A Cost-Based Analysis of Overlay Routing Geometries

Nicolas Christin and John Chuang
School of Information Management and Systems
University of California, Berkeley
102 South Hall
Berkeley, CA 94720-4600
Email: {christin, chuang}@sims.berkeley.edu

***Abstract*— In this paper, we propose a cost-based model to evaluate the resources that each node has to contribute for participating in an overlay network. Such a cost model allows to gauge potential disincentives for nodes to collaborate, and provides a measure of the "total cost" of a network, which is a possible benchmark to distinguish between different network architectures. We characterize the cost imposed on a node as a parametrized function of the experienced load and of the node connectivity, and express benefits in terms of cost reductions. We discuss the notions of social optimum and Nash equilibrium with respect to the proposed cost model. We show that the social optimum may significantly deviate from a Nash equilibrium when nodes value the resources they use to forward traffic on behalf of other nodes. Through analytical and numerical results, we then use the proposed cost model to evaluate some of the topologies recently proposed for overlay networks, and to exhibit some of the challenges systems designers may face. We conclude by outlining some of the open questions this research has raised.**

I. Introduction

Overlay networks play an increasing role in modern data communications. Examples of overlays include peer-to-peer file-sharing systems [1], ad-hoc networks [2], distributed lookup services [3], [4], application-layer multicast overlays [5]–[7], virtual private networks [8], or content delivery networks [9], to name a few.

Despite the growing popularity of overlay networks, there is no general consensus regarding how different overlay network topologies compare with each other. System architects may choose a particular overlay topology according to the graph-theoretic properties of the topology. For instance, de Bruijn graphs have recently received significant attention in the distributed lookup community [10]–[12], due to their short average routing distance and high resiliency to node failures. Other architectures, notably application layer multicast overlays, e.g., [6], [7], are usually designed so that the overlay topology exhibits desirable properties with respect to the underlying, physical, network.

This paper aims at providing a formal framework for evaluating and comparing overlay topologies. More precisely, the first contribution of this paper is a cost-based model to assess the resources that each overlay node has to contribute for being part of the overlay. We express the benefits of participating in the overlay in terms of a cost reduction. Such a cost model has several useful applications, among which, (1) providing a benchmark that can be used to compare between different proposals, (2) allowing for predicting disincentives, and designing mechanisms that ensure a protocol is *strategyproof* [13], and (3) facilitating the design of load balancing primitives.

Using the proposed cost model, our second contribution is to characterize the topologies that yield the lowest resource usage over the entire network (*social optimum*), as well as the topologies that are likely to be formed if each node is let free to select which links to maintain (*Nash equilibrium*). This study is particularly useful to assess whether allowing each participant in the overlay to adopt a rational (i.e., selfish) behavior results in an outcome desirable for all participants. Our main result is that the social optimum can significantly deviate from a Nash equilibrium when nodes value the resources they use to forward traffic on behalf of other nodes.

Our third contribution lies in the cost-based analysis of several topologies recently proposed in the context of distributed lookup services [3], [4], [10], [12], [14]. We provide analytical and numerical results to compare the costs incurred by each topology. We contrast these results with those obtained for the social optima, and discuss the implications of the observed costs on system design.

This work is not the first attempt to provide a model for the cost of participating in a network. Jackson and Wolinsky [15] proposed cost models to analyze formation strategies in social and economic networks. More recent studies [16], [17] model network formation as a non-cooperative game, where nodes have an incentive to participate in the network, but want to minimize the price they pay for doing so. Our approach extends these previously proposed cost models, by considering the load imposed on each node in addition to the distance to other nodes and degree of connectivity. Furthermore, we not only use the proposed cost model to characterize social optima and Nash equilibria, but also as a benchmark to analyze existing overlay topologies. In that respect, our work is complementary to recent graph-theoretic studies comparing topological properties of various overlays [12], [18].

The remainder of this paper is organized as follows. In Section II, we introduce our proposed cost model. In Sec-

This work is supported in part by the National Science Foundation through grants ANI-0085879 and ANI-0331659. A preliminary version of some of the material in this paper was presented at the Third International Workshop on Peer-to-Peer Systems (IPTPS'04).

tion III, we derive the social optima and Nash equilibria in the proposed cost model. In Section IV, we apply the cost model to several routing geometries used in recently proposed overlay architectures and compare analytically the costs incurred by each geometry. We illustrate and extend our analysis with numerical results obtained by simulation in Section V. Finally, we conclude the paper in Section VI, and discuss some open problems this research has uncovered.

II. Proposed Cost Model

We start with a formal description of the cost model we propose. The cost model applies to *any* (overlay) network where nodes request and serve items, or serve requests between other nodes. Let us define a routing geometry as in [18], that is, as a collection of edges, or topology, associated with a route selection mechanism. Unless otherwise noted, we assume shortest path routing in the overlay topology, and distinguish between different topologies; thus, we will interchangeably use the terms "topology" and "geometry" in the rest of this paper. Note that, a vast majority of overlay architectures, e.g., [3], [4], [6], [7], [10]–[12], [14], [19]–[21], do use shortest path routing in the overlay topology, which is quite different from using shortest path routing in the underlying physical network [6].

We define an overlay network by a quadruplet (V, E, K, F), where V, the set of nodes in the network, and E, the set of directed edges, characterize the topology used in the overlay. In addition, K is the set of items in the network, and $F : K \rightarrow V$ is the function that assigns items to nodes. Each node $u \in V$ is assigned a unique identifier (integer or string of symbols), which, for the sake of simplicity, we will also denote by u. We define by $K_u = \{k \in K : F(k) = u\}$ the set of items stored at node $u \in V$. We have $K = \bigcup_u K_u$, and we assume, without loss of generality, that the sets K_u are disjoint.[1] We characterize each request with two independent random variables, $X \in V$ and $Y \in K$, which denote the node X issuing the request, and the item Y being requested, respectively.

Consider a given node $u \in V$. Every time an item $k \in K$ is requested in the entire network, node u is in one of four situations:

Case 1: Idle. Node u does not hold or request k, and is not on the routing path of the request. Node u is not subject to any cost.

Case 2: Issuing the request. Node u requests item k. In our model, we express the benefits of participating in an overlay network in terms of latency reduction, similar to related proposals, e.g., [17]. In particular, we assume that the farther the node v holding k is from u (in a topological sense), the costlier the request is. If there is no path between nodes u and v, the request cannot be carried out, which yields an infinite cost. More precisely, we model the cost incurred by node u for requesting k as $l_{u,k}t_{u,v}$, where $t_{u,v}$ is the number of hops between nodes u and v, and $l_{u,k}$ is a (positive) proportional factor. We define the *latency cost* experienced by node u, L_u, as the sum of the individual costs $l_{u,k}t_{u,v}$ multiplied by the probability $k \in K_v$ is requested, that is

$$L_u = \sum_{v \in V} \sum_{k \in K_v} l_{u,k} t_{u,v} \Pr[Y = k] \ , \qquad (1)$$

with $t_{u,v} = \infty$ if there is no path from node u to node v, and $t_{u,u} = 0$ for any u. With this definition, to avoid infinite costs, each node has an incentive to create links such that all other nodes holding items of interest can be reached. An alternative is to store or cache locally all items of interest so that the cost of all requests reduces to $l_{u,k}t_{u,u} = 0$.

As a concrete example of the latency cost, consider the Domain Name Service (DNS, [22]). DNS can be viewed as an overlay network using a tree topology, where the leaf nodes are the DNS clients, and all other nodes are DNS servers. Consider that a client u wants to access a DNS record k so unusual that the query has to be redirected all the way to a DNS root server v. Here, we might have a relatively high value for the number of hops between u and v, say $t_{u,v} = 5$. After the query is resolved, u's primary DNS server, u', will have a copy of k, thereby reducing the latency for a request from u for k from $t_{u,v} = 5$ to $t_{u,u'} = 1$. Eqn. (1) simply captures the notion of latency as observed by u in terms of a weighted average over all possible queries u can make. The weights $l_{u,k}$ are introduced to express the relative value of one record compared to another. In our DNS example, if, from node u's perspective, the ability to resolve k = `www.google.com` is considered 100 times more valuable than the ability to resolve k' = `dogmatix.sims.berkeley.edu`, we should have $l_{u,k} = 100 \cdot l_{u,k'}$.

