

On Characterizing Affinity and Its Impact on Network Performance

Gabriel Lucas
School of Information
Management and Systems
University of California at
Berkeley

glucas@alumni.sims.berkeley.edu

Abhishek Ghose
Dept of Electrical Engineering
and Computer Sciences
University of California at
Berkeley

aghose@eecs.berkeley.edu

John Chuang
School of Information
Management and Systems
University of California at
Berkeley

chuang@sims.berkeley.edu

ABSTRACT

An important component of simulation-based network research is the selection of nodes to a member group, such as receivers in a multicast group or web clients in a content delivery network. In a seminal paper, Philips *et al.* introduce an algorithm for generating member groups with different degrees of affinity (clusteredness) and show that affinity can have a significant effect on multicast efficiency. Subsequent studies applying this algorithm have all used the algorithm's input parameter as a method for classifying and comparing affinity groups. In this paper, we propose several distance- and expansion-based analysis metrics and find them to be better measurements of the true affinity of member groups. In three separate case studies (multicast, replica placement, and sensor networks), we demonstrate the benefit of classifying member groups by their true affinity in order to predict network performance variation. By systematizing techniques for measuring affinity, we open the door for more realistic and reproducible research in studies employing affinity-based member selection techniques.

1. MOTIVATION AND RELATED WORK

Over the last decade, the field of network research has evolved tremendously, particularly in the area of topology generation. The Waxman random network model [29] was a major breakthrough, which in turn paved the way for subsequent improvements, including hierarchical and power-law models. The result is a list of generators that are becoming increasingly more realistic in their ability to model large networks, from GT-ITM [31] and TIERS [9], to BRITE [17], INET [14], and PLRG [2]. Given the link between topological characteristics and network performance [5, 18, 23, 25], each new iteration of topology modelling has been critically important.

However, topology is not the only parameter for network

analysis. Many simulations require assigning roles to subsets of nodes, such as recipients or transmitters of data. A group of receivers in a multicast communication is an example of a *member group*, which is generally defined as a subset of nodes selected for a common role. Philips *et al.* [19] recently introduce an algorithm to generate member groups with different levels of *affinity*. Affinity is a measurement of the physical closeness or clusteredness of a member group. To cluster a member group is to select nodes that are close together; to decluster is to select nodes as far apart from each other as possible¹. In their study, Philips *et al.* show that increasing the affinity level of receiver groups leads to higher multicast efficiency.

So far, however, only a few studies have considered affinity analysis. Those that have include [30] (multicast communication), [22] (replica placement on the Internet), and [11] (deployment of new routing services). One possible reason for the slow interest in affinity research is the lack of effective analysis tools. Affinity studies so far have all compared performance against different values of the selection algorithm's input parameter. One problem with this approach is that there is no guarantee that the input parameter accurately characterizes the true affinity of a member group. A larger issue is that such methodology prevents affinity analysis of not only real-world clusters, but any member group not generated using the algorithm in [19].

The goal of our paper is to refine the techniques of affinity analysis by answering the following questions:

1. Is it possible to cluster and decluster to the same extent on every network? Exactly what does extreme affinity or disaffinity mean? Can we characterize the true affinity level of a member group, without knowledge of the input parameter used to generate that group?
2. A good metric should not only describe the true affinity level of a member group, but also explain differences in application performance. Can we define a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. ACM SIGCOMM 2003 Workshops, August 25&27,2003, Karlsruhe, Germany. Copyright 2003 ACM 1-58113-748-6/03/0008...\$5.00.

¹Our use of the word *cluster* is merely a synonym for *positive affinity* among a group of nodes. By contrast, traditional *cluster analysis* involves *segmenting* a network into distinct regions. For more on traditional cluster analysis, see [15], [16], and [4].

metric that correlates with changes in network performance as we adjust member group affinity?

To summarize our findings, networks do have affinity distinctions and limitations based on different topological characteristics. We find that the measured networks we study permit the greatest variation in affinity, and that of the generated networks, the Smallworld produces affinity trends most similar to those of the measured networks. We also find that our analysis metrics are much better at characterizing the true affinity of a group than is the affinity selection parameter. Likewise, our metrics are stronger predictors of network performance in each case study than is the affinity selection parameter. While more investigation is necessary to improve on our affinity metrics, we believe that our work removes a major obstacle to affinity research.

At this point, we pause for a moment to underscore the distinction between the input parameter for the affinity selection algorithm and a general affinity analysis metric. We recognize that the input parameter in [19] is no more intended for analysis than our metrics are for selection. Nevertheless, in this paper we compare the effectiveness of our metrics in analyzing affinity against that of the input parameter, because the latter has been used in the aforementioned affinity studies to classify affinity groups when measuring network performance. Thus, absent any other affinity metric, and because the input parameter is the de facto classification tool for member groups, we feel it imperative to quantify the improvement of our metrics over the current standard.

The rest of the paper is organized as follows. In Section 2, we discuss the basic concepts of affinity selection and define candidate affinity metrics. We describe in Section 3 the networks we use and their effect on affinity selection. We also demonstrate the value of using an affinity metric to analyze member groups. In Section 4 we present three case studies, in which we compare the effectiveness of our metric versus that of the input parameter in explaining network performance differences. Finally, we present our conclusions in Section 5.

2. METRIC DEFINITIONS

In this section, we first define the relevant graph metrics to our work. We next describe the affinity selection process in [19], and finally offer several candidate metrics for affinity analysis.

2.1 General graph metrics

Given a connected graph \mathbb{G} , define N as the number of nodes and E as the number of edges. Let d_{ij} be the minimum distance from node i to node j . We use D to denote the diameter of the network. The characteristic path length of the network, $cpl_{\mathbb{G}}$, is the average distance between all pairs of nodes. That is,

$$cpl_{\mathbb{G}} = \frac{\sum_{i=1}^N \sum_{j=1}^N d_{ij}}{N(N-1)} \quad (1)$$

If \mathbb{S} is a group of nodes contained in \mathbb{G} , then let M be the size of \mathbb{S} . We denote $d_{i\mathbb{S}}$ as the minimum distance from node i to any node in \mathbb{S} , and $cpl_{\mathbb{S}}$ as the average path length of \mathbb{S} .

2.2 Affinity selection

Probabilistic affinity selection is a method in which a node is selected to \mathbb{S} based on its distance to nodes already in \mathbb{S} . The word probabilistic underscores that there is some randomness to the selection process. The affinity algorithm in [19], which we also use, has an input parameter β representing the desired level of affinity or disaffinity for \mathbb{S} . Positive values of β cluster nodes together, negative values decluster nodes, and the value $\beta = 0$ is equivalent to randomly selecting nodes. Previous studies have used the range $[-15, 15]$ for β . We do the same for our case studies, although to find the true extreme boundaries of affinity and disaffinity for each network, in Section 3 we expand the range of β to $[-50, 50]$.

