Scaling of Multicast Trees: Comments on the Chuang-Sirbu scaling law *

Graham Phillips
USC/Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
graham@isi.edu

Scott Shenker
International Computer Science Institute
1947 Center Street
Berkeley, CA 94704
shenker@icsi.berkeley.edu

Hongsuda Tangmunarunkit
USC/Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
hongsuda@isi.edu

Abstract

One of the many benefits of multicast, when compared to traditional unicast, is that multicast reduces the overall network load. While the importance of multicast is beyond dispute, there have been surprisingly few attempts to quantify multicast's reduction in overall network load. The only substantial and quantitative effort we are aware of is that of Chuang and Sirbu [3]. They calculate the number of links L in a multicast delivery tree connecting a random source to m random and distinct network sites; extensive simulations over a range of networks suggest that $L(m) \propto m^{0.8}$. In this paper we examine the function $L(m)$ in more detail and derive the asymptotic form for $L(m)$ in k-ary trees. These results suggest one possible explanation for the universality of the Chuang-Sirbu scaling behavior.

1 Introduction

IP multicast routing was proposed in [4, 5, 6]. While there have been frustrating delays in its widespread commercial deployment, experience with multicast in the MBone [2] and other experimental settings has provided convincing evidence of multicast's power and elegance. From a conceptual viewpoint, multicast offers applications a novel logical rendezvous mechanism whereby senders transmit to a logical address, and receivers join a logical group, and there is no need for coordination about group membership. From an efficiency perspective, multicast offers two main advantages. Multicast greatly alleviates the overhead on senders by allowing them to reach the entire group with the transmission of a single packet. In addition, multicast routing ensures that no more than one copy of each packet will traverse each link, thereby significantly reducing the overall network load. This overall network efficiency gain, while probably the least important of the three multicast benefits listed above, is the most obvious and the most widely cited one in the popular (*i.e.*, non-research) literature.

It is somewhat surprising that, given its centrality in the lore of multicast, there have been so few efforts to accurately characterize this reduction in overall network load. The only substantial effort we are aware of is by Chuang and Sirbu in [3]. They examined the number of links L in the multicast tree needed to reach m randomly chosen distinct network locations from a given source. The nature of this function $L(m)$ embodies the overall network efficiency properties of multicast.[1] A similarly computed quantity for unicast – the total number of link-traversals needed to reach m random network locations using unicast – yields $\hat{u}m$ where $\hat{u}$ is the length of the average unicast path; the efficiency gains of multicast are reflected in how far $L(m)$ deviates from this linear growth. The simulations presented in [3] suggest that, over a wide range of networks, $L(m)$'s behavior is reasonably described by the formula $L(m) \propto m^{0.8}$. The power law form and its apparent universality are very intriguing and somewhat surprising results. We review some of this data in Section 2.

Our goal in this paper is to determine the scaling form of $L(m)$ more precisely, and to understand the universality of this scaling behavior. To do so, we first calculate, in Section 3, the function $L(m)$ for k-ary trees. We find that the asymptotic form of $L(m)$ is not exactly a power law. Moreover, we find that the asymptotic form of $L(m)$ does not depend on the degree k of the tree (although the constants in the formula do depend on k). We suggest that this may be what gives rise to the universality of the Chuang-

*We wish to thank John Chuang for freely sharing his data, code, and his insights with us. We also wish to thank Deborah Estrin, Ramesh Govindan, and Don Towsley for helpful discussions. This research was supported at ISI by the Defense Advanced Research Projects Agency (DARPA) under grants DABT63-97-C-0049 and DABT63-98-1-0007.

[1] Here we are focusing on multicast routing algorithms that are source specific, where packets traverse the shortest path between source and receiver; we do not address the efficiency of shared tree multicast algorithms. See [12] for one such comparison.

name	type	source	nodes	links	avg. degree
ARPA	real	Early ARPANET	47	68	2.9
MBone	real	MBone (Feb 1999)	4179	8529	4.1
Internet	real	partial Internet (March 1999)	56317	77154	2.7
AS	real	AS connectivity (March 1999)	4830	9077	3.8
r100	generated	GT-ITM (random)	100	177	3.5
ts1000	generated	GT-ITM (transit-stub)	1000	1819	3.6
ts1008	generated	GT-ITM (transit-stub)	1008	3787	7.5
ti5000	generated	TIERS	5000	7084	2.8

Table 1: Description of networks used in Figure 1.

Sirbu scaling law. We then investigate the behavior of $L(m)$ on more general networks, including some real networks, in Section 4. In Section 5 we consider the effects of affinity (*i.e.*, clustering) and disaffinity of receivers. We conclude in Section 6 with a short discussion.

2 Evidence for the Chuang-Sirbu Scaling Law

We first review some experimental evidence for the Chuang-Sirbu scaling law. The data presented here is very similar in spirit to that shown in [3]. We consider the set of networks described in Table 1. The ARPA network reflects the original ARPANET topology (this topology has been used in several other studies, such as [13] and [3]); the MBone and Internet topologies are based on data collected by the SCAN project at USC/ISI; and the AS data refers to the routing connectivity between autonomous systems collected by the National Laboratory for Applied Network Research (NLANR).[2] The rest of the networks are generated either using the GT-ITM network generator [1] or the TIERS network generator [7]. The names of the generated graphs reveal their topological style: "ts" refers to transit-stub networks (generated by the GT-ITM network generator), "r" refers to random networks (generated by the GT-ITM network generator), and "ti" refers to networks generated by the TIERS network generator. These topologies represent a fairly wide variety of networks; the average degrees range from 2.7 to 7.5, the number of nodes range from 47 to 56,317, the generated networks have three different design styles, and the real networks are from different contexts and different eras. Four of these eight topologies – ARPA, r100, ts1000, and ti5000 – were used in the original Chang-Sirbu paper [3]. All topologies were "cleaned" by removing duplicate edges (most often found in the TIERS topologies) and all remaining edges were then assumed to be bi-directional.

For each network we pick a source at random. For each m, we pick $N_{\rm rcvr}$ random sets of m distinct receiver locations chosen uniformly over the network. For each random set of receiver locations we compute the size of the delivery tree[3] $L(m)$; we also compute the sum of the unicast paths for each random set of receiver locations, and average those to determine the average unicast path length $\hat{u}(m)$ for this sample of receiver locations. For each such sample we compute the ratio of the size of the delivery tree to the average unicast path length: $\frac{L(m)}{\hat{u}(m)}$. We repeat this for $N_{\rm source}$ random choices of the sources. We then average this quantity $\frac{L(m)}{\hat{u}(m)}$ (for a given m) over all random sets of receivers and random choices of source (making a total of $N_{\rm rcvr}N_{\rm source}$ data points to be averaged).[4] Typically, we set $N_{\rm rcvr} = 100$ and $N_{\rm source} = 100$ where the sources are picked with replacement.

Figure 1(a) shows data from the generated topologies, while Figure 1(b) shows data from real networks. As we can see, the fit to the relation $L(m) \propto m^{0.8}$ is by no means exact, but is remarkably good considering the variety of networks considered.

On the basis of this and similar data, Chuang and Sirbu concluded that the scaling law $L(m) \propto m^{0.8}$ was a reasonable fit to the data for a broad class of networks, and then used this expression for $L(m)$ as the basis for a multicast pricing policy. No one had previously conjectured that $L(m)$ had a universal form, much less this particular universal form, and so the existence of such a simple and nonintuitive formula capturing such a wide range of networks is a remarkable result. While clearly not a precise fit, as a rough empirical rule of thumb the relation $L(m) \propto m^{0.8}$ works quite well, and the formula is certainly sufficiently accurate for the practical purpose – a multicast pricing policy – for which it was originally intended.

