On Multicast Trees: Structure and Size Estimation

Danny Dolev, *Senior Member, IEEE*, Osnat (Ossi) Mokryn, and Yuval Shavitt, *Senior Member, IEEE*

***Abstract*—This work presents a thorough investigation of the structure of multicast trees cut from the Internet and power-law topologies. Based on both generated topologies and real Internet data, we characterize the structure of such trees and show that they obey the rank-degree power law; that most high degree tree nodes are concentrated in a low diameter neighborhood; and that the sub-tree size also obeys a power law.**

Our most surprising empirical finding suggests that there is a linear ratio between the number of high degree network nodes, namely nodes whose tree degree is higher than some constant, and the number of leaf nodes in the multicast tree (clients). We also derive this ratio analytically. Based on this finding, we develop the Fast Algorithm, that estimates the number of clients, and show that it converges faster than one round trip delay from the root to a randomly selected client.

***Index Terms*—Internet topology, multicast group size estimation.**

I. Introduction

THERE are several inhibitors to the commercial use of multicast protocols. While it is clear that multicast is beneficial for transmitting the same information to large groups, its exact gain over unicast has not yet been determined [1]–[3]. Network suppliers lack a fast and efficient way to estimate the size of large multicast groups, and the research community lacks reliable tree models.

We present here a thorough investigation we performed on the structure and characteristics of multicast trees cut from generated power law topologies and the Internet. While the exact nature of the Internet topology is in debate [4], our results show that the partial views we have from the Internet obey the power laws found by [5]. These results were also verified by [6]–[8], who conducted further investigations. Moreover, trees cut from the Internet and from the generated topologies had similar characteristics.

We found that trees cut from such topologies and the Internet obey a degree-rank and sub-tree size-rank power law distributions.[1] We also found that the distance distribution of nodes from the root node resembles a Gamma distribution, as shown previously for the Internet [8]. We observed that nodes with degree higher than five tend to be rare in the resulting trees. These high degree nodes can always be found in several adjacent rings, which reside typically at the core of the network, and in the near vicinity of the tree root.

Our most intriguing result finds a linear ratio between the number of high degree nodes in the tree and the number of clients.[2] The result is shown to be valid for trees cut from scale-free topologies that were generated with various parameters, as well as for experiments conducted on the Internet itself. We further verify this ratio analytically for power law trees. Based on the tree topological characteristics we found, we suggest the Fast Algorithm for estimating the size of large multicast groups. We analyze the algorithm's expected delay in the Internet, which sums up to less than the round trip delay from the root node of the tree to a random client at the edge of the network.

Estimating the population size of large multicast trees can improve the performance of feedback mechanisms of protocols such as RTP [9] and SRM [10]. Current feedback supression solutions for RTCP use timers at the receivers [11], [12]. Our sender based estimation produces a much faster estimation that can be propagated to the receivers and eliminate the need for such timers. Often, feedback suppression protocols are based on similar techniques as polling based estimation algorithms [13]–[15] and thus can use our faster estimation instead. Fast estimation may also be beneficial to forward error correction protocols [16].

Our suggested estimation algorithm offers an alternative approach by using the topological characteristics to obtain an estimation on the number of receivers (rather than a specific population count). It does not aggregate information at the router level, but rather polls the high degree routers in the multicast tree. Our results show that paths from the root of the tree to its receivers are very likely to pass through the core of the network; We also observed that high degree routers tend to reside within the core or in its close vicinity. Hence, the polled high degree nodes will be closer to the root than the receivers they connect. The algorithm adapts itself to dynamic topological changes, and can therefore reflect changes in the session size, as does the population sampling algorithm suggested in [17].

To the best of our knowledge, this is the first time that the existence of a power law in the underlying topology is leveraged to construct an algorithm. We believe that more such algorithms can be developed in the future for a variety of purposes.

Manuscript received December 26, 2002; revised July 13, 2004; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Liebeherr. This work was supported by the United States–Israel Binational Science Foundation (BSF), Israel, The Checkpoint PhD Fellowship Grant, Israel, The Intel COMM Grant-Internet Network/Transport Layer and QoS Environment (IXA), Israel, The Israel Science Foundation (ISF) Center of Excellence Program (grant number 8008/03), and by a grant from the EU 6th FP, IST Priority, Proactive Initiative "Complex Systems Research", as part of the EVERGROW integrated project. A shorter version of this paper was presented at the IEEE INFOCOM 2003, San Francisco, CA.

D. Dolev is with the School of Engineering and Computer Science, Hebrew University, Jerusalem, Israel (e-mail: dolev@cs.huji.ac.il).

O. Mokryn was with the Department of Electrical Engineering—Systems, Tel-Aviv University, Tel-Aviv, Israel. She is now with the Technion IIT, Haifa, Israel (e-mail: ossi@ee.technion.ac.il).

Y. Shavitt is with the Department of Electrical Engineering—Systems, Tel-Aviv University, Tel-Aviv, Israel (e-mail: shavitt@eng.tau.ac.il).

Digital Object Identifier 10.1109/TNET.2006.876195

[1]Note that rank-degree and frequency-degree power laws can be derived from each other [8].

[2]We denote by clients the group of routers that directly attach clients.

TABLE I
TYPE OF UNDERLYING TOPOLOGIES USED

Name	Type	Parameters	No. of Nodes	Avg. Node degree
VS	generated	$a = 1; p \in 0 : 0.05 : 0.5$	10000	$1.99 - 3.98$
IS	generated	$a = 2; p \in 0 : 0.05 : 0.5$	10000	$3.99 - 7.9$
LS	generated	$a = 3; p \in 0 : 0.05 : 0.5$	10000	$5.98 - 12.04$
Big IS	generated	$a = 1.5, 2; p = 0.1$	50000;100000	3.3,4.4
BL[1,2]	real data	–	Internet	3.2 [4]
LC	real data	–	Internet	3.2 [5]

[4]based on [22]
[5]based on [23]

The paper is organized as follows. Section II discusses our topological findings on trees cut from power law topologies. In Section III we outline our found receiver group size estimation method and prove it both empirically and analytically. Section IV suggests two algorithms that leverage the found method for session size estimation and analyzes their performance. Additionally, we outline simulation results of the Fast Algorithm. Section V discusses the accuracy of the found method in details. We conclude with our conclusions and discussion of future work.

II. EMPIRICAL CHARACTERISTICS OF MULTICAST TREES

This section details our findings on the structure of multicast trees cut from generated power law topologies, as well as the Internet. These findings are the basis for the estimation method we present in Section III, and are of interest in their own right.

Little work has been done on modeling and characterizing multicast trees. Chalmers and Almeroth [3] investigated the branching characteristics of Internet multicast trees on the MBone and their impact on multicast efficiency. They found that multicast trees tend to have low average internal degree that grows logarithmically with the number of receivers in the tree, and a maximum height of approximately 23 nodes. They also found a high frequency of "relay" nodes that have a degree of two throughout the tree. In previous work, Pansiot and Grad, who constructed trees from a graph based on true routing paths in the Internet, also showed a high frequency of relay nodes in the tree graphs [18]. Chuang and Sirbu [1] found a power law between the number of links in a multicast delivery tree connecting a random source to m random and distinct network sites; Philips *et al.* [2] developed a mathematical explanation for the Chuang–Sirbu scaling law, for networks with an exponential reachability function.

A. Topology and Tree Generation

Our method for producing trees is the following. First, we generate power law topologies based on the Notre-Dame model [19] which has been shown to reflect the Internet topology quite well despite its limitations [20]. The model specifies four parameters: a_0, a, p, and q,[3] where a_0 is the initial number of detached nodes, and a is the initial connectivity of a node. When a link is added, one of its end points is chosen uniformly, and the other with probability that is proportional to the node degree. This reflects the fact that new links often attach to popular (high degree) nodes. The growth model is the following: with probability p, a new links are added to the topology. With probability q, a links are rewired, and with probability $1-p-q$ a new node with a links is added. The rewiring parameter, q, is intended to corporate local events and increase the small world effect. An analysis of the rewiring parameter effect showed that the degree distribution approaches an exponential distribution for large q values [19]; our measurements showed that small q values do not affect any of our results and measurements. Hence, for simplicity, we take $q = 0$ in the generated topologies. Note that a, p and q determine the average degree of the nodes. We created a vast range of topologies, but concentrated on several parameter combinations that can be roughly described as very sparse (VS), Internet like sparse (IS) and less sparse (LS). Table I summarizes the main characteristics of the topologies used in this paper.