Once the first node is selected to \mathbb{S} , then for each node i , the probability p_i of selecting it to \mathbb{S} is

$$p_i = \frac{a}{(d_{i\mathbb{S}})^\beta} \quad (2)$$

The constant a is chosen so that $\sum_{i=1}^N p_i = 1$. Each time a new node joins \mathbb{S} , the p_i 's are recalculated. Observe that for $\beta > 0$, p_i is maximized when $d_{i\mathbb{S}}$ is small, so \mathbb{S} is likely to be clustered. Conversely, for $\beta < 0$, p_i is maximized when $d_{i\mathbb{S}}$ is large, increasing the chance that nodes in \mathbb{S} will be spread apart. In general, the number of nodes selected is kept relatively small, for the effects of clustering or declustering become less noticeable as the size of \mathbb{S} increases².

2.3 Affinity metrics

Following are the candidate metrics we consider for analyzing the true affinity of \mathbb{S} .

2.3.1 λ

We define λ as the average path length of a selection group, divided by the characteristic path length. In other words,

$$\lambda = \frac{cpl_{\mathbb{S}}}{cpl_{\mathbb{G}}} \quad (3)$$

The range of values for λ is the interval $[\frac{1}{cpl_{\mathbb{G}}}, \frac{D}{cpl_{\mathbb{G}}}]$. When \mathbb{S} is clustered, the average path length of \mathbb{S} is small, so λ is also small. As nodes become more spread apart in \mathbb{S} , λ increases.

Note that as $M \rightarrow N$, $cpl_{\mathbb{S}} \rightarrow cpl_{\mathbb{G}}$, so $\lambda \rightarrow 1$. Also, the extreme values for λ occur when $M = 2$, although the maximum value could occur for larger M , also.

2.3.2 ρ

We define ρ as the number of paths in \mathbb{S} longer than the characteristic path length of the network, divided by the number of paths in \mathbb{S} . That is,

$$\rho = \frac{2L_{\mathbb{S}}}{M(M-1)} \quad (4)$$

²There are ways to modify the affinity selection algorithm. For example, if M members are to be selected, [21] first selects \sqrt{M} members with extreme disaffinity, and then for each of those members selects $\sqrt{M} - 1$ members with extreme affinity. The intention is to create \sqrt{M} clusters, each one being as far apart from the others as possible. One logical extension of this idea would be to vary the number of initial declustered members, from 1 to $\frac{M}{2}$.

where L_S is the number of path lengths in \mathbb{S} greater than $cpl_{\mathbb{G}}$. The range of values for ρ is the interval $[0, 1]$. Like λ , small values of ρ indicate more clustered groups.

2.3.3 ψ

We define ψ as the diameter of \mathbb{S} divided by the diameter of \mathbb{G} . That is,

$$\psi = \frac{D_{\mathbb{S}}}{D_{\mathbb{G}}} \quad (5)$$

The range of values for ψ is the interval $[\frac{1}{D}, 1]$. Like the previous two metrics, small values of ψ indicate more clustered groups.

2.3.4 χ

Let $E_{\mathbb{S}}(p)$ be the average number of hops required for a node in \mathbb{S} to reach $p\%$ of the nodes in \mathbb{S} . Let $E_{\mathbb{G}}(q)$ be the average number of hops required for a node in \mathbb{G} to reach $q\%$ of all nodes in \mathbb{G} . Then

$$\chi = \frac{E_{\mathbb{S}}(50\%)}{E_{\mathbb{G}}(50\%)} \quad (6)$$

The range of possible values for χ is $(0, D]$. When $\chi < 1$, nodes in \mathbb{S} reach each other more quickly than do nodes in \mathbb{G} , indicating that \mathbb{S} is clustered. Conversely, when $\chi > 1$, nodes in \mathbb{S} must travel farther to reach each other, so \mathbb{S} is declustered.

χ is inspired by expansion analysis in [25]. Expansion calculates the reachability of nodes within h hops of each other, where h varies from 1 to D . Our goal for χ is to summarize expansion into a single number by examining the shift of the cumulative distribution graphs at the median point. Of course, a value other than 50% could also be compared.

2.3.5 Candidate metric selection

After investigating and comparing each metric, the results suggest that λ and χ are the most effective, in terms of analyzing affinity and explaining network variation. We also discover that λ and χ are highly correlated. The reason is that χ , although intended to measure expansion, is a function of median path distances. In the interest of brevity, we present results only for λ in the remaining sections of this paper. We believe, however, that significantly more work is needed to identify additional affinity metrics, particularly those that can better capture expansion tendencies of a network.

3. TOPOLOGICAL ANALYSIS

In this section, we describe the topologies that we use for analysis, and then analyze affinity groups across different networks with respect to λ and β .

3.1 Networks

In general, we consider three types of networks: *measured*, *generated*, and *canonical*. We treat all networks as undirected graphs with edges of equal length.

The measured networks are Mercator [10], NLANR [1], UCB [24], and UMich [7]. The Mercator network [10] is a router-level Internet map obtained through hop-limited traceroute-like probes. The NLANR map [1] is obtained by processing BGP routing tables collected from a route server with

BGP connections to multiple geographically distributed target operational routers. The UCB network [24] is an AS-level Internet map obtained in 2001 by combining partial views of the AS graph from multiple strategically located BGP routers across the globe. The UMich network [7] is an AS-level connectivity graph inferred from data obtained through the Oregon Route-views project.

We generate four networks, each with an average node degree of five. Power Law and Hierarchical networks come from the BRITE topology generator [17]. The Power Law topology is the output of the Router Barabasi-Albert (BA) [3] model. Because we are not concerned with physical placement or varying bandwidth, the only parameter we adjust is that which guarantees average degree of five. Hierarchical is created using the top-down approach, with 100 autonomous systems containing 100 nodes each. Both intra and inter domain connectivity is assigned using the Router Waxman model.

We generate two other networks on our own: Smallworld, and Random. The Smallworld network is based on the techniques developed by Watts *et al.* [28]. To create a small-world, we start with a ring-like network, in which each node is connected to its five nearest neighbors. Then, for each edge we reconnect it with probability p to another node in the network. We select the value for p that maximizes the small-world effect: reduce the overall characteristic path length, but keep nodes connected primarily to their local neighbors. In our case, the optimum value for p is 0.01. Finally, the Random network is generated by randomly assigning edges to pairs of nodes, provided that each node ultimately has degree greater than zero and that the network comprises only one component.

