The Hopcount to an Anycast Group

Piet Van Mieghem

Delft University of Technology[*]

Abstract

The probability density function of the number of hops to the most nearby member of the anycast group consisting of m members (e.g. servers) is analysed. The results are applied to compute a performance measure η of the efficiency of anycast over unicast and to the server placement problem. The server placement problem asks for the number of (replicated) servers m needed such that any user in the network is not more than j hops away from a server of the anycast group with a certain prescribed probability. Two types of shortest path trees are investigated: the regular k-ary tree and the irregular uniform recursive tree. Since these two types of trees indicate that the performance measure $\eta \approx 1 - a \log m$ where the real number a depends on the details of the tree, it suggests that for trees in real networks (as the Internet) a same logarithmic law applies. An order calculus on exponentially growing tree further supplies evidence for the conjecture that $\eta \approx 1 - a \log m$ for small m.

1 Introduction

IPv6 possesses a new address type, anycast, that is not supported in IPv4. The anycast address is syntactically identical to a unicast address. However, when a set of interfaces is specified by the same unicast address, that unicast address is called an anycast address. The advantage of anycast is that a group of interfaces at different location is treated as one single address. For example, often the information on servers is duplicated over several secondary servers at different locations for reasons of robustness and accessibility. Changes are only performed on the primary servers which are then copied onto all secondary servers to maintain consistency. If both the primary and all secondary servers have a same 'anycast' address, a query from some source towards that anycast address is routed towards the most nearby server of the group. Hence, instead of routing the packet to the root server (primary server) anycast is more efficient.

In this article, the distribution of the number of hops to the most nearby server of the anycast group is analyzed. The main focus is thus on quantifying the performance of the anycasting paradigm rather than on discussing the implementation or protocol-related impact of anycast. Suppose there are m (primary plus all secondary) servers and that these m servers are uniformly distributed over the Internet. The number of hops from the querying device A to the most nearby server is the minimum number of hops, denoted by $h_N(m)$, of the set of shortest paths from A to these m servers in a network

[*] Faculty of Electrical Engineering, Mathematics and Computer Science, P.O Box 5031, 2600 GA Delft, The Netherlands. Email: P.VanMieghem@ewi.tudelft.nl

with N nodes. In order to solve the problem, the shortest path tree rooted at node A, the querying device, needs to be investigated. We assume in the sequel that one of the m uniformly distributed servers can possibly coincide with the same router to which the querying machine A is attached. In that case, $h_N(m) = 0$.

Clearly, if $m = 1$, the problem reduces to the hopcount of the shortest path from A to one uniformly chosen node in the network and we have that

$$h_N(1) = h_N,$$

where h_N is the hopcount of the shortest path in a graph with N nodes. The other extreme for $m = N$ leads to

$$h_N(N) = 0$$

because all nodes in the network are servers. In between these extremes, there holds

$$h_N(m) \leq h_N(m-1)$$

since one additional anycast group member (server) can never increase the minimum hopcount to the root.

The hopcount to an anycast group is a stochastic problem. Even if the network graph is exactly known, an arbitrary node A views the network along a tree. Most often it is a shortest path tree where the precise optimization criterion is here irrelevant. Although the sequel emphasizes 'shortest path trees', the presented theory is equally valid for any type of tree. The node A's perception of the network is very likely different from the view of another node A'. Nevertheless, shortest path trees in a same graph possess to some extent related structural properties which allow us to treat the problem by considering certain types or classes of shortest path trees. Hence, instead of varying the arbitrary node A over all possible nodes in the graph and computing the shortest path tree at each different node, we vary the structure of the shortest path tree rooted at A over all possible shortest path trees of a certain type. Of course, the confinement of the analysis then lies in the type of tree that is investigated. In this article, we will only consider the regular k-ary tree and the irregular uniform recursive tree (URT). Earlier for multicast [17], we found that 'real' shortest path trees in Internet possess properties similar to these trees and that scaling laws observed in both these two types of trees also apply to the Internet.