The focus of this paper, however, is more abstract and esoteric than practical. Our goal is to determine the form of $L(m)$ more precisely and to understand why $L(m)$ appears to be fairly universal. To this end, we initially narrow our focus on the more tractable case of k-ary trees with all receivers placed at the leaves.

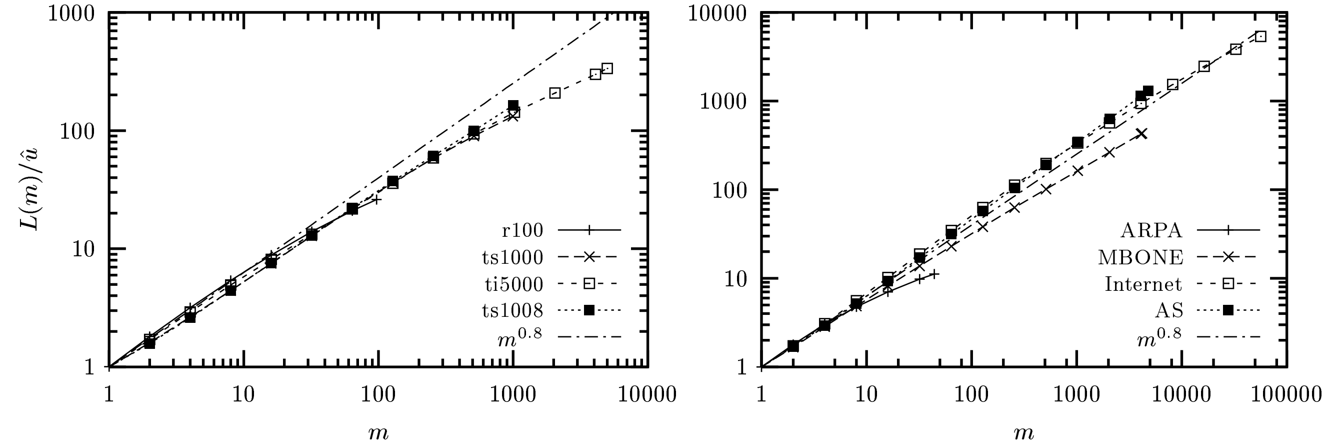
3 Calculation for k-ary Trees

Consider a k-ary tree of depth D, with the source at the root of the tree and all receivers placed at the leaves. We randomly pick n leaves, *not* necessarily unique, and compute $\tilde{L}(n)$ which is the size of the delivery tree that results; because we need not ensure that the receiver sites are distinct, the function $\tilde{L}(n)$ is easier to analyze than $L(m)$. However, we can relate the number of selected leaves n to the average number of distinct sites $\hat{m}$ that results. Let $M = k^D$ denote the total number of possible receiver locations; then $\hat{m} = M(1 - (1 - \frac{1}{M})^n)$. Defining $x = \frac{n}{M}$ and $y = \frac{\hat{m}}{M}$ and taking the limit of large M and fixed x and y, we have

[2] http://moat.nlanr.net/Routing/rawdata/ASconnlist.990324.922272001

[3] We merely count the number of links, we do not weight the links by their length or bandwidth.

[4] Note that we use a slightly different methodology than in [3]; there, for the networks created by network generators, there are also $N_{\rm network}$ random creations of each such network.



(a) generated network topologies. (b) real network topologies.

Figure 1: $\ln \frac{L(m)}{\hat{u}}$ versus $\ln m$ for several network topologies, compared to the line $\frac{L(m)}{\hat{u}} = m^{0.8}$.

$y = 1 - e^{-x}$. As M and n become large, with a fixed ratio $\frac{n}{M}$, the distribution of resulting m values is tightly centered around $\hat{m}$ and so we can (in this limit) approximate $L(m)$ in terms of $\tilde{L}(n)$ as follows:

$$L(m) \approx \tilde{L}\left(\frac{\ln(1-\frac{m}{M})}{\ln(1-\frac{1}{M})}\right) \tag{1}$$

which becomes, in the limit of large M and fixed x and y

$$L(m) \approx \tilde{L}(-M\ln(1-\frac{m}{M})) \tag{2}$$

3.1 Basic Expression

We first derive the basic expression for $\tilde{L}(n)$. Consider some link at level l at the tree; there are k^l such links in total. Now select n not necessarily distinct leaf sites. Each such selection will require a path through one of the k^l links at level l, and so picking one of the leaf sites at random is equivalent to picking one of the k^l links at level l at random. Thus, the probability that this particular link is in the delivery tree after n sites have been selected is given by:

$$1-(1-k^{-l})^n \tag{3}$$

We can treat all such probabilities as independent since the sum of the averages is independent of correlations, and so the average number of links in the delivery tree at level l is just $k^l\left(1-(1-k^{-l})^n\right)$. The average number of links in the entire delivery tree is thus given by:

$$\tilde{L}(n) = \sum_{l=1}^{D} k^l\left(1-(1-k^{-l})^n\right) \tag{4}$$

We now seek to evaluate this expression by looking at its (discrete) derivatives. Define $\Delta\tilde{L}(n) \equiv \tilde{L}(n+1) - \tilde{L}(n)$ so:

$$\Delta\tilde{L}(n) = \sum_{l=1}^{D}(1-k^{-l})^n \tag{5}$$

Furthermore, define $\Delta^2\tilde{L}(n) \equiv \Delta\tilde{L}(n+1) - \Delta\tilde{L}(n) = \tilde{L}(n+2) + \tilde{L}(n) - 2\tilde{L}(n+1)$ and so we have:

$$\Delta^2\tilde{L}(n) = -\sum_{l=1}^{D} k^{-l}(1-k^{-l})^n \tag{6}$$

3.2 Asymptotic Behavior of $\Delta^2\tilde{L}(n)$

We now derive the asymptotic form of $\Delta^2\tilde{L}(n)$ in the limit of large n and M but fixed $x = \frac{n}{M}$. We start by approximating the sum by an integral as follows:

$$\Delta^2\tilde{L}(n) \approx -\int_{\frac{1}{2}}^{D+\frac{1}{2}} dl\ k^{-l}(1-k^{-l})^n \tag{7}$$

Setting $z = k^{-l}$ we can rewrite this as:

$$\begin{aligned}\Delta^2\tilde{L}(n) &\approx \frac{1}{\ln k}\int_{k^{-1/2}}^{k^{-D-1/2}} dz\ (1-z)^n \\ &= \frac{1}{(n+1)\ln k}\left((1-k^{-1/2})^{n+1} - (1-k^{-D-1/2})^{n+1}\right)\end{aligned} \tag{8}$$

In the limit of large n and M but fixed $x = \frac{n}{M}$, we can simplify this as:

$$\Delta^2\tilde{L}(n) \approx \frac{-e^{-\frac{n+1}{Mk^{1/2}}}}{(n+1)\ln k} \approx \frac{-e^{-xk^{-1/2}}}{(xM+1)\ln k} \tag{9}$$