We construct three canonical networks: Ring, 2D, and K-ary. In the ring, each node is connected only to its two nearest neighbors on either side. 2D places nodes in rows and columns that form a square grid. K-ary is a tree network in which each node is connected to four children below and one parent above, with the exception of the root node at the top and leaf nodes at the bottom. Canonical networks are important because they are similar to topologies constructed by many overlay networks.

For our case studies, we generate networks in which N is 10000, although for a few networks, N varies to preserve the canonical structure of the graph. Table 1 lists the networks and their summary statistics.

Past affinity studies have varied β within the interval $[-15, 15]$. For our case studies, we do the same by generating member groups for $\beta = (-15, -3, -1, 0, 1, 3, 15)$ for each network. We vary M from 0.1% to 10% of N , although we focus on smaller receiver groups when affinity variations are most noticeable. For each M_i and β_j , we generate ten independent member groups and report the averages for our affinity metrics and network performance measurements. Thus, when we plot network performance against either β or λ for a given M , we will have exactly seven data points for each network, although each of those data points is itself an average of ten independent values.

Network	N	E	D	cpl	Degree
MEASURED					
Mercator	27646	67310	26	8.28	4.87
NLANR	6475	13895	9	3.71	4.29
UCB	13836	29076	10	3.66	4.20
UMich	10901	31180	9	3.57	5.72
GENERATED					
Power Law	10000	25000	6	3.64	5.00
Hierarchical	10000	25000	33	16.04	5.00
Random	10000	25000	12	5.96	5.00
Smallworld	10020	25050	56	22.09	5.00
CANONICAL					
2D	10000	19800	198	66.67	3.96
K-ary	5461	5460	12	10.67	2.00
Ring	10000	10000	5000	2500.25	2.00

Table 1: Networks used in case studies

3.2 General metrics analysis by topology

Previous studies have examined network performance for different values of β . One problem with this approach is that it implies an equivalence of clusteredness among selection groups with the same β from different topologies. Furthermore, if β is unknown or was not used, then an analysis metric becomes essential to characterize group affinity.

We examine maximum and minimum values of λ for $M = 0.1\%$ and $M = 1\%$ and present them in Table 2. For this analysis, we expand β beyond the usual interval $[-15, 15]$ and instead let β vary between $[-50, 50]$, to determine whether the boundaries of affinity selection are uniform across all networks.

From Table 2, we observe very different ranges of λ across the various networks, both in terms of size and absolute value. When $M = 0.1\%$, the measured networks have the four highest values of λ_{max} , each of which is at or near 2. Therefore, the effect of declustering would be most noticeable on these networks. Power Law has by far the lowest observed range of λ for either M , which indicates that on this network extreme affinity and disaffinity groups are only moderately different from each other. Also, increasing M from 0.1% to 1% does not have a uniform effect across all networks, although in general declustering is more adversely affected than is clustering. Smallworld has the most similar range of λ to the measured networks.

A high value for λ_{min} suggests that clustering is not significantly different from random selection. When $M = 0.1\%$, for example, the average path length of the most clustered Power Law group is only 63% smaller than $cpl_{\mathbb{G}}$. We compare that result to the Ring network, in which $cpl_{\mathbb{S}}$ can be made more than 600 times smaller than $cpl_{\mathbb{G}}$.

Conversely, a low value for λ_{max} suggests that declustering is not significantly different from random selection. For example, on the Ring network, λ_{max} is barely greater than 1, indicating that even when $\beta = -50$, groups appear randomly distributed. However, when $M = 2$, $\lambda_{max} = \frac{5000}{2500.25} \approx 2$, so declustering is possible on the Ring but only for very small M .

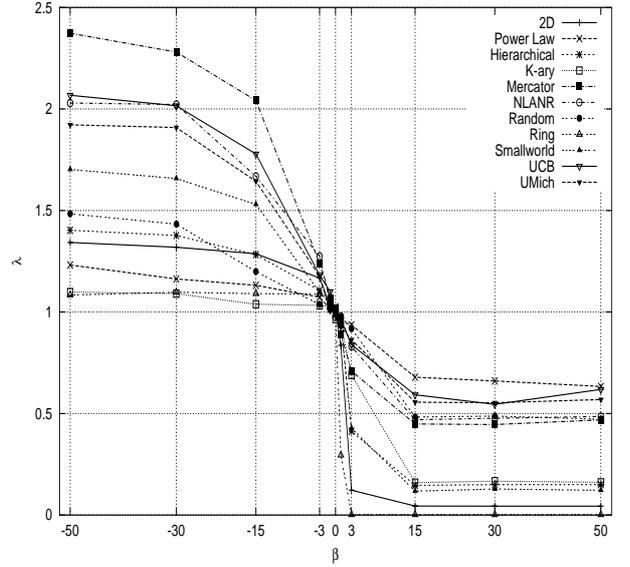


Figure 1: λ vs β for $M = 0.1\%$

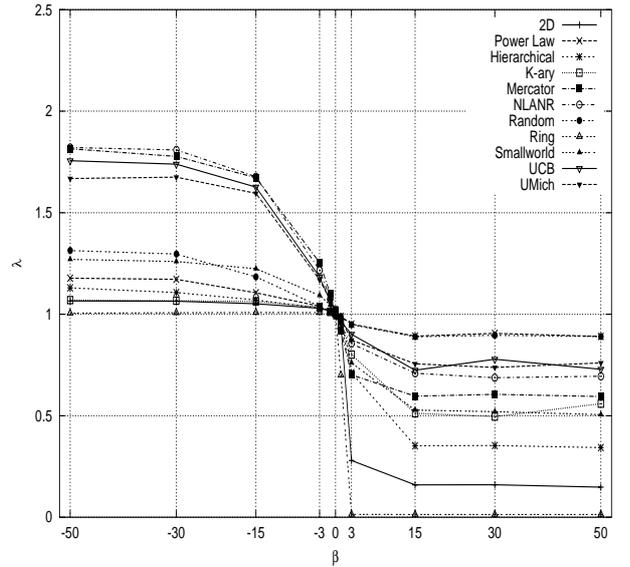


Figure 2: λ vs β for $M = 1\%$

The diameter of a network can have a big effect on the range of values for λ . When the diameter is small, declustering is difficult because members cannot get sufficiently far apart from each other. Likewise, extreme affinity is limited since nodes in \mathbb{G} are already very close to each other.

However, diameter is not the only relevant factor. The Ring network, for instance, has the largest diameter, yet all other networks attain higher values of λ_{max} . Thus, overall topology plays an important role in the affinity selection process.