The presented analysis allows us to address at least two different issues. First, for a same class of tree (or topologies), the efficiency of anycast over unicast defined in terms of a performance measure η

$$\eta = \frac{E\left[h_N(m)\right]}{E\left[h_N(1)\right]} \leq 1$$

is quantified. The performance measure η indicates how much hops (or link traversals or bandwidth consumption) can be saved, on average, by anycast. Alternatively, η also reflects the gain in end-to-end delay or how much faster than unicast, anycast finds the desired information. Second, the so-called *server placement problem* can be treated. More precisely, the question "How many servers m are needed to guarantee that any user request can access the information within k hops with probability $\Pr\left[h_N(m) > k\right] \leq \epsilon$, where ϵ is a certain level of stringency" can be answered. The server placement problem is expected to gain increased interest especially for real-time services where end-to-end QoS (e.g. delay) requirements are desirable. In the most general setting of this server placement problem, all nodes (routers) are assumed to be equally important in the sense that user requests are generated equally likely at any router in the network with N nodes. The validity of this assumption has been justified by Philips *et al.* [12] and later by Chalmers and Almeroth [4]. In this case of uniform user requests, the best strategy is to place servers also uniformly over the network.

We start presenting a general analysis, valid in any graph, in section 3. As an example, two types of shortest path trees are analyzed: the regular k-ary tree in section 4 and the irregular uniform or recursive tree in section 5. An order calculus of the performance measure η in exponentially growing graphs, that includes the graph of the Internet, is presented in section 6. Mathematical derivations are found in [20].

2 Related Work.

The server placement problem as defined above is a probabilistic analogon of the 'cache location problem' [9], [11], [2] also known as '(constrained) mirror placement' [7]. Most articles on the cache location problem assume that the underlying network topology $G(N, E)$, where N are the number of nodes and E the number of vertices, is known and propose an algorithm for the following graph theoretical problem: "Given $G(N, E)$ and m servers (caches), place these m servers at a subset of nodes of the graph G in order to optimize some criterion (and often subject to constraints, e.g. only at given locations servers can be placed [7])". The usual criterion is either a minimization of the round-trip-time or of the maximum distance from a querying node to the most nearby server (Min K-center algorithm) or it attempts to distribute the total server load as equal as possible over the m servers. Unfortunately, the algorithm to the above mentioned graph theoretical problem is NP-complete which naturally leads to the proposal of heuristics [7], [11] or to dynamic programming solutions [9]. Instead of formulating the cache location problem into a graph theoretical framework, sometimes a cache architecture or a strategy is proposed which is then evaluated by analysis or measurements in terms of latency, bandwidth usage or load and disk space (see e.g. [13]).

While most of the literature concentrates on *strategies* to place the m servers optimally, our work assumes uniform or random placement and targets to gain insight in how the performance measure η, the gain (in hops) of using m servers instead of 1, scales with m and N in different trees. In particular, our analysis suggests that, for large N and small m, the performance measure η obeys the law $\eta \approx 1 - a \log m$. Jamin *et al.* [7] showed quickly diminishing benefits of placing additional mirrors (servers), which is a less precise and not quantified formulation of the our claimed law $\eta \approx 1 - a \log m$. Some of their figures that plot performance measures based on round-trip-time measurements versus the number of mirrors m seem to decrease logarithmically in m.

The difficult and always debatable point in nearly all work concerns the assumption of the underlying network topology. The Internet topology is not a static graph but continuously changing over time. Usually random graphs (Waxman graphs [19]) or "Internet-like" graphs are simulated or measurements in (a particular part of) the Interent are presented. Although many measurements and analyses (for references see [16]) have been and are still being performed, relatively little insight in the topological properties of the Internet has been gained. For example, it would be desirable to have the Internet graph categorized as a member of some particular class of graphs. Perhaps the most cited paper is that of the Faloutsos *et al.* [5]. They show that the degree distribution of the Internet graph follows a power law. However, if one constructs the shortest path tree based on trace-routes to a small number m, that tree resembles a URT surprisingly well as shown below. That shortest path tree which is the union of IP-traces measured by the trace-route utility from a root to m other nodes is deemed relevant for the hopcount to an anycast group with relatively small m. Hence, the URT models the hopcount and the degree of the shortest path deduced from trace-routes accurate enough to deserve due analytic treatment as presented here. At last, together with the regular trees, the URT seems one of the very few stochastic trees that permits analytic modeling as presented here.

