

Characterizing Selfishly Constructed Overlay Routing Networks

Byung-Gon Chun, Rodrigo Fonseca, Ion Stoica, and John Kubiatowicz

Computer Science Division

University of California, Berkeley

Email: {bgchun,rfonseca,istoica,kubitron}@cs.berkeley.edu

Abstract—We analyze the characteristics of overlay routing networks generated by selfish nodes playing competitive network construction games. We explore several networking scenarios—some simplistic, others more realistic—and analyze the resulting Nash equilibrium graphs with respect to topology, performance, and resilience. We find a fundamental tradeoff between performance and resilience, and show that limiting the degree of nodes is of great importance in controlling this balance. Further, by varying the cost function, the game produces widely different topologies; one parameter in particular—the relative cost between maintaining an overlay link and increasing the path length to other nodes—can generate topologies with node-degree distributions whose tails vary from exponential to power-law. We conclude that competitive games can create overlay routing networks satisfying very diverse goals.

I. INTRODUCTION

Overlay routing networks [1], [2], [3], [4], [5], [6], [7] have become increasingly popular over the last few years. They form supporting technology for diverse application domains such as multicast, object location, and secure data dissemination. Overlays are easy to deploy and flexible, and can be resilient to faults. To achieve desired properties, however, most overlay systems assume that nodes *cooperate* with one another by following well-defined protocols, regardless of the costs incurred.

In reality, however, nodes may behave selfishly—seeking to maximize their own benefit. For instance, when parties in different domains utilize their own resources (overlay nodes) to participate in an overlay network, they have clear incentives to create links that maximize the benefit to their domain, possibly at the expense of globally optimum behavior. It is an open question whether these networks can have desirable global properties, in spite of the distinct local interests of the participating nodes.

Inspired by the game-theoretical model in [8], we study selfishly constructed networks by modeling network formation as a non-cooperative game. In this game, each node chooses its overlay neighbors to maximize its benefit and to minimize its linking cost. Consequently, nodes can have conflicting goals: on the one hand, they want to have low cost paths to other nodes in the network by establishing more links, and on the other hand, they may not want to establish many links, which may turn out to be costly. The outcome of the game is a

network topology, which is a Nash equilibrium¹.

In this paper, we simulate the network creation game in two different physical domains. The first involves a fully-connected physical topology with unit distance between nodes; this exposes the non-trivial effects of linking cost. The second involves more realistic topologies and physical limits—allowing an analysis of a *practical* overlay construction protocol. Our cost model extends the one suggested in [8] in several ways: the cost to establish a link is a general function, distance can represent any metric such as latency, and the neighbors of a node can be constrained.

We characterize the Nash equilibria obtained with metrics such as stretch, resilience to failures and attacks, and node degree distribution, and link these properties with variations in the game. We show that variations in a single parameter—the relative cost between maintaining an overlay link and increasing the path length to other nodes—can generate diverse topologies. We find a fundamental tradeoff between performance and resilience in the selfishly constructed networks. We show that varying parameters in the game model, such as limiting the degree of nodes, is of great importance in controlling this balance. A surprising fact, for example, is that by restricting the maximum number of links any node can establish, the game produces networks that are more resilient to attacks. Due to the selfish nature of the game, when degree is unrestricted, a few nodes establish many links, and most of the others free ride, by linking to the highly connected nodes. We conclude that, given the appropriate setting for the game, desirable *global* properties may be obtained by selfish optimization by local nodes.

The rest of the paper is organized as follows. In Section II we discuss related work. Section III discusses details of the network creation game, while Section IV describes the domain of our investigation. In Section V we explore properties of the networks formed by the games. Then, Section VI probes the resilience of these networks. We summarize the results and discuss directions for future work in Section VII. Finally, Section VIII concludes.

II. RELATED WORK

There has been considerable research on overlay routing networks such as RON [3], CAN [4], Chord [5], Pastry [6], and

¹A Nash equilibrium is a set of strategies with the property that no player can benefit by changing its strategy, while the other players keep their strategies unchanged [9].

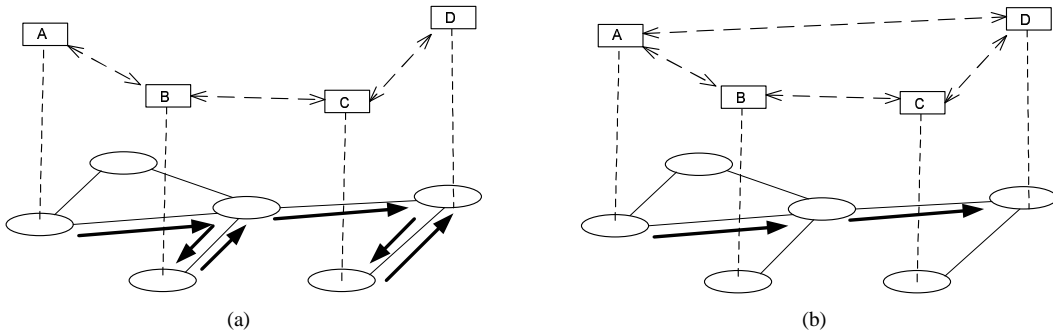


Fig. 1. Overlay routing networks. The ellipses are physical nodes and the rectangles are overlay nodes labeled A, B, C, and D. The first overlay network (a) has virtual links AB, BC, and CD. In (a), A’s messages are sent to D through B and C. The path length is 6 physical hops. When A decides to add a link to D, the resulting network is the network (b). A incurs cost to establish this new link, but the distance reduces to 2 physical hops due to the virtual link AD. This example shows the importance of selections of links in individual nodes.

Tapestry [7]. These overlay protocols assume obedience to the protocol and ignore participants’ incentives. Our work starts from the assumption that nodes are selfish and characterizes the networks created by such local optimization procedures. Overlays such as Narada [1] and Gossamer [2] use a mesh optimization process, which can be modeled as a specialization of our game. The detailed description of these protocols is presented in Section III-C.

The open problem of the characteristics of overlay networks created by selfish nodes, which is our focus of the study, is first mentioned by Feigenbaum and Shenker in [10]. A game-theoretic approach to the network creation problem is introduced in [8]. Their model is simplified for mathematical analysis. In their model, the cost of establishing an edge is the same for all nodes and the distance between two nodes is the number of hops. They investigate the price of anarchy. This concept, introduced by Papadimitriou in [11] is the ratio of the social costs of the worst-case Nash equilibrium and the social optimum, i.e., the price the participants pay as a group, for being selfish. They prove upper and lower bounds for this ratio, and also present a conjecture that says that, if the relative cost of establishing links is high enough, all Nash equilibria of the game are trees. Our work bridges this theoretical study to a practical network creation protocol (e.g., Narada), thus improving the understanding of selfishly constructed routing networks.

We analyze the failure and attack tolerance of the networks created by games. Albert, Jeong, and Barabási first studied the failure and attack tolerance of the scale-free networks [12]. Park et al. also analyzed the Internet’s susceptibility to faults and attacks with new metrics [13], one of which we use in our characterization.