From these underlying topologies, we create the trees in the following manner. For each predetermined size of client population we choose a root node and a set of clients. Using Dijkstra's algorithm we build the shortest path tree from the root to the clients. To create a set of trees that realistically resemble Internet trees, we defined four basic tree types. These types are based on the rank of the root node and the clients nodes. The rank of a node is its location in a list of descending degree order, in which the lowest rank, one, corresponds to the node with the highest degree in the graph. For the case of a tree rooted at a big ISP site, we choose a root node with a low rank, thus ensuring the root is a high degree node with respect to the underlying topology. Then, we either choose the clients as high ranked nodes, or at random, as a control group. Note, that due to the characteristic of the power law distribution, a random selection of a rank has a high probability of choosing a low degree node. The next two tree types have a high ranked root, which corresponds to a multicast session from an edge router. Again, the two types differ by the clients degree distribution, which is either low, or picked at random.

The tree client population is chosen at the range [50,4000] for the 10 000 node generated topology, [50,10 000] for the 100 000 node generated topology, and [500,50 000] for the trees cut from real Internet data. For each client population size, 14 instances were generated for each of the four tree types. All of our results are averaged over these instances. The variance of the results was always negligible.

There are two underlying assumptions made in the tree construction. The first, is that the multicast routing protocol delivers

[3]The notations in [19] are m_0, m, p and q, respectively.

TABLE II
LINEAR FIT OF DEGREES AND FREQUENCIES

	a	p	Y	ACC
topology	2	0.1	-2.50X + 4.49	0.9721
	High degree root, low degree clients		Root and clients chosen randomly	
Receivers	Y	ACC	Y	ACC
50	-2.76X + 2.25	0.9337	-3.27X+2.68	0.9752
100	-2.64X + 2.42	0.9613	-2.96X+2.71	0.9611
300	-2.50X + 2.73	0.9730	-2.64X+2.85	0.9717
500	-2.58X + 2.97	0.9732	-2.58X+2.96	0.9654
750	-2.57X + 3.12	0.9825	-2.59X+3.09	0.9609
1000	-2.56X +3.23	0.9785	-2.59X+3.21	0.9728
1500	-2.64X + 3.45	0.9812	-2.56X+3.32	0.9741
2000	-2.58X + 3.52	0.9858	-2.60X+3.44	0.9620
2500	-2.65X + 3.66	0.9817	-2.63X+3.57	0.9731
3000	-2.66X+ 3.75	0.9851	-2.58X+3.57	0.9670
4000	-2.70X + 3.90	0.9825	-2.64X+3.73	0.9611

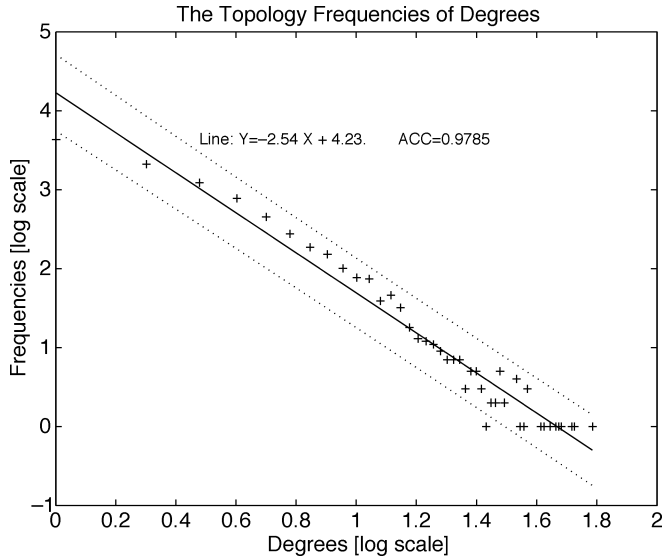


Fig. 1. Frequency of degrees for a 10 000 node topology with $a_0 = 6$, $a = 1$, $p = 0.3$, $q = 0$.

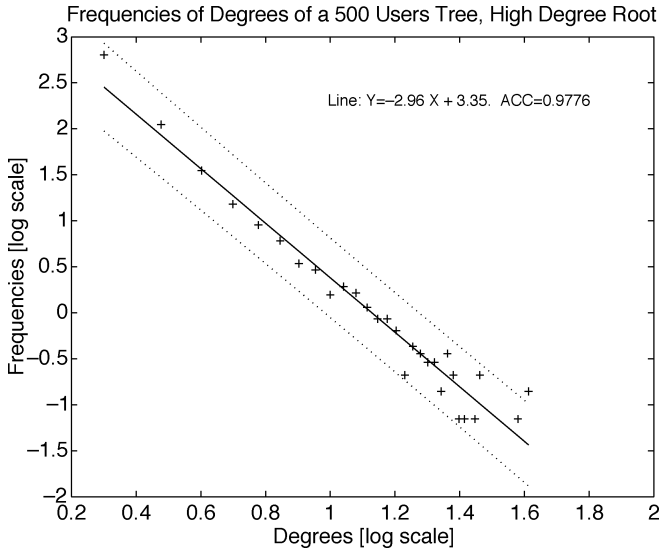


Fig. 2. Tree with 500 low degree clients, high degree root. Cut from topology $a_0 = 6$, $a = 1$, $p = 0.3$, $q = 0$.

a packet from the source to each of the destinations along a shortest path tree. This scenario conforms with current Internet routing. For example, IP packets are forwarded based on the reverse shortest path, and multicast routing protocols such as Source Specific Multicast [21] deliver packets along the shortest path route. In addition, we assume that client distribution in the tree is uniform, as has been shown by [2], [3]. In addition, all trees were tested and validated for the Chuang–Sirbu law [1].

B. Tree Characteristics

1) Degree-Rank and Size-Rank Power Laws: Our results show that trees cut from a power law topology obey a similar power law. Specifically, we compared the degree-frequency power law found by [5]. Fig. 1 shows in log-log scale the degree frequency plot for 10 000 nodes topology generated with the parameter set $a_0 = 6$, $a = 1$, $p = 0.3$, $q = 0$. The dotted lines here, and in the rest of the linear fit figures, mark the 95% confidence interval.

Fig. 2 shows the same plot for a multicast tree with 500 low degree clients and a root with a high degree. In Table II we summarize the best linear fit parameters in a log-log scale for all trees generated for the topology set $a_0 = 6$, $a = 2$, $p = 0.1$, $q = 0$. It can be seen that the power law holds even for very small trees, e.g., for a tree with 50 multicast clients that has on the average around 200 nodes. The same phenomenon appears in all the trees cut from all topologies, regardless of the way the root and the client nodes were chosen.

These findings conform with the findings of [3] and [18] who found a very large frequency of relay nodes in the trees, i.e., nodes with a degree of two. In a power law relationship of frequency and degree, the frequency of two degree nodes is the highest in the tree. Leaf nodes are determined by clients, and are a subset of the clients.

We also found that the distribution of degrees at a specific distance from the root, i.e., in a certain depth ring, also showed a power law distribution of degree-rank, but with different slopes.