3 General Analysis.

Let us consider a particular shortest path tree T rooted at node A. Denote by $\left\{X_N^{(k)}\right\}$ the k-th level set of T or the set of nodes in the tree T at hopcount k from the root A in a graph with N nodes and by $X_N^{(k)}$ the number of elements in the set $\left\{X_N^{(k)}\right\}$. Then, we have $X_N^{(0)} = 1$ because the zeroth level can only contain the root node A itself. For all $k > 0$, holds that $0 \leq X_N^{(k)} \leq N - 1$ and that

$$\sum_{k=0}^{N-1} X_N^{(k)} = N \tag{1}$$

Another consequence of the definition is that, if $X_N^{(n)} = 0$ for some level $n < N - 1$, then all $X_N^{(j)} = 0$ for levels $j > n$. Clearly, in such a case, the longest possible shortest path in the tree has a hopcount of n. The level set

$$L_N = \left\{1, X_N^{(1)}, X_N^{(2)}, \ldots, X_N^{(N-1)}\right\}$$

of a tree T is defined as the set containing the number of nodes $X_N^{(k)}$ at each level k. An example of a tree organized per level is drawn in Figure 1.

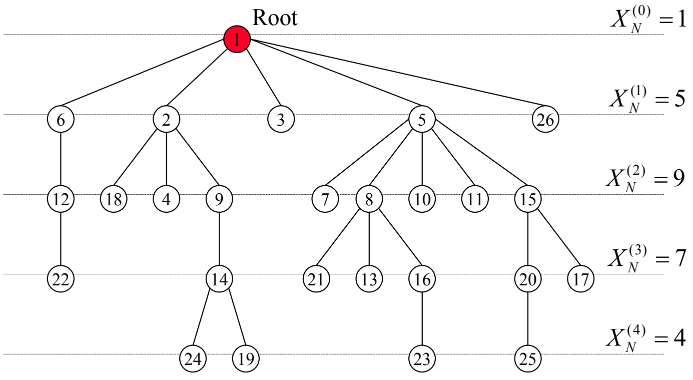

Figure 1: A tree with $N = 26$ organized per level $0 \leq k \leq 4$.

Further, suppose that the result of uniformly distributing m anycast group members over the graph leads to a number $m^{(k)}$ of those anycast group member nodes that are k hops away from the root. These $m^{(k)}$ distinct nodes all belong to the set $\left\{X_N^{(k)}\right\}$. Similarly as for $X_N^{(k)}$, some relations are immediate. First, $m^{(0)} = 0$ means that none of the m anycast group members coincides with the root node A or $m^{(0)} = 1$ means that one of them (and at most one) is attached to the same router A as the querying device. Also, for all $k > 0$, holds that $0 \leq m^{(k)} \leq X_N^{(k)}$ and that

$$\sum_{k=0}^{N-1} m^{(k)} = m \tag{2}$$

Given the tree T specified by the level set L_N and the anycast group members specified by the set $\left\{m^{(0)}, m^{(1)}, \ldots, m^{(N-1)}\right\}$, we will derive the lowest non-empty level $m^{(j)}$, which is equivalent to $h_N(m)$.

Let us denote by e_j the event that all first $j+1$ levels are not occupied by an anycast group member,

$$e_j = \left\{m^{(0)} = 0\right\} \cap \left\{m^{(1)} = 0\right\} \cap \ldots \cap \left\{m^{(j)} = 0\right\}$$

The probability distribution of the minimum hopcount, $\Pr\left[h_N(m) = j | L_N\right]$, is then equal to the probability of the event $e_{j-1} \cap \left\{m^{(j)} > 0\right\}$. Since the event $\left\{m^{(j)} > 0\right\} = \textit{not} \left\{m^{(j)} = 0\right\}$, using the conditional probability yields

$$\begin{aligned} \Pr\left[h_N(m) = j | L_N\right] &= \Pr\left[\left\{m^{(j)} > 0\right\} | e_{j-1}\right] \Pr\left[e_{j-1}\right] \\ &= \left(1 - \Pr\left[\left\{m^{(j)} = 0\right\} | e_{j-1}\right]\right) \Pr\left[e_{j-1}\right] \end{aligned} \tag{3}$$