The topological design of (physical) networks has been widely studied as a centralized optimal design problem [14]. Since the network design is NP-hard, approximation algorithms and heuristics such as genetic algorithms are explored. Our network creation problem is different in that the optimization process is performed among distributed selfish nodes.

III. ROUTING NETWORK CREATION

A. Overlay Routing Network

The overlay routing problem is to find a path with certain properties (e.g., shortest path) in overlay networks. To see how routing is determined given the overlay topology, Figure 1 shows two examples of overlay networks over the same physical topology. Suppose A wants to communicate with D. The routing path of messages from A to D is different in the two networks. A’s messages go through B, C, and D in the network (a) and the path length is 6 physical hops. A has an incentive to set up a link to D, if the benefit (reduction in distance) is more than the linking cost. Network (b) is the network after A sets up the link to D. The path length is only 2 physical hops. As one can see, the topology of the network significantly affects the routing performance. Understanding the formation of overlay networks is the domain of our study in this paper. Unless a centralized entity dictates how to set up links to each overlay node, each node needs to determine neighbors to establish links. Our focus is specifically to understand the formation of the overlay routing networks when nodes select links selfishly.

Since the routing is not the focus of our study, we assume that overlay nodes run a routing protocol to route packets. For example, in a scenario we investigate, nodes run the link state protocol.

B. Routing Network Creation Game

We model the network creation game problem as a non-cooperative game with n nodes (i.e., players) whose strategies are to select which nodes to connect to. In the game, each player changes its pure strategy² to minimize its cost. Our focus is to investigate resulting network topologies, which are the Nash equilibria of the games.

The cost model is the most important part of the game. We generalize the model of [8] in the following ways.

- 1) The cost to connect to a node j is not constant, but a function of j . This allows us to represent, for instance, congested links, in the sense that it may be more expensive to connect to popular or congested nodes.

²A pure strategy is one where players deterministically choose their moves.

- 2) The “distance” between two nodes may be represented by other functions than the number of hops (examples include the monetary cost of the path or the latency).
- 3) The possible neighbors that a node can connect to are a subset of all the nodes, which can increase the tractability of the model.

The cost of a node is a function of the cost paid for links and the distances from the node to the other nodes. The *strategy* of a node, at a given time, is the choice that the node can make, i.e., the subset of the other nodes in the graph the node chooses to connect to. Let the set of nodes be V , and let the set of feasible strategies of node i be A_i . A_i is constrained to the neighbor candidate set $NC_i (\subseteq V)$ where node i can establish links (i.e., $A_i = 2^{NC_i}$). The total cost incurred by node i , given a strategy $s = (s_1, s_2, \dots, s_n)$ where $s_i \in A_i$ for all $i \in V$ is given by

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

where $NB_i (\subseteq NC_i)$ is the set of neighbors of node i , t_j is the cost incurred to connect to node j , and $d_{G[s]}(i, j)$ is the distance from node i to node j in the graph $G[s]$. The distance from node i to node j is ∞ when node j is unreachable from node i . This condition prevents the generated graphs from being disconnected. In this model, α can be thought of as the relation between the cost of establishing a link and the change in distance to other nodes caused by the link. Notice that its magnitude depends on the units used in these costs, and also on the size of the network.

The total cost of the graph G is then defined as

$$C(G) = \sum_{i=0}^{n-1} C_i(s), \quad (2)$$

where $C_i(s)$ is the cost incurred by node i . This cost is used to compare Nash equilibria with the social optimum, which is the graph with smallest total cost.

C. Game Optimization Process

To study the equilibrium graphs created by overlay routing network creation game, we use an iterative procedure in which players make decisions based on their current view of the network in order to select which actions to perform (change links). We start the game from a connected random graph, and in each round, each player changes its link configuration to minimize its cost as given by Equation 1. We employ two variants of this model in our simulations. The first is characterized by an exhaustive search of the (exponential) strategy space, which can only feasibly be used in rather small topologies. For larger topologies we use a randomized local search strategy, in which we restrict the set of neighboring strategies nodes can choose. These procedures are described in detail below.

a) Exhaustive search: Regarding how nodes choose their next strategy given the current configuration, ideally all of the strategy space should be examined, i.e., the node should verify all possible configurations of the edges existing or not, to all

Algorithm 1 Link Addition for node i

```

Randomly select node  $j$  not in the neighborhood of  $i$ 
Compute  $Cost_{new}$  with  $j$  included
if  $Cost_{old} - Cost_{new} > 0$  then
    Add the link

```

Algorithm 2 Link Dropping for node i

```

NodeToDrop = -1
MinCost = Costold
for all node  $j$  in the neighborhood of  $i$  do
    Compute  $Cost_{new}$  without  $j$ 
    if  $MinCost - Cost_{new} > 0$  then
        MinCost =  $Cost_{new}$ 
        NodeToDrop =  $j$ 
if NodeToDrop  $\neq -1$  then
    Drop the link between  $i$  and NodeToDrop.

```

other neighbors. Thus, for each node, at each step, there are $2^{(n-1)}$ different strategies. The time complexity of running the game in this fashion is exponential in the number of nodes, and indeed the problem is NP-hard [8].

For small number of nodes, we use exhaustive search to find Nash equilibria. One node at a time, each node examines all possible configuration of links that it can drop or add, and chooses the one with the least cost, given the current configuration of the network. At each step the nodes take turns in adding and removing links, using a fixed ordering of action established in the beginning.

b) Randomized local search: To allow us to run larger simulations, we replace the exponential-time strategy search with a local search, a practical overlay protocol, which we describe next.

We use a randomized local search that can be implemented in practice. It should be noted that this search may converge to network configurations that do not meet the Nash condition. In the randomized local search, the game update procedure is as follows. We assume each node runs the link state (LS) protocol. Each node periodically performs the link drop and link addition procedures. The link drop procedure computes the minimum of the latency from the node to the other nodes when each link in the neighborhood is dropped. If the difference between the minimum value and the old latency value is less than a threshold (α), we drop the link that gives the minimum value. Then, the link addition procedure randomly selects a node that is not the previously dropped link and is not in the neighborhood. It fetches the link state and the cost t_j of the contacted node. If the latency improves by more than α , the link is added in the neighborhood. In addition, each node has a maximum degree bound. If the chosen node has already reached the maximum degree bound, the link cannot be established. The reason for which the procedures for adding and removing links are different is the information that is available to the node making the decision. For removing links, the node has information about all of its neighbors, and can select the best one. For adding links, information is

not available for all potential new neighbors, and this is why a random decision is made regarding which node to probe. Below we describe the algorithms more precisely.

Link Addition: We randomly choose one node that is not in the neighborhood of node i , and fetch its link state and cost t_j . We add the link if the cost of node i is reduced by linking to node j . The algorithm is presented as Algorithm 1.

Now, we explain the meaning of the if condition in the algorithm. By replacing $Cost_{old} - Cost_{new}$ with the equation 1, we get the following inequality when the maximum degree bound is met.

$$Cost_{old} - Cost_{new} > 0$$

$$\text{iff } \sum_{j=0}^{n-1} (d_{G_{old}}(i, j) - d_{G_{new}}(i, j)) > \alpha t_j$$

where the $Cost_{old}$ and $d_{G_{old}}(i, j)$ are the cost and the distance before the new link is added, respectively and $Cost_{new}$ and $d_{G_{new}}(i, j)$ are the cost and the distance after the new link is added, respectively. The cost reduction means that the latency reduction is greater than αt_j .

