy consumption and to improve response times. Nodes in MANETs are mobile, meaning that the underlying topology is constantly changing as users move about. Existing P2P topology control schemes, designed for wired networks, are able to accommodate a changing topology due to the expectation that peers will constantly be joining and quitting. However, nodes are assumed to be disconnected once they are unreachable in non-mobile networks. In MANETs, this assumption is incorrect since nodes may have simply moved to a different location. Mobile devices are powered by battery. The overlay

* Corresponding author at: Telecommunications Research Lab, School of Computing, Queen's University, Kingston, Ontario, Canada, K7L 3N6.
E-mail addresses: mawji@cs.queensu.ca (A. Mawji), hossam@cs.queensu.ca (H. Hassanein), xiang@cs.queensu.ca (X. Zhang).

1574-1192/$ – see front matter
doi:10.1016/j.pmcj.2011.04.003

control algorithm needs to be computationally simple and ensure that MANET nodes forward as less traffic as possible. On the one hand, it is preferable for peers to have a small degree on the overlay to reduce control traffic. On the other hand, it is preferable for the overlay to be well-connected so the traffic between two overlay peers traverses as few MANET nodes as possible. Because of the lack of infrastructure, a fully distributed overlay topology control algorithm is preferred.

Most P2P overlay topology control algorithms assume that peers are cooperative. Unfortunately, peers are selfish in many cases. They seek to minimize their own costs, in the case of MANETs, to minimize the number of links to other peers and the distance to all destinations. Several studies [2–4] investigate the impact of selfish peers on the topology in the context of non-mobile networks. However, they study the theoretical bounds or require peers to have global knowledge to construct the overlay. No practical overlay topology control algorithm, even for non-mobile networks, exists. As well, a study of the impact of selfish peers in the context MANETs is needed.

In this paper, we study the construction of the P2P overlay by selfish peers in the context of MANETs, and propose a fully decentralized, computationally feasible, heuristic algorithm based on game theory. In our P2P–MANET creation game, peers seek to maintain as few neighbors as possible while minimizing their distances to all destinations. We compare the topology of our heuristic to that of random local search algorithms. We find that Nash equilibria are difficult to find or may not even exist. Minimum cost topologies are typically highly connected and thus resilient, but are not in a Nash equilibrium state. Our heuristic is fairly stable relative to the minimum cost algorithm, and when the degree–constraint is relaxed, it approximates the minimum cost. The lack of global knowledge of the overlay and underlying network at individual peers does not allow each peer to fully exploit such information for its own best interest, and actually may reduce the total cost network-wide.

The remainder of the paper is organized as follows. Section 2 provides related work on P2P overlay topology control algorithms. Section 3 discusses the game-theoretic heuristic overlay topology control algorithms in detail. In Section 4 we compare the topologies formed by the heuristic and benchmark algorithms. Section 5 provides some conclusions.

2. Related work

P2P overlays may be separated into two classes: *structured* and *unstructured* [5]. Hora et al. [6] have found that in MANETs, unstructured protocols are more resilient than structured ones at the cost of higher energy and delay. Liu [7] showed that in an unstructured overlay, more than 70% of the links do not reflect the underlay topology due to the selection of random neighbors.

Fabrikant et al. [2] introduce a network creation game to model the construction of the Internet, which has inspired our overlay topology control algorithm. In the game, nodes are required to pay for the links that they establish to others and benefit from shorter distances to all destinations. Nodes are selfish, trying to have fewer direct neighbors while keeping the entire network as close as possible. The cost to establish a link is constant; the distance metric is the hop count. A system parameter, α, catches the trade-off between the cost and benefit. The authors show that determining the Nash equilibrium is an NP-hard problem and give an upper and lower bound of the "price of anarchy".

Moscibroda et al. [3] extend the work of Fabrikant et al., examining the effect of the network creation game on P2P topologies. The hop count is replaced by the *stretch* as the distance metric and overlay links are directional; otherwise the model remains the same. The authors provide an upper and lower bound on the price of anarchy. They go on to prove that there exist cases for which no Nash equilibrium exists, meaning that even in the absence of peer churn, the topology will never stabilize.

Chun et al. [4] simulate a P2P overlay creation game using Fabrikant et al.'s model. The cost to each neighbor may be different (e.g., congested links have a higher cost), and the number of neighbors is bounded. The authors use a heuristic random local search algorithm to calculate the Nash equilibrium graphs. They show that without constraining the node degree, the resulting topologies are near-star configurations, in which a small number of nodes will maintain a large number of connections. This allows other nodes to "free ride" off of those nodes by connecting to them and achieving both a small number of connections and also a low distance metric. This results in less resilient topologies because a small number of node failures will result in a large degree of disconnection. Therefore, the authors conclude that there is a fundamental trade-off between performance and resilience.

Our work differs from all the above in that we study the construction of P2P overlays by selfish nodes on an infrastructureless MANET and propose a computationally feasible heuristic algorithm that accommodates node mobility and peer churn and does not require peers to have global knowledge. By contrast, [2,3] analytically study the upper and lower bounds of the price of anarchy, and [4] uses a heuristic algorithm, which requires peers to have global knowledge, to simulate the construction of P2P overlay on wired networks.

3. P2P–MANET overlay topology control

Given a MANET and a subset of nodes that are participating in the P2P overlay, the problem of finding an optimal topology can be formulated in game-theoretic fashion using a network creation game. In this section, we first introduce the P2P–MANET creation game, then describe two random local search algorithms that will be used as the benchmark, and finally introduce the computational feasible heuristic.

3.1. A P2P–MANET creation game

The network creation game has n players, $P = \{p_0, p_1, \ldots, p_{n-1}\}$, each being a member of the P2P overlay. The strategy space of player p_i is the set S_i; the strategy chosen by p_i, $s_i \in S_i$, is a subset of $P \setminus p_i$. If $p_j \in s_i$ then peer i is establishing a link to peer j in the strategy s_i. Since the link is undirected, peer j also establishes a link to peer i. Otherwise, no link is established between i and j in the graph. The combination of the strategies $s = (s_0, s_1, \ldots, s_{n-1}) \in S_0 \times \cdots \times S_{n-1}$ form the undirected graph $G[s] = (P, \bigcup_{i=0}^{n-1}(p_i \times s_i))$.

We define $s^{-p_i,p_j} = (s_0, s_1, \ldots, s_i \setminus p_j, \ldots, s_j \setminus p_i, \ldots, s_{n-1}) \in S_0 \times \cdots \times S_{n-1}$ as the strategy with the link between p_i and p_j removed from s. We also define $s^{+p_i,p_j} = (s_0, s_1, \ldots, s_i \cup p_j, \ldots, s_j \cup p_i, \ldots, s_{n-1}) \in S_0 \times \cdots \times S_{n-1}$ as the strategy with the link between p_i and p_j added to s. We define a maximum degree, $\deg_{\max}$, associated with each peer. For ease of description, we assume peers are homogeneous (in terms of computation, communication, and energy level); our algorithms can be extended to the heterogeneous case.