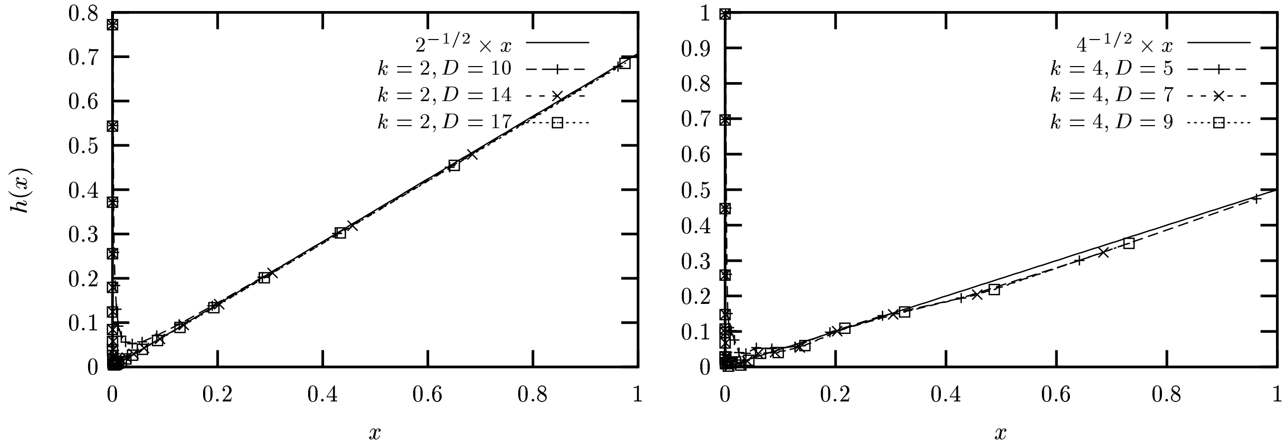
Note that D is exactly the average unicast path length $\hat{u}$. To best compare $\tilde{L}(n)$ to unicast we should normalize by the number of links in the average unicast path:

$$\frac{\Delta^2\tilde{L}(xk^D)}{\hat{u}} \approx \frac{-e^{-xk^{-1/2}}}{x}\frac{1}{M\ln M} \tag{10}$$

Define

$$h(x) \equiv -\ln\left(-x(M\ln M)\frac{\Delta^2\tilde{L}(xM)}{\hat{u}}\right) \tag{11}$$

Note that $h(x)$ is defined only in terms of $\Delta^2\tilde{L}$, the size of the network M, and the average unicast path length $\hat{u}$; the



(a) $k = 2$ and $D = 10,\ 14,\ 17$.

(b) $k = 4$ and $D = 5,\ 7,\ 9$.

Figure 2: $h(x)$ versus x for k-ary trees for $k = 2$ and $k = 4$ with receivers at leaves, compared to the line $h(x) = xk^{-1/2}$.

definition does not contain any terms that refer to the degree of the tree explicitly. When we combine the two preceding equations we find:

$$h(x) \approx xk^{-1/2} \tag{12}$$

Thus, the effect of the degree of the tree is to merely change the multiplicative constant in $h(x)$. Therefore, the degree of the network does not change the basic *form* of the asymptotic expression for $\Delta^2\tilde{L}(n)$, it merely rescales it. This is perhaps a sign that indeed the asymptotic form of $\tilde{L}(n)$ may be independent of the particular network topology.

Note that this approximate expression for $h(x)$ clearly fails for very small x, since $\Delta^2\tilde{L}(n)$ reaches a finite limit for small n (based on Equation 6) and so $h(x)$ as defined diverges in the limit of small x. However, we are not particularly interested in the extremely small x regime ($x < \frac{1}{M}$ means there is less than one receiver!). We now verify the accuracy of the above approximations outside the regime of very small x. We calculate $h(x)$ as defined in Equation 11 using the exact expressions in Equation 6 to compute $\Delta^2\tilde{L}(n)$. Figure 2 shows the results for $k = 2$ and $k = 4$ (both for three values of D), along with the curve $h(x) = xk^{-1/2}$. The $k = 2$ data fits the approximation $h(x) \approx xk^{-1/2}$ quite well (as long as $x > \frac{5}{M}$). The $k = 4$ data obeys the long-term linear trend, but oscillates at the beginning before converging toward linear behavior. Calculations for higher k show an increasing oscillatory trend. These oscillations are due to the discrete nature of the sum in Equation 6; the term $z(1-z)^n$ is a single-peaked function of z, and so the sum depends on how close k^{-l} for some l comes to the peak. These oscillations are washed out by the continuum approximation. However, these oscillations won't matter when we integrate $\Delta^2\tilde{L}(n)$ to derive $\tilde{L}(n)$, since the integral will average them out.

3.3 Asymptotic Behavior of $\tilde{L}(n)$

We now convert the approximate expression for $\Delta^2\tilde{L}(n)$ into an approximate one for $\tilde{L}(n)$. We first make the following approximation to simplify the analysis:

$$\frac{-e^{-\frac{n+1}{Mk^{1/2}}}}{(n+1)\ln k} \approx \begin{cases} \frac{-1}{(n+1)\ln k} & \text{if } n+1 \le Mk^{1/2} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Although this approximation is crude, we are interested mainly in the asymptotic behavior and it appears (as shown below) that this approximation does not alter the asymptotics. Using the boundary conditions that $\tilde{L}(0) = 0$ and $\tilde{L}(1) = D$, we then have $\Delta\tilde{L}(n) \approx D - \frac{\ln(n+1)}{\ln k}$ and so

$$\tilde{L}(n) \approx nD - \frac{1}{\ln k}\left((n+1)\ln(n+1) - (n+1)\right) \tag{14}$$

In the limit of large n and M with fixed $x = \frac{n}{M}$, we find:

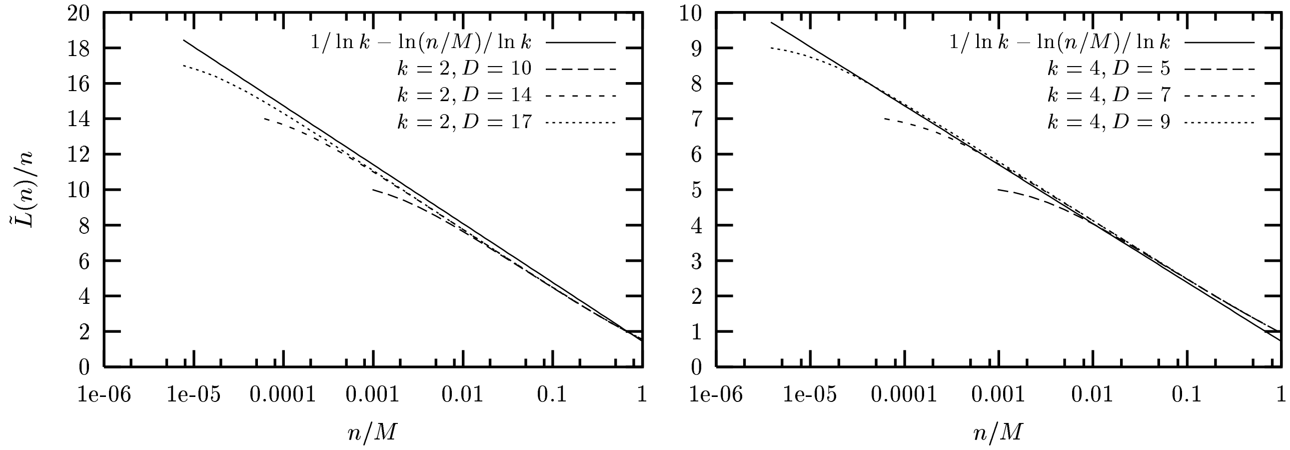
$$\frac{\tilde{L}(n)}{n} \approx (D + \frac{1}{\ln k}) - \frac{\ln n}{\ln k} \tag{15}$$

In terms of $x = \frac{n}{M}$ we get:

$$\frac{\tilde{L}(xM)}{xM} \approx \frac{1}{\ln k} - \frac{\ln x}{\ln k} \tag{16}$$