Case 3: Serving the request. Node u holds item k, and pays a price $s_{u,k}$ for serving the request. For instance, in an overlay file-sharing network, a node uses some of its upload capacity to serve a file requested by other nodes. We define the *service cost* S_u incurred by u, as the expected value of $s_{u,k}$ over all possible requests. That is,

$$S_u = \sum_{k \in K_u} s_{u,k} \Pr[Y = k] \ .$$

Going back to our earlier DNS example, copying the record k to the server u' implies that u' has to use some resources to store the copy of the record k, which our cost model characterizes by an increase in the service cost $S_{u'}$. In the DNS example, for a given DNS server, the cost of serving a DNS record k is the same for all k, so that we have for all k, $s_{u',k} = s_{u'}$, which corresponds to the cost of storing one record.

Case 4: Forwarding the request. Node u does not hold or request k, but has to forward the request for k, thereby paying a price $r_{u,k}$. The overall *routing cost* R_u suffered by node u is the average over all possible items k, of the values of $r_{u,k}$ such that u is on the path of the request. That is, for $(u, v, w) \in V^3$, we consider the binary function

$$\chi_{v,w}(u) = \begin{cases} 1 & \text{if } u \text{ is on the path from } v \text{ to } w, \\ & \text{excluding } v \text{ and } w \\ 0 & \text{otherwise,} \end{cases}$$

[1] If an item is stored on several nodes (replication), the replicas can be viewed as different items with the exact same probability of being requested.

and express R_u as

$$R_u = \sum_{v \in V} \sum_{w \in V} \sum_{k \in K_w} r_{u,k} \Pr[X = v] \Pr[Y = k] \chi_{v,w}(u) \ . \tag{2}$$

In our DNS example, the routing cost denotes the resources used by a server which receives a query for k, cannot resolve it, and has to redirect the query to a DNS server higher up in the tree, averaged over all possible queries.

In addition to the latency, service and routing costs, each node keeps some state information so that the protocol governing the overlay operates correctly. In most overlay protocols, each node u has to maintain a neighborhood table and to exchange messages with all of its neighbors, that is, the nodes v for which an edge (u, v) exists. Denoting by $\mathcal{N}(u)$ the set of neighbors of u, we characterize a *maintenance cost* M_u, as

$$M_u = \sum_{v \in \mathcal{N}(u)} m_{u,v} \ ,$$

where $m_{u,v} \geq 0$ characterizes the cost incurred by node u for maintaining a link with its neighbor $v \in \mathcal{N}(u)$. Returning to the DNS example, the maintenance cost characterizes the resources used by the DNS server u to maintain information about all the other servers u might contact (or refer to) when a query cannot be answered locally.

Adding the latency, service, routing, and maintenance costs for a node u, we can define the *individual cost* imposed on node u, C_u, as

$$C_u = L_u + S_u + R_u + M_u \ .$$

We can in turn use C_u to compute the *total cost of the network*, $C = \sum_{u \in V} C_u$.

Last, the expression of C_u only makes sense if S_u, R_u, M_u, and L_u are all expressed using the same unit. Thus, the coefficients $s_{u,k}$, $r_{u,k}$, $l_{u,k}$ and $m_{u,v}$ have to be selected appropriately. For instance, $l_{u,k}$ is given in monetary units per hop per item, while $m_{u,v}$ is expressed in monetary units. We next rely on our definition of the individual cost at a node u and of the total cost of the network to compute the social optima and Nash equilibria.

III. Social Optima and Nash Equilibria

In this section, we characterize the geometries that constitute a social optimum or a Nash equilibrium in the proposed cost model. The *social optimum* is defined as the routing geometry that minimizes the *total cost* C. A (pure) *Nash equilibrium* corresponds to a routing geometry where no node u can decrease its *individual cost* C_u by (deterministically) creating or removing a link. In other words, the social optimum is the outcome a system designer is likely to desire, while the Nash equilibrium describes the outcome that is likely to result from each node acting in its best interest. Thus, from a system designer's perspective, an ideal situation occurs when the Nash equilibrium and the social optimum correspond to the same topology. Conversely, when the social optimum is not a Nash equilibrium, one might need to devise mechanisms to realign the incentives of each individual node with a desirable global outcome. Studying Nash equilibria and social optima appears particularly useful in the context of self-forming networks, such as ad-hoc networks, or in describing peering relationships between Internet service providers, where individual nodes choose which links to maintain.

We next discuss a few simplifications useful to facilitate our analysis, before characterizing some possible social optima, and describing how they relate to the Nash equilibria.

A. Assumptions

For the remainder of this paper, we consider a network of $N > 0$ nodes, where, for all $u \in V$ and $k \in K$, $l_{u,k} = l$, $s_{u,k} = s$, $r_{u,k} = r$, and for all $u \in V$ and $v \in V$, $m_{u,v} = m$. In other words, we assume that the costs associated with incurring a one-hop latency, serving one request, routing one request, or maintaining one link, are the same on all nodes, irrespective of the item requested or served.[2] We suppose that the network is in a steady-state regime, i.e., nodes do not join or leave the network, so that the values l, s, r and m are constants. We also suppose that requests are uniformly distributed over the set of nodes, that is, for any node u, $\Pr[X = u] = 1/N$. For the time being, we make a further simplification by choosing the mapping function F such that all nodes have an equal probability of serving a request. In other words, $\sum_{k \in K_u} \Pr[Y = k] = 1/N$, which implies

$$S_u = \frac{s}{N} \ ,$$

regardless of the geometry used. (This assumption will be removed in Section V.) Moreover, if we use $E[x]$ to denote the *expected value* of a variable x, Eqs. (1) and (2) reduce to

$$L_u = lE[t_{u,v}] \ ,$$

and

$$R_u = rE[\chi_{v,w}(u)] \ ,$$

respectively. Also, because each node u has $\deg(u)$ neighbors, we immediately obtain

$$M_u = m \deg(u) \ .$$

Last, we assume that no node is acting maliciously.

B. Full Mesh

In our investigation of possible social optima, let us first consider a full mesh, that is, a network where any pair of nodes is connected by a bidirectional edge, i.e., $t_{u,v} = 1$ for any $v \neq u$. Nodes never any route any traffic and $\deg(u) = N - 1$. Thus, for all u, $R_u = 0$, $L_u = l(N-1)/N$, and $M_u = m(N-1)$. With $S_u = s/N$, we get $C_u = s/N + l(N-1)/N + m(N-1)$, and, summing over u,

$$C(\text{full mesh}) = s + l(N-1) + mN(N-1) \ . \tag{3}$$

Let us remove a link from the full mesh, for instance the link $0 \rightarrow 1$. The maintenance cost at node 0, M_0, decreases by m.

[2] While very crude in general, this simplification is relatively accurate in the case of a network of homogeneous nodes and homogeneous links containing fixed-sized keys such as used in distributed hash tables.

However, to access the items held at node 1, node 0 now has to send a request through another node (e.g.,[3] node 2): as a result, L_0 increases by l/N, and the routing cost at node 2, R_2, increases by r/N^2. So, removing the link $0 \rightarrow 1$ causes a change in the total cost $\Delta C = -m + l/N + r/N^2$. If $\Delta C \geq 0$, removing a link causes an increase of the total cost, and the full mesh is the social optimum. In particular, the full mesh is the social optimum if the maintenance cost is "small enough," that is, if

$$m \leq \frac{l}{N} + \frac{r}{N^2} \ . \tag{4}$$

Note that, as $N \rightarrow \infty$, the condition in Eqn. (4) tends to $m = 0$. In fact, we can also express $\Delta C \geq 0$ as a condition on N that reduces to $N \leq \lfloor l/m + r/l \rfloor$ when $m \ll l^2/r$, using a first-order Taylor series expansion.

We can draw a parallel with the DNS example of Section II to illustrate condition (4). As long as the number of Internet hosts remained reasonably small, each host used a large `HOSTS.TXT` file to directly resolve hostnames into IP addresses, effectively creating a full mesh for the naming overlay: each node knew about all the other nodes.[4] DNS was only introduced when the number of hosts on the Internet grew large enough to render maintaining all information in a single, distributed file impractical.