Link Drop: The link drop procedure is presented in Algorithm 2. It computes the node's cost of a new graph when a particular link is dropped. It chooses the neighbor that leads to the minimum cost value that is less than the old value. Note that the new cost should be reduced if a player wants to move in the game.

Likewise, we explain the meaning of $Cost_{old} - Cost_{new}$. In this case, the cost reduction means that the latency will increase by less than αt_j when the node drops a link. The following relation can be derived.

$$Cost_{old} - Cost_{new} > 0$$

$$\text{iff } \sum_{j=0}^{n-1} (d_{G_{new}}(i, j) - d_{G_{old}}(i, j)) < \alpha t_j$$

This strategy selection represents a generalization of two similar protocols for an overlay network creation which have been proposed in the literature, Narada[1] and Gossamer[2]. In Narada, nodes self-organize into an overlay mesh using a distributed protocol. The protocol optimizes the efficiency of the overlay by the adaptation mechanisms of individual nodes. Each node adds a link or drops a link by evaluating the utility of each link periodically. There are thresholds to add or drop a link. The threshold values are dependent on the number of nodes. The difference between our protocol and Narada is that our protocol also exchanges linking cost values in addition to routing information. Furthermore, Narada uses the distance vector (DV) protocol. Since it's not possible to compute the increase of the latency when a link is dropped, the Narada relies on a heuristics to choose a link to drop. If the LS protocol is used, we can compute the latency increases when a link is dropped.

IV. CASE STUDIES

Given the large parameter space of the general game model presented in Section III-B, we divided our study in two

| Scenario | $d_G(i, j)$ | Explored Parameters | Strategy Selection |
|-----------|--------------------------------|-------------------------|-------------------------|
| Simple | Number of Hops | α , Linking Cost | Exhaustive Search |
| Realistic | Latency from physical topology | α , Max Degree | Randomized Local Search |

TABLE I
SUMMARY OF THE SCENARIOS INVESTIGATED

| Cost Model | Linking Cost (t_j) |
|------------------------|------------------------|
| <i>Unit-Countout</i> | 1 |
| <i>Exp-Countout</i> | c_j |
| <i>Unit-Nodedegree</i> | degree(j) |

TABLE II

COST MODELS WE EXAMINE. j IS A NODE LINKED TO BY i . $\text{degree}(j)$ IS THE NUMBER OF EDGES OF j , AND c_j IS THE COST TO CONNECT TO NODE j , DRAWN FROM AN EXPONENTIAL DISTRIBUTION OF MEAN 1.

scenarios, which are summarized in Table I and described in more detail in the following subsections. In Section VII we describe other variations of the model, which we leave as directions for future work. The simulation results are presented in Sections V and VI.

A. Simple Scenario

In the first scenario, which we refer to as the 'simple scenario', we assume a 'simple' underlying physical topology where all pairs of physical nodes are connected and the distance between them is one. In other words, the underlying topology is a complete graph where all links have the same cost. The only reason not to establish a connection, in this case, is the cost incurred in maintaining many connections. We use this scenario to investigate different linking cost functions (t_j), described below, and summarized in Table II.

Unit-Countout: In this cost function, $t_j = 1, \forall j$, and the node that initiates the connection pays the total cost of the connection. This is the linking cost studied in [8].

Exp-Countout: In this function, we explore the effect of node heterogeneity. Each node pays a different amount to establish a connection depending on the node that is being connected to. The linking costs are generated from an exponential distribution of mean 1.0. Like the previous model, the node that initiates connection pays the cost of the link.

Unit-Nodedegree: In this cost function, cost incurred by a node to create a link depends on the node degree of the node to connect to. This can be thought of as relating to the *congestion* at a node: if there are many links already using a node, then there is a higher cost (that may represent a higher expected delay or loss rate), associated with connecting to this node. The linking cost dynamically changes as the graph changes. As the node degree of a node grows, it is less likely to acquire more links. This has the effect of balancing node degrees in the graph, which may be very desirable, as we show later.

B. Realistic Scenario

In the second scenario, which we call the 'realistic scenario', we use a more realistic underlying topology modeling the Internet. The strategy selection we use is the randomized local search described in Section III-B. The linking cost function we

use in this scenario (Equation 3) is a generalization of *Unit-Countout*, in that it allows for a restriction in the maximum degree nodes can have in the graph. The linking cost t_j when node i requests to connect to node j , is defined as

$$t_j = \begin{cases} 1 & \text{if } \text{degree}(j) < \text{MaxDegree} \text{ and} \\ & \text{degree}(i) < \text{MaxDegree} \\ \infty & \text{otherwise.} \end{cases} \quad (3)$$

where the $\text{degree}(j)$ is the degree of node j and MaxDegree is the maximum degree bound of the node. As we shall see later, limiting the degree has a positive effect in the resilience of the network.

The underlying topology we use consists of 1000 node transit-stub physical network topologies generated using the GT-ITM library [15]. Given the physical topology, we randomly choose nodes in stub domains from the underlying topology in which to place overlay nodes, and compute the shortest path latency between all pairs of overlay nodes. We use these as the distances between nodes in the overlay network creation game. The delays in the overlay topology thus obtained are approximately distributed according to a normal distribution, with mean delay around 328(ms). The distance $d_G(i, j)$ in the model is the sum of the latencies in the overlay path between node i and node j in this setting.

V. ANALYSIS OF NETWORK FORMATION

In this and in the next section we describe the results of the simulations for the two scenarios discussed in Section IV, exploring the parameter space. Here we look at topological and performance metrics of the equilibrium graphs, and in Section VI we examine the resilience of the created networks to failures and attacks.

For each set of parameters, we averaged simulation results over 50 different initial overlay configurations for the simple scenario and 100 different initial overlay configurations for the realistic scenario. In the plots, the errorbars represent one standard deviation in the value shown. We examine the graphs in this section through metrics described below.

Graph Cost: Defined in Equation 2, this metric is the total cost of the graph, i.e., the sum over all nodes of C_i . We use it to compare the graphs generated with the social optimum graphs.

Node Degree Distribution: This metric is important for its direct impact on the link stress of the network [1], on the state maintained by the node, and also, as we show in Section VI, on the resilience properties of the network.

Characteristic Path Length [16]: Defined as the average shortest distance among all pairs of nodes, this metric is important as a performance metric because the latency grows with the distance to reach other nodes.

Stretch: Defined as the average ratio of the shortest path latency in the overlay network to the shortest path latency in the physical network, this is also a performance metric. It measures, given the underlying topology, the efficiency of the overlay paths.

A. Simple Scenario

We analyze the equilibrium graphs in the simple scenario for the three linking cost functions as we increase α . We ran simulations involving 20 nodes using the exhaustive strategy search described in Section III, varying linking cost functions (t_j) and α values. The characteristics for the smaller networks followed similar trends. We leave the study of larger networks for the realistic scenario. Figure 2 shows sample equilibrium graphs we obtained, and the diverse topologies that can be generated by the game. Figure 3 shows the histogram of the degree distribution for sample graphs, for different parameter values. We refer to these figures throughout the evaluation of the graph characteristics.