The cost to each peer has two components. The first component incorporates the energy level of the neighboring peers, since it is preferable to connect to peers which have longer lifetimes. The second component is the sum of distances from the peer to all others. A parameter, α, is used to capture the trade-off between establishing a link and the change in distances to peers caused by the establishment of that link. We adopt a notion of distance commonly used in P2P networks, referred to as the *stretch*. We define the stretch as the ratio between the total number of physical hops traversed and the shortest physical distance possible. A peer has the minimum stretch of 1 to its overlay neighbors. This definition is used because in MANETs all traffic must ultimately be sent via the wireless links, in a hop-by-hop fashion. For a single peer, shorter distances to all destinations reduce response time and improve reliability, since compared with a wired network, MANETs have a large delay and are less reliable. Shorter distances also reduce network-wide energy usage. Specifically, the cost to peer i of strategy s is

$$C_i(s) = \alpha \sum_{j \in N_i} e_j + \sum_{j=0}^{n-1} d_{G[s]}(i, j) \tag{1}$$

where N_i is the set of neighbors of peer i, e_j is the complimentary energy level of peer j, and $d_{G[s]}(i, j)$ is the stretch from peer i to peer j in graph $G[s]$. We assume peers are fully charged to their maximum energy level at the beginning, so e_j equals the energy peer j has consumed so far. For convenience, the consumed and residual energy is expressed as a fraction of the maximum energy level.

The total cost is the sum of all peer costs

$$C(G[s]) = \sum_{i=0}^{n-1} C_i(s). \tag{2}$$

Given Eq. (2), the lowest possible total cost results in a global optimal minimum cost topology. However, this solution may not satisfy a Nash equilibrium. Each peer is attempting to maximize its own payoff, which in this case is given by Eq. (1). A pure Nash equilibrium of this game is a strategy s such that for each player i and for all s' that differ from s only in the ith component, $c_i(s') \geq c_i(s)$. That is, no player has an incentive to unilaterally deviate from its selected strategy since this would increase its cost. This strategy is known as the *best response*. The difference between the minimum cost solution and the Nash equilibrium is referred to as the *price of anarchy*.

3.2. Benchmark algorithms

Finding both the minimum cost solution and the Nash equilibrium of the network creation game have been shown to be NP-hard [2]. Random local search algorithms provide the closest possible approximation to both problems. Therefore, we make use of two random local search algorithms to examine the solution space and use the results as the benchmark. The random local search algorithm requires that peers have *a priori* global knowledge of all MANET nodes, their energy level, and the overlay and underlay distances between them. Peers' degrees on the overlay are not bounded. In each round, the algorithms explore a potential solution within the solution space. The algorithm to find a global optimum has a complexity of $O(n^3)$ in each round; the algorithm for each peer to find Nash equilibrium has a complexity of $O(n^2)$. To examine an acceptable portion of the solution space, random local search algorithms need to run a long time.

Our random local search algorithm to find the minimum cost operates as follows (see Algorithm 1)). We start from a random topology and note its cost. Next, a random peer is selected, and its connection to each of the other peers is examined in turn. If the connection does not exist, the total cost of the entire topology is determined as if it were connected. If this results in a decrease in the total cost, then the link is established. If the connection exists, the total cost of the entire topology is determined as if it were not connected. If this results in a decrease in the total cost, then the link is disconnected. When the change in cost results in only a small difference, ϵ, a new random peer is selected and the process is started anew. This continues until the maximum time allowed, $T_{\max}$, has elapsed. When the algorithm terminates, the graph $G[s]$ is the overlay topology of the lowest cost graph found, and its cost is $C(G[s])$.

Our random local search algorithm for determining the Nash equilibrium is similar, and is given in Algorithm 2. Again, an initial, random topology is created as a starting point. Every neighbor of every peer in the network is now examined in turn. For a given peer, p_i, the effect of disconnecting each established link on the cost to p_i, and p_i alone, is observed. If p_i's

Algorithm 1 Random local search to find the minimum cost

```
G[s] ← random topology
C_min ← C(G[s])
while t < T_max do
  select random peer, p_i
  for each p_j ≠ p_i do
    if p_j ∈ N_i then
      if |C_min − C(G[s^{−p_i,p_j}])| < ϵ then
        break
      else if C(G[s^{−p_i,p_j}]) < C_min then
        N_i ← N_i \ p_j
        C_min ← C(G[s^{−p_i,p_j}])
      end if
    else
      if |C_min − C(G[s^{+p_i,p_j}])| < ϵ then
        break
      else if C(G[s^{+p_i,p_j}]) < C_min then
        N_i ← N_i ∪ p_j
        C_min ← C(G[s^{+p_i,p_j}])
      end if
    end if
  end for
  increment t
end while
```

cost decreases, the link is severed. Next, a random peer $p_k \notin N_i$ is selected. p_i then determines its cost if it were to establish a connection to p_k. If the cost decreases, the link is established. This process is repeated for every peer in the overlay until no peer changes any of its links or the maximum time, $T_{\max}$ has been exceeded. In the former case, the resulting topology is a Nash equilibrium.

Algorithm 2 Random local search to find the Nash equilibrium

```
G[s] ← random topology
C_i^{Nash} ← C_i(s)
while t < T_max do
  unchanged ← true
  for each p_i do
    for each p_j ≠ p_i do
      if p_j ∈ N_i then
        if C_i(s^{−p_i,p_j}) < C_i^{Nash} then
          N_i ← N_i \ p_j
          C_i^{Nash} ← C_i(s^{−p_i,p_j})
          unchanged ← false
        end if
      end if
    end for
    select a random p_k ∉ N_i
    if C_i(s^{+p_i,p_k}) < C_i^{Nash} then
      N_i ← N_i ∪ p_k
      C_i^{Nash} ← C_i(s^{+p_i,p_k})
      unchanged ← false
    end if
  end for
  if unchanged = true then
    break
  end if
  increment t
end while
```

We now discuss the complexity of determining the cost of the topology of n peers. To determine the cost for a particular peer p_i, each of the remaining $n - 1$ destinations must be examined. Determining the energy of a peer requires a simple lookup. However, determining the stretch of the resulting graph requires an examination of the overlay hop distance to each of the $n-1$ possible destinations. For each destination p_j, the minimum overlay distance must be determined. This requires that each possible path from p_i to p_j be examined using the remaining $n-2$ peers as a potential first hop. Therefore, determining the total stretch of the resulting distance graph requires $O(n^3)$ time. In the case of the Nash equilibrium, the time complexity is $O(n^2)$ for each peer.