C. Star Network

Suppose now that Eqn. (4) does not hold, and consider a star network. Let $u = 0$ denote the center of the star, which routes all traffic between peripheral nodes. That is, $\chi_{v,w}(0) = 1$ for any $v \neq w$ $(v, w > 0)$. One can easily show that $R_0 = r(N-1)(N-2)/N^2$, $L_0 = l(N-1)/N$ and $M_0 = m(N-1)$, so that the cost C_0 incurred by the center of the star is

$$C_0 = m(N-1) + \frac{s}{N} + l\frac{N-1}{N} + r\frac{(N-1)(N-2)}{N^2} \ . \tag{5}$$

Peripheral nodes do not route any traffic, i.e., $R_u = 0$ for all $u > 0$, and are located at a distance of one from the center of the star, and at a distance of two from the $(N-2)$ other nodes, giving $L_u = l(2N-3)/N$. Further, $\deg(u) = 1$ for all peripheral nodes. Hence, $M_u = m$, and the individual cost imposed on nodes $u > 0$ is

$$C_u = m + \frac{s}{N} + l\frac{2N-3}{N} \ . \tag{6}$$

Proposition 1: $C_0 = C_u$ can only hold when N is a constant, or when $l = r = m = 0$.

Proof: By identification. (See [23].) ■

Since the difference $C_0 - C_u$ quantifies the (dis)incentive to be a priori in the center of the star, Proposition 1 tells us that there is a (dis)incentive to be in the center of the star in a vast majority of cases.

Next, we compute the total cost of the star, and determine under which condition it is a social optimum. Summing Eqs. (5) and (6), we obtain

$$C(\text{star}) = 2m(N-1) + s + 2l\frac{(N-1)^2}{N} + r\frac{(N-1)(N-2)}{N^2} \ . \tag{7}$$

Proposition 2: For any number of nodes $N \geq 3$, the star is a social optimum, if (i) Eqn. (4) does not hold and (ii) all links are bidirectional, i.e., for any $u \in V$ and $v \in V$, if $(u \rightarrow v) \in E$ then $(v \rightarrow u) \in E$.

Proof: Let us start from a full mesh. Every time we remove a (directed) link $u \rightarrow v$, we reduce M_u, and thus the total cost of the network, by m. However, at the same time, removing the link $u \rightarrow v$ imposes that traffic going from u to v has to go through at least one intermediary node w. So, L_u increases by at least l/N, and there is at least one node w for which R_w increases by r/N^2. In other words, every time we remove a link from a full mesh the change in cost is at least $\Delta C \geq -m + l/N + r/N^2$. (By hypothesis, the right term of the inequality is negative, so that there is potentially an advantage of removing a link from the full mesh.) Now, remark that all N nodes must be connected for the total cost C to remain finite. Further observe that one always need at least $(N-1)$ directed links to ensure that all N nodes are connected. So, under the assumption that all links must be bidirectional, we need at least $2(N-1)$ directed links to ensure all N nodes are connected. Differently stated, since the full mesh has $N(N-1)$ links, we can at most remove $(N-2)(N-1)$ links from the full mesh and still have a connected network. Assume that we can select the $(N-2)(N-1)$ links to be removed so that we realize the maximum savings $\Delta C = -m + l/N + r/N^2 < 0$ for each link we remove. Hence, we obtain the following lower bound on the cost of the social optimum, $C(\text{s. opt.})$:

$$\begin{aligned} C(\text{s. opt.}) \quad \geq \quad & C(\text{full mesh}) - (N-2)(N-1)m \\ & + \tfrac{(N-2)(N-1)l}{N} + \tfrac{(N-2)(N-1)r}{N^2} \ . \end{aligned}$$

From Eqs. (3) and (7), it follows that the right term in the above inequality is in fact equal to $C(\text{star})$. In other words, we have shown the total cost of a star network is smaller than or equal to the cost of the social optimum, from which we conclude that the star is a social optimum. ■

Let us make two remarks regarding Proposition 2. First, Proposition 2 does not guarantee that the star is a unique social optimum. In fact, in the limit case where $m = l/N + r/N^2$, adding any number of "shortcuts" between peripheral nodes of a star still results in a social optimum. Second, the assumption that the links are bidirectional is crucial for Proposition 2 to hold for any N. For instance, if we allow for unidirectional links, it can be shown that, if m is large enough and N remains small,[5] the unidirectional ring $0 \rightarrow 1 \rightarrow \ldots \rightarrow N \rightarrow 1$ has a lower cost than the star network. However, while finding the social optimum when unidirectional links are allowed is an open problem, we conjecture that the star network still plays a predominant role, and that geometries such as the unidirectional ring may only appear under very stringent conditions. More concisely, the above analysis tells us that, when the number of links to maintain becomes too high to make a full

[3] The actual mechanism that informs node 0 of which node to contact to send a request to node 1 is irrelevant to this discussion. One can for instance assume without loss of generality that nodes periodically advertise their list of neighbors.

[4] Note that we are here only concerned with name resolution. Updating and disseminating the `HOSTS.TXT` file is a separate issue, and was actually done in a centralized manner [22].

[5] More precisely, if $m > 0.5(N-1)(N-2)(l/N + r/N^2)$.

mesh an attractive solution, a centralized network is generally optimal from the point of view of resource consumption.

D. Nash Equilibria

Assume now that each node can choose which links it maintains, but does not have any control over the items it holds, and honors all routing requests. In other words, each node is selfish when it comes to link establishment, but is obedient once links are established. When each node u is (perfectly) rational, i.e., tries to minimize its individual cost C_u given the behavior of all other nodes, the resulting topology constitutes a Nash equilibrium. Even though the existence or uniqueness of a Nash equilibrium is in general not guaranteed, the following results yield some insight on the possible equilibria that may occur in our proposed cost model.

Proposition 3: If $m < l/N$, the full mesh is a unique (pure) Nash equilibrium.

Proof: In a fully connected network, no node can create additional links. If a given node u removes one of its links, $\deg(u)$ decreases from $(N-1)$ to $(N-2)$, but, at the same time, one of the nodes $u' \neq u$ is now at a distance of 2 from u. Thus, $E[t_{u,v}]$ increases from 1 to

$$E[t_{u,v}] = \frac{N-1}{N} + \frac{2}{N} = 1 + \frac{1}{N} \ ,$$

and the difference in utility for node u, between the strategy of removing one link and the strategy consisting in maintaining all links, is $m - l/N$. To have a Nash equilibrium, we therefore need to have $m - l/N \leq 0$, which is true if and only if $m \leq l/N$.

Suppose now that we have $m < l/N$, and a network that is not fully connected. In particular, consider that a node u can decide whether to create a link to another node $u' \neq u$. Before addition of the link $u \to u'$, u' is at a distance $2 \leq t_{u,u'} \leq N-1$ of u. After creation of the link $u \to u'$, u' is at a distance 1 of u. Thus, by creating the link $u \to u'$, $E[t_{u,v}]$ at least decreases by $(2-1)/N = 1/N$. Adding the link $u \to u'$ also results in $\deg(u)$ increasing by one, so that that the addition of the link $u \to u'$ eventually results in a change in the node u's utility equal to $-m + l/N$, which, by hypothesis, is strictly positive. Hence, node u always has an incentive to add links to nodes it is not connected to. Using the same reasoning for all nodes, we conclude that the fully connected network is the unique Nash equilibrium if $m < l/N$. ■

Proposition 4: If $m > l/N$, the star network is a pure Nash equilibrium.

Proof: Suppose, without loss of generality, that the central node is node 0. Node 0 is fully connected to the rest of the network, and therefore cannot create additional links. If node 0 removes one of its links, one of the $N-1$ other nodes becomes unreachable, which implies $E[t_{0,v}] \to \infty$, and $u_0 \to -\infty$. Thus, node 0 has no incentive in modifying its set of links. Likewise, peripheral nodes do not remove their (only) link to the central node, to avoid having their cost $C_u \to -\infty$.

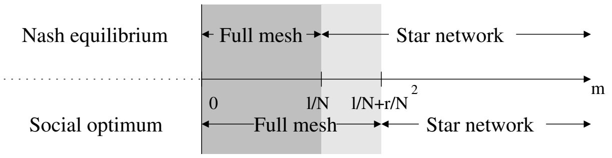


Fig. 1. Social optimum and Nash equilibrium. Incentives of individual nodes are not aligned with the social optimum in the interval $[l/N, l/N + r/N^2]$.