The total cost of the graphs is shown in Figure 4. As the nodes optimize for their own cost, one can examine how the total cost of the resulting networks compares to that of the social optimum. The curves labeled ‘Optimum’ in the figure are the results of the nodes optimizing for the *total cost* of the graph, rather than for their own, an approximation of the social optimum. The cost of the graphs produced by the *Unit-Countout* game does not seem to asymptotically differ from the cost found with the global optimization by more than a constant. This is not true for the case of the *Unit-Nodedegree* model, and this fact is an indicative that it is important to consider factors such as the dynamic and heterogeneous characteristics of networks when studying the price of anarchy in these systems. The cost of *Exp-Countout* grows with high α , the reason being that with low α the links are always created to the nodes with the least cost.

We now move to examining the topological properties of the graphs. At a high level, the first two linking cost functions, *Unit-Countout* and *Exp-Countout* have the tendency to form star-shaped graphs, in which one or a few nodes have (very) high connectivity, and the others have no incentive to form additional links. This can be easily seen in Figure 2. The graphs in the *Unit-Nodedegree* cost model never form stars, and always have more balanced degree distributions. This does, however, come at the cost of longer path lengths. Also, for sufficiently high values of α , all equilibrium graphs for the three cases are trees, albeit with different properties. These findings are discussed in more detail below, for the three linking cost functions.

Unit-Countout: We observe 4 main phases as α increases. Our results are in line with the theoretical results in [8], since their model is equivalent to our simple scenario with this linking cost function.

When $\alpha < 1$, all equilibrium graphs are complete graphs, as adding an edge is always cheaper than having to traverse at least two hops to a non-neighbor node. This can be seen in Figure 5, which shows the characteristic path length of the graphs. We can see that the characteristic path length is one for this range of α in the *Unit-Countout* curve. We can also see that all nodes, for *Unit-Countout*, with $\alpha = 0.5$, have degree of 19 (Figure 3(a)).

When $1 \leq \alpha < 2$, all equilibrium graphs have path length of at most 2 [8]. We obtained dense graphs, with characteristic path lengths between 1 and 2 (Figure 5). An example such

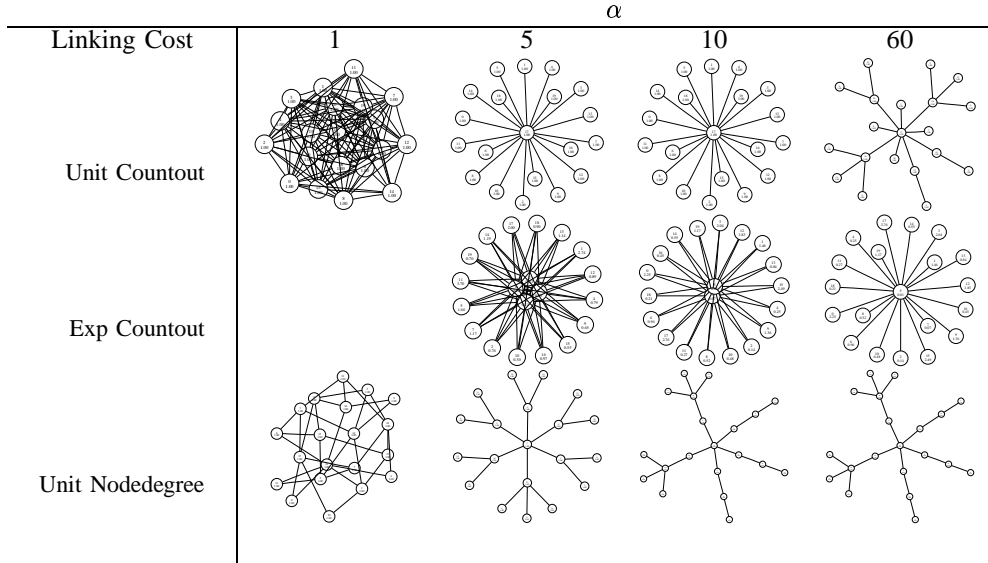


Fig. 2. Sample equilibrium graphs for the simple scenario with 20 nodes, for the different cost models and values of α . With *Exp-Countout* and $\alpha = 1$, we could not find an Nash equilibrium in our experiments.

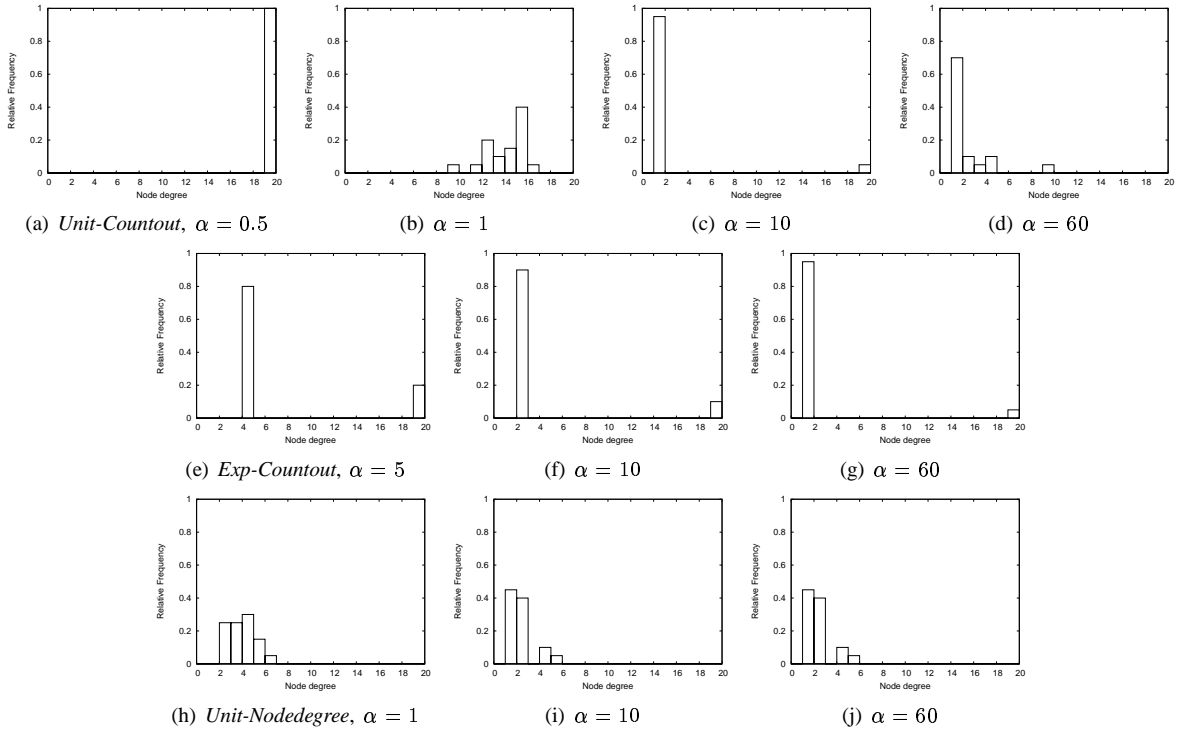


Fig. 3. Histograms of the node degree distributions for the simple scenario. On the top, the phases of *Unit-Countout*: complete graph (a), dense graph (b), star (c), and ‘leafy’ tree (d). *Exp-Countout* (e-g) goes through k -core stars, and *Unit-Nodedegree* (h-k) produces much more degree-balanced graphs.

graph can be seen in Figure 2, for *Unit-Countout* and $\alpha = 1$.