Since $e_j = e_{j-1} \cap \left\{m^{(j)} = 0\right\}$, the probability of the event e_j can be decomposed as

$$\Pr\left[e_j\right] = \Pr\left[\left\{m^{(j)} = 0\right\} | e_{j-1}\right] \Pr\left[e_{j-1}\right] \tag{4}$$

The assumption that all m anycast group members are uniformly distributed enables us to compute $\Pr\left[\left\{m^{(j)} = 0\right\} | e_{j-1}\right]$ exactly. Indeed, by the uniform assumption, the probability equals the ratio of the favorable possibilities over the total possible. The total number of ways to distribute m items over $N - \sum_{k=0}^{j-1} X_N^{(k)}$ positions (the latter constraints follows from the condition e_{j-1}), equals $\binom{N-\sum_{k=0}^{j-1} X_N^{(k)}}{m}$. Likewise, the favorable number of ways to distribute m items over the remaining levels higher than j, leads to

$$\Pr\left[\left\{m^{(j)} = 0\right\} | e_{j-1}\right] = \frac{\binom{N-\sum_{k=0}^{j} X_N^{(k)}}{m}}{\binom{N-\sum_{k=0}^{j-1} X_N^{(k)}}{m}} \tag{5}$$

The recursion (4) needs an initialization, given by $\Pr\left[e_0\right] = \Pr\left[m^{(0)} = 0\right] = 1 - \frac{m}{N}$, which follows from $\Pr\left[m^{(0)} = 0\right] = \frac{\binom{N-1}{m}}{\binom{N}{m}}$ and equals $\Pr\left[\left\{m^{(0)} = 0\right\} | e_{-1}\right]$ (although the event e_{-1} is meaningless). Observe that $\Pr\left[m^{(0)} = 1\right] = \frac{m}{N}$ holds for any tree such that

$$\Pr\left[h_N(m) = 0\right] = \frac{m}{N}$$

By iteration of (4), we obtain

$$\Pr\left[e_j\right] = \prod_{s=0}^{j} \frac{\binom{N-\sum_{k=0}^{s} X_N^{(k)}}{m}}{\binom{N-\sum_{k=0}^{s-1} X_N^{(k)}}{m}} = \frac{\binom{N-\sum_{k=0}^{j} X_N^{(k)}}{m}}{\binom{N}{m}} \tag{6}$$

where the convention in summation is that $\sum_{k=a}^{b} f_k = 0$ if $a > b$. Finally, combining (3) with (5) and (6), we arrive at the general (conditional) probability for the minimum hopcount to the anycast group,

$$\Pr\left[h_N(m) = j | L_N\right] = \frac{\binom{N-\sum_{k=0}^{j-1} X_N^{(k)}}{m} - \binom{N-\sum_{k=0}^{j} X_N^{(k)}}{m}}{\binom{N}{m}} \tag{7}$$

Clearly, while $\Pr\left[h_N(0) = j|L_N\right] = 0$ since there is no path, we have for $m = 1$,

$$\Pr\left[h_N(1) = j|L_N\right] = \frac{X_N^{(j)}}{N}$$

It directly follows from (7) that

$$\Pr\left[h_N(m) \leq n|L_N\right] = 1 - \frac{\binom{N-\sum_{k=0}^{n} X_N^{(k)}}{m}}{\binom{N}{m}} \tag{8}$$

If $N - \sum_{k=0}^{n} X_N^{(k)} < m$ or, equivalently, $\sum_{k=n+1}^{N-1} X_N^{(k)} < m$, then (8) shows that $\Pr\left[h_N(m) > n|L_N\right] = 0$. The maximum possible hopcount of a shortest path to an anycast group strongly depends on the specifics of the shortest path tree or the level set L_N. Yet, a general result is worth mentioning,

Theorem 1 *For any graph holds that*

$$\Pr[h_N(m) > N - m] = 0$$

In words, the longest shortest path to an anycast group with m members can never possess more than $N - m$ hops.