Suppose now that a peripheral node u creates an additional link to another peripheral node $u' \neq u$. An argument identical to that used in the proof of Proposition 3 shows that the addition of the link $u \to u'$ results in a change in the node u's utility equal to $-m + l/N$. Here, however, $m > l/N$, so that $-m + l/N < 0$, and node u has no incentive in adding the link $u \to u'$. Thus, the star network is a pure Nash equilibrium. ■

Propositions 3 and 4, tell us that, if maintaining links is cheap, or if the network is small, the only Nash equilibrium is the full mesh. If maintaining links is more expensive, or if the network is large, a star network is a possible Nash equilibrium; we cannot guarantee unicity of the equilibrium, however. For instance, in the limit case $m = l/N$, any network created by adding an arbitrary number of links between peripheral nodes of a star constitutes a Nash equilibrium.

E. Interpretation

We summarize our findings in Fig. 1, where we discriminate between social optima and Nash equilibria according to the value of m. For $m < l/N$, represented as a dark gray area in the figure, the full mesh is both a Nash equilibrium and a social optimum; for $m > l/N + r/N^2$ (white area), the star network is both a Nash equilibrium and a social optimum. In both cases, the incentives of each node are aligned with the most efficient overall usage of the resources in the network.

The most interesting region in Fig. 1 is perhaps the light gray area, in which individual incentives and overall resource usage are conflicting. This area corresponds to the parameter range $l/N < m < l/N + r/N^2$, whose size solely depends on r. Stated differently, under the assumption that all nodes have an identical probability of serving a request, *the social optimum may significantly deviate from a Nash equilibrium as soon as nodes value the resources they use to forward traffic on behalf of other nodes*.

As a corollary, a network where “forwarding comes for free” (i.e., $r = 0$), e.g., where bandwidth and computational power are extremely cheap, is ideal from the system designer's perspective, because individual incentives should produce a socially optimal solution. Unfortunately, in most networks, the price paid for forwarding data cannot be neglected, which suggests that our cost model is better suited at capturing possible disincentives than previous models solely based on node degree (i.e., maintenance costs) and hop count (i.e., latency costs).

IV. Analysis of Some Proposed Overlay Routing Geometries

In the discussion in the previous section, we have ignored robustness against attacks, fault-tolerance, or potential performance bottlenecks. All these factors pose practical challenges in a centralized approach, as does providing an incentive to occupy (or relinquish) the central position of a star. Using a full mesh avoids most of these concerns, but, as we have seen, is only a solution for a modest number of nodes.

Many research efforts have been directed at designing overlay geometries that provide reasonable performance, while addressing the aforementioned robustness concerns. In this section, we use our cost model to evaluate a few of the routing geometries that have been recently proposed for overlay networks in the networking literature. We focus on de Bruijn graphs, D-dimensional tori, PRR trees, and Chord rings. We derive analytically the various costs experienced by a node in each geometry. We will later compare our results with those obtained in our study of the social optima and Nash equilibria.

A. De Bruijn Graphs

De Bruijn graphs are used in algorithms such as Koorde [10], Distance-Halving [11], or ODRI [12], and present very desirable properties, such as short average routing distance and high resiliency to node failures [12]. In a de Bruijn graph, any node u is represented by an identifier string $(u_1, \dots, u_D)$ of D symbols taken from an alphabet of size Δ. The node represented by $(u_1, \dots, u_D)$ links to each node represented by $(u_2, \dots, u_D, x)$ for all possible values of x in the alphabet. The resulting directed graph has a fixed out-degree Δ, and a diameter D.

Denote by V' the set of nodes such that the identifier of each node in V' is of the form $(h, h, \dots, h)$. Nodes in V' link to themselves, so that $M_u = m(\Delta - 1)$ for $u \in V'$. For nodes $u \notin V'$, the maintenance cost M_u is $M_u = m\Delta$. The next two lemmas will allow us to show that the routing cost at each node also depends on the position of the node in the graph.

Lemma 1: With shortest-path routing, nodes $u \in V'$ do not route any traffic, and $R_u = 0$.

Proof: (By contradiction.) Consider a node $u \in V'$ with identifier $(h, h, \dots, h)$, and suppose u routes traffic from a node v to a node w. The nodes linking to u are all the nodes with an identifier of the form $(x, h, \dots, h)$, for all values of x in the alphabet. The nodes linked from u are all the nodes of the form $(h, \dots, h, y)$ for all values of y in the alphabet. Therefore, there exists x_0 and y_0 such that traffic from node v to node w follows a path $\mathcal{P} = (x_0, h, \dots, h) \to (h, h, \dots, h) \to (h, h, \dots, y_0)$. But, because, in a de Bruijn graph, there is an edge between $(x_0, h, \dots, h)$ and $(h, h, \dots, y_0)$, traffic using the path $\mathcal{P}$ between v and w does not follow the shortest path. We arrive to a contradiction, which proves that u does not route any traffic. ■

Lemma 2: The number of routes ρ_u passing through a given node u, or node loading, is bounded by $\rho_u \le \rho_{\max}$ with

$$\rho_{\max} = \frac{(D-1)(\Delta^{D+2} - (\Delta-1)^2) - D\Delta^{D+1} + \Delta^2}{(\Delta-1)^2} \ .$$

The bound is tight, since it can be reached when $\Delta \ge D$ for the node $(0, 1, 2, \dots, D-1)$.

Proof: The proof follows the spirit of the proof used in [24] to bound the maximum number of routes passing through a given edge. In a de Bruijn graph, by construction, each node maps to an identifier string of length D, and each path of length k hops maps to a string of length $D + k$, where each substring of D consecutive symbols corresponds to a different hop [12]. Thus, determining an upper bound on the number of paths of length k that pass through a given node u is equivalent to computing the maximum number, l_k, of strings of length $D + k$ that include node u's identifier, $\sigma_u = (u_1, \dots, u_D)$, as a substring. In each string of length $D + k$ corresponding to a paths including u, where u is neither the source nor the destination of the path, the substring σ_u can start at one of $(k-1)$ positions $(2, \dots, k)$. There are Δ possible choices for each of the k symbols in the string of length $D + k$ that are not part of the substring σ_u. As a result,

$$l_k \le (k-1)\Delta^k \ .$$

With shortest path routing, the set of all paths going through node u include all paths of length $D + k$ with $k \in [1, D]$. So,

$$\begin{aligned} \rho_u &\le \sum_{k=1}^{k=D} l_k \le \sum_{k=1}^{k=D} (k-1)\Delta^k \\ &\le \frac{(D-1)\Delta^{D+2} - D\Delta^{D+1} + \Delta^2}{(\Delta-1)^2} \ . \end{aligned} \tag{8}$$

We improve the bound given in Eqn. (8) by considering the strings of length $2D$ that are of the form $\sigma^*\sigma^*$, where σ^* is a string of length D. Strings of the form $\sigma^*\sigma^*$ denote a cycle $\sigma^* \to \sigma^*$, and therefore, never characterizes a shortest path in a de Bruijn graph. Hence, we can subtract the number of the strings $\sigma^*\sigma^*$ from the bound in Eqn. (8). Because $\sigma_u = (u_1, \dots, u_D)$ is a substring of $\sigma^*\sigma^*$ of length D, σ^* has to be one of the D circular permutations of σ_u, for instance $(u_{D-1}, u_D, u_1, \dots, u_{D-2})$. Since u does not route any traffic when u is the source of traffic, $\sigma^* \ne \sigma_u$. Thus, there are only $(D-1)$ possibilities for σ^*, and $(D-1)$ strings $\sigma^*\sigma^*$. Subtracting $(D-1)$ from the bound in Eqn. (8) yields $\rho_{\max}$. ■

From Lemmas 1 and 2, we infer that, in a de Bruijn graph, for any u, v and w, $0 \le \Pr[\chi_{v,w}(u) = 1] \le \rho_{\max}/N^2$. Because $\chi_{v,w}(u)$ is a binary function, $\Pr[\chi_{v,w}(u) = 1] = E[\chi_{v,w}]$, and we finally obtain $0 \le R_i \le R_{\max}$ with

$$R_{\max} = \frac{r\rho_{\max}}{N^2} \ .$$

We next compute upper and lower bounds on the latency cost. To derive a tight upper bound on L_u, consider a node $u \in V'$. Node u links to itself and has only $(\Delta-1)$ neighbors. Each neighbor of u has itself Δ neighbors, so that there are $\Delta(\Delta-1)$ nodes v such that $t_{u,v} = 2$. By iteration and substitution in Eqn. (1), we get, after simplification, $L_u \le L_{\max}$, with

$$L_{\max} = l\frac{D\Delta^{D+1} - (D+1)\Delta^D + 1}{N(\Delta-1)} \ ,$$

and $L_u = L_{\max}$ for nodes in V'.