3.3. A topology control heuristic algorithm

We now propose a heuristic algorithm which approximates the optimal solution. It allows the topology to be built in steps and accommodate node mobility and peer churn, and does not require global knowledge of the underlay or overlay. The algorithm has a complexity of $O(n)$. A peer's degree on the overlay is bounded by $\deg_{\max}$.

The heuristic algorithm presented utilizes a two-phase mechanism for each peer and uses a slightly modified cost function. In the first phase, a MANET node chooses the initial neighbors it wishes to connect to. From the node's point of view, this is the culmination of the bootstrapping process. Once successfully joined to the network, it actively maintains its links, changing them as needed to decrease its costs. This second phase performs the topology maintenance.

The cost function used in the heuristic is similar to Eq. (1), except that only the distance to a subset of P is determined

$$C_i^{\mathrm{heur}}(s) = \alpha \sum_{j \in N_i} e_j + \sum_{j \in H_i} d_{G[s]}(i,j) \tag{3}$$

where N_i is the set of neighbors of peer i, e_j is the complimentary energy level of peer j, $d_{G[s]}(i,j)$ is the stretch from peer i to peer j in graph $G[s]$, and H_i is a subset of P referred to as a *hot list*. The hot list consists of the $\deg_{\max}$ peers that p_i sends and receives the most messages to and from, and will be explained further in this section.

When a node n_i wishes to join the overlay, it determines which peers it would prefer to connect to. The set of best peers for a node to join is determined by the Cobb–Douglas utility function [8]

$$P = h^{\beta} \times e^{1-\beta} \tag{4}$$

where h is the hop distance of the MANET node in the substrate network, and e is the complimentary energy level of the potential neighbor. Both h and e are normalized, and β is a parameter indicating the preference of a nearer peer to a longer-lived one. When $\beta = 0$, this indicates that the node wishes to select the peer with the most remaining energy regardless of its distance. This may result in an overlay topology which does not closely reflect the underlying network topology, however the connection to the peer is likely to remain for a longer period of time. On the other hand, when $\beta = 1$, this indicates that the node wants a neighbor that is closer, no matter what its energy level is. n_i cannot use Eq. (3) because it does not know how the topology is connected and therefore cannot determine a value for the stretch.

If the node is the first in the overlay, it is bootstrapped as the sole peer. Nodes that join later proceed as follows. When a node n_i wishes to join an overlay network, it must connect to at least one overlay member, and preferably several more. The utility function (Eq. (4)) is applied to all known peers in the overlay, and the list is sorted. n_i then contacts the $\deg_{\max}$ top peers and requests a connection with them, one by one. When a peer p_j is contacted by n_i, it evaluates Eq. (3) and will agree to connect to n_i only when the resulting cost is less than or equal to its current cost. Peers are associated with a maximum degree, $\deg_{\max}$, so if p_j already has $\deg_{\max}$ neighbors, it will also reject n_i's request.

The very first peer p_j that n_i contacts will always agree to the connection request, provided it does not result in exceeding $\deg_{\max}$. This is necessary because n_i will not be on any peer's hot list as it is joining and will always result in a greater cost to p_j. Therefore, to successfully bootstrap n_i, one peer must agree to allow n_i to connect to it. n_i makes the connection requests one by one because this provides an accurate picture of its connectivity to the potential neighbors. Once it has contacted the top $\deg_{\max}$ peers, n_i, now p_i, has successfully bootstrapped itself into the overlay if it has at least one neighbor. At this point, the second phase of the heuristic algorithm, topology maintenance, begins for p_i.

For the lifetime of the overlay, peers verify a link's status via a mechanism similar to Gnutella's "ping–pong" technique. During this process, p_i sends information to all $p_j \in N_i$ to inform them of the remaining energy level and stretch to all other peers p_i knows of. p_i may also send these messages to its neighbors when it learns of new information, such as a change in link status or a new peer that has joined the overlay.

Given the new information received by p_i from its neighbors, p_i will execute Algorithm 3 in an attempt to lower its cost. This algorithm serves to provide topology maintenance, and works as follows: for each $p_j \in N_i$, p_i examines the change to its cost if it were to drop p_j as a neighbor. If the cost decreases, the link is severed, otherwise it is maintained.

p_i also maintains a hot list, which contains information regarding how much data has been transferred between itself and every other peer in the last epoch. The *epoch* is defined as the time period between ping–pong messages. This list is sorted such that the peer with which p_i has sent and received the most messages, labeled as h_0, appears at the top, and the peer with which p_i has communicated with the least, labeled as h_{n-1} appears at the bottom. The list is then truncated to size $\deg_{\max}$.

If $h_j \notin N_i$, p_i will determine its cost if h_j is added to N_i. If p_i's cost drops, then h_j is added to N_i, providing $\deg_i < \deg_{\max}$. One possible downside to focusing on the hot list for potential neighbors is that it may result in cliques which are poorly connected to one another. Therefore, p_i will also attempt to connect to the nearest peers by underlay hop distance which are not already its neighbors. If connecting to $p_m \notin N_i$ such that p_m is within η underlay hops results in a reduced overlay cost to p_i, and does not cause $\deg_{\max}$ to be exceeded, p_m is added to N_i.

Finally, to allow the chance discovery of better neighbors, p_i will select a random peer p_r that it has not yet considered in the process already described, and evaluate the cost of connecting to it. Once again, if it results in a reduced cost, p_r is added to N_i. This process allows the topology to be dynamically updated with the most recent information, providing resilience to peer churn and node mobility, and also allows p_i to focus on the peers it communicates with most.

Algorithm 3 Overlay topology maintenance

Require: $C_i^{\min} \leftarrow$ current cost for p_i
  sort and truncate p_i's hot list
  for each $p_j \in N_i$ **do**
    if $C_i(s^{-p_i,p_j}) < C_i^{\min}$ **then**
      $N_i \leftarrow N_i \setminus p_j$
      $C_i^{\min} \leftarrow C_i(s^{-p_i,p_j})$
    end if
  end for
  for each $h_j \notin N_i$ **do**
    if $C_i(s^{+p_i,h_j}) < C_i^{\min}$ and $\deg_i < \deg_{\max}$ **then**
      $N_i \leftarrow N_i \cup h_j$
      $C_i^{\min} \leftarrow C_i(s^{+p_i,h_j})$
    end if
  end for
  for each $p_m \notin N_i$ within η hops of p_i **do**
    if $C_i(s^{+p_i,p_m}) < C_i^{\min}$ and $\deg_i < \deg_{\max}$ **then**
      $N_i \leftarrow N_i \cup p_m$
      $C_i^{\min} \leftarrow C_i(s^{+p_i,p_m})$
    end if
  end for
  select random $p_r \notin N_i$
  if $C_i(s^{+p_i,p_r}) < C_i^{\min}$ and $\deg_i < \deg_{\max}$ **then**
    $N_i \leftarrow N_i \cup p_r$
    $C_i^{\min} \leftarrow C_i(s^{+p_i,p_r})$
  end if

The heuristic algorithm uses Eq. (3), which calculates the cost to the top $\deg_{\max}$ peers of the hot list only, for each peer. This results in a reduced time complexity of $O(n \times \deg_{\max})$ to determine the stretch. The messages exchanged by the topology control heuristic include connection and disconnection messages, as well as periodic ping–pong messages to verify the link status. These messages piggyback information regarding energy status and overlay distances to other peers. These distances, in conjunction with information determined by the routing protocol, are used to compute the stretch.