As previously mentioned, yet another effect on λ is M , which we graphically demonstrate in Figures 1 ($M = 0.1\%$) and 2 ($M = 1\%$). We observe that as $M \rightarrow N$, λ decreases when $\beta < 0$, and λ increases when $\beta > 0$. Hence, λ is converging

Network	$M = 0.1\%$		$M = 1\%$	
	λ_{max}	λ_{min}	λ_{max}	λ_{min}
Mercator	2.3730	0.4691	1.8150	0.5947
UCB	2.0677	0.6188	1.7558	0.7288
NLANR	2.0296	0.4858	1.8209	0.6944
UMich	1.9224	0.5689	1.6683	0.7603
Smallworld	1.7022	0.1217	1.2700	0.5053
Random	1.4837	0.4724	1.3134	0.8898
Hierarchy	1.4027	0.1500	1.1296	0.3429
2D	1.3427	0.0436	1.0653	0.1480
Power Law	1.2312	0.6336	1.1778	0.8905
K-ary	1.0983	0.1612	1.0697	0.5595
Ring	1.0837	0.0015	1.0053	0.0135

Table 2: Maximum and minimum values of λ when $M = 0.1\%$ and $M = 1\%$, sorted by λ_{max} for $M = 0.1\%$

to 1. However, the exact change in λ appears to be network-specific.

From these figures, we make two other important observations. First, the range of β that produces groups with distinguishable affinity varies for each network. As an example, when $\beta = -15$ all four measured networks have a negative slope, indicating more extreme declustering is still possible by lowering β to -30 or even -50 . By comparison, the slope for most other networks is much closer to 0, and in some cases has already reached 0. Thus, the rate of change in λ as β varies is not constant across all networks. Second, the actual value of λ for a given β depends on the network. Both these observations underscore the limitation of β as a classification tool for affinity groups, even when it is known.

Finally, to demonstrate the importance of having an affinity metric for real-world data, we calculate λ for the sets of Web clients as described in [22]. We observe that for each of these sets, λ is very close to 1, indicating that these groups appear randomly distributed. Given that those clients are not known to have any affiliation with each other, these results seem intuitive.

In summary, to answer our first set of questions from Section 1, it is not possible to generate selection groups with the same degree of affinity or disaffinity on every network. Extreme affinity and disaffinity do not appear to be universal values, but rather network-specific boundaries that are a function of topology. Finally, we can characterize true affinity by using a metric like λ to analyze characteristics of the selection group.

One could argue that because many of the networks have so little in common, it is not surprising that affinity varies widely for a fixed β . While this may be true, we again point out that prior work has avoided this issue, instead classifying member groups from different networks by their respective β values. Moreover, in past research, the upper and lower bounds of β have been fixed across all networks. We have demonstrated that achieving the full range of affinity variation requires choosing network-specific bounds. In the following section, we demonstrate three examples in which affinity analysis metrics are not only relevant but also useful in explaining variation of application performance.

4. CASE STUDIES

In this section, we present three case studies to demonstrate the value of affinity analysis in explaining application performance. We compare the predictive power of λ against that of β in an attempt to quantify this value.

4.1 Multicast

In its most abstract form, multicast is a method for lowering the cost of information distribution. Instead of having a communication source send data individually to each recipient, multicast enlists intermediate network routers or receivers to pass along data to other receivers farther away. Multicast has the potential to improve the efficiency and cost of numerous content-rich transmissions, including webcasts and teleconferences.

Beginning with [8] and [19], several studies have developed methods to quantify the savings of multicast. To do so requires first constructing the shortest path tree that connects the source and all receivers, and then calculating the cost to send one packet to all receivers. Most studies assume a unit cost over any edge, so that the cost from node i to node j equals d_{ij} ; we do the same here. Although many possible performance metrics exist, we use δ as defined in [6]. If L_m is the number of hops in the shortest path tree connecting S , and L_u is the sum of all unicast distances from the source to each receiver, then

$$\delta = 1 - \frac{L_m}{L_u} \quad (7)$$

Note that δ is bounded between 0 and 1. Moreover, small values of δ indicate that the multicast tree is close in size to the sum of individual unicast paths, suggesting minimal if any efficiency gains from multicast. Conversely, as δ approaches 1, the multicast tree becomes much smaller, so efficiency gains are more substantial.

Since multicast efficiency is a function of receiver distribution, it is logical to examine application performance against receiver group affinity. It is not difficult to show that clustered receivers improve multicast efficiency, as [19] has already done. A simple explanation for this conclusion lies in graph theory; it requires a larger tree to connect scattered receivers than it does clustered ones. Our goal in this case study is to see whether λ can help explain the observed differences in δ for receiver groups with different degrees of