We now verify the accuracy of this expression by comparing it with numerical calculations using the exact expression in Equation 4. Figures 3(a) and 3(b) show the quantity $\frac{\tilde{L}(n)}{n}$ plotted versus $\ln\frac{n}{M}$ for $k = 2$ and $k = 4$ (for three values of D); also shown is the curve $\frac{\tilde{L}(n)}{n} = \frac{1}{\ln k} - \frac{\ln\frac{n}{M}}{\ln k}$. We observe three clear trends in these figures. First, the curves are reasonably linear for intermediate values of x; for very small values of $\frac{n}{M}$, roughly $\frac{5}{M} > \frac{n}{M}$, the curves have a pronounced concavity, and for $\frac{n}{M} \approx 1$ the curves have a very slight convexity (more visible in the $k = 2$ data, but it becomes increasingly apparent for smaller values of k).[5] Second, the slopes of the linear portions of the curves are quite close to the predicted values of $\frac{1}{\ln k}$. Third, the intercept of the linear portions of the curves deviate slightly from the predicted values of $\frac{1}{\ln k}$. Thus, in the linear regime $5 < n < M$, Equation 16 captures the behavior to within

[5] Note that k is merely a parameter in this calculation and so we can vary it continuously towards the limit of $k = 1$.



(a) $k = 2$ and $D = 10,\ 14,\ 17$.

(b) $k = 4$ and $D = 5,\ 7,\ 9$.

Figure 3: $\frac{\tilde{L}(n)}{n}$ versus $\ln \frac{n}{M}$ for k-ary trees and receivers at leaves, compared to the line $\frac{\tilde{L}(n)}{n} = \frac{1}{\ln k} - \frac{\ln \frac{n}{M}}{\ln k}$.

an additive constant; the additive error is not surprising given the many approximations embedded in Equation 16. In general, then, it appears that in the regime of interest

$$\tilde{L}(n) \approx n \left(c - \frac{\ln \frac{n}{M}}{\ln k} \right) \tag{17}$$

for some constant c. However, this approximation is clearly not valid in the limit of large $\frac{n}{M}$; when $\ln \frac{n}{M} > c \ln k - 1$, the approximation is decreasing in n. Our calculations suggest that the approximation is reasonably accurate for $5 < n < M$ and so, in this regime, the function $\tilde{L}(n)$ is roughly linear with a logarithmic correction.

We now have an asymptotic form for $\tilde{L}(n)$; what about our original target of focus, $L(m)$? We can easily re-express Equation 17 in terms of m using Equation 1:

$$L(m) \approx \frac{\ln(1 - \frac{m}{M})}{\ln(1 - \frac{1}{M})} \left(c - \frac{\ln \left(\frac{\ln(1-\frac{m}{M})}{M \ln(1-\frac{1}{M})} \right)}{\ln k} \right) \tag{18}$$

This function is most decidedly not of the form $L(m) \propto m^{0.8}$. Somewhat surprisingly, though, it produces results quite similar to the Chuang-Sirbu formula. Using Equations 4 and 1, we can approximate $L(m)$ for k-ary trees. Figure 4 depicts $\ln \frac{L(m)}{\hat{u}}$ plotted versus $\ln m$ for $k = 2$ and $k = 4$; even though the form of the function $L(m)$ is rather different than $m^{0.8}$, the agreement with the Chuang-Sirbu scaling law is remarkably good. Thus, if our results about $\tilde{L}(n)$ apply to more general networks, as we address in Section 4, then the Chuang-Sirbu scaling law will also remain reasonably accurate for those general networks.

3.4 Putting Receivers at Non-Leaf Sites

Above, to make the calculation of $\tilde{L}(n)$ simpler, we assumed that receivers were located only at leaf sites. We now relax that assumption and allow receivers to be spread uniformly over the entire tree (excepting the root, where the source is located). In this case, the probability that a particular link at level l is in the delivery tree becomes:

$$1 - \left(1 - k^{-l} \frac{k^{D+1} - k^l}{k^{D+1} - k} \right)^n = 1 - \left(1 - \frac{k^{D-l+1} - 1}{k^{D+1} - k} \right)^n \tag{19}$$

The form of this expression is due to the fact that for a choice of receiver to use a particular link at level l, the receiver has to be located at or below level l (probability $\frac{k^{D+1} - k^l}{k^{D+1} - k}$) and the receiver has to be located below that particular link (conditional probability k^{-l}).

In the limit of large D, for a fixed l, this probability becomes

$$1 - (1 - k^{-l})^n \tag{20}$$

which is the same as the leaf-only expression. The expression for $\tilde{L}(n)$ for this case, analogous to Equation 4 for the leaf-only case, is:

$$\tilde{L}(n) = \sum_{l=1}^{D} k^l \left(1 - \left(1 - \frac{k^{D-l+1} - 1}{k^{D+1} - k} \right)^n \right) \tag{21}$$

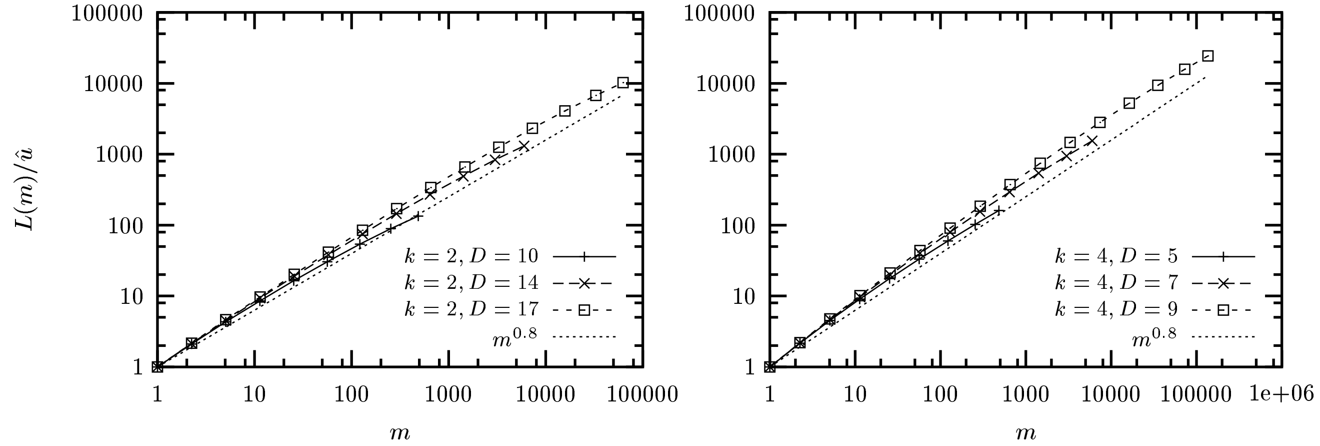
Figure 5 shows the results for this expression for our canonical cases of $k = 2$ and $k = 4$ (with three values of D). As we can see, the curves still show the same behavior of $\tilde{L}(n) \approx n(c - \frac{\ln \frac{n}{M}}{\ln k})$ but the value of c has changed from when the receivers were located only on the leaves.

4 More General Networks

We have used k-ary trees as a simple test case where the computation of $\tilde{L}(n)$ is reasonably tractable. The real networks of interest, such as those we discussed in Section 2, are far more complicated. We now ask whether the asymptotic form for $\tilde{L}(n)$ we derived in Section 3 applies to these more general networks. We begin by deriving an approximate expression for $\tilde{L}(n)$ for a more general class of networks.