Given the above findings, it is important to note the following. Cohen *et al.* [24] showed that the maximal node degree in a graph of N nodes is proportional, for Internet-like topologies, to approximately the square root of the number of nodes. More

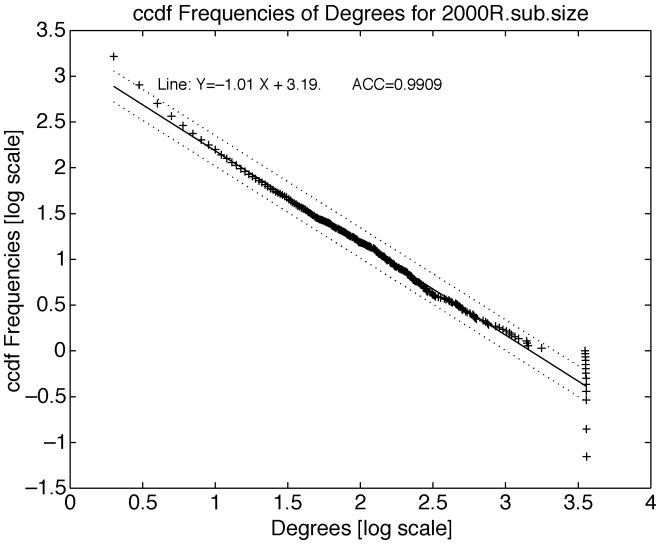


Fig. 3. Sub-tree size CCDF distribution, for a 2000 node tree cut from topology $a_0 = 6, a = 2, p = 0.1, q = 0$.

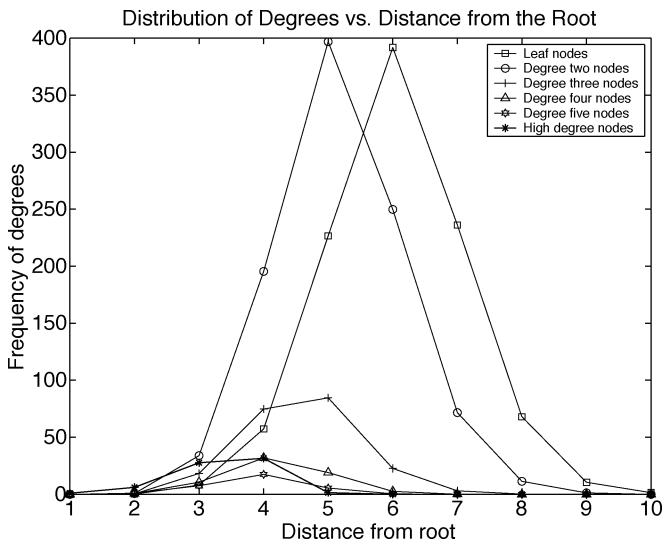


Fig. 4. Distribution of the distance of high degree, two to five degree and leaf nodes in a tree cut from topology $a_0 = 6, a = 1, p = 0.3, q = 0$.

precisely, $D_{\max} \sim N^{1/(\alpha-1)}$, where α is the exponent of the degree-frequency power law of the topology. Hence, all resulted degree-frequency graphs of finite sizes exhibit a cut-off at the tail. This holds true for partial views taken from the Internet, with the cut-off being a result of the partiality as well as from the finite size of the Internet itself.

The second power law we found for the trees is of frequency and size of the sub-trees in each tree. Namely, the self similarity holds not only for the degree distribution in the tree, but also for its inner structure. Fig. 3 shows the excellent fit of the complementary cumulative distribution function of the sub-tree sizes of a 2000 node tree. The tree, with a high degree root, is cut from a 10 000 node topology with the parameter set $a_0 = 6$, $a = 2$, $p = 0.1$, $q = 0$. The size distribution differs from the degree distribution in that the big sub-trees, although almost similar in size, may differ by one or two nodes, which is negligible compared to their overall size. Thus, we give the CCDF graph, which plots the probability that the observed values are greater than the ordinate. It can be seen that the fit to a power law is over 99%. The slope computed for the PDF graph without the tail, resembles the one of the degree distribution.

2) Per Degree Distance Distribution: Cheswick *et al.* [8] found that the distribution of the number of nodes at a certain distance from a point in the Internet is similar to the Gamma distribution. Our results show that the distribution of distance from the root of nodes of a certain degree seems close to a gamma distribution, although we did not determine its exact nature. Fig. 4 shows the distribution of the distance of two to five degree, leaf and high degree nodes, where high degree nodes are nodes with a degree six and higher. In this case the root is a low degree node, and the tree has 1000 low degree clients. As can be seen, the high degree nodes tend to reside much closer to the root than the low degree nodes, and in adjacent rings. In this example, most of them are in the second to forth depth rings around the root.

This phenomenon was even more obvious when the root was a high degree node. We found the following observation with regard to power law generated topologies. The high degree nodes seem to form a "core" with a low diameter (around five hops for trees cut from the generated topologies, and seven for trees cut from Internet data) and most of the other nodes in the network are not distanced more than three to five hops away from this core. Subramanian *et al.* [25] observed a similar phenomenon at the Internet AS topology, although obtained from directed BGP routing tables.

The distribution of client distances from the tree root is given by the leaves distances in Fig. 4. Note that the longest path to a client is the tree height. Our results show that the less connected the underlying topology, the taller is the average tree cut from the topology.

3) Empirical Results From Internet Data: We verify the above findings with results obtained from real Internet data. Our results are verified on two different data sets. The first is an Internet partial view at the routers level, obtained from the Lucent Internet Mapping Project [23]. We used this data set as the underlying topology, from which we cut trees in the same manner described in Section II-A. We denote this topology by LC.

For the second data set, we use the client population of http://www.bell-labs.com, which is a medium size web site. This may represent the potential audience of a multicast of a program with scientific content (such as the livecast of the INFOCOM conference). From this set two lists of clients were obtained, and traceroute was used to determine the paths from the root to the clients. It is important to note, that the first three levels of the tree consist of routers that belong to the site itself, and therefore might be treated as the root point of the tree, although in these graphs they appear separately. We denote this tree as the BL tree.

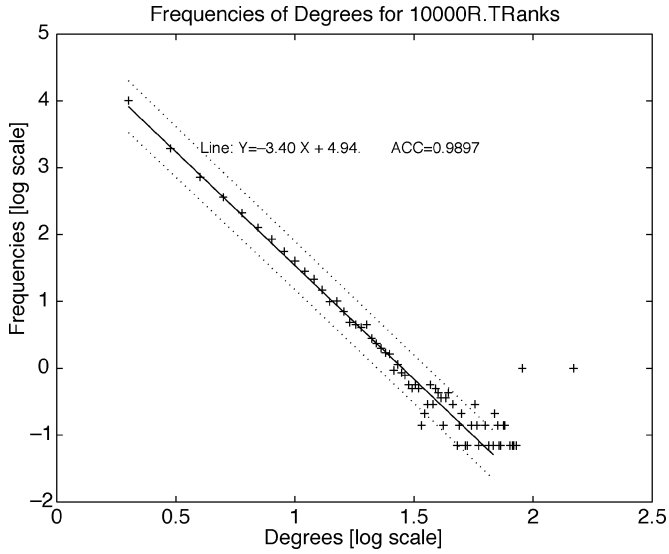


Fig. 5. Frequency of degrees of a 10 000 node tree cut from the LC Internet data.

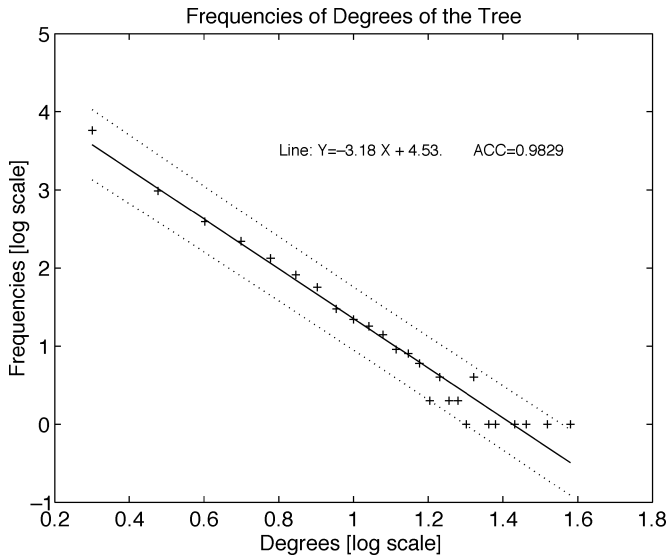


Fig. 6. Frequency of degrees of the BL Internet tree.