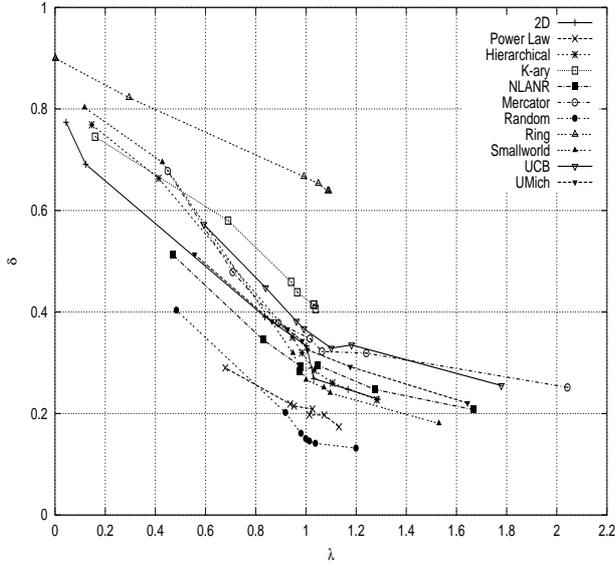


Figure 3: δ vs λ for $M = 0.1\%$

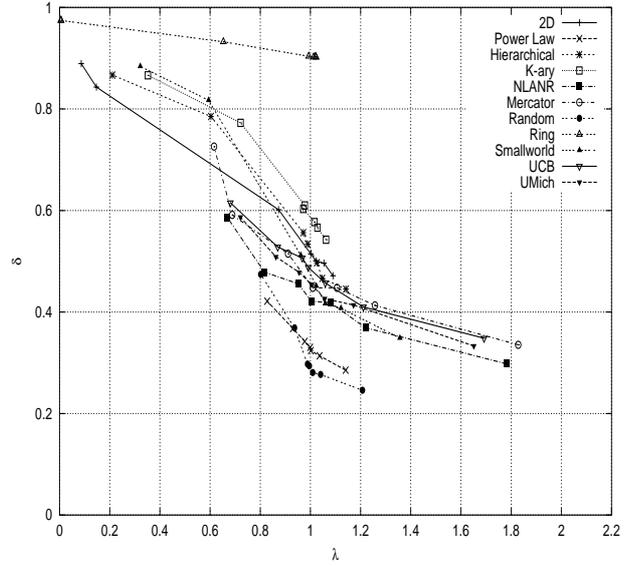


Figure 5: δ vs λ for $M = 0.4\%$

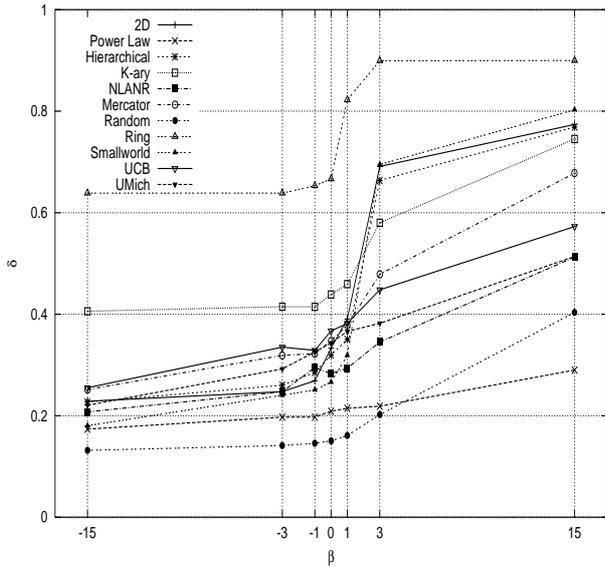


Figure 4: δ vs β for $M = 0.1\%$

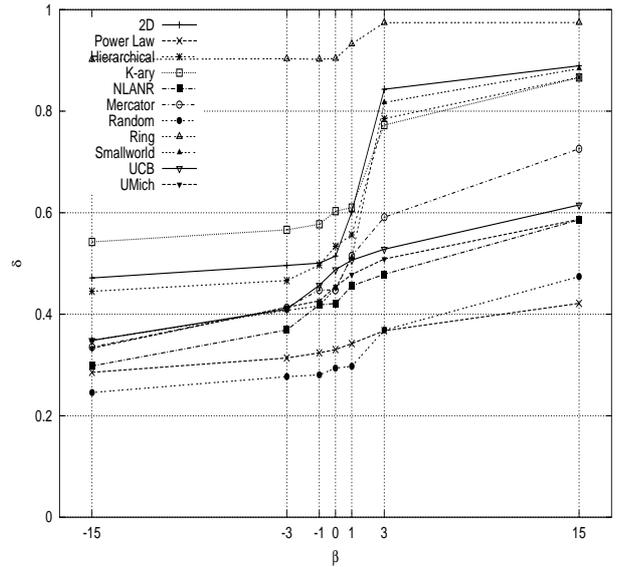


Figure 6: δ vs β for $M = 0.4\%$

affinity.

For each receiver group, we average the results for ten different sources to minimize the effect of source placement. Figures 3 and 4 plot δ versus both λ and β for $M = 0.1\%$; Figures 5 and 6 do the same for $M = 0.4\%$.

We observe reasonably good linear fits between δ and λ across all networks. We also observe a knee in the graphs when $\lambda = 1$. This knee indicates that for declustered receivers, increasing disaffinity reduces efficiency at a lower rate than increasing affinity improves efficiency for clustered groups.

When we examine δ versus β , we see a different picture. For

some networks, making β more negative has virtually no effect. Additionally, the actual value for δ across all networks varies widely for a given β .

To compare the predictive powers of λ and β , we run regression on the simple linear models $\delta = a\lambda + b$ and $\delta = c\beta + d$ for $M = 0.1\%$ and $M = 0.4\%$. We test the goodness of fit for each model by comparing the R^2 values; we present the results in Table 3. In Table 4, we examine the aggregated fit for the system of topology-specific linear models.

In every case λ is a better predictor for explaining efficiency variations than is β . That is, knowing something about the clusteredness of a member group is preferable to merely knowing the parameter that was used to generate the group.

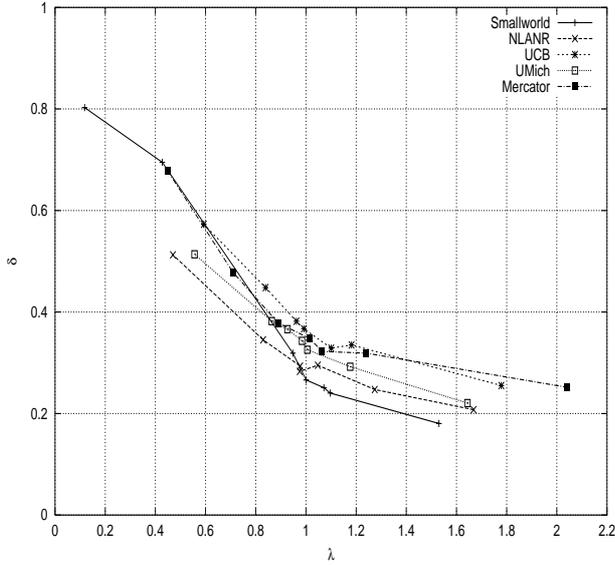


Figure 7: δ vs λ for Measured and Smallworld

Finally, in Table 2, we saw that Smallworld and measured networks had similar ranges of λ . In Figure 7 we continue this comparison further by plotting δ versus λ for just these networks when $M = 0.1\%$. We observe that the expected multicast efficiency for a given λ is remarkably similar between the measured and Smallworld networks.

Model	R^2 ($M = 0.1\%$)	R^2 ($M = 0.4\%$)
$\delta = a\lambda + b$	0.5618	0.4690
$\delta = c\beta + d$	0.2375	0.1403

Table 3: R^2 values for one model across all networks

Model	R^2 ($M = 0.1\%$)	R^2 ($M = 0.4\%$)
$\delta = a_i\lambda + b_i$	0.9208	0.9094
$\delta = c_i\beta + d_i$	0.8178	0.8856

Table 4: R^2 values for network-specific models

4.2 Replica placement

Server replica placement algorithms are an essential component in the design of Content Delivery Networks, which aim to improve network performance. Replica placement algorithms determine the appropriate location of storage nodes, where Internet content is duplicated, in order to minimize the cost of improving network performance. Numerous studies have investigated the impact of different replica placement methods on network performance, notably [13], [20], and [22]. Qiu *et al.* propose a *greedy* placement strategy in [20], achieving performance comparable to the computationally expensive optimal situation. *Fanout-based* placement strategies, proposed in [13] and [22], utilize information gleaned from the underlying network topology.