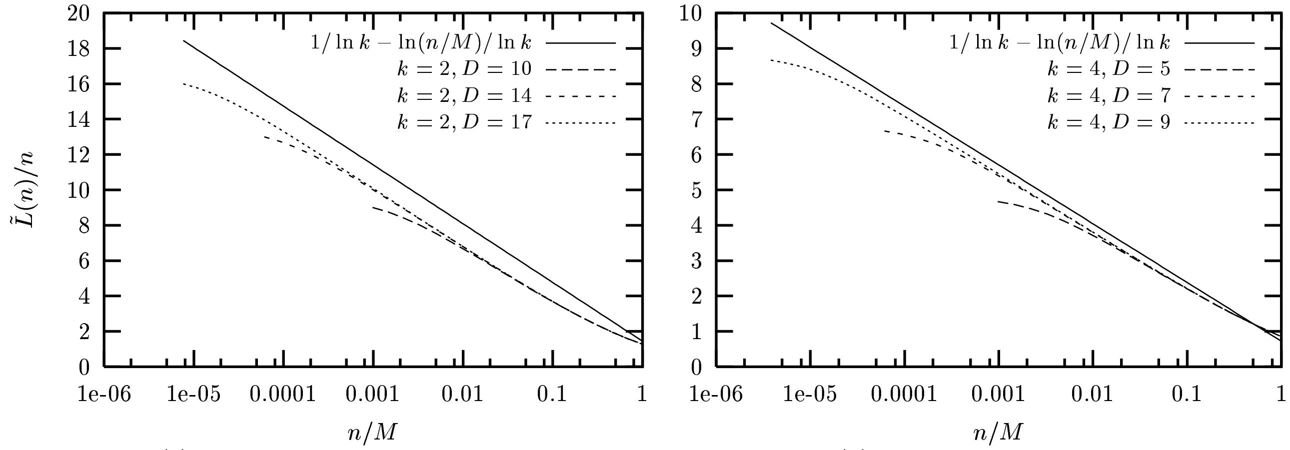
4.1 Basic Analysis

For a given graph with a chosen source, let the reachability function $S(r)$ denote the number of distinct sites that are



(a) $k=2$ and $D = 10,\ 14,\ 17.$ (b) $k=4$ and $D = 5,\ 7,\ 9.$

Figure 4: $\ln \frac{L(m)}{\hat{u}}$ versus $\ln m$ for k-ary trees with receivers at leaves, compared to the line $\frac{L(m)}{\hat{u}} = m^{0.8}$.



(a) $k=2$ and $D = 10,\ 14,\ 17.$ (b) $k=4$ and $D = 5,\ 7,\ 9.$

Figure 5: $\frac{\tilde{L}(n)}{n}$ versus $\ln \frac{n}{M}$ for k-ary trees with receivers throughout, compared to the line $\frac{\tilde{L}(n)}{n} = \frac{1}{\ln k} - \frac{\ln \frac{n}{M}}{\ln k}$.

exactly r hops from the source. Initially assume that all receivers are on leaf sites a distance D from the source (and so $\hat{u} = D$), define $M = S(D)$, and consider the set of links that are r hops (counting themselves) from the source. If we assume that the receivers are equally likely to be downstream from any of these links, then we can approximate the probability that these links are on the delivery tree by the expression:

$$1 - \left(1 - \frac{1}{S(r)}\right)^n \tag{22}$$

Assuming that all such calculations are independent, the average number of links in the delivery tree is given by:

$$\tilde{L}(n) = \sum_{r=1}^{D} S(r)\left(1 - \left(1 - \frac{1}{S(r)}\right)^n\right) \tag{23}$$

Following the analysis in Section 3, we find:

$$\Delta^2 \tilde{L}(n) = -\sum_{r=1}^{D} \frac{1}{S(r)}\left(1 - \frac{1}{S(r)}\right)^n \tag{24}$$

Consider the limit of large n and large D, with $\frac{n}{S(D)}$ fixed, we approximate the sum by an integral as we did in Section 3:

$$\Delta^2 \tilde{L}(n) \approx -\int_{1/2}^{D+1/2} dr\ \frac{1}{S(r)}\left(1 - \frac{1}{S(r)}\right)^n \tag{25}$$

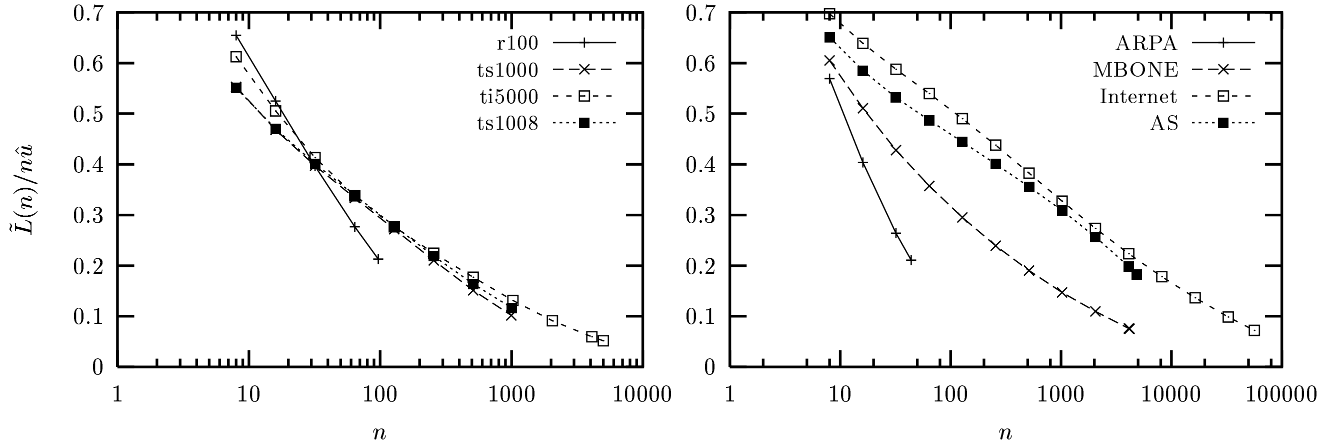
Setting $z = \frac{1}{S(r)}$ we can rewrite this and then take the limit of large n:

$$\Delta^2 \tilde{L}(n) \approx \int_{S(1/2)^{-1}}^{S(D+1/2)^{-1}} dz\ \frac{S(r)}{S'(r)}(1-z)^n \tag{26}$$

We now consider several different possibilities for $S(r)$.

4.2 Exponential Case

Random graphs and k-ary trees have the property that $S(r)$ is exponentially increasing. More generally, an exponential increase in $S(r)$ results if the increase in the number of sites reachable as we go from r to $r+1$ is proportional to the number of sites reachable in r hops (*i.e.*, $\frac{S(r+1)-S(r)}{S(r)}$ is roughly



(a) generated network topologies.

(b) real network topologies.

Figure 6: $\frac{\tilde{L}(n)}{n\hat{u}}$ versus $\ln n$ for several networks.

constant). When we initiated this study we expected that such exponential increase in $S(r)$ was the generic behavior in real networks. Accordingly, we first considered the case where $S(r) \approx e^{\lambda r}$ for some λ. Then

$$\Delta^2 \tilde{L}(n) \approx \frac{-e^{-\frac{n}{S(D+1/2)}}}{(n+1)\lambda} \tag{27}$$

Using the function $h(x) \equiv -\ln\left(-x(M\ln M)\frac{\Delta^2\tilde{L}(xM)}{\hat{u}}\right)$ as in Section 3 yields:

$$h(x) \approx xe^{-\frac{\lambda}{2}} \tag{28}$$

Thus, if the function $S(r)$ is exponential, we find the same form for the function $\Delta^2\tilde{L}(n)$ as we did in Section 3 for k-ary trees with the parameter λ playing the role of $\ln k$. Continuing to follow the derivation in Section 3, we find

$$\frac{\tilde{L}(xM)}{xM} \approx \frac{1}{\lambda} - \frac{\ln x}{\lambda} \tag{29}$$

We now ask whether this approximate analysis matches what we see in the networks we first considered in Section 2. Before doing so, we generalize our treatment to include putting receivers at non-leaf sites since that is the more realistic case. Following the derivation of Equation 21, we find that the analogous expression for general networks is:

$$\tilde{L}(n) = \sum_{l=1}^{D} S(l)\left(1 - \left(1 - \frac{T(D) - T(l-1)}{S(l)T(D)}\right)^n\right) \tag{30}$$

where $T(r) \equiv \sum_{j=1}^{r} S(j)$ is the number of sites reachable in r or fewer hops. $T(D)$ is thus the total number of sites in the network (excluding all sites that are farther than D hops from the source), and $T(D) - T(r-1)$ is the number of sites reachable in r or more hops.