4. Performance evaluation

We now evaluate the performance of our proposed heuristic overlay topology control algorithm, and compare its performance to that of the benchmark algorithms described in Section 3.2. The heuristic algorithm was implemented in the network simulator *ns-2* 2.33 [9]. The underlay topologies resulting from the simulation were then fed into the benchmark algorithms' solver, which are written in the C++ programming language, to determine the minimum cost and Nash equilibria. We first discuss the simulation model, followed by a discussion and comparison of the results.

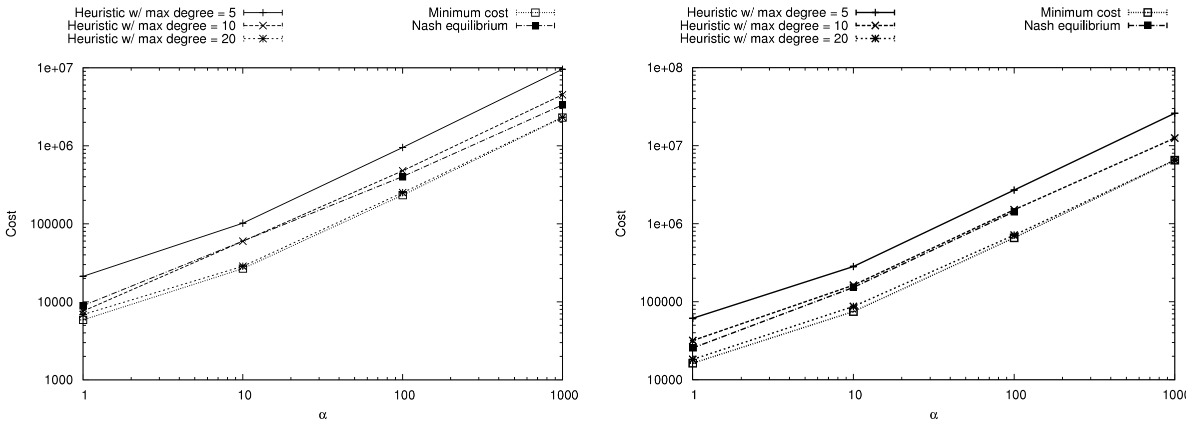
4.1. Simulation model

In our simulations, we use 100 MANET nodes, with the number of nodes participating in a P2P overlay varying from 50 to 100 in increments of 10. The network area is 1500 m × 1500 m, the transmission rate is 54 Mbps, and the communication range is 240 m by default. The two-ray ground radio propagation model along with an omnidirectional antenna are used by all nodes. We use the random waypoint mobility model, with all nodes evenly distributed in the simulation area. Nodal velocities are distributed according to a uniform distribution, with a minimum 1 m/s and a maximum 3 m/s, and a uniformly distributed pause time with mean 60 s, thus mimicking a moderate walking pace with infrequent stops. The AODV [10] routing protocol is used for unicast routing.

The energy consumption model used in the simulations is the linear model proposed by Feeney [11]. Each MAC layer operation takes a certain amount of power as defined by cost $= m \times \text{size} + b$ where m is the incremental cost of the operation, b is the fixed cost, and size is the amount of data sent or received. The constants are obtained by physical measurements for a Lucent IEEE 802.11 WaveLAN PC Card from [11] and are summarized in Table 1.

Table 1
Energy consumption constants used in simulations.

m_{send}	1.89 μWs/byte
b_{send}	246 μWs
m_{recv}	0.494 μWs/byte
b_{recv}	56.1 μWs
$b_{sendctl}$	120 μWs
$b_{recvctl}$	29.0 μWs



(a) 60 peers. (b) 100 peers.

Fig. 1. Total cost.

The heuristic experiments run for two simulation hours. Constant bit rate (CBR) traffic of 1000 byte packets sent 100 times per second was sent between members of the P2P overlay. The traffic started and stopped at random times, between random peers. This created overlay traffic, which was used to determine peer energy levels, as well as the hot list information. Peer energy levels were initially set to a maximum value of 100,000 J, which decreased as the CBR traffic flowed. The epoch value was set to 60 s. The α parameter was set to the values 1, 10, 100, and 1000 to determine its effects.

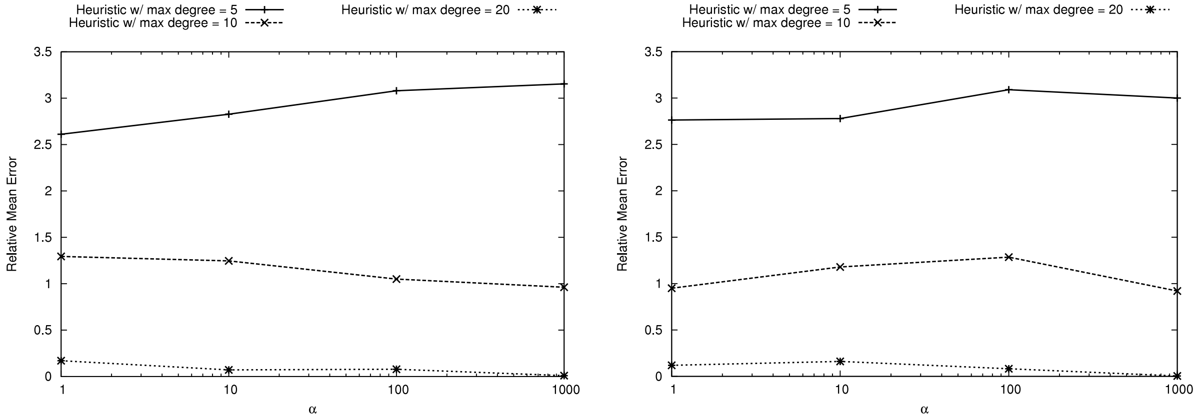
The input topology to the solver is the underlay topology at the end of the simulation along with the resulting energy values. We perform a random local search of the solution space using Algorithms 1 and 2. An initial random point of the solution space is examined, and we proceed to walk to progressively better results in the area as discussed in Section 3.2. When the difference between successive results is less than 1%, we jump to a new random point of the solution space and begin walking again. Random local search algorithms require a long running time to examine a large solution space. We run the solver for a maximum period of 24 h to determine the lowest cost, and another maximum period of 24 h to determine the Nash equilibrium. The minimum cost and Nash equilibria are not constrained by a maximum degree in the solver, i.e., $\deg_{max} = n$.