It is plausible that the relative degree of clustering of web-clients should affect the network performance. Analogous to multicast, clustered clients should improve the performance of replica-placement algorithms even further. Hence we think it is important to examine in some detail the effect

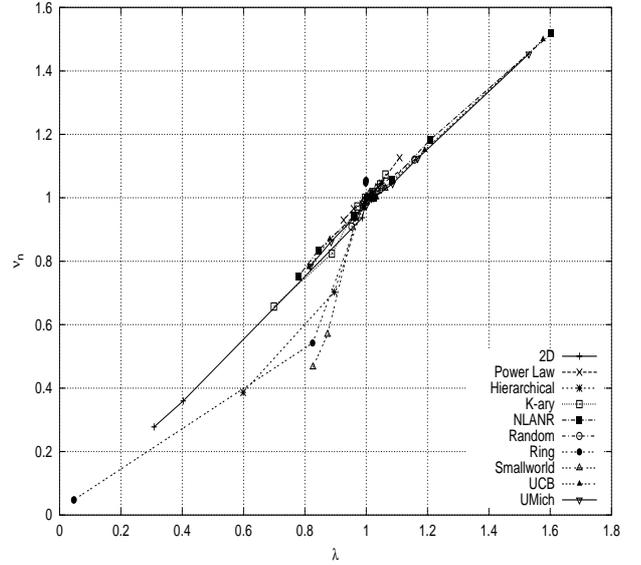


Figure 8: ν_n vs λ for $M = 4\%$

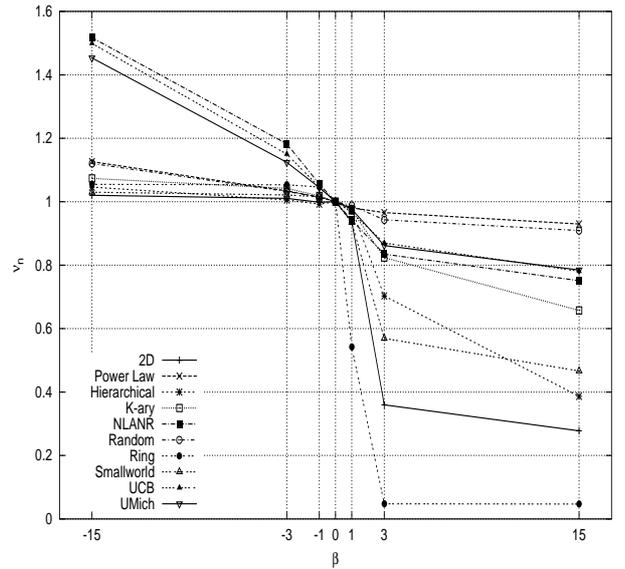


Figure 9: ν_n vs β for $M = 4\%$

of client-group affinity and the underlying network topology on the performance of replica placement algorithms.

An intuitive metric for quantifying the performance of replica placement algorithms is the total network overhead, which we denote ν . Let M be the number of web-clients and k be the number of replicas. We assume that each client i selects the closest replica $r(i)$ and that its contribution to the overhead is the distance $d_{i,r(i)}$. Hence the total overhead is given by [22]:

$$\nu(\beta) = \sum_{i=1}^M d_{i,r(i)} \quad (8)$$

To facilitate meaningful comparison between the ν values obtained for the various network topologies we have studied,

it is necessary to normalize ν to neutralize the effect of the underlying topology on the absolute value of the overhead. Thus, we propose a normalized overhead metric ν_n :

$$\nu_n(\beta) = \frac{\nu(\beta)}{\nu(0)} \quad (9)$$

The replica placement study uses the same networks and β values from the multicast study, for $M = 0.4\%$. In the interest of brevity, we only present results for the *greedy* placement strategy [20], but we expect that our results can be easily extended to other replica placement algorithms. We fix the value of k at 5 because Jamin *et al.* [13] show that increasing the number of replicas improves performance only for small values of k .

Figure 8 shows the variation of ν_n with λ ; Figure 9 does the same for β . For the β graph, we observe that the various topologies can be divided into two sets: the measured topologies, which show a significant improvement in performance as β increases ($\frac{\nu(\beta=15)}{\nu(\beta=1)} \approx 2$), and the generated/canonical topologies, for which performance improvement is significantly less. There is, however, significant variation in these trends across the various networks. In other words, β does not appear to be a strong predictor for ν_n . In contrast, Figure 8 suggests a very good linear fit between ν_n and λ across all networks. Like the multicast case, we observe a knee in the graphs at $\lambda = 1$.

We have confirmed the power of λ versus that of β in predicting ν_n by computing the R^2 values of regressions and present the results in Tables 5 and 6. The R^2 values for λ are significantly greater than those for β . We conclude that for the case of replica placement, λ outperforms β in explaining network overhead variations as client affinity changes.

Model	R^2 ($M = 4.0\%$)
$\nu = a\lambda + b$	0.8020
$\nu = c\beta + d$	0.4580

Table 5: R^2 values for one model across all networks

Model	R^2 ($M = 4.0\%$)
$\nu = a_i\lambda + b_i$	0.9282
$\nu = c_i\beta + d_i$	0.6513

Table 6: R^2 values for network specific models

4.3 Sensor networks

Our third case study examines affinity as it relates to the performance of wireless sensor networks. Wireless sensor networks have emerged as a promising solution for a large number of monitoring applications, in fields as diverse as climatic monitoring, tactical surveillance, and earthquake detection. Such networks often have finite lifetimes because nodes have limited battery power. Thus, unlike the previous two case studies, sensor network topologies can change over time as nodes fail or become disconnected. A typical sensor network consists of a relatively large number of wireless sensing *sources*, which use hop-by-hop radio communication to send their readings back to one or more base station *sinks*.

Parameter	Value
Transmit power	660 mW
Receive power	395 mW
Idle power	35 mW
Transmit time	50 ms
Receive time	65 ms
Initial sensor energy	100 J
Radio range	40m

Table 7: Sensor network simulation parameters

Numerous performance improving protocols have been suggested, such as Directed Diffusion [12], in which energy expenditure is lowered by using in-network data aggregation and positive reinforcement of cost-effective paths.

The dynamic nature of sensor networks also suggests that non-random source placement is a relevant analysis consideration. Suppose, for example, that a large number of sensors are clustered in a network that implements limited or no data aggregation. Then, the resulting high data rate of transmissions originating from the cluster could lead to quicker failures and increased packet loss in that region, as well as higher overall energy costs as shortest path lengths between sensors and sinks increase. While we know of one sensor network study that has considered non-random source placement [26], we are not aware of any studies that have sought to quantify the functional relationship between source affinity and network performance.