For $\alpha \geq 2$, a sharp transition occurs, as the social optimum shifts from being the complete graph to the star [8]. From Figure 5, we notice a flat region for α between 2 and 15, with characteristic path length of 2. Indeed, if we look at the corresponding sample graphs in Figure 2, and at the node degree distributions in Figure 3, we see examples of stars.

Finally, for even larger α values, another shift happens,

when even the star becomes expensive to maintain, and tree topologies in which most of the nodes are leaves start to appear. These have longer characteristic path lengths, but the distribution of degrees is still very skewed. We discuss the implications of this topology, and of the star, for the resilience of the network in Section VI.

Exp-Countout: In this cost function, as described in Section IV, nodes are not created equal: the cost of linking

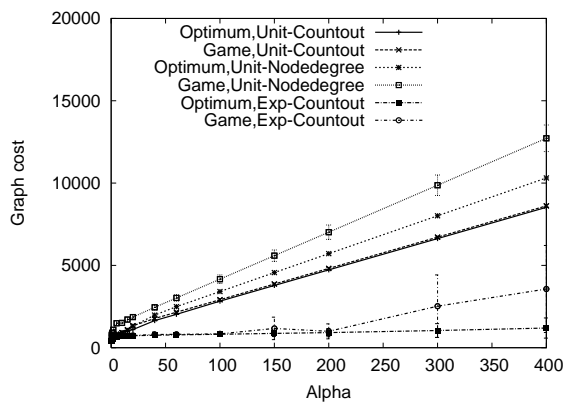


Fig. 4. Total graph cost for the three linking cost functions. Note that for the three linking cost functions, the cost asymptotes to a linear function of α .

to nodes is distributed exponentially. We observe a very strong tendency towards the formation of star-like topologies, specifically topologies we call k -core stars³. What happens is that all nodes choose to connect to the k nodes with the least linking cost, and the number k of links a node can afford to establish depends on α . There is a smooth transition from the complete graph to a star. At any point, the nodes that form the core are always the cheapest nodes. This can be clearly observed in Figures 2 and 3, in the middle row.

Unit-Nodedegree: In this last linking cost function, we recall that the cost to link to a node is an increasing function of the degree of the node, which may be thought of as representing the desire to avoid congestion. Across all values of α we observe a more even distribution of node degrees, concentrated around a central value, as can be seen in the bottom row of Figure 3. In the transition from complete graphs to trees, as we increase α , the graphs are never stars. In fact, as we observed for $\alpha = 2$, with the 20 nodes topology, many graphs had all nodes with the same node degree of 3. Another observed characteristics of the graphs, as can be seen in Figure 2, is that they tend to form longer cycles. We verified that for $\alpha \geq 1$, no graphs have cycles of length ≤ 3 (triangles).

By looking at Figure 5, the characteristic path length for *Unit-Nodedegree* grows as α increases. When α is high enough, we observe trees that tend to have longer branches and no nodes with too high degree. In comparison with the trees formed by *Unit-Countout*, there are more internal nodes in the trees and less leaves. As we discuss in Section VI, these trees with longer chains are more resilient to attacks than more leafy trees.

B. Realistic Scenario

In this section, we consider a more realistic network setting and show that it exhibits the same qualitative behavior as the simple scenario. We ran simulations involving 50 and 100 overlay nodes varying α and the maximum node degree bound

³These are graphs with a clique of size k that forms a core to which all other ‘leaf’ nodes are connected.

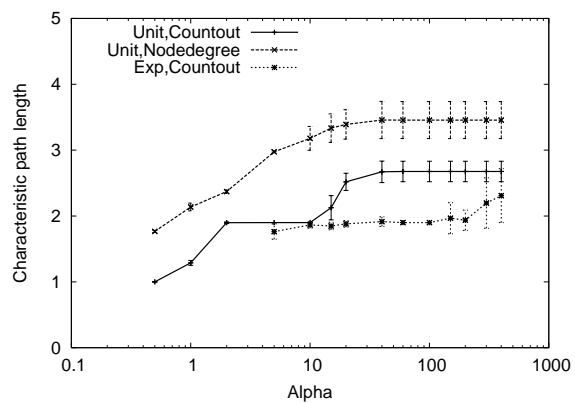


Fig. 5. Characteristic path length (CPL) for simple game scenario, for the three linking cost functions and different values of α . Complete graphs have CPL of 1, stars present CPL close to 2, and graphs with longer cycles or chains present higher CPL.

(*MaxDegree*) with 1000 node transit-stub physical network topologies. We present simulation results for 100 nodes, since the trends of results for 50 nodes are similar. We varied α from 200 to 20000 and we varied *MaxDegree* between 10 and 100. Since the distance metric in the cost model is latency, in the game, the unit of α is ms. Figure 6 shows sample equilibrium graphs we obtained, and Figure 7 shows the histogram of the degree distribution for sample graphs, for different parameter values. We refer to these figures throughout the evaluation of the graph characteristics for the realistic scenario.

We first look at the stretch. The stretch decreases as α decreases and it decreases as *MaxDegree* increases (Figure 8). Specifically, when α is small and *MaxDegree* is large, the stretch approaches the optimum value. In the overlay protocol, if α is small, the node is likely to add links, since it has a higher chance of finding a new node that reduces the distance enough. However, the node is not likely to drop links, since it has a higher chance of increasing the distance over α . On the contrary, if α is large, the node is not likely to add links, since it is hard to find a new node that reduces the distance by large amount. However, it is likely to drop a link to the neighbor, since the increase in the distance is likely to be less than α . Therefore, when α is small, the protocol tends to have more links, thus decreasing the stretch. When *MaxDegree* is large, the game process can find the nodes near the center that many peripheral nodes can benefit by establishing links to. These core nodes can be shortcuts to other nodes, thus reducing the stretch.