4.2. Simulation results

In this section, we present the results of our heuristic algorithm, with values of 5, 10, and 20 for $\deg_{max}$. The simulation results obtained in all experiments in this chapter have a 95% confidence level based on 10 independent runs. Though overlay sizes of 50, 60, 70, 80, 90, and 100 nodes were tested, due to the similarity in the results and space constraints, only the values for 60 and 100 overlay peers are given below. We compare the results to that of the solver, including both the minimum cost and Nash equilibrium, where it could be obtained.

The total cost, $C(G[s])$ for various values of α and different overlay sizes are provided in Fig. 1. The figures show that the heuristic algorithm with $\deg_{max} = 20$ performs very well relative to the benchmark algorithm for minimum cost; the cost is even smaller than the Nash equilibrium when peers have global knowledge. This shows that the lack of global knowledge of peers actually reduces the network-wide cost. As the value of $\deg_{max}$ decreases, the results move further away from the optimum value. Fig. 2 provides the relative mean error (RME) for the heuristic case as compared to the minimum cost. It can be seen that the RME tends to be fairly stable over all values of α and overlay sizes, indicating a constant difference in cost. This shows that our heuristic algorithm performs rather stably for various α and overlay sizes, and as the degree constraint is relaxed, approximates the minimum cost topology. Note that the total cost of our heuristic shown in the figures is calculated over the cost of the entire overlay, not just the hot list, in order to make for a fair comparison.

In more than half the experiments shown, no Nash equilibria could be found within the time constraints. Even in those cases where Nash equilibria were found, no more than 6 out of the 10 test runs were successful in determining the equilibrium value. This indicates that finding the Nash equilibrium is difficult at best, and may not even exist in most cases.



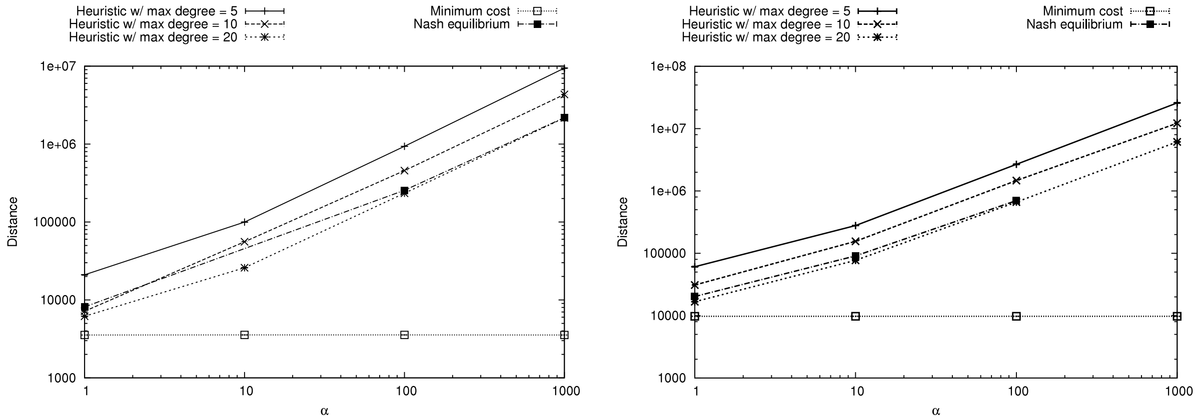
(a) 60 peers.

(b) 100 peers.

Fig. 2. Relative mean error.

Table 2
The price of anarchy.

$n = 60, \alpha = 1$	1.35
$n = 60, \alpha = 100$	1.29
$n = 60, \alpha = 1000$	1.08
$n = 70, \alpha = 1$	1.88
$n = 70, \alpha = 100$	1.31
$n = 80, \alpha = 1$	1.64
$n = 80, \alpha = 100$	1.62
$n = 90, \alpha = 1$	1.47
$n = 100, \alpha = 1$	1.37
$n = 100, \alpha = 10$	1.58
$n = 100, \alpha = 100$	1.63



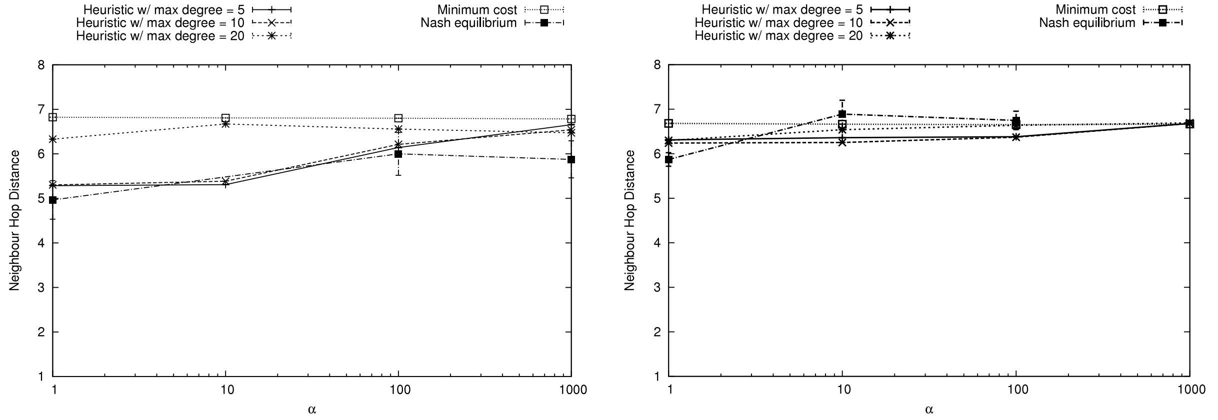
(a) 60 peers.

(b) 100 peers.

Fig. 3. Total stretch.

The absence of Nash equilibria may not be a major issue for a P2P–MANET in which peers are both mobile and churning. For those cases where a Nash equilibrium was found, it tends to be closest to the $\deg_{max} = 10$ case (see Fig. 1). This indicates that the minimum cost result does not converge to a stable topology. Table 2 provides the price of anarchy for those cases where a Nash equilibrium could be found. The price of anarchy was found to be between 1 and 2, indicating that the stable topology may be as much as twice the cost of the minimum cost one. The values in Table 2 have no discernible trend, this result, together with the fact that Nash equilibria were found in less than half the experiments, show that a random search of the solution space for Nash equilibrium will produce unpredictable results.