Numerous metrics exist for analyzing network performance, including energy efficiency, packet loss ratio, node failure rate, and overall network lifetime. However, our goal for this case study is to develop a metric analogous to δ from the multicast study. δ is a robust tool for comparing network performance between different scenarios because it is both protocol-independent and dimensionless. We therefore define a decay-based metric γ as follows. Let f_1 be the time at which the first source node fails. Let f_m be the time when the network fails. Then,

$$\gamma = \frac{f_1}{f_m} \quad (10)$$

Node failure means either loss of battery power or disconnection from the sink, while network failure signifies that the sink no longer receives messages from any source. Thus, at time f_m one or more source nodes could still be alive, but because of intermediate node failures, their messages no longer reach the sink.

The range of values for γ is the interval $(0, 1]$. When γ is close to 0, the first source failure occurs early relative to overall network lifetime. This early decay implies a long period of sub-optimal network performance. On the other hand, when γ is close to 1, the first source failure occurs near the end of the network lifetime. Thus, the network performs optimally for a longer period, followed by a brief period of severe decay.

For this study, we build a lightweight simulator without radio details. We assume stationary nodes with equal start-

ing energy levels in a network with a single access point or *sink*. The simulation parameters listed in Table 7 are obtained from the Smartdust project [27]. We simulate periodic packet transmissions from each sensor to the sink. Each time an intermediate node fails, we reroute packets along the shortest available path. For simplicity, we assume that nodes expend energy only to send and receive data packets from sources; at all other times, nodes are idle. Thus, network administration, such as shortest path recalculation, occurs automatically without additional energy expenditure from individual nodes.

We run our simulations over two types of topologies: Random and 2D grids, the most common placement strategies seen in sensor network literature. Each network contains 900 nodes; we vary the area of the sensor network to generate multiple topologies with distinct node densities. Table 8 gives summary information about each network.

Since γ is a function of the first node failure, it is subject to outlier behavior effects. Therefore, we average γ not only across ten different sink placements, but also across ten different sets of sources for each of those sinks. Thus, γ is actually an average of 100 different iterations for a given β , M , and network area.

We examine how the clusteredness of sources in a sensor network affects decay by plotting the performance metric γ versus both λ and β in Figures 10 and 11. As in the previous two studies, we observe that the range of values for the performance metric are dependent on the topology. Here, because less dense networks have larger diameters, these networks produce source groups with wider ranges of affinity and disaffinity, thereby causing larger ranges for γ .

We see that the length of a network's decay period varies depending on source affinity. Declassified sources increase the percentage of time that the network performs optimally. Clustering sources causes earlier decay relative to total net-

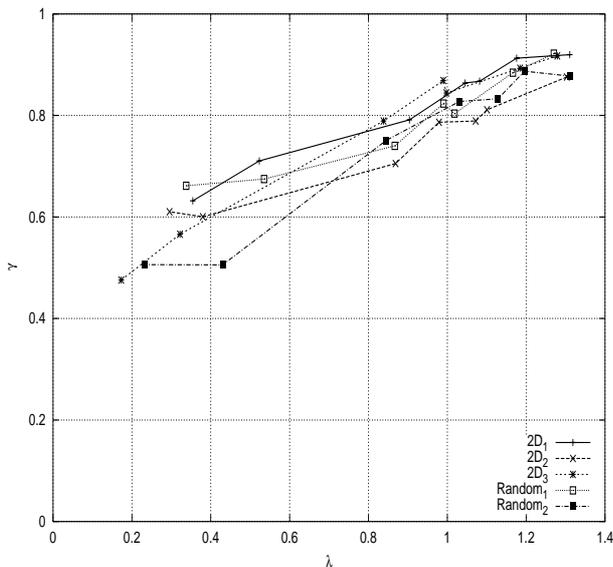


Figure 10: γ vs λ for $M = 1\%$

Model	R^2 Value
$\gamma = a\lambda + b$	0.9056
$\gamma = c\beta + d$	0.6161

Table 9: R^2 values for one model across all networks

Model	R^2 Value
$\gamma = a_i\lambda + b_i$	0.9403
$\gamma = c_i\beta + d_i$	0.6650

Table 10: R^2 values for network specific models

work lifetime. Thus, λ and γ are positively correlated; moreover, this relationship appears to be reasonably linear for all networks. On the other hand, the relationship between β and γ is less clear. In particular, for values of β near 0, the rate of change in decay does not appear equivalent for all networks.

We investigate the linearity of λ and γ , along with the relative predictive strength of λ and β , by running regression for simulation results when $M = 1\%$. We summarize the results in Tables 9 and 10 respectively.

We determine that λ does a better job than does β in explaining the length of decay of a sensor network. We observe an extremely good fit between λ and γ when each network is analyzed separately. Although our results are at an abstract level, it seems that affinity affects sensor network performance, and that affinity analysis explains some of those variations.

5. CONCLUSIONS

In this paper, we have defined several distance and expansion based metrics to quantify the true *affinity* level of member groups. We demonstrate the effectiveness of one in particular: λ , which normalizes the average path length of a group by the characteristic path length of the network. In

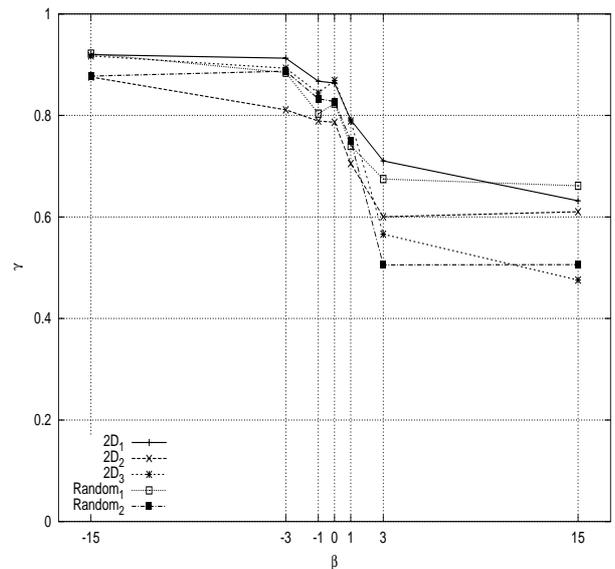


Figure 11: γ vs β for $M = 1\%$

Network	Node Density (nodes/m ²)	N	E	D	cpl	Degree
2D ₁	7.12×10^{-3}	900	14670	15	5.62	32.60
2D ₂	3.16×10^{-3}	900	8350	20	7.70	18.56
2D ₃	1.41×10^{-3}	900	3422	29	14.00	7.60
Random ₁	7.12×10^{-3}	900	14734	14	5.56	32.74
Random ₂	3.16×10^{-3}	900	6770	23	8.88	15.04

Table 8: Networks used for sensor network case study

doing so, we have illustrated the limitations of clustering and declustering on various topologies. We have shown that knowing the value of λ helps to better predict network performance in three different case studies: multicast, replica placement, and sensor networks. We have compared λ against β , demonstrating the limitations of using β for classifying affinity groups. Moreover, when β is unknown or not used, an analysis metric like λ is the only option to classify and compare groups by affinity. Future work includes identifying additional analysis metrics, refining ones we have proposed, and conducting more detailed case studies to understand even better the effect of affinity on network performance. In the long run, network research will benefit as affinity modelling improves, and we believe that our work will enable more realistic and comprehensive affinity techniques.