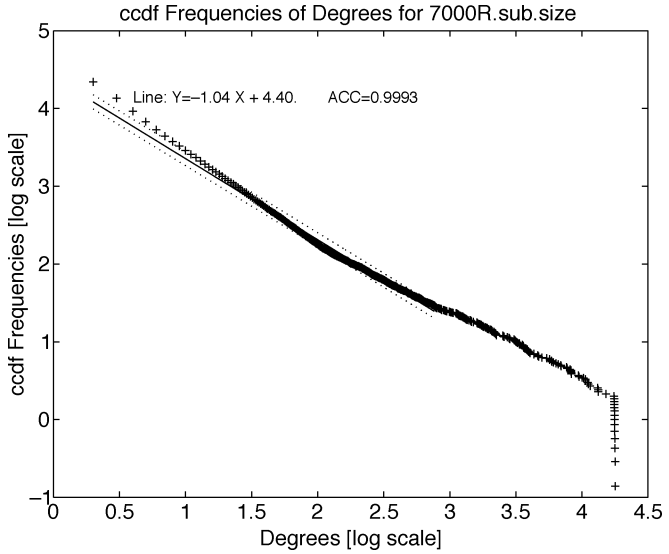


Fig. 7. Size distribution of a 7000 clients tree cut from the LC data.

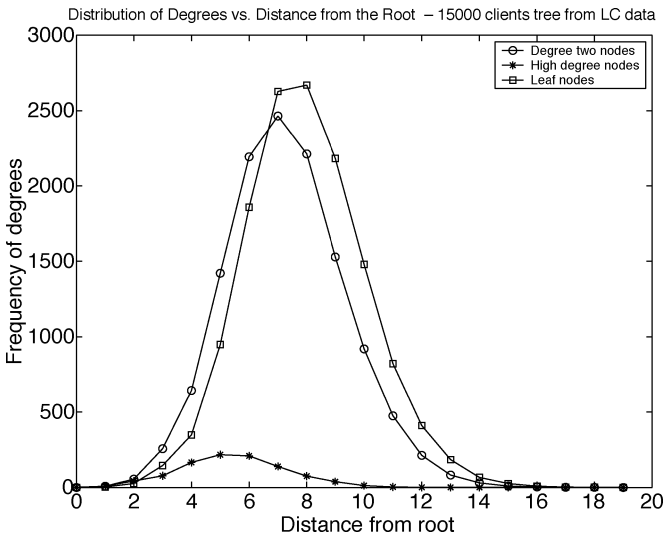


Fig. 8. Distance of two, high degree and leaf nodes from the root of a 15000 client tree cut from the LC Internet data.

Fig. 5 shows the frequency of degrees for a 10 000 node tree cut from the LC topology. The tree, which is an average of 14 instances, exhibits a clear degree-frequency power law with a good fit.[4] The tree was chosen with a high degree root, and low degree leaf nodes. The variance of the instances of each tree was negligible, and the same result was obtained for each of the generated trees, with as low as 1000 clients and as high as 50 000. Fig. 6 shows the frequency of degrees for the BL tree. The linear fit of the log-log ratio is excellent, with a correlation coefficient of 0.9829.

[4]We fit the data for the points above the line $\mathrm{Y} = 0$ which capture all the degrees that appear on average, at least, once in every tree. To extend the fit below this line we need more trees. If we want to get rid of the noisy tail all together we need to generate, at least, an order of 10^4 trees as our fit predicts that the highest degree points will appear on the average in less than one of every 10^3 trees.

Fig. 7 shows the CCDF of the sub-tree sizes of a tree with 7000 clients cut from the LC data. The root is a high degree node, and the clients are low degree nodes. Note that every point in the graph is the result of an average of 14 instances therefore the tail was omitted from the fit. The size-rank power law appears in all the trees cut from this data.

Fig. 8 shows the distribution of the distance of two degree, leaf and high degree nodes, for a 15000 client tree, cut from the LC data. The majority (90%) of the high degree nodes reside within a distance of eight hops from the root, while the clients are distanced up to 18 hops from the root.

III. Receiver Group Size Estimation Method

While all of the above observations are interesting and help in our understanding of multicast trees, we were intrigued whether

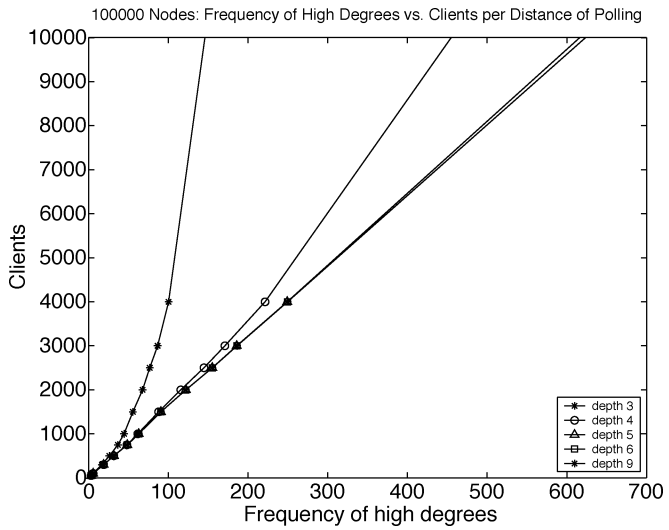


Fig. 9. Clients versus frequency of high degree nodes. Cut from a 100 000 nodes topology with $a_0 = 6, a = 1.5, p = 0.1, q = 0$.

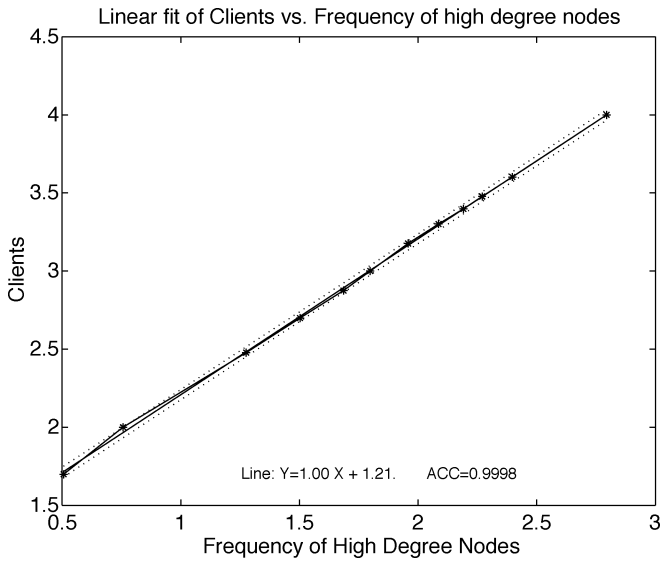


Fig. 10. Linear fit of Clients versus frequency of high degree nodes. Cut from a 100 000 nodes topology with $a_0 = 6, a = 1.5, p = 0.1, q = 0$.

we can use any of this knowledge to evaluate the size of a multicast tree. We compared the degree of the nodes in the tree to their degree in the topology, and focused on the high degree nodes. Interestingly, we found that while some nodes had a tree degree that is significantly smaller than their degree in the underlying network topology, other nodes seemed to have a tree degree close to their network degree. We then compared the frequency of nodes with degree i and above (high degree nodes) to the number of clients in the tree, and found a linear ratio with a correlation coefficient of not less than 0.99. We term this ratio the HCN_i ratio (hubs-to-client number ratio).

Next, we outline our findings on HCN_i ratio for both simulated trees and trees cut from the real Internet. We proceed by giving a mathematical analysis of our results for power law trees.