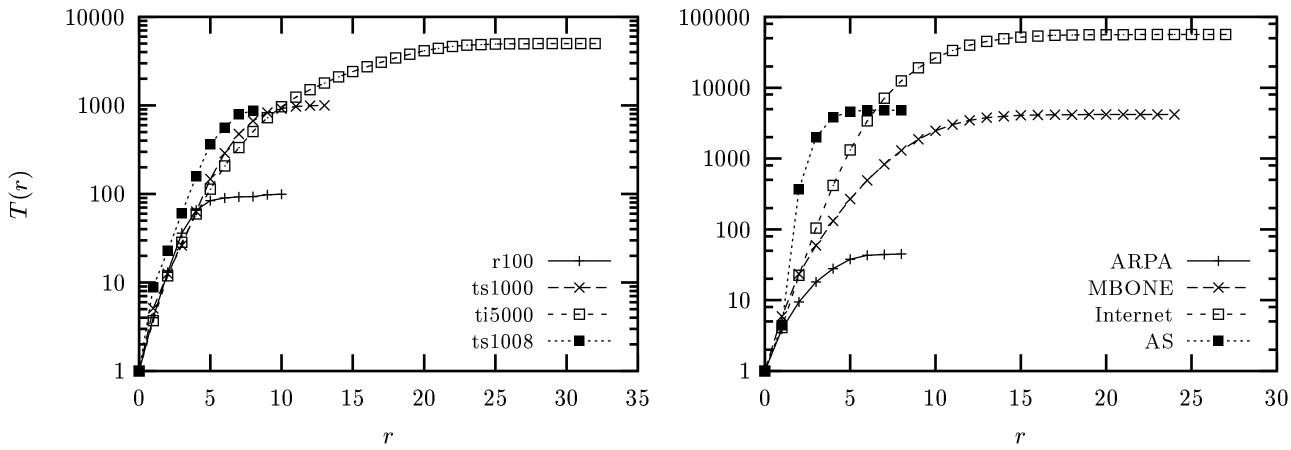
Figures 6(a) and 6(b) shows the plot of $\frac{\tilde{L}(n)}{n\hat{u}}$ versus $\ln n$ for the generated and real networks we considered in Section 2. The generated networks r100, ts1000, and ts1008 all appear to fit the predicted linear behavior quite well; the curve for the ti5000 network is significantly less linear. In addition, it is a bit surprising that the two transit stub networks, ts1000 and ts1008, have such similar slopes even though they have very different average degrees (3.6 and 7.5). Turning to the data from real networks, the Internet and AS topologies give rise to fairly linear curves. The ARPA data is slightly less linear, but the MBone data is significantly less linear.

To understand these results a bit better, we examined the reachability functions $S(r)$ for these networks. Figure 7 depicts $T(r) = \sum_{j=1}^{r} S(j)$ for these various networks (averaged over the N_{source} choices for the source). The r100, ts1000, and ts1008 exhibit exponential growth before reaching the saturation point where $T(r) \approx M$. Note that the two transit-stub networks have similar growth rates despite their having different degrees; this may be due to the philosophy of the transit-stub portion of the ITM network generator [1] which constructs portions of the graph randomly while constraining the gross structure. This similarity in exponential growth rates, whatever its origin, is likely the reason that the slopes of the curves in Figure 6 associated with the two transit-stub topologies are so similar. In contrast to the random and transit-stub networks, the ti5000 curve in Figure 7 has a significant degree of concavity for a wide range of r values, suggesting sub-exponential growth.

The corresponding data from real networks has a similar but less clear-cut dichotomy. The curves in Figure 7 from the Internet and AS maps exhibit exponential growth before saturating.[6] The ARPA curve has significant concavity, and the MBone data has a slight concavity, which would be consistent with sub-exponential growth for these two networks.

In both the real and generated networks, when $T(r)$ exhibits exponential behavior (the ts1000, ts1008, r100, Internet, and AS) the predicted form for $\tilde{L}(n)$ appears to hold. In the cases where $T(r)$ appears to be sub-exponential (ti5000, MBone, and ARPA) the predicted form applies less well. Our original intuition that real networks have exponential

[6] However, the existence of exponential growth in real networks is the subject of some controversy; see [8] for a different reading of similar data.



(a) generated network topologies.

(b) real network topologies.

Figure 7: $\ln T(r)$ versus r for several networks.

$T(r)$ appears to hold for the direct Internet measurements (Internet and AS maps), but applies less well to the MBone. The MBone remains partially an overlay network, which may affect the nature of $T(r)$. The nature of $S(r)$ for real networks is an interesting, and open, research question: see [9] and [8] for some related work on this topic.

4.3 Non-exponential Cases

While we expect that $S(r) \approx e^{\lambda r}$ for some λ for many networks, we have already seen that there are cases where the reachability function $S(r)$ has somewhat different behavior. Let's consider two other extremes: $S(r) \approx r^{\lambda}$ and $S(r) \approx e^{\alpha r^{\gamma}}$. The first grows more slowly, and the second grows more quickly, than exponential.

To simplify matters, we can assume that receivers are at the leaves of the network and can use Equation 23 to calculate $\tilde{L}(n)$. Figure 8 shows the curves for an exponential $S(r) = 2^r$ along with the two non-exponential cases we are considering: $S(r) \approx r^{\lambda}$ and $S(r) \approx e^{\alpha r^{\gamma}}$. The constants were normalized so that $S(D)$ is the same for all three networks. As can be seen, the non-exponential cases have quite different behavior than the exponential case. This shows that the asymptotic form we derived for the exponential case does not apply to these other kinds of networks. If it turns out that real networks do not exhibit the exponential reachability functions $S(r)$ then our analysis here is rendered moot.

5 Receiver Affinity and Disaffinity

Up to now we have assumed that receivers are uniformly distributed in the network.[7] However, the distribution of such receivers will, in practice, deviate from this uniform distribution. There are many situations, like teleconferences, where the participants will tend to cluster. There will be others – such as sites in a sensor network in which the sensors are evenly spread out – where the participating sites will tend to be apart. We use the terms affinity and disaffinity for these tendencies: receiver affinity means the receivers like to cluster together, and disaffinity means that they have a bias towards spreading out.

In this section we investigate the implications of receiver affinity and disaffinity through the use of a simple model. We do not claim that our model is in any way an accurate representation of reality, merely that it captures the notion of clustering (and spreading out) sufficiently well so that we can look at how affinity and disaffinity affect $\tilde{L}(n)$.