6. ACKNOWLEDGMENTS

We would like to thank Pavlin Radoslavov for providing us with both data to validate our replica placement case study, as well as the groups of real-world web clients used for metric validation (Section 3.2). This research was supported at the University of California at Berkeley by the National Science Foundation under Grant No. 0085879.

7. REFERENCES

- [1] NLANR global as interconnectivity graphs. <http://moat.nlanr.net/AS/>, 2003.
- [2] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *Proceedings of the ACM Symposium on Theory of Computing*, pages 171–180, 2000.
- [3] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, Oct. 1999.
- [4] P. Barford, J.-Y. Cai, and J. Gast. Cache placement methods based on client demand clustering. In *Proceedings of IEEE INFOCOM*, pages 41–50, Apr. 22–26 2001.
- [5] T. Bu and D. Towsley. On distinguishing between internet power law topology generators. In *Proceedings of IEEE INFOCOM*, pages 1587–1596, June 23–27 2002.
- [6] R. Chalmers and K. Almeroth. Modeling the branching characteristics and efficiency gains of global multicast trees. In *Proceedings of IEEE INFOCOM*, pages 449–458, Apr. 22–26 2001.
- [7] H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger. Towards capturing representative AS-Level internet topologies. In *Proceedings of ACM SIGMETRICS*, pages 280–281, June 15–19 2002.
- [8] J. C.-I. Chuang and M. A. Sirbu. Pricing multicast communication: A cost-based approach. *Telecommunication Systems*, 17(3):281–297, 2001.
- [9] M. Doar. A better model for generating test networks. In *Proceedings of IEEE GLOBECOM*, Nov. 10–12 1996.
- [10] R. Govindan and H. Tangmunarunkit. Heuristics for internet map discovery. In *Proceedings of IEEE INFOCOM*, pages 1371–1380, Mar. 26–30 2000.
- [11] X. He and C. Papadopoulos. A framework for incremental deployment strategies for router-assisted services. In *Proceedings of IEEE INFOCOM*, pages 1301–1310, Apr. 1–3 2003.
- [12] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of ACM MOBICOM*, pages 56–67, Aug. 6–11 2000.
- [13] S. Jamin, C. Jin, A. Kurc, Y. Shavitt, and D. Raz. Constrained mirror placement on the internet. In *Proceedings of IEEE INFOCOM*, pages 31–40, Apr. 22–26 2001.
- [14] C. Jin, Q. Chen, and S. Jamin. Inet: Internet topology generator. Technical Report Report CSE-TR443-00, Department of EECS, University of Michigan, 2000.
- [15] B. Krishnamurthy and J. Wang. On network-aware clustering of web clients. In *Proceedings of ACM SIGCOMM*, pages 97–110, 2000.
- [16] B. Krishnamurthy and J. Wang. Topology modeling via cluster graphs. In *Proceedings of the First ACM SIGCOMM Internet Measurement Workshop (IMW-01)*, pages 19–24, Nov. 1–2 2001.
- [17] A. Medina, I. Matta, and J. Byers. BRITE: A flexible generator of internet topologies. Technical Report 2000-005, CS Department, Boston University, Jan. 21 2000.
- [18] M. Mihail, C. Gkantsidis, A. Saberi, and E. Zegura. On the semantics of internet topologies. Technical Report GIT-CC-02-07, Georgia Institute of Technology, 2002.
- [19] G. Phillips, S. Shenker, and H. Tangmunarunkit. Scaling of multicast trees: Comments on the chuang-sirbu scaling law. In *Proceedings of ACM SIGCOMM*, pages 41–52, 1999.

- [20] L. Qiu, V. Padmanabhan, and G. Voelker. On the placement of web server replicas. In *Proceedings of IEEE INFOCOM*, pages 1587–1596, Apr. 22–26 2001.
- [21] P. Radoslavov. *The Relationship Between Topology and Protocol Performance: Case Studies*. PhD thesis, University of Southern California, 2001.
- [22] P. Radoslavov, R. Govindan, and D. Estrin. Topology-Informed internet replica placement. In *Proceedings of the Sixth International Workshop on Web Caching and Content Distribution*, pages 229–238, June 20–22 2001.
- [23] P. Radoslavov, H. Tangmunarunkit, H. Yu, R. Govindan, S. Shenker, and D. Estrin. On characterizing network topologies and analyzing their impact on protocol design. Technical Report USC-CS-TR-00-731, University of Southern California, 2000.
- [24] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the internet hierarchy from multiple vantage points. In *Proceedings of IEEE INFOCOM*, June 23–27 2002.
- [25] H. Tangmunarunkit, R. Govindan, S. Shenker, S. Jamin, and W. Willinger. Network topology generators: Degree-based vs. structural. In *Proceedings of ACM SIGCOMM*, pages 147–160, Aug. 19–23 2002.
- [26] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman. Infrastructure tradeoffs for sensor networks. In *Proceedings of the ACM 1st International Workshop on Sensor Networks and Applications (WSNA '02)*, pages 65–77, Sept. 16–17 2002.
- [27] B. Warneke, M. Last, B. Liebowitz, and K. Pister. Smart dust: Communicating with a cubic-millimeter computer. *IEEE Computer*, pages 44–51, Jan. 2001.
- [28] D. J. Watts and S. H. Strogatz. Collective dynamics of small world networks. *Nature*, 393(6684):397–498, June 1998.
- [29] B. M. Waxman. Routing of multipoint connections. *IEEE Jour. Selected Areas in Communications (Special Issue: Broadband Packet Communications)*, 6(9):1617–1622, Dec. 1988.
- [30] T. Wong, R. H. Katz, and S. McCanne. An evaluation on using preference clustering in large-scale multicast applications. In *Proceedings of IEEE INFOCOM*, pages 451–460, Mar. 26–30 2000.
- [31] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee. How to model an internetwork. In *Proceedings of IEEE INFOCOM*, volume 2, pages 594–602, March 1996.