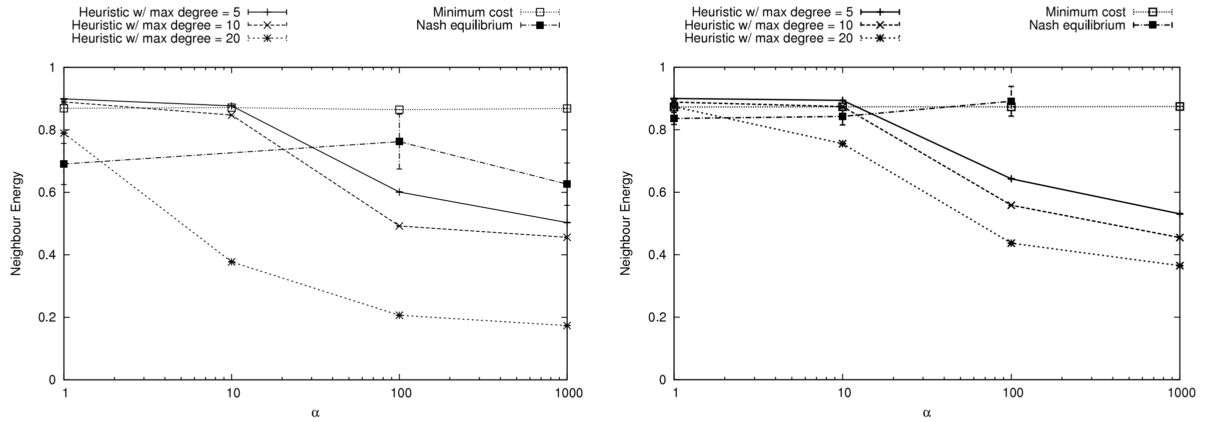
Fig. 3 shows the total stretch values obtained. This is a component of the total cost figure, but is worth examining on its own as well. As α increases, the cost of maintaining more links increases, and so the heuristic algorithm begins reducing the number of neighbors (see Fig. 6). This results in the stretch increasing because more hops are needed on average.



(a) 60 peers.

(b) 100 peers.

Fig. 4. Average neighbor hop distance.



(a) 60 peers.

(b) 100 peers.

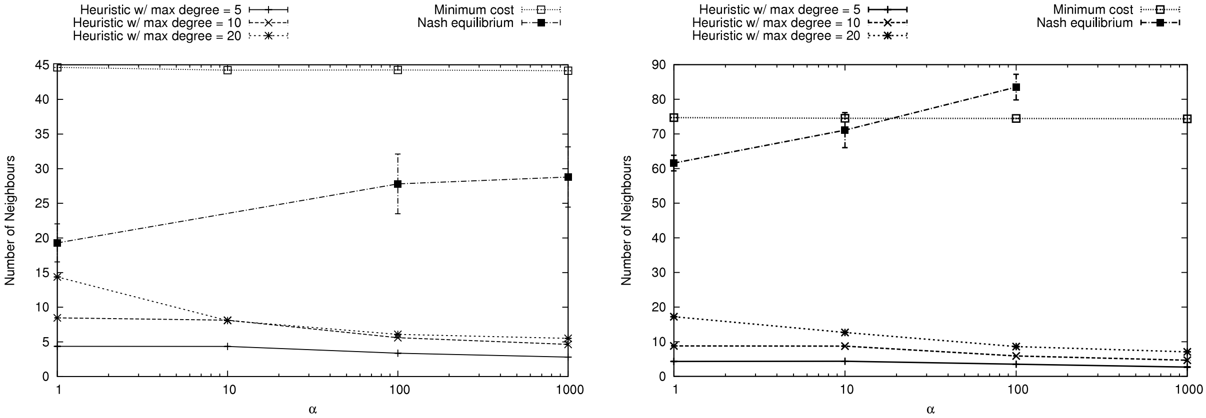
Fig. 5. Average neighbor energy.

This analysis also applies to the Nash equilibrium, though it tends to keep more neighbors than the heuristic algorithm, resulting in a stretch that is closest to $\deg_{\max} = 20$. Interestingly, the minimum cost topology maintains a very large number of neighbors, even though the cost increases. This results in the stretch value not changing as α increases. In fact, the stretch of the minimum cost is very close to the minimum possible stretch. This results in a very highly interconnected network, in which most peer-pairs are connected.

Fig. 4 provides the underlay hop distance a peer's neighbors are from it, averaged across all the peers. Lower values mean that the peer has selected closer nodes, which indicates that the overlay is similar in topology to the underlay. The heuristic algorithm tends to be connected to closer neighbors since it is constrained by $\deg_{\max}$. Since it is trying to minimize the cost to its hot list, the neighbors it keeps are those that keep the distance to the hot list smallest, and it tends to avoid nearer peers that it does not communicate with. As the overlay size increases, the differences between the various algorithms decreases, and the α value has less effect. With more overlay peers, hot list peers come from more locations around the network, and the average distance to them increases. The minimum cost algorithm connects to nearly all peers, resulting in higher overall neighbor distances, for all overlay sizes and values of α.

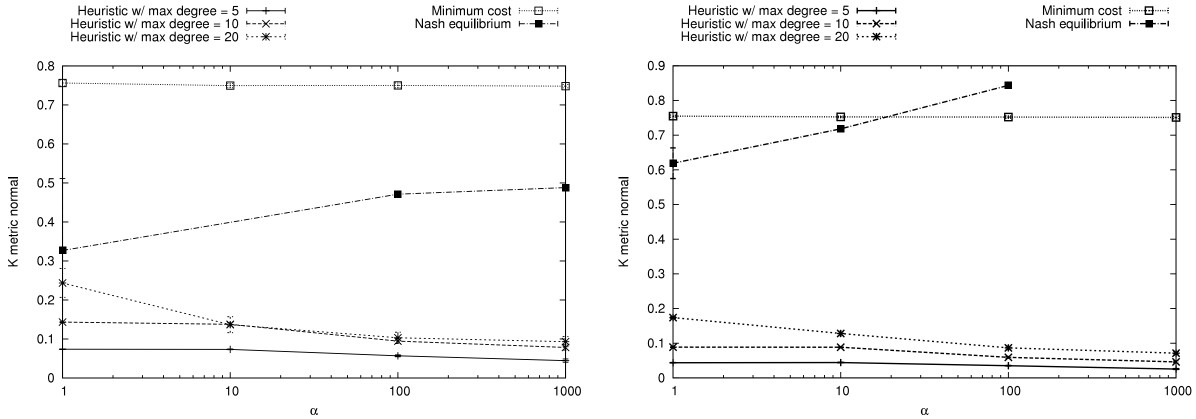
Fig. 5 provides the average normalized energy consumed for each peer's neighbors. Lower values mean that the peer is connected to peers with more energy remaining. The heuristic algorithm tends to have longer-lived neighbors, mainly because the minimum cost algorithm connects to nearly all peers. As α increases, the minimum cost algorithm does not change, while the heuristic tends to connect to fewer, but longer-lived neighbors. This happens because when α increases, it becomes more costly to maintain neighbors, so the heuristic tries to maintain connections to only the most important peers, as defined by the hot list. As the overlay size increases, the energy consumed by each node increases on average due to the larger amount of network traffic. As a result, the average neighbor energy consumed rises with increasing overlay size.