To better understand why stretch is different, we look at the node degree distribution and the sample equilibrium graphs. Figure 7 shows the histogram of node degree. When α is small (e.g., 200), the node degree distribution becomes skewed due to the maximum degree bound. When *MaxDegree* is 10, the peak is 10, and the frequency decreases rapidly as degree value decreases. When *MaxDegree* is 30, the distribution is multi-modal. When there is no constraint on the degree bound, the degree distribution resembles exponential distribution with several nodes of high degree. As there are more high degree nodes, there are more short paths to other nodes. As α becomes

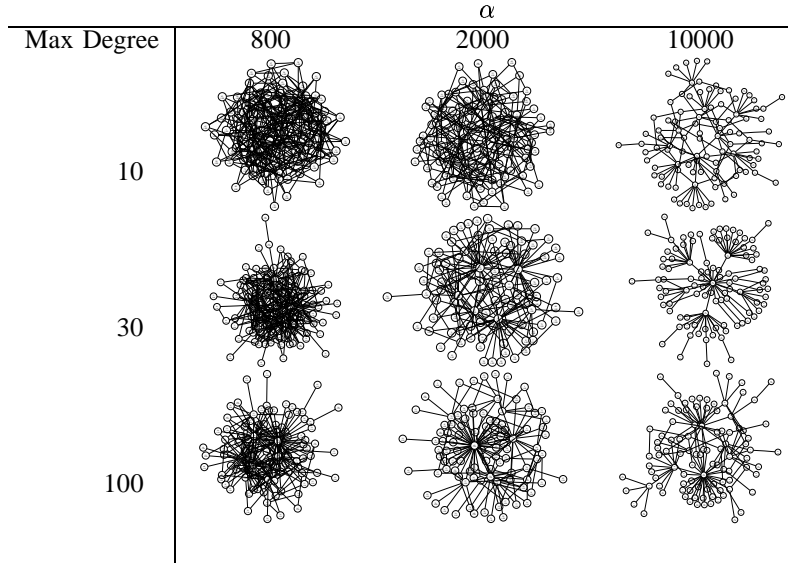


Fig. 6. Sample equilibrium graphs created in the realistic scenario. The network has 100 nodes, and we vary the maximum degree bound, as well as the α parameter.

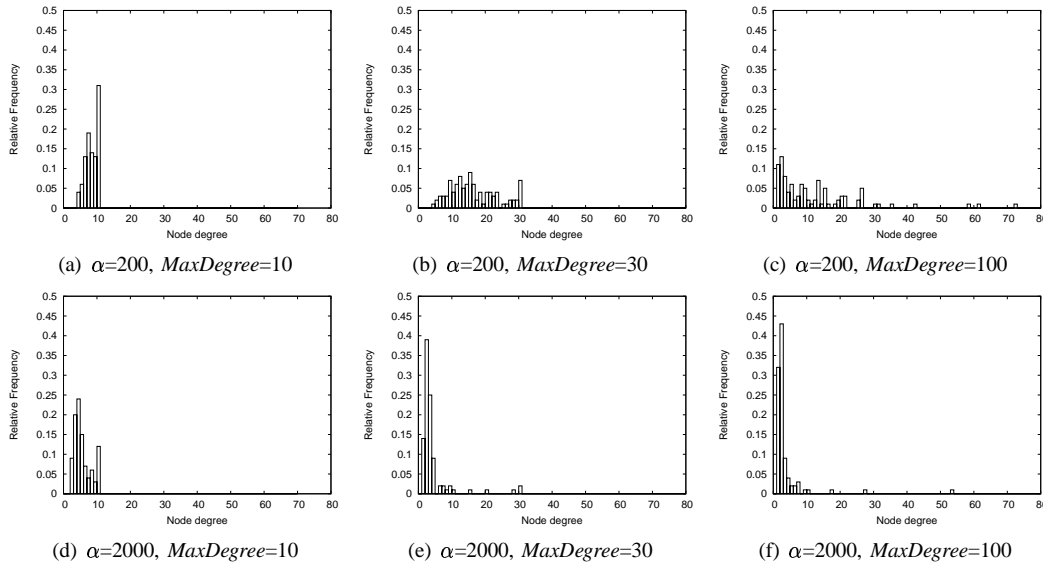


Fig. 7. Histogram of node degree distributions for the realistic scenario, for different values of α and $MaxDegree$.

large (e.g., 2000), the degree distribution becomes very skewed with the peak value of 3 or 4. Therefore, the stretch becomes larger. In the case of $MaxDegree$ of 100, there are still a few nodes of high degree, which reduces latency. For further understanding, we present the sample equilibrium graphs in Figure 6. When $MaxDegree$ is 100, the graph has a few cores that connect most of the nodes. However, when $MaxDegree$ is 10, each node has multiple links to other nodes and there are no central points unless α is very large. In this case, more nodes need to establish many links (e.g., more than 3) to maintain fair stretch. The overall trends of the graph forms are similar to those of the Nash equilibria produced by the *Unit-Countout* linking cost in the simple scenario, although degree bound is imposed. As α decreases, the graphs at Nash

equilibria transform from trees to almost k -regular graphs.⁴

To characterize the resulting graph topologies by comparing with widely known topologies, we also examined the tail of the node degree distribution. Figure 9 shows the tail distribution of the node degree when the game does not constrain degree bound, i.e., $MaxDegree$ is 100. We plot three lines, for α of 200, 800, and 2000. The graph is shown in a log-log scale, and the curves indicate, for a given node degree d , the fraction of nodes with degree greater than d . We also show, for reference, the curves for the tail of an exponential distribution, and of a Pareto (power-law) distribution. For $\alpha=200$, the distribution is not very skewed, and its tail decays in a similar fashion to the exponential distribution. When $\alpha=2000$, the decay in the distribution is much more pronounced, and exhibits similar

⁴ k -regular graphs are ones in which all nodes have the same degree $k(\leq n)$.

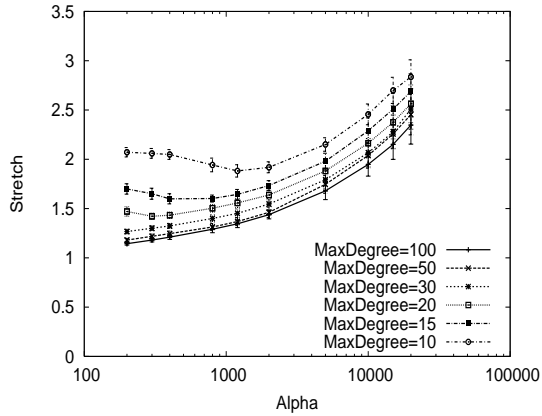


Fig. 8. Stretch for the realistic scenario, for different values of α and $MaxDegree$. The stretch is lowest for high $MaxDegree$, and always increases with α .

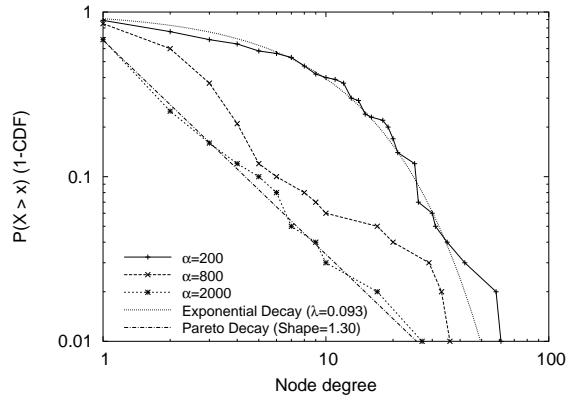


Fig. 9. Tail of the distribution of node degree, for the realistic scenario, and unrestricted node degree. For a low value of α , the tail has exponential decay, while for a high value of α the decay approximately follows a power-law.

behavior to that of the Pareto distribution. This is an important finding, since the game can be a way to create power-law graphs. These results are obtained when we create overlay networks on top of the GT-ITM topology. We are currently investigating the degree distributions for other underlying topologies. While we do not have enough evidence to prove such behavior, we conjecture that these distributions are not an artifact of the particular network used here. We leave more complete investigations on this subject, including using real Internet measurements for latencies, for future work.