5.1 Simple Model of Receiver Affinity/Disaffinity

We model the tendency for receivers to cluster (or spread out) by assuming that receivers have an affinity (or disaffinity) for being near other sites in the group. Consider some network and a multicast group with m distinct receiver sites and a given sender. Let $A(m)$ be the set of all receiver configurations (*i.e.*, set of receiver placements) with m distinct receiver sites. Consider some specific configuration $\alpha \in A(m)$ and let L_α denote the size of the delivery tree for this configuration. Let β denote some parameter reflecting the degree of affinity or disaffinity; $\beta = 0$ is the uniform distribution of receivers we have focused on so far, $\beta = \infty$ is infinite affinity (receivers pack as closely as possible), and $\beta = -\infty$ is infinite disaffinity (receivers spread out as much as possible). We can then model affinity or disaffinity by assigning weights $W_\alpha(\beta)$ to the various receiver configurations (with $\sum_{\alpha \in A(m)} W_\alpha(\beta) = 1$) and then take a weighted average, as follows:

$$L_\beta(m) = \sum_{\alpha \in A(m)} W_\alpha(\beta) L_\alpha \qquad (31)$$

We could similarly define $\tilde{L}_\beta(n)$, where n is the number of not necessarily distinct receiver locations, by defining the set $\tilde{A}(n) \equiv \cup_{m \leq n} A(m)$, which contains configurations with multiple receivers at a given site. Note, however, that the relationship between n and m developed in Section 3 no longer applies when $\beta \neq 0$ (because there is a bias either for or against receivers to be at the same site). Therefore we cannot simply study one of these functions and convert it into the other, as we have done so far.

[7] At times we have limited receivers to leaf sites, as in Section 3, but so far in all cases the receivers are distributed over the possible sites in a uniform manner.

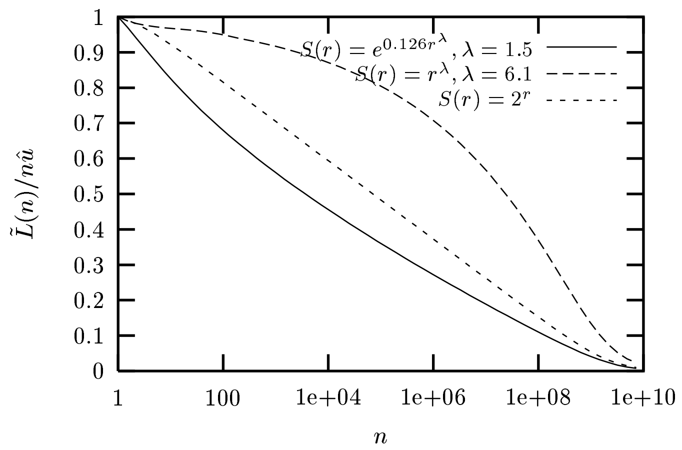


Figure 8: $\frac{\tilde{L}(n)}{n\hat{u}}$ versus $\ln n$ for several reachability functions $S(r)$.

The key aspect to affinity or disaffinity is that receivers are either more or less likely to situate themselves *near* each other, for some definition of *near*. For convenience, we measure the distance d between two receivers in terms of the number of hops in the shortest path between them. For a given receiver configuration let $\hat{d}(\alpha)$ be the average inter-receiver distance for configuration α. We then define $W_\alpha(\beta)$ as follows:

$$W_\alpha(\beta) = \rho e^{-\beta \hat{d}(\alpha)} \tag{32}$$

where ρ is chosen so that $\sum_{\alpha \in A(m)} W_\alpha(\beta) = 1$. We now ask what happens in the limit of extreme affinity ($\beta = \infty$) and disaffinity ($\beta = -\infty$), and then review some simulation results for intermediate values of β. For simplicity, in the theoretical discussion in Sections 5.2 and 5.3 we restrict ourselves to k-ary trees with receivers at the leaves. For the simulations in Section 5.4 we allow receivers to be at all sites.

5.2 Extreme Disaffinity

Consider the case where the receivers stay as far apart as possible. This is equivalent to adding them in an order that maximizes the number of links added to the tree at each step. Set $L_{-\infty}(0) = 0$. Then, we have the sequence:

$$\begin{aligned}
\Delta L_{-\infty}(0) &= D, \ldots,\ \Delta L_{-\infty}(k-1) = D, \\
\Delta L_{-\infty}(k) &= D-1, \ldots,\ \Delta L_{-\infty}(k^2-1) = D-1, \\
\Delta L_{-\infty}(k^2) &= D-2, \ldots,\ \Delta L_{-\infty}(k^3-1) = D-2, \\
\Delta L_{-\infty}(k^3) &= D-3, \ldots,\ \Delta L_{-\infty}(k^4-1) = D-3, \\
\Delta L_{-\infty}(k^4) &= D-4, \ldots
\end{aligned}$$

Therefore, $\Delta^2 L_{-\infty}(m)$ is given by:

$$\Delta^2 L_{-\infty}(k^l - 1) = -1 \text{ for all } l > 0 \tag{33}$$

with $\Delta^2 L_{-\infty}(m) = 0$ for all others values of m. Smoothing this out, we find:

$$\Delta^2 L_{-\infty}(m) \approx \frac{-1}{m(k-1)} \tag{34}$$

Thus,

$$\Delta^2 L_{-\infty}(xk^D) \approx \frac{-1}{k^D(k-1)} \frac{1}{x} \tag{35}$$

Returning to the original formula, we can write:

$$\begin{aligned}
L_{-\infty}(1) &= D, \\
L_{-\infty}(k) &= kD, \\
L_{-\infty}(k^2) &= kD + k(k-1)(D-1) \\
L_{-\infty}(k^3) &= kD + k(k-1)(D-1) + k^2(k-1)(D-2)
\end{aligned}$$

More generally,

$$\begin{aligned}
L_{-\infty}(k^l) &= D + \sum_{i=0}^{l-1} k^i (k-1)(D-i) \\
&= Dk^l - \frac{k}{k-1}\left(k^{l-1}(lk-k-l)+1\right)
\end{aligned} \tag{36}$$

Thus, for $m = k^l$ for some l, we have:

$$L_{-\infty}(m) = mD - \frac{k}{k-1}\left(\left(\frac{\ln m}{\ln k} - 1\right)m - \frac{m \ln m}{k \ln k} + 1\right) \tag{37}$$

The slope starts out high (D) and then decreases slowly as sites are added. Note that in this case, since receivers won't occupy the same site unless necessary, $\tilde{L}_{-\infty}(n) = L_{-\infty}(n)$ for all $n \le M$, and $\tilde{L}_{-\infty}(n) = L_{-\infty}(M)$ for all $n > M$.

5.3 Extreme Affinity

Consider the case where the receivers tend to cluster. This is equivalent to adding them in such a way as to minimize the number of links added to the tree at each step. Consider a binary tree. Then, we have the sequence:

$$\Delta L_\infty(0) = D,\ \Delta L_\infty(1) = 1,\ \Delta L_\infty(2) = 2,$$

$$\Delta L_\infty(3) = 1,\ \Delta L_\infty(4) = 3,\ \Delta L_\infty(5) = 1,\ \Delta L_\infty(6) = 2,$$