A. Empirical Findings

We have found that an HCN_6 ratio of 1:16 is a very good predictor for trees cut from the Internet, and most generated topologies. Fig. 9 shows the HCN_6 ratio in trees cut from a 100 000 node topology. The topology parameters are $a_0 = 6$, $a = 1.5$, $p = 0.1$, $q = 0$, and the root node of all trees is a high degree node. The linear ratio is obtained after gathering the information from not more than five depth rings around the root. We plotted the frequency of high degree nodes obtained after scanning three, four, five, six and nine depth rings around the root. As can be seen from the graph in Fig. 9, the entire information was obtained until the sixth depth ring—the following rings did not add any more information. The HCN_6 ratio was found to be 16. Fig. 10 shows the excellent fit of the HCN_6 ratio with a correlation coefficient of 0.9998. When we plotted the data for trees cut from this topology with a low degree root, we obtained very similar results. The ratio was again 16, with a correlation coefficient of 0.9996. However, another depth ring was needed to obtain accurate results, since the root was not as close to the core of high degree nodes as in the previous case.

We verified our results using actual Internet data on the client population of the Bell Labs web site described in Section II,

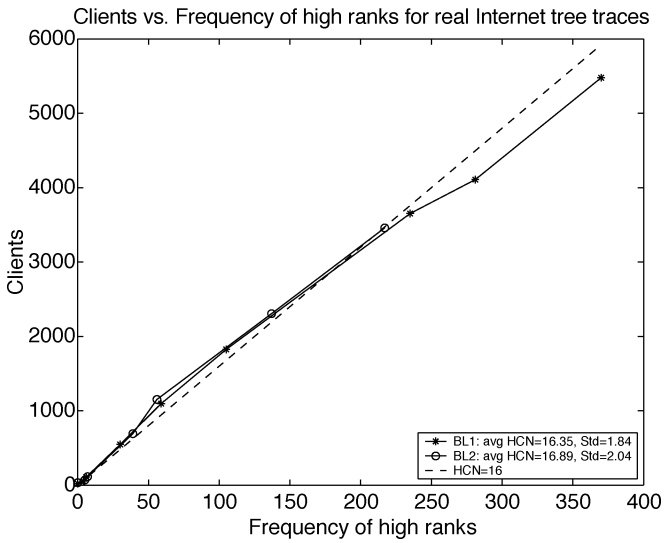


Fig. 11. Clients versus high degree nodes and the HCN predictor for the BL[1,2] trees.

and on trees cut from the data from Cheswick's Lucent Internet mapping project, noted LC, also described there. The Bell Labs client population data contains two log files. The first, denoted BL1, has 10897 clients and the second, BL2, has 7356. We created subsets of clients by randomly selecting entries from the log files, and cut the corresponding trees for these subsets from the original trees. Fig. 11 shows the ratio between the 16 predictor and the actual number of clients in the generated trees. For BL1 the ratio was $1{:}16^{0.99}$ with a fit of 99.75%, for BL2 the ratio was $1{:}16^{1.04}$ with a fit of 99.72%. For client populations larger than roughly 1500 clients the predictor of 16 gives an excellent estimate—within 9% of the actual number of clients.

The LC data gives a partial view of the Internet at the router level with more than 110 000 routers. From this topology, we cut trees in the same manner described in Section II. Again, each result is averaged over 14 instances. Fig. 12 shows the ratio

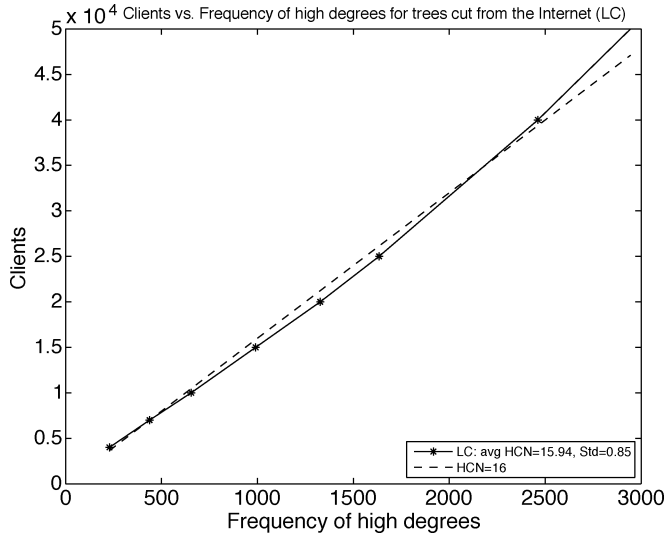


Fig. 12. Clients versus high degree nodes and the HCN predictor for the trees cut from the LC topology.

between the number of clients and high degree nodes, compared with the predicted value from the simulations, 16. The average value of the ratio is 15.89, with a standard of deviation of 0.9. Hence, a 16 predictor for the ratio gives a very good estimation for this data also.

For the generated topologies and the Internet experiments, our results are less definite for very small trees. We found that HCN_6 ratio $= 16$ is accurate when client population is at least 0.1% the size of the underlying topology. Nevertheless, for the Internet, our experiments yielded very good results for group sizes of 1500 clients and more. Note that when the group size is small enough, exact counting of the clients can be done with a reasonable cost.

While a predictor of 16 was shown to be a very good predictor for large groups, it becomes less scalable when the group size is extremely large. For example, in the case of a multicast tree with a million clients, the expected number of high degree nodes is 62500. A good solution for this problem is to increase the degree of the sample nodes. For example, in the case of very large groups, counting the number of nodes with degree higher than nine will produce an accurate prediction, with a ratio of 1:48, namely HCN_{10} ratio $= 48$. Note that sampling nodes with a larger degree gives us a coarser estimation. Our experiments show that when we sample nodes of degree ten and above the estimation is accurate only for group sizes of at least 1.5% the size of the underlying topology. Remember that sampling nodes of degree 6 and above yields a good estimation for trees as small as 0.1% of the network.

B. Analytical Derivation of HCN_6 Ratio

In this section, we derive the HCN_6 ratio for trees in power law topologies. Our experiments have shown that the group of leaf nodes of a tree closely approximates the tree's client population. For simplicity we take the exponent of the underlying topology degree probability instead of the tree's, but these are fairly close.

Given a tree with N nodes, we denote by L the number of leaf nodes and by $\tilde{N}$ the number of non leaf nodes. Let $\tilde{\mathcal{N}}$ be the group of non leaf nodes. The average internal degree is defined by: $r = (\sum_{j\in\tilde{\mathcal{N}}} d_j)/\tilde{N}$ where d_j is the degree of node j. But by its definition it also holds that $\sum_{j\in\tilde{\mathcal{N}}} d_j = 2\tilde{N} + L - 1 \approx 2\tilde{N} + L$, and $\sum_{j\in\tilde{\mathcal{N}}} d_j = N + \tilde{N} - 1 \approx N + \tilde{N}$. Given all the above, we can write

$$L = N \cdot \frac{r-2}{r-1} \tag{1}$$

which holds for any tree.

Given that p_i is the probability to find a node with degree i in the tree, we can rewrite the above expression for r:

$$r = \frac{1}{1-p_1} \cdot \sum_{i=2}^{N} i \cdot p_i \tag{2}$$

and the probability conservation equation

$$\frac{L}{N} + \sum_{i=2}^{N} p_i = 1. \tag{3}$$

Substituting (1) in (2) and (3), and given that the degree distribution obeys the power law $p_i = c \cdot i^{-\alpha}$, we get that

$$r = \frac{S_1}{S_2}, \quad c = \frac{r}{S_1 \cdot (r-1)} \tag{4}$$

where $S_1 = \sum_{i=2}^{N} i^{-(\alpha-1)}$ and $S_2 = \sum_{i=2}^{N} i^{-\alpha}$.

The HCN_6 ratio is defined by

$$\text{HCN}_6^{-1} = \frac{\sum_{i=6}^{i=N} p_i \cdot N}{L}. \tag{5}$$

Plugging (1) and (4) in (5) yields

$$\text{HCN}_6^{-1} = \frac{(1-S_3)\cdot(r-1)}{r-2} - 1 \tag{6}$$

where $S_3 = \sum_{i=2}^{i=5} i^{-\alpha}$.