Now, consider that each node u has at most Δ neighbors. Then, node u has at most Δ^2 nodes at distance 2, at most Δ^3 nodes at distance 3, and so forth. Hence, there are at least

$\Delta^D - \sum_{k=0}^{D-1} \Delta^k$ nodes at the maximum distance of D from node u. We get

$$L_u \geq \frac{l}{N} \left(\sum_{k=1}^{D-1} k\Delta^k + D \left(\Delta^D - \sum_{k=0}^{D-1} \Delta^k \right) \right) \ ,$$

which reduces to $L_u \geq L_{\min}$, with

$$L_{\min} = \frac{l}{N} \left(D\Delta^D + \frac{D}{\Delta - 1} - \frac{\Delta(\Delta^D - 1)}{(\Delta - 1)^2} \right) .$$

It can be shown that $L_u = L_{\min}$ for the node $(0, 1, \ldots, D-1)$ when $\Delta \geq D$.

Note that, the expressions for both $L_{\min}$ and $L_{\max}$ can be further simplified for $N = \Delta^D$, that is, when the identifier space is fully populated.

B. *D*-dimensional Tori

We next consider D-dimensional tori, where each node is represented by D Cartesian coordinates, and has $2D$ neighbors, for a maintenance cost of $M_u = 2mD$ for any u. This type of routing geometry is for instance used in CAN [3].

Routing at each node is implemented by greedy forwarding to the neighbor with the shortest Euclidean distance to the destination. We assume here that each node is in charge of an equal portion of the D-dimensional space. This constraint could also be expressed using the slightly stronger assumption that $N^{1/D}$ is an integer, and that all possible sets of Cartesian coordinates $(u_1, \ldots, u_D)$ (where each u_i maps to an integer in $[0, N^{1/D} - 1]$) map to a node. In other words, we assume the identifier space $(u_1, \ldots, u_D)$ is fully populated.

From [12], we know that the average length of a routing path is $(D/4)N^{1/D}$ hops for N even, and $(D/4)N^{1/D} + D/4 - o(1)$ for N odd. Because we assume that the D-dimensional torus is equally partitioned, by symmetry, we conclude that for all u,

$$L_u = l\frac{DN^{1/D}}{4} \ ,$$

using the same approximation as in [3] that the average length of a routing path is almost equal $(D/4)N^{1/D}$ hops even for N odd.

To determine the routing cost R_u, we compute the node loading as a function $\rho_{u,D}$ of the dimension D. With our assumption that the D-torus is equally partitioned, $\rho_{u,D}$ is the same for all u by symmetry.

Lemma 3: In a D-torus completely populated with N nodes, the node loading at any node u is given by

$$\begin{aligned} \rho_{u,D} = 1 + \quad & N^{\frac{D-1}{D}} \Bigg(-N^{\frac{1}{D}} + D \Bigg(N^{\frac{1}{D}} - 1 \\ & + \left(\left\lfloor \frac{N^{\frac{1}{D}}}{2} \right\rfloor - 1 \right) \left(\left\lceil \frac{N^{\frac{1}{D}}}{2} \right\rceil - 1 \right) \Bigg) \Bigg) . \end{aligned} \tag{9}$$

Proof: By induction on the dimension D. (See [23] for details.) ■

For all u, R_u immediately follows from $\rho_{u,D}$ with

$$R_u = r\frac{\rho_{u,D}}{N^2} \ .$$

C. PRR Trees

We next consider the variant of PRR trees [25] used in Pastry [14] or Tapestry [19]. Nodes are represented by a string $(u_1, \ldots, u_D)$ of D digits in base Δ. Each node is connected to $D(\Delta - 1)$ distinct neighbors of the form $(u_1, \ldots, u_{i-1}, x, y_{i+1}, \ldots, y_D)$, for $i = 1 \ldots D$, and $x \neq u_i \in \{0, \ldots, \Delta - 1\}$. The resulting maintenance cost is $M_u = mD(\Delta - 1)$.

Among the different possibilities for the remaining coordinates $y_{i+1}, \ldots, y_D$, the protocols generally select a node that is nearby according to a proximity metric. We here assume that the spatial distribution of the nodes is uniform, and that the identifier space is fully populated, which enables us to pick $y_{i+1} = u_{i+1}, \ldots, y_D = u_D$. Thus, two nodes u and v at a distance of n hops differ in n digits. There are $\binom{D}{n}$ ways of choosing which digits are different, and each such digit can take any of $(\Delta - 1)$ values. So, for a given node u, there are $\binom{D}{n}(\Delta - 1)^n$ nodes that are at distance n from u. Multiplying by the total number of nodes $N = \Delta^D$, and dividing by the total number of paths N^2, we infer that, for all u, v, and w, we have

$$\Pr[t_{u,v} = n] = \frac{\binom{D}{n}(\Delta - 1)^n}{N} \ . \tag{10}$$

Now, for any u and v such that $t_{u,v} = n$, because routes are unique, there are exactly $(n - 1)$ different nodes on the path between u and v. So, the probability that a node w picked at random is on the path from u to v is

$$\Pr[\chi_{u,v}(w) = 1 | t_{u,v} = n] = \frac{n - 1}{N} \ . \tag{11}$$

The total probability theorem tells us that

$$\begin{aligned} & \Pr[\chi_{u,v}(w) = 1] \\ & \quad = \textstyle\sum_{n=1}^{D} \Pr[\chi_{u,v}(w) = 1 | t_{u,v} = n] \cdot \Pr[t_{u,v} = n] \ . \end{aligned}$$

Substituting with the expressions obtained for $\Pr[t_{u,v} = n]$ and $\Pr[\chi_{u,v}(w) = 1 | t_{u,v} = n]$ in Eqs. (10) and (11) gives:

$$\Pr[\chi_{u,v}(w) = 1] = \frac{1}{N^2} \sum_{n=1}^{D} (n - 1) \binom{D}{n} (\Delta - 1)^n \ , \tag{12}$$

which, expressing the right-hand side as a function of the derivative of a series, and using the binomial theorem, reduces to

$$\Pr[\chi_{u,v}(w) = 1] = \frac{\Delta^{D-1}(D(\Delta - 1) - \Delta) + 1}{N^2} \ .$$

Multiplying the above expression for $\Pr[\chi_{u,v}(w) = 1]$ by r eventually gives us the routing cost,

$$R_u = r\frac{\Delta^{D-1}(D(\Delta - 1) - \Delta) + 1}{N^2} \ . \tag{13}$$

To compute the access cost L_u, we use the relationship $L_u = lE[t_{u,v}]$. We have

$$E[t_{u,v}] = \sum_{n=1}^{D} k \Pr[t_{u,v} = n] \ ,$$

(Δ, D)	$L_{\min}$	$L_{\max}$	$\frac{L_{\max}}{L_{\min}}$	$R'_{\min}$	$R_{\max}$	$\frac{R_{\max}}{R'_{\min}}$
(2, 9)	7.18	8.00	1.11	3.89	17.53	4.51
(3, 6)	5.26	5.50	1.04	2.05	9.05	4.41
(4, 4)	3.56	3.67	1.03	5.11	13.87	2.71
(5, 4)	3.69	3.75	1.02	1.98	5.50	2.78
(6, 3)	2.76	2.80	1.01	5.38	9.99	1.86

TABLE I

ASYMMETRY IN COSTS IN A DE BRUIJN GRAPH $(l = 1, r = 1,000)$

which, using the expression for $\Pr[t_{u,v} = n]$ given in Eqn. (10), and relying, here again, on the binomial theorem, leads us to

$$E[t_{u,v}] = \frac{D\Delta^{D-1}(\Delta-1)}{N} \ .$$

Multiplying by l to obtain L_u, we eventually get, for all u,

$$L_u = l\frac{D\Delta^{D-1}(\Delta-1)}{N} \ . \qquad (14)$$

(Note that, for $N = \Delta^D$, Eqn. (14) reduces to $L_u = lD(\Delta - 1)/\Delta$.)

D. Chord Rings

In a Chord ring [4], nodes are represented using a binary string (i.e., $\Delta = 2$). When the ring is fully populated, each node u is connected to a set of D neighbors, with identifiers $((u+2^p) \mod 2^D)$ for $p = 0 \ldots D-1$. An analysis similar to that carried out for PRR trees yields R_u and L_u as in Eqs. (13) and (14) for $\Delta = 2$. Simulations confirm this result [4].