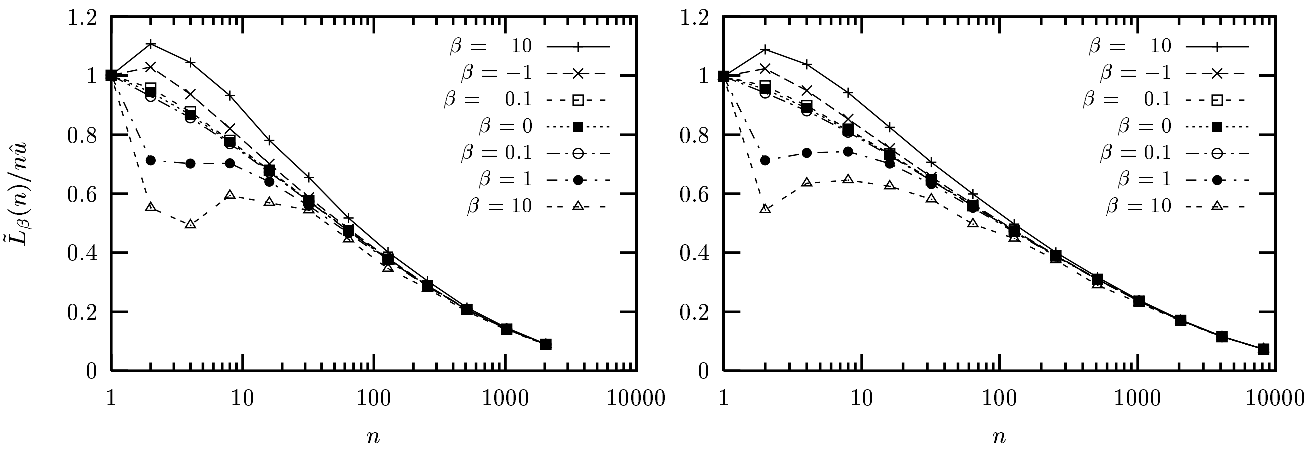
$$\Delta L_\infty(7) = 1, \ldots$$

More generally, when $m = k^l$ then

$$L_\infty(k^l) = D - l + \frac{k^{l+1} - k}{k-1} = D - l + (k^l - 1)\frac{k}{k-1} \tag{38}$$

Thus, $L_\infty(m)$ is roughly linear (with an additive logarithmic correction) when restricted to $m = k^l$ for some l.

In the extreme affinity case, $\tilde{L}_\infty(n) = D$ for all n (all receivers cluster at the same location).



(a) binary tree with depth $D = 10$.

(b) binary tree with depth $D = 12$.

Figure 9: $\frac{\tilde{L}_\beta(n)}{n\hat{u}}$ versus $\ln n$ for a binary tree and various values for β.

5.4 Intermediate β

Figure 9 depicts $\frac{\tilde{L}_\beta}{n\hat{u}}$ for $\beta = -10,\ -1, -0.1,\ 0,\ .1,\ 1,\ 10$ for binary trees with depth $D = 10$ and $D = 12$ respectively. As expected, Figure 9(a) shows that increasing affinity decreases the size of the multicast tree $\tilde{L}_\beta$; the effects are most obvious for smaller n. When we increase the size of the network by a factor of 4, going from $D = 10$ to $D = 12$, the effect of a change in β for a fixed n (*i.e.*, $\frac{\tilde{L}_{\beta_1}(n)}{n\hat{u}} - \frac{\tilde{L}_{\beta_2}(n)}{n\hat{u}}$) remains relatively constant. This suggests that if we take the limit of large D and fixed $x = \frac{n}{2^D}$ then the effect of affinity on $\frac{\tilde{L}(n)}{n\hat{u}}$ will vanish:

$$\lim_{D\to\infty}\left(\frac{\tilde{L}_{\beta_1}(x2^D)}{x2^D\hat{u}} - \frac{\tilde{L}_{\beta_2}(x2^D)}{x2^D\hat{u}}\right) = 0 \tag{39}$$

We therefore conjecture that in the asymptotic limit of large networks and fixed fractional population, affinity does not disturb the basic form for $\frac{\tilde{L}(n)}{n}$ described in Equation 17. However, for a fixed n, affinity and disaffinity do significantly affect the size of the delivery tree $\tilde{L}(n)$.

6 Discussion

Inspired by the intriguing results of Chuang and Sirbu, we have focused on a very narrow question: what is the asymptotic nature of the function $\tilde{L}(n)$ which describes the number of links in a multicast tree connecting a single source to n randomly located receivers? We were able to analyze k-ary trees, deriving an exact expression (Equation 4) and showed that, asymptotically, $\tilde{L}(n) \approx n(c - \frac{\ln \frac{n}{M}}{\ln k})$. We argued that this form should apply to any network whose reachability function $S(r)$ exhibited exponential behavior. This prediction then led to an expression for $L(m)$ which was quite different than the Chuang-Sirbu scaling law in form, but which was not too dissimilar in behavior.

To test these predictions, we looked at several more realistic networks. Our simulations results were not entirely conclusive. The results for several of the networks agreed quite closely with the predicted form, while the results for the ARPA, MBone, and ti5000 topologies were somewhat less in agreement. This lack of agreement may be due to the non-exponential behavior of these networks' reachability function $S(r)$. Thus, we still conjecture that the form $\tilde{L}(n) \approx n(c - \frac{\ln \frac{n}{M}}{\ln k})$ will apply to networks with exponential reachability functions. The question then becomes: do real networks (current or future ones) have exponential reachability functions $S(r)$? More investigations of artificially generated networks [1, 7, 11, 10] and real networks (see [9, 8]) would shed light on this question, and this will be the subject for our future research.

If exponential behavior in $S(r)$ is generic then we think our analysis helps explain the asymptotic nature of $\tilde{L}(n)$, and the universality of the Chuang-Sirbu scaling law. However, we should emphasize that this result is of little practical importance. The Chuang-Sirbu scaling law is sufficiently accurate for the purposes for which it was intended, and our analysis merely suggests an explanation for its origin.

References

[1] Calvert, K. L., Doar, M. B., and Zegura, E. W. Modeling internet topology. *IEEE Communications Magazine 35*, 6 (June 1997), 160–163.

[2] Casner, S., and Deering, S. First IETF Internet audiocast. *ACM Computer Communication Review 22*, 3 (July 1992), 92–97.

[3] Chuang, J., and Sirbu, M. Pricing multicast communications: A cost-based approach. In *Proceedings of the INET'98* (1998).

[4] Deering, S. E. Multicast routing in internetworks and extended LANs. In *ACM SIGCOMM* (Stanford, California, Aug. 1988), ACM, pp. 55–64. also in *Computer Communication Review* 18 (4), Aug. 1988 and as Report STAN-CS-88-1214, Department of Computer Science, Stanford, California.

[5] Deering, S. E. *Multicast routing in a datagram internetwork*. PhD thesis, Stanford University, Palo Alto, California, Dec. 1991.

[6] Deering, S. E., and Cheriton, D. R. Multicast routing in datagram internetworks and extended LANs. *ACM Trans. Comput. Syst. 8*, 2 (May 1990), 85–110.

[7] Doar, M. B. A better model for generating test networks. In *IEEE GLOBECOM* (London, England, Nov. 1996), J. Crowcroft and H. Schulzrinne, Eds., IEEE, pp. 86–93.

[8] Faloutsos, M., Faloutsos, P., and Faloutsos, C. On power-law relationships of the Internet topology. In *ACM SIGCOMM* (Cambridge, Massachusetts, USA, Sept. 1999), ACM.

[9] Siamwalla, R., Sharma, R., and Keshav, S. Discovering internet topology. preprint, 1998.

[10] Waxman, B. M. Routing of multipoint connections. *IEEE Journal of Selected Areas in Communication 6*, 9 (Dec. 1988), 1617–1622.

[11] Waxman, B. M. Models for multipoint connections in gigabit networks. In *Proc. of Gigabit Networking Workshop* (Boston, Massachusetts, Apr. 1995).

[12] Wei, L., and Estrin, D. A comparison of multicast trees and algorithms. In *IEEE Infocom* (Toronto, Canada, June 1994), IEEE.

[13] Wei, L., and Estrin, D. The trade-offs of multicast trees and algorithms. In *International Conference on Computer Communications and Networks* (1994).