Lastly, we present control overhead to evaluate the state maintenance overhead of the created networks compared to complete graphs. The control overhead is defined as the number of messages exchanged in the network created by the game normalized by the number of messages exchanged in the complete graph. The overhead is proportional to the number of links in the network. In Figure 10, the overhead is maximum 8% and decreases to 1%. The created networks have good properties even though they use only a small percentage of links among the total number of possible links. Overall, networks under high $MaxDegree$ have lower overhead than those under low $MaxDegree$, since fewer number of edges are established in the networks under high $MaxDegree$. When the low degree bound is applied, the control overhead is saturated as α becomes small. At this stage, the most of nodes in the network have links of $MaxDegree$.

VI. ANALYSIS OF FAILURE AND ATTACK TOLERANCE

In this section, we analyze the failure and attack tolerance of the Nash equilibrium networks. To evaluate the tolerance, we use the K metric presented in [13]. K is the ratio of all connected node-pairs in the network over the total number of distinct node-pairs in the network. This is used to characterize the whole network connectivity under certain percentages of node failures or attacks. For failures, we remove a fraction of randomly selected nodes. For attacks, we remove a fraction of nodes, starting from the node with the largest degree, followed by the next largest, and so on.

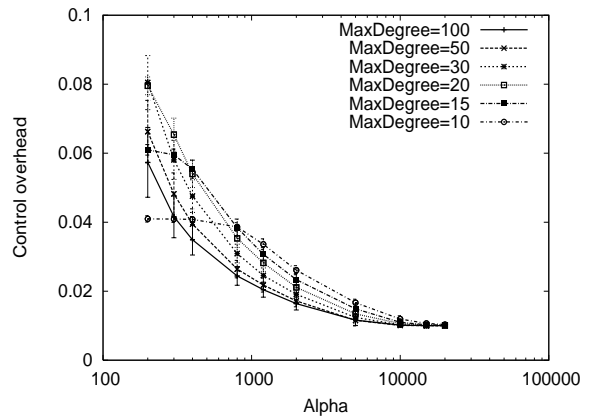


Fig. 10. Control overhead. In general, networks under higher $MaxDegree$ have lower overhead. We also observe the overhead is saturated under small α and low $MaxDegree$.

A. Simple Scenario

In the simple scenario, we evaluate the effects that failures and attacks of nodes have on the graphs produced by the different linking cost functions, for different values of α . Looking again at Figure 2, we can gain intuition in the resilience properties to be expected. Let us look at the three graphs for $\alpha = 10$, for example. *Unit-Countout* and *Exp-Countout* show star-like topologies, while *Unit-Nodedegree* shows a graph with long chains. If nodes are taken down randomly, the star topologies will most likely suffer the least, since a very small fraction of the nodes are really important for connectivity. However, if the most connected nodes are attacked, it is easy to see that the star topologies will collapse, while the *Unit-Nodedegree* generated topology will still have a good fraction of its nodes connected among themselves.

Figure 11(a) shows the value of K when 10% of the nodes are subject to failure. The results for other fractions of nodes taken down are very similar. The k -core star topologies created by the *Exp-Countout* model are the most resilient to failures, as we just discussed. Similarly, the star and the ‘leafy’ tree

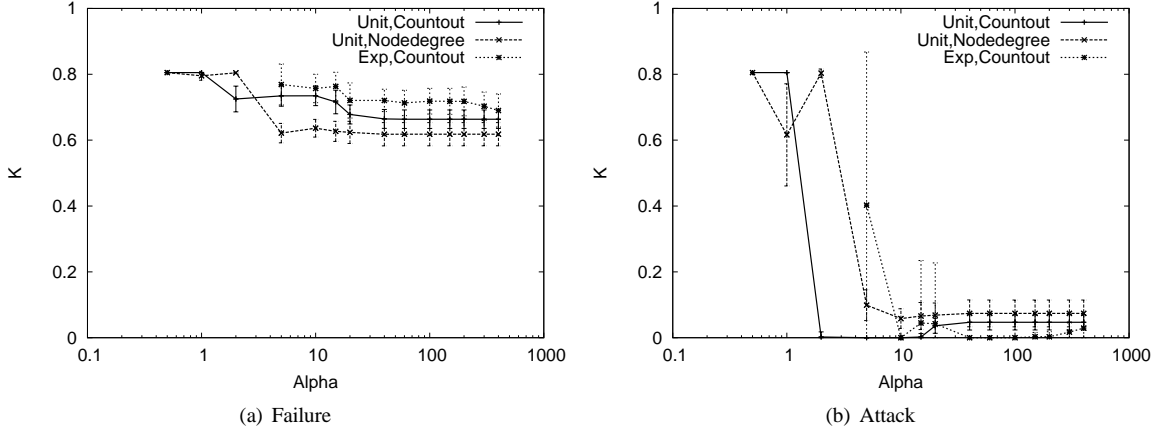


Fig. 11. Failure and attack tolerance (a) K when 10% of nodes fail, (b) K when 10% of nodes are attacked

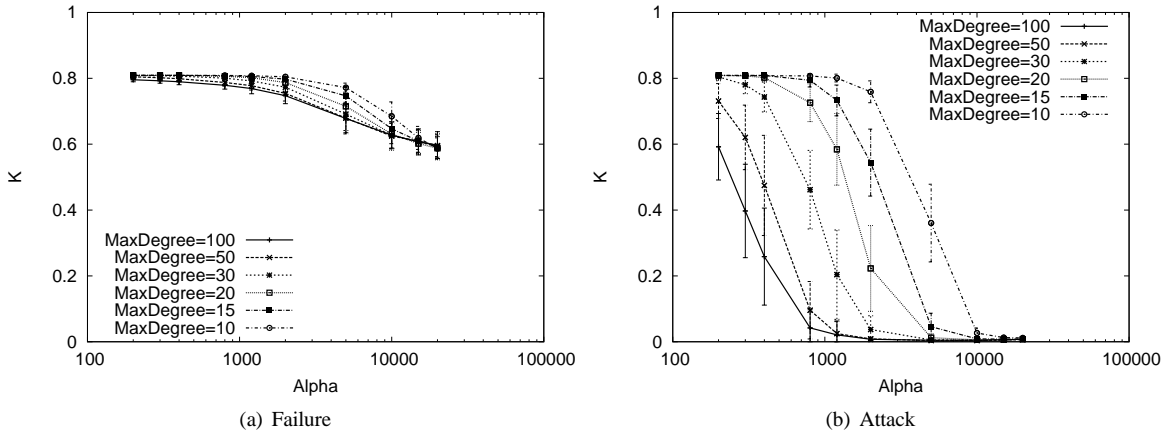


Fig. 12. Failure and attack tolerance (a) K when 10% of nodes fail, (b) K when 10% of nodes are attacked

topologies created by *Unit-Countout* are resilient to failures, since most nodes are leaves.