Fig. 6 shows how many neighbors each peer has on average. As evidenced in previous figures, the minimum cost algorithm tends to maintain a very high number of neighbors, regardless of α. The Nash equilibrium figures available show that it



(a) 60 peers. (b) 100 peers.

Fig. 6. Average number of neighbors.



(a) 60 peers. (b) 100 peers.

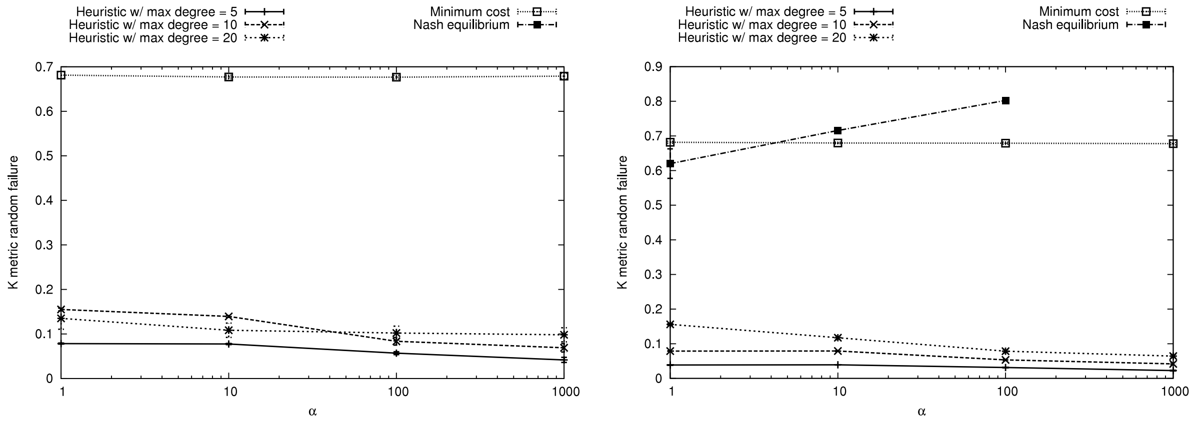
Fig. 7. *K* metric normal.

also maintains a high number of neighbors, which actually increases as it becomes more costly to maintain neighbors. This indicates that the peers are penalized for being selfish, because it would be expected that they would maintain fewer neighbors. But only a small number of them can do so before the others are forced to compensate by increasing their degree, resulting in a higher average degree. The Nash equilibria also have larger confidence intervals due to the more volatile nature of the equilibria. The heuristic algorithm has close to $\deg_{max}$ peers when α is low, but as expected, this number falls as α increases since the cost of maintaining that many peers is too high. As the overlay size increases, the heuristic algorithm keeps more neighbors because there are more to choose from that result in a lower cost, while the minimum cost algorithm does not change.

The next three figures provide K metric values. These are a measure of the topology's resilience [12]. The K is the ratio of all connected peer-pairs in the network divided by the total number of distinct peer-pairs in the network. We refer to this value, calculated for the overlay topology selected, as K metric$_{\text{normal}}$. We call K metric$_{\text{random-failure}}$ the K when 10% of peers are randomly selected and removed from the topology. Finally, we call K metric$_{\text{popular-failure}}$ the K when the 10% of the peers with the highest degree are removed.

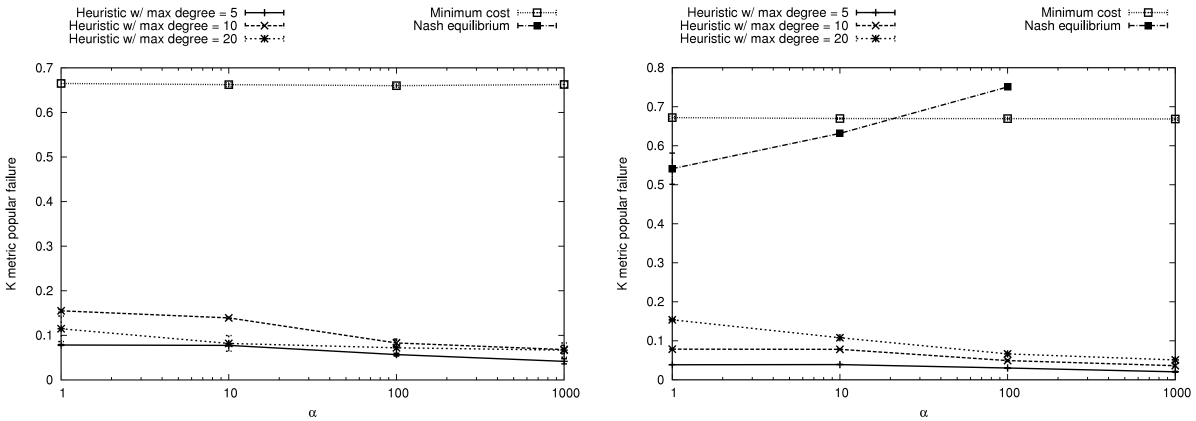
Fig. 7 shows the K metric for the topologies that were determined. A value of 1 means that each peer-pair is connected, as would be the case in a fully connected mesh. We see that the minimum cost algorithm has a very high K metric because such a large share of the peers are interconnected. The heuristic algorithm, restricted by $\deg_{max}$ had much lower K values, with lower $\deg_{max}$ values resulting in lower K values, as expected.

Fig. 8 shows the K metric assuming that a random 10% of the peers fail, possibly due to high energy consumption, peer mobility, or voluntarily leaving the overlay. Here, the minimum cost algorithm shows a reduced K metric, but as is expected, the value is still very high due to the very high level of connectedness of the topology. The heuristic algorithm also shows a reduced K metric, as expected, with the reduction in value very slight. Because the heuristic maintains fewer connections, a given peer is less likely to be affected by a random peer failure.



(a) 60 peers. (b) 100 peers.

Fig. 8. *K* metric random failure.



(a) 60 peers. (b) 100 peers.