Fig. 13 shows how the HCN_6 ratio in (6) changes with α. For $3 \le \alpha \le 4$ the HCN_6 ratio changes between 14.5 and 19. Hence, a precise value for the tree's α will yield an excellent evaluation of the number of leaf nodes in the tree, and hence a good estimation to the client population (see Section V for a discussion on how to obtain a more accurate α value). Nevertheless, our results show that for the shortest path trees cut from the Internet, as well as from most of our generated topologies, HCN_6 ratio $= 16$ gives a very good estimation. Understanding the precise correlation between our empiric and analytical results may lead to a deeper understanding of the Internet topology, and is the subject of our next work.

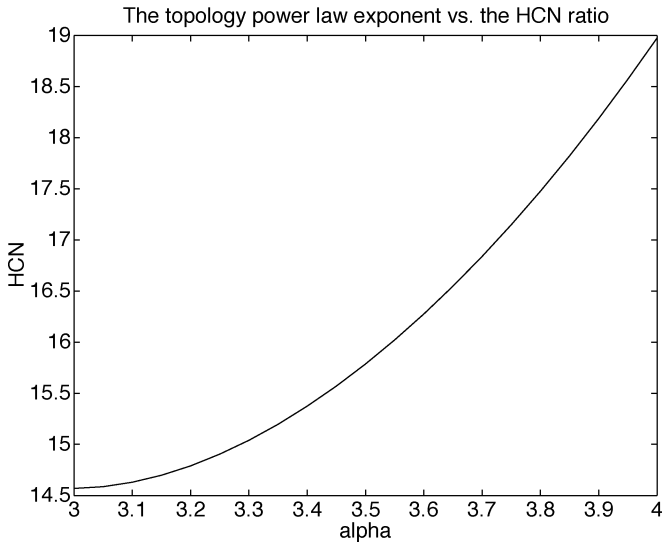


Fig. 13. Change in HCN_6 ratio with α.

Algorithm 1 (Basic)

1. Send $Req(d)$
2. $n \leftarrow 0$
3. Activate $\text{Timer}_1(T_{d_1})$

When *Rep* arrives

4. $n{+}{+}$

When TimeOut_1

5. return$(c_d \cdot n)$

Fig. 14. Formal description of the basic algorithm for the root node.

IV. Estimation Algorithms

A. A Basic Algorithm

The findings in the previous section give rise to an algorithm for estimating the number of clients in a multicast tree, in which the number of nodes with five or more child nodes is counted. The main idea, given formally in Fig. 14, is that the root multicasts a feedback request, Req, along the multicast tree. The request carries the parameter d, which indicates the minimal node degree that needs to report back. Such a node, upon receiving the request, replies with a UDP Rep packet sent directly to the root. The root waits for a time long enough to ensure that most replies are accepted. The root then counts the number of different replies it receives, and by multiplying with the appropriate coefficient produces the estimate.

Note that for the Internet, T_{d_1}, the time the root waits for the replies to arrive, should be quite large. Specifically, T_{d_1} needs to be long enough such that the vast majority of slow responses due to round trip and processing delays are not lost. (We assume that T_{d_1} of several seconds satisfies these requirements.)

B. Fast Algorithm

The Fast Algorithm, formally presented in Fig. 15, is motivated by the need to obtain a fast estimation on the client population. We would like to determine the termination rule in a way that guarantees that a significant portion of the Rep messages has already arrived. In the basic algorithm we achieve this by setting a very large timeout. Here, we monitor the Rep message arrival process to achieve this goal.

Algorithm 2 (Basic)

1. Send $Req(d)$
2. $n \leftarrow 0$
3. $ndt \leftarrow 0$
4. Activate $\text{Timer}_1(T_{d_1})$

When TimeOut_1

5. if $ndt = 0$ then
6. return(0)
7. else
8. Activate $\text{Timer}_2(T_{d_2})$
9. $n \leftarrow ndt$
10. $ndt \leftarrow 0$

When TimeOut_2

11. $n+ = ndt$
12. if $ndt \leq K_{th} \cdot n$
13. return$(c_d \cdot n)$
14. else
15. Activate $\text{Timer}_2(T_{d_2})$
16. $ndt \leftarrow 0$

When *Rep* arrives

17. $ndt{+}{+}$

Fig. 15. Formal description of the Fast Algorithm for the root node.

We start the algorithm with an *initial sampling period*, T_{d_1}, whose purpose is to enable responses from the high degree nodes in the k-neighborhood of the root to arrive back at the root. If by the end of the initial sampling period the root receives no replies, it assumes the group is either very small or inactive. If the root receives Rep messages, a shorter sampling period termed the *iterative sampling period* is activated repeatedly until the termination condition is satisfied. The purpose of the iterative sampling period, noted T_{d_2}, is to enable the algorithm to converge to a good estimate within a short time.

There are several options to determine a termination condition based on the Rep message arrival process. We can choose a threshold and stop when the message arrival rate drops below it. This solution, however, is not immune to network jams, and is very sensitive to the threshold's value. Another option is to stop when the rate keeps dropping for several successive iterative sampling periods. In this case, the algorithm is very sensitive to the length of the iterative sampling period. If it is too short the algorithm might terminate too early with a large estimation error. On the other hand, a long iterative sampling period might cause the algorithm to be less practical.

Thus, we devised a termination rule (see line 12 in Fig. 15) that can self-tune according to the arrival process. Under reasonable conditions it will guarantee termination within a preset estimation error. The algorithm terminates when the number of replies received at the root during one of the iterative sampling periods *does not* improve the estimation by more than K_{th}, where K_{th} is the estimation error. For example, setting the iterative sampling period to the average two-hop delay and the initial sampling period to $2T$, causes the algorithm to terminate when the replies gathered from the $T + i$th depth ring, at the ith iterative sampling period, do not improve the estimation by more than K_{th}. Under reasonable network conditions, about half of the replies from this depth ring reach the root node by the end

TABLE III
FAST ALGORITHM TIME AND PREDICTION

Clients	300	500	750	1000	1500	2000	3000	4000
ND prediction	304	512	736	992	1472	2000	2992	4000
ND time	10.0	12.0	10.0	10.0	10.0	10.0	12.0	12.0
ED prediction	256	400	672	960	1456	1920	2736	3856
ED time	12.0	12.0	18.0	14.0	20.0	18.0	16.0	20.0

of the ith iterative sampling period. Thus, the termination condition enables the algorithm to stop when it identifies the end of the adjacent depth rings around the root.

1) Performance Evaluation of the Fast Algorithm: In this section we estimate the delay of the Fast Algorithm and define the average values for T_{d_1} and T_{d_2}. The delay of a packet traversing a single link, d, is comprised of two components: $d = \Delta + q$, where Δ is the fixed minimum link delay and q is a random variable representing the queueing delay, which is exponentially distributed. We would like to derive the distribution of the queueing delay of a packet traveling h links. The density function of the delay, $d_h(t)$, is a convolution of the density functions of $q(t - h\Delta)$, h times:

$$d_h(t) = q(t - h\Delta) * q(t - h\Delta) * \cdots q(t - h\Delta). \quad (7)$$

Let us define, for simplicity

$$\tau = t - h\Delta. \quad (8)$$

Thus, $d_h(\tau)$ is a gamma random variable with parameters h and λ. Namely

$$d_h(\tau) = \frac{\lambda^h \tau^{h-1} e^{-\lambda\tau}}{(h-1)!} \quad (9)$$

where λ^{-1} is the average queueing delay. Assuming that all high degree nodes reside within h hops from the root node of the tree, and let the probability of a high degree node to reside at distance h from the root be $p_{hd}(h)$, from (7) and (9) we get that the probability distribution function of the *total* delay is

$$D(\tau) = \sum_{i=0}^{h} D_h(\tau) p_{hd}(i) = \sum_{i=0}^{h} \frac{\lambda^i \tau^i \Gamma(i, i\tau)}{(i\tau)^i} p_{hd}(i) \quad (10)$$

where $\Gamma(\cdot,\cdot)$ is the incomplete gamma function [26, sec. 1.2.11]. Plugging back (8) in (10) we get that the final form of the total delay probability distribution function is

$$D(t) = \sum_{i=0}^{h} \frac{\lambda^i \Gamma\left(i, i(t - h\Delta)\right)}{i^i} p_{hd}(i). \quad (11)$$

The values of T_{d_1} and T_{d_2} need to be established in a way that will ensure that the majority of the replies are gathered. For example we can select T'_{d_1} to be the value of t that minimizes $D(t) = 0.5$, meaning that ensures that on the average we wait for half of the replies to be done waiting at queues.