More degree-balanced graphs produced by *Unit-Nodedegree* are less resilient to failures, as more nodes are important for connectivity. However, when it comes to resilience to attacks, the cost in longer paths and lower resilience to failure pays off. Figure 11(b) shows the value of K when the 10% most connected nodes are attacked. The situation is reversed in this case, as the stars are very susceptible to attacks: some nodes are very important for connectivity and the network collapses when they are taken down. Comparing Figures 5 and 11(b), we see that exactly when the graphs are stars (and the path length is close to 2), K is equal to 0 (since removing the center node disconnects all others).

There is an interesting phenomenon in the resilience to attacks of the graphs created by the *Unit-Nodedegree* games. The graphs for $\alpha = 2$ are the most resilient to both failures and attacks. By examining the topologies and the node degree distributions, we see that when $\alpha = 1$, the equilibrium graphs have long cycles, and some nodes have a tendency to have more links than others (see Figure 3(h)). When $\alpha = 2$, however, the distribution of node degrees becomes much more centered around a single value. We observed graphs in which all nodes had degree 3. In this case, the attack is equivalent

to failure, since all nodes have the same degree.

B. Realistic Scenario

The equilibrium networks achieve good performance (low stretch) at the cost of lower resilience of the networks. Figure 12 shows K when 10% of nodes fail and K when 10% of nodes are attacked in the realistic scenario. The networks restricted by *MaxDegree* of 10 have the highest failure tolerance (Figure 12(a)). When α is 5000, the difference among different *MaxDegree*'s in the failure tolerance is the largest, but the difference diminishes as α gets smaller or larger. As shown in Figure 7, if a node is randomly selected for removal, the chance of removing the very high degree nodes is small, thus the difference in the tolerance decreases. However, the attack tolerance shows striking results (Figure 12(b)). The networks with the lowest stretch have the worst attack tolerance. They lose their connectivity among nodes rapidly as α increases. The attack tolerance graphs shift right as we decrease *MaxDegree*. Imposing node degree bounds is key to create networks with good attack tolerance property. The node degree distribution and sample graphs show the effects of attack in a straightforward way. When α is 2000 and *MaxDegree* is 100, by taking down a few high degree nodes, the network is completely disconnected, since most of the nodes have low degree and they are connected to the

high degree nodes. On the contrary, when α is 2000 and *MaxDegree* is 10, each node is well connected to several other nodes. There is no vulnerable core that significantly affects the connectivity of the network under attacks.

VII. DISCUSSION AND FUTURE WORK

We show that wide variety of linking cost functions create overlay networks with diverse properties such as low stretch, balanced degree distribution, and high failure or attack resilience. Some examples are k -regular graphs, k -core stars, trees, and graphs whose degree distributions have tails with decay ranging from exponential to power-law.

Exp-Countout creates graphs with cores comprising nodes with small linking cost values. Such nodes can be more powerful nodes than others, and games can exploit the heterogeneity. On the contrary, *Unit-Nodedegree* can produce graphs with balanced degrees without hot spots and nodes too vulnerable to attacks. Depending on the goals, one can use other variations of linking cost (t_j) functions, which is an interesting research question. For example, the product of a heterogeneous constant linking cost term and the node degree may get the benefit of both *Exp-Countout* and *Unit-Nodedegree*, exploiting heterogeneities with balanced load distribution among nodes. We also observe interesting relationships between linking cost functions. *Unit-Nodedegree* with small α creates k -regular graphs that can be produced by *MaxDegree* bound. Indeed, one can use an increasing function of node degree as an approximation for *MaxDegree*.

The exploration of Nash equilibria in the realistic scenario was restricted due to the limits on computation. If the interaction among agents are constrained, the computation depends on the size of the neighborhood candidate set, even if the network size is large. We intend to study the effect of constrained neighborhood candidate sets on the Nash equilibria.

We also found out that there is an important tradeoff between performance and resilience in the networks. The more the node degrees are balanced, the more resilient the network is to attacks. In the simple scenario, *Unit-Nodedegree* produces graphs with longer paths, more balanced degree distributions, and higher resilience to attacks. In the realistic scenario, limiting the degree achieves higher resilience, for similar reasons, at the cost of higher stretch. An overlay routing network among multiple parties can be created to meet its target goals by exploiting this tradeoff. If the most important requirement of the overlay routing network is low stretch, one can choose high *MaxDegree* and low α . To be resilient to attacks and failures, the game should impose low *MaxDegree*, forcing nodes to establish redundant links.

There are plenty of other interesting directions for future work. We want to further study the relationship between characteristics of the underlying topology and the produced overlay topologies. The current game is run with a fixed number of players. We want to examine the game in a dynamic network where the total number of nodes changes over time due to node joins and leaves. Another interesting area is to take traffic into consideration. If there is no traffic between two nodes, there may be little incentive to set up a link. We

also intend to apply the model to other scenarios, e.g., overlay networks created by minimizing the path failure probabilities. Finally, we want to examine mechanism designs to provide nodes no incentive to tell lies about linking cost values.

VIII. CONCLUSIONS

In this work we characterize selfishly constructed overlay routing networks. We use a non-cooperative game model to evaluate such networks and examine the effects of the underlying topology and different linking cost functions in the resulting Nash equilibria of the game. We find that the games can produce widely different networks, from complete graphs to trees with different properties. For the realistic scenario, varying the α parameter in the cost model produces networks with degree distribution with tails ranging from exponential to power-law distributions. We show that the networks obtained can present desirable properties, with respect to stretch and resilience, even though nodes are not interested in such global properties. We also find that there is a fundamental tradeoff between these two metrics, and that it can be controlled by restricting the maximum node degree. This tradeoff can be exploited to construct overlay routing networks that satisfy diverse requirements.

REFERENCES

- [1] Y.-H. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," in *Proceedings of ACM SIGMETRICS*, Santa Clara, CA, June 2000.
- [2] Y. Chawathe, "Scattercast: An architecture for Internet broadcast distribution as an infrastructure service," 2000.
- [3] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proceedings of SOSP*. ACM, 2001.
- [4] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of SIGCOMM*. ACM, August 2001.
- [5] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," in *Proceedings of SIGCOMM*. ACM, August 2001.
- [6] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," *Lecture Notes in Computer Science*, vol. 2218, 2001.
- [7] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," in *Journal on Selected Areas in Communications*. IEEE, January 2004.
- [8] A. Fabrikant, A. Luthra, E. Maneva, C. Papadimitriou, and S. Shenker, "On a network creation game," in *Proceedings of ACM PODC*, 2003.
- [9] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, 1994.
- [10] J. Feigenbaum and S. Shenker, "Distributed algorithmic mechanism design: Recent results and future directions," in *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 2002.
- [11] C. Papadimitriou, "Algorithms, games, and the Internet," in *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing*, 2001.
- [12] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," in *Nature* 406, July 2000.
- [13] S.-T. Park, A. Khrabrov, D. M. Pennock, S. Lawrence, C. L. Giles, and L. H. Ungar, "Static and dynamic analysis of the Internet's susceptibility to faults and attacks," in *Proceedings of IEEE INFOCOM*, 2003.
- [14] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner Tree Problem*, *Annals of Discrete Mathematics*, 1992.
- [15] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proceedings of IEEE INFOCOM*, 1996.
- [16] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," in *Nature* 393, June 1998.