E. Discussion

The analytical results we have derived in this section can serve a number of purposes. First, they confirm that all of the routing geometries considered here have the same asymptotic behavior: the routing costs decrease in $\log N$, while the latency costs grow with $\log N$. Second, while these asymptotic results are well known (see for instance [3], [4], [12], [18]), the main advantage of the above analysis is to provide closed-form equations that can be used for tuning configuration parameters such as Δ or D in function of the relative importance of each cost, e.g., routing cost vs. latency cost. Such a study of the configuration parameters is, however, outside the scope of the present paper. Third, our analysis provides us with a baseline we can use in a comparison with (1) the social optima and/or Nash equilibria and (2) more realistic scenarii where the identifier space is sparsely populated or where some items are more popular than others, which is the object of the next section.

V. NUMERICAL RESULTS

We present here some simulation results to validate and illustrate the analysis presented in Section IV. We complement the analysis by investigating numerically the effect of relaxing the assumptions that all items have identical popularity, and that the identifier space is fully populated.

A. Illustration of the Analysis

Let us first illustrate numerically the analysis of Section IV. In Table I, we consider five de Bruijn graphs with different values for Δ and D, and X and Y i.i.d. uniform random variables. Table I shows that while the latency costs of all nodes are comparable, the ratio between $R_{\max}$ and the second best case routing cost,[6] $R'_{\min}$, is in general significant. Thus, if $r \gg l$, there can be an incentive for the nodes with $R_u = R_{\max}$ to defect. For instance, these nodes may leave the network and immediately come back, hoping to be assigned a different identifier $u' \neq u$ with a lower cost. Additional mechanisms, such as enforcing a cost of entry to the network, may be required to prevent such defections.

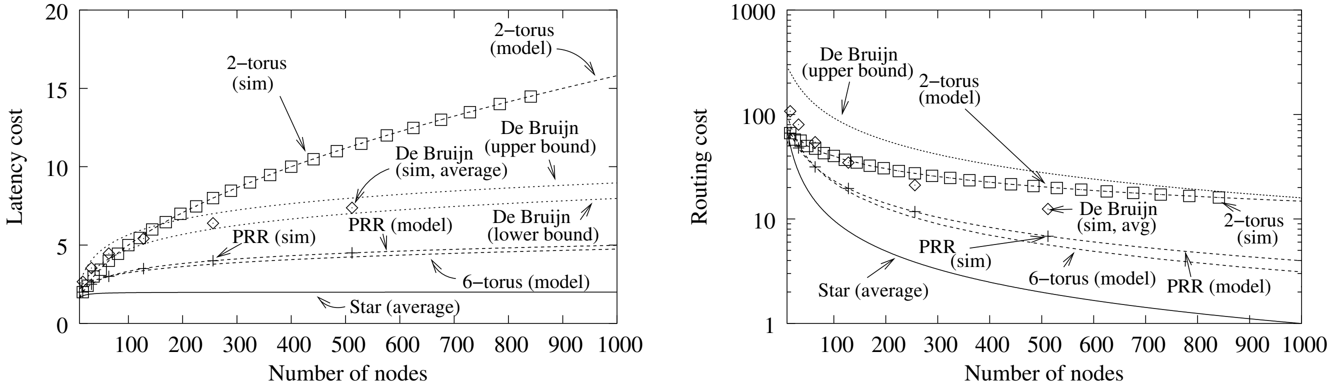
We next simulate the costs incurred in the different geometries we discussed. We choose $\Delta = 2$, for which the results for PRR trees and Chord rings are identical. We choose $D = \{2, 6\}$ for the D-dimensional tori, and $D = \log_\Delta N$ for the other geometries. We point out that selecting a value for D and Δ common to all geometries may inadvertently bias one geometry against another. We emphasize that we only illustrate a specific example here, without making any general comparison between different geometries.

We vary the number of nodes between $N = 10$ and $N = 1,000$, and, for each value of N run ten differently seeded simulations, consisting of 100,000 requests each, with X and Y i.i.d. uniform random variables. We plot the latency and routing costs averaged over all nodes and all requests in Fig. 2. The graphs show that our analysis is validated by simulation, and that the star provides a lower average cost than all the other geometries. This result is consistent with our earlier finding that the star is, in many cases, a social optimum, which may be more desirable to the community as a whole than a distributed solution. Note however, that our cost model does not take into account factors such as scalability and resiliency, both of which are cause for serious concerns in a completely centralized architecture. Additionally, while we have shown that the star network was potentially a Nash equilibrium, we nevertheless need incentive mechanisms (e.g., monetary rewards) to compensate for the asymmetry of a star network, and to convince a node to occupy the central position in the first place.

B. Asymmetry in Item Popularity

We investigate next how relaxing the assumption that all items have identical popularity impacts the results we have obtained so far. To that effect, we run a set of experiments, where items have a popularity that follows a Zipf-like distribution defined as follows. Assume the existence of a (bijective) function Rank : $V \rightarrow \{1, \ldots, N\}$, that orders the nodes $u \in V$ by decreasing probability that a given item k is held by u. For instance, if $\text{Rank}(u) = 1$, the probability that node u holds an arbitrary item k is strictly higher than the probability that any node $v \neq u$ holds k. Given $\text{Rank}(u)$, we characterize the

[6] That is, the minimum value for R_u over all nodes but the Δ nodes in V' for which $R_u = 0$.



(a) Latency cost $(l = 1)$

(b) Routing cost $(r = 1,000)$

Fig. 2. Latency and routing costs. Curves marked "sim" present simulation results. The full mesh, for which the latency cost is constantly equal to 1, and the routing cost is constantly equal to 0, is omitted for readability purposes.

probability that u has to serve a given request as:

$$\sum_{k \in K_u} \Pr[Y = k] = \frac{\Omega}{(\text{Rank}(u))^\alpha} \ , \tag{15}$$

where $\Omega = \left(\sum_{i=1}^{N} i^\alpha\right)^{-1}$. We immediately obtain $S_u = s\Omega/(\text{Rank}(u))^\alpha$.

In the case $\alpha = 1$, Eqn. (15) characterizes a Zipf distribution. Our motivation for using the distribution in Eqn. (15) stems from the observation that on the one hand, web caching, and more generally, content delivery networks, are one of the most deployed applications of network overlays;[7] on the other hand, measurement studies such as [27], show that web pages requests follow the distribution given in Eqn. (15) for $0.6 \leq \alpha \leq 0.9$.

In this set of experiments, we use $\alpha = 0.75$, and we simulate a network of size $N = 512$ nodes. We select $D = 3$ for the D-torus, and $\Delta = 2$ and $D = 9$ for the other geometries. Because the function Rank(.) is a permutation of the node indices, we should run $N!$ different experiments to exhaust all possible experimental cases, which is impractical. Instead, we pick 1,024 different orderings at random, and run one simulation experiment for each ordering. Our hope is that the sample size of 1,024 experiments is large enough to give a relatively accurate overall picture of the results one can expect. Each experiment consists of 100,000 requests. The source of the request X is a uniform random variable, and the requested item Y is determined according to Eqn. (15).

Because Y is not a uniform random variable anymore, different nodes experience different latency and routing costs. In each experiment, we collect the ratios between the highest ($L_{\max}$ and $R_{\max}$) and lowest ($L_{\min}$ and $R'_{\min}$) latency and routing costs observed over all nodes. Since in de Bruijn graphs, some nodes do not route any traffic, we use again $R'_{\min} = \min_{u \in V}\{R_u > 0\}$. In Table II, we present the average ratios $L_{\max}/L_{\min}$ and $R_{\max}/R'_{\min}$, averaged over all 1,024 experiments. Numbers in parentheses denote the corresponding standard deviation. The results indicate that, for all geometries, the latency costs of all nodes are relatively similar, but, the routing costs present significant differences. We explain the higher degree of asymmetry of the de Bruijn graph by the disparities in the node loadings (see Section IV), that magnify inequalities in routing costs. As a comparison to the social optima, we point out that in a star or a full mesh, the routing and latency costs are similar regardless of the popularity of the different items.