Alternatively, we should choose T_{d_1} to be long enough for each node to at least reach the core, preferably its center. Let us define by r_c the estimated radius of the core, in which we have established that most high degree nodes reside. Let us define by r_e the average distance from an edge node to the core. Then

$$T_{d_1} = 2(r_c + r_e)(\Delta + \bar{q}) \quad (12)$$

thus ensuring that T_{d_1} is sufficient for the request to reach the core vicinity and for some of the replies of high degree nodes to arrive back to the root. In the same manner, setting

$$T_{d_2} = 2(\Delta + \bar{q}) \quad (13)$$

yields an iterative sampling period of one hop round trip delay, thus enabling the algorithm to obtain most of the information from the next hop. From our experiments, as described in Section II, we discovered that the values of $r_c = 7$ and $r_e = 6$ are sufficient for today's Internet.

In Table III, we summarize the simulation results of the Fast Algorithm. We denote by τ the average one hop delay. The hop delay is either normally distributed (ND) or exponentially distributed (ED). The length of the initial sampling period is 8τ, and the length of the iterative sampling period is 2τ. The results in this table are obtained for trees cut from topology $a_0 = 6$, $a = 1$, $p = 0.3$, $q = 0$, and the Fast Algorithm was executed with an estimation error of $K_{th} = 2\%$. All the high degree nodes in the generated trees reside within five depth rings from the root. Time units are in $[\tau]$. Note that due to the long tail of the exponential distribution, an iterative sampling period of 2τ is shown to be too short, since the exponential case represents a bursty network. However, when the delay is normally distributed with variance τ, the algorithm counts all of the high degree nodes in the tree within less than 12τ time units, which is less than the measured average clients' round trip delay of 16τ for these trees.

V. DISCUSSION

Our results, which show a strong correlation between the number of high degree nodes and the number of clients, hold for all tree types over all tested power laws topologies. As stated before, all of the results obtained from the simulations as well as the LC data were averaged over 14 instances. When degrees 6 and higher are chosen (i.e., $d = 6$), we found that 16 is a very good predictor in the average case. In this section, we discuss the accuracy of this result for specific trees.

We examined the specific predictors of the 14 instances of a 7000 clients tree cut from the LC data. The smallest ratio was 15.52 and the largest 16.78, yielding a maximal error of 5%. Fig. 16 shows our results for 14 trees that were cut from a 100 000 node topology. The root is a randomly chosen high degree node and the clients are chosen uniformly. The figure

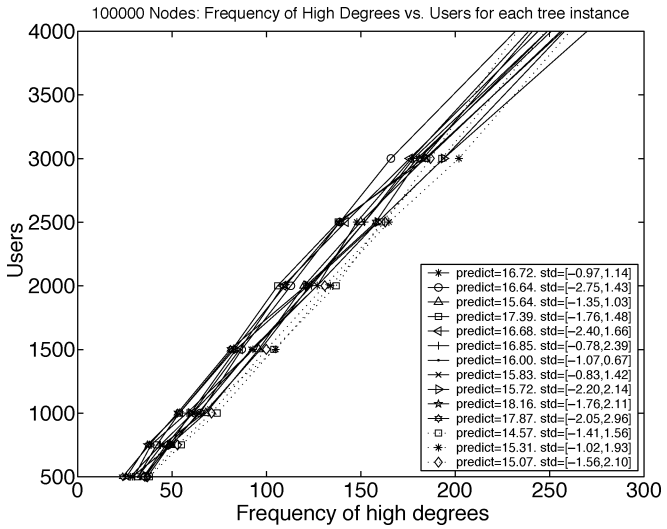


Fig. 16. Clients versus high degree nodes for each of the 14 instances of the tree.

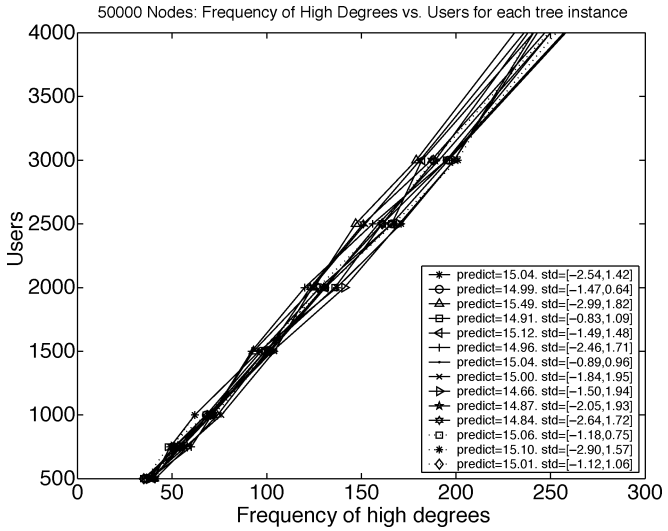


Fig. 17. Clients versus high degree nodes for each of the 14 instances of the tree.

legend details for each of the trees its specific slope, i.e., its average ratio between the number of clients and the high degree nodes over all points. It also specifies for each tree the maximal and minimal deviation points, i.e., the ratio at the points which are furthest from the average for that tree. We can see that the slopes of most of the trees are within 10% of the average predictor. This phenomenon can be seen throughout the different tree types. The worst deviation from the average predictor of a slope was 12.5%. A few points diverge up to 30% from the estimation, yet this should be expected, given the statistical nature of the estimation method.

We found that the reliability of the prediction increases with the group size. According to our findings, described in Section III, the found predictor is accurate only for medium to large groups. When group size exceeds 1000 clients, the average predictor yields very good estimations, with not more than a 10% error. For the general case, for all group sizes, the vast majority of the individual test points are within a marginal limit of 15%. For our analysis on Internet logs the estimation error was no more than 15% in almost all cases. The single exception was for a group of size 1153, which exhibited a 22% estimation error.

We have found that instances of a tree with the same root node tend to have a more stable behavior. Thus, a root can calibrate the estimator for its trees by counting the number of clients and the number of high degree nodes when the trees are reasonably small, and use the more accurate estimator when the trees grow. Fig. 17 demonstrates this for 14 trees that were generated with the same root. It is clear that the best estimator for these trees is around 15 and the deviation is less than 4% (compared with 12.5% for the general case). The individual point estimates here are also much better—within 16% of the calibrated estimate, 15.

VI. Conclusion

We presented our findings on the characteristics of shortest path trees cut from power law topologies. We base our conclusions on extensive simulations, and real Internet topologies from two different sources: The Bell Labs web site logs, and the Cheswick–Burch Internet mapping project. All of the empirical and the simulation reuslts agree. Our results may not hold, however, in cases where the group of receivers was generated with different affinities (clusteredness) [27] or with a client population from a specific region of the Internet.

Our findings may improve our understanding of multicast trees and therefore may help theoretical and practical research done in this area. We have shown that the structure of such trees follows power laws of rank-degree and rank-size, and that high degree nodes tend to reside in a low diameter neighborhood. We found a linear ratio between the number of high degree nodes and the number of multicast tree leaves. We also proved this ratio analytically, and devised the Fast Algorithm that uses this ratio to estimate the tree client population in less than the Internet round trip delay.