Fig. 9. *K* metric popular failure.

Fig. 9 shows the K metric under the scenario that the most highly connected 10% of peers are removed from the overlay. This can occur because they are required to forward the most traffic for others, resulting in excessive energy consumption. Once again, the minimum cost algorithm performs very well because since nearly all peer-pairs are connected, the effect of removing the most connected 10% is minimal. Furthermore, these values are all unchanging with α. The Nash equilibria, as expected from previous results, show an increasing K metric with increasing α since more peers become connected. This results in a K metric that is sometimes higher than the minimum cost algorithm. The heuristic algorithm has a fairly low K metric, due to the $\deg_{max}$ constraint. As popular peers leave, the K metric falls since the relatively small number of connections falls further.

Overall, the heuristic algorithm has performed very well relative to the minimum cost algorithm, particularly with higher $\deg_{max}$. It results in a similar total cost, even though it maintains far fewer neighbors. This results in poorer K metric figures, demonstrating weaker resilience, though this could be addressed by raising the value of $\deg_{max}$.

5. Conclusion

P2P networks are immensely popular and may be used for many applications. Combining MANETs and P2P networks so that a P2P overlay runs on a cooperative MANET is a natural evolution. In many cases, peers are selfish in that they try to reduce their energy consumption, while minimizing their distances to all destinations. This paper provides an examination of overlay construction for P2P–MANETs with selfish peers. We propose a computationally feasible heuristic algorithm for neighbor selection that can adapt to peer churn and node mobility. We find that minimum cost topologies are highly connected and thus resilient, but in most cases the topologies are not in a stable state even without peer mobility or churn. This is demonstrated through the lack of existence of Nash equilibria, and when they exist, they show no discernible trend with respect to overlay size and the trade-off between the cost of maintaining neighbors and benefits of short distance to all destinations. Our heuristic is fairly stable relative to minimum cost algorithms, and when the degree–constraint is relaxed, the cost approximates the minimum. We also show the trade-off between the degree–constraint and resilience. When deg_{max} is low, the topology is sparsely connected, removing popular nodes results in significantly poorer connectivity of the overlay.

Acknowledgments

We would like to thank the anonymous reviewers. Their comments significantly helped us to improve the quality of this paper. This work is supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] Wi–Fi Alliance, Wi–Fi Alliance announces groundbreaking specification to support direct Wi–Fi connections between devices, http://www.wi-fi.org/news_articles.php?f=media_news&news_id=909, 2009.
[2] A. Fabrikant, A. Luthra, E. Maneva, C.H. Papadimitriou, S. Shenker, On a network creation game, in: ACM Symposium on Principles of Distributed Computing, 2003, pp. 347–351.
[3] T. Moscibroda, S. Schmid, R. Wattenhofer, On the topologies formed by selfish peers, in: 5th International Workshop on Peer-to-Peer Systems, February 2006.
[4] B.-G. Chun, R. Fonseca, I. Stoica, J. Kubiatowicz, Characterizing selfishly constructed overlay routing networks, in: IEEE INFOCOM, vol. 2, 2004, pp. 1329–1339.
[5] S. Androutsellis-Theotokis, D. Spinellis, A survey of peer-to-peer content distribution technologies, ACM Computing Surveys 36 (4) (2004) 335–371.
[6] D.N. da Hora, D.F. Macedo, L.B. Oliveira, I.G. Siqueira, A.A. Loureiro, J.M. Nogueira, G. Pujolle, Enhancing peer-to-peer content discovery techniques over mobile ad hoc networks, Computer Communications 32 (2009) 1445–1459.
[7] Y. Liu, A two-hop solution to solving topology mismatch, IEEE Transactions on Parallel and Distributed Systems 19 (11) (2008) 1591–1600.
[8] H.R. Varian, Intermediate Microeconomics, a Modern Approach, 6th ed., W.W. Norton & Company, 2003.
[9] ns 2, The network simulator, http://www.isi.edu/nsnam/ns/.
[10] C. Perkins, E. Belding-Royer, S. Das, Ad hoc on-demand distance vector (AODV) routing, RFC 3561, July 2003.
[11] L.M. Feeney, An energy-consumption model for performance analysis of routing protocols for mobile ad hoc networks, Mobile Networks and Applications 6 (3) (2001) 239–250.
[12] S.-T. Park, A. Khrabrov, D. Pennock, S. Lawrence, C. Giles, L. Ungar, Static and dynamic analysis of the Internet's susceptibility to faults and attacks, in: IEEE INFOCOM, vol. 3, 2003, pp. 2144–2154.

Afzal Mawji completed his Ph.D. at the Telecommunications Research Lab in the School of Computing at Queen's University. He received his MASc and BASc from the University of Waterloo. His research interests lie in the areas of mobile computing, ad hoc and mesh networks, pervasive computing and peer-to-peer computing.

Hossam Hassanein is with the School of Computing at Queen's University working in the areas of broadband, wireless and variable topology networks architecture, protocols, control and performance evaluation. Dr. Hassanein obtained his Ph.D. in Computing Science from the University of Alberta in 1990. He is the founder and director of the Telecommunication Research (TR) Lab in the School of Computing at Queen's. Dr. Hassanein has more than 350 publications in reputable journals, conferences and workshops in the areas of computer networks and performance evaluation. He has delivered several plenary talks and tutorials at key international venues, including Unconventional Computing 2007, IEEE ICC 2008, IEEE CCNC 2009, IEEE GCC 2009, IEEE GIIS 2009, ACM MSWIM 2009 and IEEE Globecom 2009. Dr. Hassanein has organized and served on the program committee of numerous international conferences and workshops. He also serves on the editorial board of a number of International Journals. He is a senior member of the IEEE, and is currently chair of the IEEE Communication Society Technical Committee on Ad hoc and Sensor Networks (TC AHSN). Dr. Hassanein is the recipient of Communications and Information Technology Ontario (CITO) Champions of Innovation Research award in 2003. He received several best paper awards, including at IEEE Wireless Communications and Network (2007), IEEE Global Communication Conference (2007), IEEE International Symposium on Computers and Communications (2009), IEEE Local Computer Networks Conference (2009) and ACM Wireless Communication and Mobile Computing (2010). Dr. Hassanein is an IEEE Communications Society Distinguished Lecturer.

Xiangyang Zhang received his B.E. degree and the M.E. degree in Electrical Engineering from Harbin Institute of Technology, Harbin, China in 1992 and 1995, respectively. He was with Huawei Technologies, from 1995 to 2007, where he was a senior engineer and research manager in the data communication division. He is now pursuing a Ph.D. degree at the School of Computing, Queen's University, Canada. His research interests span the fields of video communication, peer-to-peer networks, Internet infrastructure, and telecom networks.