	$\frac{L_{\max}}{L_{\min}}$	$\frac{R_{\max}}{R'_{\min}}$
3-torus	1.2675 (±0.0442)	5.2845 (±0.3516)
De Bruijn	1.2453 (± 0.0265)	30.7275 (±9.5970)
PRR tree	1.2591 (±0.0420)	9.2154 (±0.6590)

TABLE II

ASYMMETRY IN COSTS IN A NETWORK WHERE ITEM POPULARITY FOLLOWS A ZIPF-LIKE DISTRIBUTION.

	Corr(R, L)	Corr(R, S)	Corr(L, S)
3-torus	-0.3133	-0.0166	-0.0960
De Bruijn	-0.3299	-0.0112	-0.0981
PRR tree	-0.2278	-0.0128	-0.1027

TABLE III

CORRELATION BETWEEN ROUTING, LATENCY, AND SERVICE COSTS IN A NETWORK WHERE ITEM POPULARITY FOLLOWS A ZIPF-LIKE DISTRIBUTION.

We next determine whether asymmetries in routing costs compensate asymmetries in latency costs, or, more significantly, in service costs. To that effect, we compute the correlation coefficient (denoted as Corr(x, y) for two variables x and y) between R and L, R and S, and L and S, computed over the 512 nodes × 1,024 experiments= 524,288 data points available for the triplet (R, L, S), and present our findings in Table III. For all three geometries, Table III indicates that

[7] In all fairness, a Zipf distribution may only be a very rough approximation of the request distribution in a file-sharing network such as KaZaA [26]. We conjecture however that the request patterns observed in file-sharing networks is more of an anomaly than a rule that can be generalized to all overlays.

there is almost no correlation[8] between S and R or L. In other words, the service cost S incurred by a node has almost no incidence on R or L. The correlation between R and L is also very weak, which indicates that different nodes may have, in the end, completely different costs.

In other words, with all three routing geometries considered, an asymmetry in the popularity of the items can cause a significant disparity in the costs incurred by different nodes. The disparity in costs itself results in some nodes being overloaded, or at least having strong incentives to leave and re-join the network to get a "better spot." This result emphasizes the importance of efficient load-balancing primitives for protocols relying on any of these routing geometries.

C. Sparse Population of the Identifier Space

So far, we have assumed that the identifier space is fully populated. For instance, a PRR tree with $\Delta = 2$ and $D = 9$ would necessarily contain $N = 512$ nodes. In practice however, the identifier space is likely to be relatively sparsely populated, especially during the deployment phase of a new overlay service or protocol. Here, we investigate the effects of a sparse population of the identifier space on the various costs incurred by different nodes.

Because routing geometries generally assume that the identifier space is fully populated, one has to address how to deal with identifiers that do not map to any node. In general, different overlay protocols use different solutions to the problem of handling a sparsely populated identifier space. Since, in this paper, we are interested in comparing geometries rather than specific protocols, we use a common technique for all of the routing geometries we study. The technique we use bears some similarity to the solutions proposed in [3], [4], [12], [14]. Each identifier v that does not map to a node is assigned to the node with the identifier u the closest to v according to an arbitrary norm in the identifier space. Thus, each node u may be assigned more than one identifier. In particular, if node u is assigned the identifier that would correspond to a node v in a fully populated identifier space, node u links to all the nodes v would link to. As a result, different nodes may have different maintenance costs M_u. In the computation of M_u, we consider that there is at most one link from one node to another, i.e., we discount duplicate links that may result from nodes holding multiple identifiers.

We run the following simulations. For each geometry, we consider a fixed number of nodes $N = 512$. We start with a fully populated identifier space, with $\Delta = 2$ and $D = 9$ for both de Bruijn graphs and PRR trees, and gradually increase D up to $D = 15$. For the D-torus, we use $D = 3$, so that each node u is represented by a set of coordinates (u_x, u_y, u_z). We allow each coordinate to take integer values between 0 and n. Initially, we select $n = 8$, so that each possible set of coordinates corresponds to a given node (because $n^D = N$), and we then gradually increase n up to $n = 32$. In other words, for all three topologies, we increase the identifier space from 512 to 32,768 identifiers. Identifiers that initially do not map to any node are selected using a uniform random variable. For each value of D (resp. n) we run 100 experiments with different random seeds, corresponding to 100 different ways of populating the identifier space. Each experiment consists of 100,000 requests, where X and Y are i.i.d. uniform random variables.

In Fig. 3, for each geometry, we plot the average value of the ratios $R_{\max}/R'_{\min}$, $L_{\max}/L_{\min}$, and $M_{\max}/M'_{\min}$ averaged over the 100 experiments corresponding to a given number of identifiers, as well as their worst-case (i.e., maximum) value over the same 100 experiments. For all geometries, we observe that the imbalance in latency costs remains relatively modest in a sparsely populated identifier space. The imbalance in maintenance costs is more significant, but the main observation is that the imbalance in routing costs can become very large. This observation emphasizes the urgent need for efficient load balancing algorithms.

Last, in Fig. 4, we plot the correlation coefficients between R and L, R and M, and L and M, as a function of the number of identifiers. Our main finding is that a sparsely populated identifier space has the effect of making the different costs correlated. This confirms the intuition that the routing and latency costs of a given node are largely dependent on how well the node is connected to the rest of the network, which is expressed by the maintenance cost.

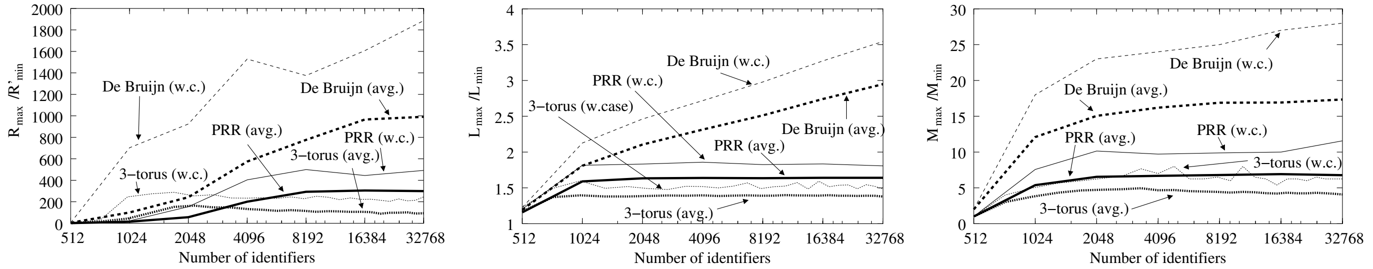
VI. Discussion and Conclusions

We proposed a model, based on experienced load and node connectivity, for the cost incurred by each node to participate in an overlay network. We argue such a cost model is a useful complement to topological performance metrics [12], [18], in that it allows to predict disincentives to collaborate (nodes refusing to serve requests to reduce their cost), discover possible network instabilities (nodes leaving and re-joining in hopes of lowering their cost), identify hot spots (nodes with high routing load), and characterize the efficiency of a network as a whole.

We believe our cost model can be used beyond the context of overlay networks, and can in fact apply to most networked systems with competing entities. Indeed, by adopting different values for the parameters (l, s, r, m) the model can indifferently apply to interconnections between Internet service providers, peer-to-peer file sharing networks, or mobile ad-hoc networks, to name a few examples. One of our main results is that, if nodes value the resources they use to forward traffic on behalf of other nodes, letting nodes choose which links they wish to maintain can yield a sub-optimal network with respect to overall resource usage.

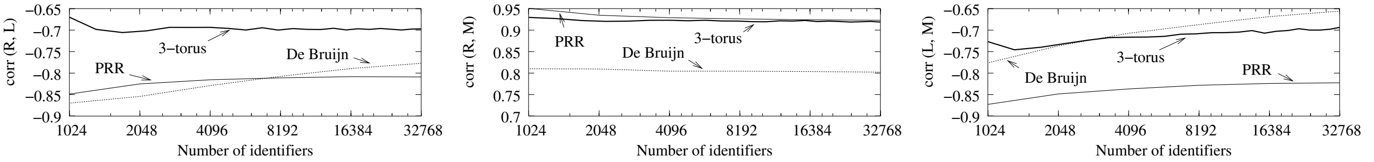
When individual incentives are not aligned with a desirable social outcome, which is the case in the context of most overlay networks, one may want to design rules to limit the effects of individual selfishness. Among the different type of rules that a designer can impose, we focused in this paper on network topology. We showed that, when the number of nodes is small, fully connected networks are generally the most

[8] The correlation coefficient actually only tests for a linear correlation. Additional tests, such as the η-test (or correlation ratio) are generally required to confirm the lack of correlation between two variables. We omit these tests here, but point out that additional data (e.g., scatter plots) confirm the lack of correlation between the variables.