The Fast algorithm, when used as an initial estimator to polling based counting algorithms such as [13] and [15], enables these algorithms to converge much faster, especially for medium and large groups. Note, that these algorithms performance is improved significantly with a tight initial group size estimation. It is also beneficial for transport layer feedback suppression algorithms and control algorithms which need to know the session size such as RTCP [11]. Finally, the Fast Algorithm can be used by network providers in calculating the gain from multicast with metrics such as the one suggested by Chuang and Sirbu [1]. As part of our future work, we intend to include an addition to the Fast Algorithm that enables the root to receive online updates on the changes of the branching characteristics of the trees. These online updates sent by nodes going in or out of the *high degree nodes* group, enable efficient tracking over time of the multicast group size.

In general, we have found only a few examples where the estimator was off by more than 15%. When the estimator was calibrated to a specific root node the accuracy was a factor of four better.

This work presents a novel way for leveraging topological characteristics of a tree to obtain important knowledge such as

its size. A further understanding of the exact ratio between the trees and the underlying topology characteristics is the subject of our future work.

REFERENCES

[1] J. Chuang and M. Sirbu, "Pricing multicast communication: a cost based approach," presented at the INET'98 Geneva, Switzerland, 1998.

[2] G. Philips, S. Shenker, and H. Tangmunarunkit, "Scaling of multicast trees: comments on the Chuang-Sirbu scaling law," presented at the ACM SIGCOMM Cambridge, MA, 1999.

[3] R. C. Chalmers and K. Almeroth, "Modeling the branching characteristics and efficiency gains in global multicast trees," presented at the IEEE INFOCOM'01 Anchorage, AK, 2001.

[4] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "The origin of power-laws in Internet topologies revisited," presented at the IEEE INFOCOM New York, Apr. 2002.

[5] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," presented at the ACM SIGCOMM Boston, MA, 1999.

[6] R. Govindan and H. Tangmunarunki, "Heuristics for Internet map discovery," in *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, Mar. 2000, pp. 1371–1380.

[7] A. Medina, I. Matta, and J. Byers, "On the origin of power laws in Internet topologies," *ACM Comput. Commun. Rev.*, vol. 30, no. 2, pp. 18–28, Jan. 1, 1998.

[8] W. Cheswick, J. Nonnenmacher, C. Sahinalp, R. Sinha, and K. Varadhan, Modeling Internet Topology. Lucent Technologies, Tech. Rep. Technical Memorandum 113410-991116-18TM, 1999.

[9] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF RFC 1899, Jan. 1, 1996.

[10] S. Floyd, V. Jacobson, S. McCanne, C. Liu, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," presented at the ACM SIGCOMM, New York, 1995.

[11] J. Rosenberg and H. Schulzrinne, "Timer reconsideration for enhanced scalability," presented at the IEEE INFOCOM, San Francisco, CA, Mar. 1998.

[12] J. Nonnenmacher and E. W. Biersack, "Scalable feedback for large groups," *IEEE/ACM Trans. Netw.*, vol. 7, no. 3, pp. 375–386, Jun. 1999.

[13] J.-C. Bolot, T. Turletti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the Internet," in *Proc. ACM SIGCOMM*, London, U.K., Sep. 1994, pp. 58–67.

[14] J. Nonnenmacher and E. W. Biersack, "Optimal multicast feedback," presented at the IEEE INFOCOM, San Francisco, CA, Mar. 1998.

[15] T. Friedman and D. Towsley, "Multicast session membership size estimation," presented at the IEEE INFOCOM, New York, Mar. 1999.

[16] D. Rubenstein, J. Kurose, and D. Towsley, "Real-time reliable multicast using proactive forward error correction," presented at the NOSSDAV'98, Berlin, Germany, 1998.

[17] A. Alouf, E. Altman, and P. Nain, "Optimal online estimation of the size of a dynamic multicast group," presented at the IEEE INFOCOM, New York, 2002.

[18] J. Pansiot and D. Grad, "On routes and multicast trees in the Internet," *ACM Comput. Commun. Rev.*, vol. 28, no. 1, pp. 41–50, Jan. 1, 1998.

[19] R. Albert and A.-L. Barabási, "Topology of evolving networks: local events and universality," *Phys. Rev. Lett.*, vol. 85, no. 24, pp. 5234–5237, Dec. 11, 2000.

[20] T. Bu and D. Towsley, "On distinguishing between Internet power-laws topologies," presented at the IEEE INFOCOM 2002, New York, Apr. 2002.

[21] H. Holbrook and B. Cain, "Source-Specific Multicast for IP," Nov. 2002, IETF Internet draft, draft-ietf-ssm-arch-01.txt.

[22] P. Krishnan, D. Raz, and Y. Shavitt, "The cache location problem," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 568–582, Oct. 2000.

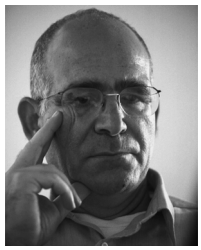
[23] H. Burch and B. Cheswick, "Mapping the Internet," *IEEE Computer*, vol. 32, no. 4, pp. 97–98, 102, Apr. 1999.

[24] R. Cohen, K. Erez, D. Ben Avraham, and S. Havlin, "Resilience of the Internet to random breakdowns," *Phys. Rev. Lett.*, vol. 4626, pp. 85–89, 2000.

[25] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz, "Characterizing the Internet hierarchy from multiple vantage points," presented at the IEEE INFOCOM, New York, Apr. 2002.

[26] D. E. Knuth, *The Art of Computer Programming*, 3rd ed. Boston, MA: Addison-Wesley, 1997, vol. 1.

[27] G. Lucas, A. Ghose, and J. Chuang, "On characterizing affinity and its impact on network performance," presented at the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research (MOMETOOLS), Karlsruhe, Germany, Aug. 2003.



Danny Dolev (SM'89) received the B.Sc. degree in mathematics and physics from the Hebrew University, Jerusalem, in 1971. His M.Sc. thesis in applied mathematics was completed in 1973, at the Weizmann Institute of Science, Israel. His Ph.D. thesis was on synchronization of parallel processors (1979).

He was a Post-Doctoral Fellow at Stanford University, Stanford, CA, from 1979 to 1981, and an IBM Research Fellow from 1981 to 1982. He joined the Hebrew University in 1982. From 1987 to 1993, he held a joint appointment as a Professor at the Hebrew University and as a Research Staff Member at the IBM Almaden Research Center. He is currently a Professor at the Hebrew University of Jerusalem. His research interests are all aspects of distributed computing, fault tolerance, and networking—theory and practice.

Osnat (Ossi) Mokryn received the Ph.D. degree in computer science from the Hebrew University of Jerusalem, Israel, in 2004. She is currently a Post Doctorate at the electrical engineering faculty, Technion IIT, Israel.

Her recent research focuses on BGP routing and Internet structure, topology and measurements.

Yuval Shavitt (S'88–M'97–SM'00) received the B.Sc. degree in computer engineering (*cum laude*), the M.Sc. degree in electrical engineering, and the D.Sc. degree from the Technion, Israel Institute of Technology, Haifa, in 1986, 1992, and 1996, respectively.

From 1986 to 1991, he served in the Israel Defense Forces first as a System Engineer and the last two years as a Software Engineering Team Leader. After graduation, he spent a year as a Postdoctoral Fellow at the Department of Computer Science at Johns Hopkins University, Baltimore, MD. Between 1997 and 2001, he was a Member of Technical Stuff at the Networking Research Laboratory at Bell Laboratories, Lucent Technologies, Holmdel, NJ. Starting October 2000, he is a Faculty Member in the School of Electrical Engineering at Tel-Aviv University, Tel-Aviv, Israel. His recent research focuses on Internet measurement, mapping, and characterization and on wireless networks.

Dr. Shavitt served as a TPC member for INFOCOM 2000–2003 and 2005, IWQoS 2001 and 2002, ICNP 2001, IWAN 2002–2005, tridentcom 2005–2006, and more, and on the executive committee of INFOCOM 2000, 2002, and 2003. He was an Editor of Computer Networks 2003–2004, and served as a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and the *Journal of the World Wide Web*.