(a) $R_{\max}/R'_{\min}$ (b) $L_{\max}/L_{\min}$ (c) $M_{\max}/M_{\min}$

Fig. 3. Ratios between maximum and minimum routing, latency, and maintenance costs experienced at a given node in function of the number of identifiers used. Curves marked "avg." indicate average results over all experiments in a given set, while curves marked "w.c." denote the maximum ratio, or worst case, observed over all experiments in a given set.



(a) Corr(R, L) (b) Corr(R, M) (c) Corr(L, M)

Fig. 4. Correlation between R_u, M_u, and L_u in function of the number of identifiers used.

cost-efficient solution. When the number of nodes is large, star networks are desirable from the point of view of overall resource usage. This result leads us to conjecture that, when feasible, centralized networks, where the "center" consists of a few fully connected nodes can be an interesting alternative to completely distributed solutions, provided that incentive mechanisms to handle network asymmetries are in place.

Using analysis and simulations, we characterized the costs incurred with some of the recently proposed topologies for network overlays. The main finding is that, while very appealing from the point of view of resiliency and scalability, all of the geometries we analyzed can potentially create large imbalances in the load imposed on different nodes. We also showed that, assuming that all nodes have approximately the same degree of connectivity to the rest of the network, different types of imbalance (e.g., routing load vs. experienced latency) are generally independent. As a result, we concluded that designing very efficient load-balancing primitives is a must to avoid favoring some nodes at the expense of others, which can potentially create network instability. It is worth noting that several papers have attempted to tackle the problem of load-balancing, notably in the context of distributed hash tables, e.g., [28], [29]. However, the load balancing algorithms proposed in the literature usually try to compensate for asymmetries in item popularity, while our study has shown that asymmetries in node connectivity arising from a sparsely populated identifier space were also a potential source of large imbalance.

We believe that this paper has sparked a number of avenues for future work. In particular, we only analyzed a handful of routing geometries, and even omitted interesting geometries such as the butterfly [20], geometries based on the XOR metric [21], or interconnected star networks, as used in file-sharing networks such as FastTrack or eDonkey. We believe that using the framework described in this paper will be useful in determining which type of topology is more appropriate for a specific application. A related open problem consists in obtaining a meaningful set of values for the parameters (l, s, r, m) for a given class of applications (e.g., file sharing between PCs, ad-hoc routing between energy-constrained sensor motes). To that effect, we plan on gathering measurement data from deployed networks, such as file-sharing systems, content delivery networks, or deployed ad-hoc and (centralized) wireless networks. Last, we point out that a possible alternative to load balancing primitives is to devise incentive mechanisms that make it desirable for nodes to forward as much traffic as possible. Incentive mechanisms have started to receive attention from the systems community (e.g., [13], [30], [31]) and one of our hopes for the present paper is to foster more research in that direction.

ACKNOWLEDGMENTS

The authors would like to thank Paul Laskowski and the anonymous reviewers for their insightful comments on earlier versions of this paper.

REFERENCES

[1] T. Klingberg and R. Manfredi, "Gnutella 0.6," June 2002, http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html.

[2] C. Perkins (editor), *Ad hoc networking*. Boston, MA: Addison-Wesley, 2000.

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of ACM SIGCOMM'01*, San Diego, CA, Aug. 2001, pp. 161–172.

[4] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for Internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, Feb. 2003.

[5] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proceedings of ACM SIGCOMM'02*, Pittsburgh, PA, Aug. 2002, pp. 205–217.

[6] Y.-H. Chu, S. Rao, and H. Zhang, "A case for endsystem multicast," in *Proceedings of ACM SIGMETRICS'00*, Santa Clara, CA, June 2000, pp. 1–12.

[7] J. Liebeherr, M. Nahas, and W. Si, "Application-layer multicast with Delaunay triangulations," *IEEE Journal of Selected Areas in Communications*, vol. 20, no. 8, pp. 1472–1488, Oct. 2002.

[8] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter, "Layer two tunneling protocol "L2TP"," IETF RFC 2661, August 1999.

[9] A. Vakali and G. Pallis, "Content delivery networks: Status and trends," *IEEE Internet Computing*, vol. 7, no. 6, pp. 68–74, Nov. 2003.

[10] F. Kaashoek and D. Karger, "Koorde: A simple degree-optimal distributed hash table," in *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03)*, Berkeley, CA, Feb. 2003, pp. 323–336.

[11] M. Naor and U. Wieder, "Novel architectures for P2P applications: the continuous-discrete approach," in *Proceedings of ACM SPAA'03*, San Diego, CA, June 2003, pp. 50–59.

[12] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh, "Graph-theoretic analysis of structured peer-to-peer systems: routing distances and fault resilience," in *Proceedings of ACM SIGCOMM'03*, Karlsruhe, Germany, Aug. 2003, pp. 395–406.

[13] C. Ng, D. Parkes, and M. Seltzer, "Strategyproof computing: Systems infrastructures for self-interested parties," in *Proceedings of the 1st Workshop on the Economics of Peer-to-Peer Systems*, Berkeley, CA, June 2003.

[14] A. Rowston and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large scale peer-to-peer systems," in *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platform (Middleware'01)*, Heidelberg, Germany, Nov. 2001, pp. 329–350.

[15] M. Jackson and A. Wolinsky, "A strategic model for social and economic networks," *Journal of Economic Theory*, vol. 71, no. 1, pp. 44–74, Oct. 1996.

[16] B.-G. Chun, R. Fonseca, I. Stoica, and J. Kubiatowicz, "Characterizing selfishly constructed overlay networks," in *Proceedings of IEEE INFOCOM'04*, Hong Kong, Mar. 2004.

[17] A. Fabrikant, A. Luthra, E. Maneva, C. Papadimitriou, and S. Shenker, "On a network creation game," in *Proceedings of ACM PODC'03*, Boston, MA, July 2003, pp. 347–351.

[18] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, "The impact of DHT routing geometry on resilience and proximity," in *Proceedings of ACM SIGCOMM'03*, Karlsruhe, Germany, Aug. 2003, pp. 381–394.

[19] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 41–53, Jan. 2004.

[20] D. Malkhi, M. Naor, and D. Ratajczak, "Viceroy: a scalable and dynamic emulation of the butterfly," in *Proceedings of ACM PODC'02*, Monterey, CA, July 2002, pp. 183–192.

[21] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the XOR metric," in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, Cambridge, MA, Feb. 2002, pp. 53–65.

[22] P. Mockapetris and K. Dunlap, "Development of the domain name system," in *Proceedings of ACM SIGCOMM'88*, Stanford, California, Aug. 1988, pp. 123–133.

[23] N. Christin and J. Chuang, "On the cost of participating in a peer-to-peer network," University of California, Berkeley, Tech. Rep., Dec. 2003, http://p2pecon.berkeley.edu/pub/TR-2003-12-CC.pdf. See also: arXiv:cs.NI/0401010.

[24] K. Sivarajan and R. Ramaswami, "Lightwave networks based on de Bruijn graphs," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 70–79, Feb. 1994.

[25] G. Plaxton, R. Rajamaran, and A. Richa, "Accessing nearby copies of replicated objects in a distributed environment," *Theory of Computing Systems*, vol. 32, no. 3, pp. 241–280, June 1999.

[26] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," in *Proceedings of ACM SOSP'03*, Bolton Landing, NY, Oct. 2003, pp. 314–329.

[27] L. Breslau, P. Cao, L. Fan, G. Philips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proceedings of IEEE INFOCOM'99*, New York, NY, Mar. 1999, pp. 126–134.

[28] J. Byers, J. Considine, and M. Mitzenmacher, "Simple load balancing for distributed hash tables,," in *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03)*, Berkeley, CA, Feb. 2003, pp. 80–87.

[29] B. Godfrey and I. Stoica, "Heterogeneity and load balance in distributed hash tables," in *Proceedings of IEEE INFOCOM'05*, Miami, FL, Mar. 2005.

[30] B. Cohen, "Incentives build robustness in BitTorrent," in *Proceedings of the First Workshop on the Economics of Peer-to-Peer Systems*, Berkeley, CA, June 2003.

[31] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," in *Proceedings of the Fifth ACM Conference on Electronic Commerce (EC'04)*, New York, NY, June 2004, pp. 102–111.