Proof. This general Theorem 1 follows from the fact that the line topology is the tree with longest hopcount $(N-1)$ and only in case all m last positions (with respect to the source or root) are occupied by the m anycast group members, the maximum hopcount is $N - m$. ■

For the URT in Sec. 5, $\Pr[h_N(m) = N - m]$ is computed exactly in (17).

Corollary 2 *For any graph holds that*

$$\Pr[h_N(N-1) = 1] = \frac{1}{N}$$

Proof. This Corollary follows from Theorem 1 and the law of total probability. Alternatively, if there are $N - 1$ anycast members in a network with N nodes, the shortest path can only consist of 1 hop if none of the anycast members coincides with the root node. This chance is precisely $\frac{1}{N}$. ■

Since for any discrete random variable Y holds that $E[Y] = \sum_{k=0}^{\infty} \Pr[Y \geq k]$, it is immediate from (8) that

$$E\left[h_N(m)|L_N\right] = \frac{1}{\binom{N}{m}} \sum_{n=0}^{N-2} \binom{N - \sum_{k=0}^{n} X_N^{(k)}}{m} \tag{9}$$

from which we find,

$$E\left[h_N(1)|L_N\right] = \frac{1}{N} \sum_{k=1}^{N-1} k X_N^{(k)}$$

Thus, given L_N, a performance measure η for anycast over unicast can be quantified as

$$\eta = \frac{E\left[h_N(m)|L_N\right]}{E\left[h_N(1)|L_N\right]} \leq 1$$

Using the law of total probability, the distribution of the minimum hopcount to the anycast group is

$$\Pr\left[h_N(m)=j\right]=\sum_{\text{all } L_N}\Pr\left[h_N(m)=j|L_N\right]\Pr\left[L_N\right] \tag{10}$$

or,

$$\Pr\left[h_N(m)=j\right]=\sum_{\sum_{k=1}^{N-1}x_k=N-1}\frac{\binom{\sum_{k=j}^{N-1}x_k}{m}-\binom{\sum_{k=j+1}^{N-1}x_k}{m}}{\binom{N}{m}}\Pr\left[X_N^{(1)}=x_1,X_N^{(2)}=x_2,\ldots,X_N^{(N-1)}=x_{N-1}\right]$$

where $x_k\geq 0$ for all k. This expression explicitly shows the importance of the level structure L_N of the shortest path tree T. The level structure L_N entirely determines the *shape* of the tree T. Unfortunately, a general form for $\Pr\left[L_N\right]$ or $\Pr\left[h_N(m)=j\right]$ is difficult to obtain.

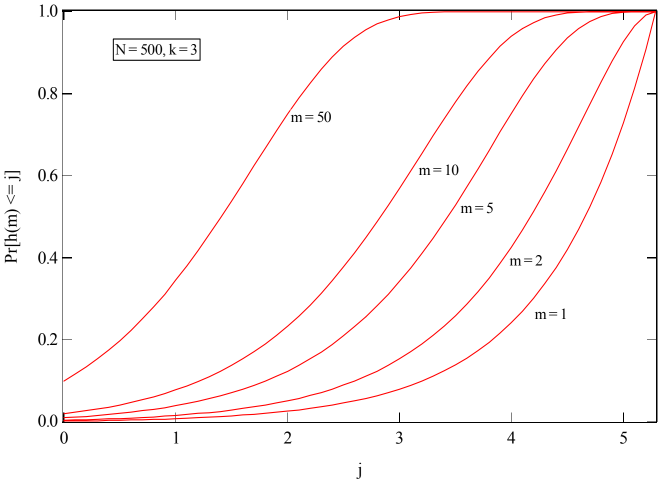

Figure 2: The distribution function of $h_{500}(m)$ versus the hops j for various sizes of the anycast group in a k-ary tree with $k=3$ and $N=500$

4 The k-ary tree.

For regular trees explicit expressions are possible because the summation in (10) simplifies considerably. For example, for the k-ary tree,

$$X_N^{(j)}=k^j$$

Provided the set L_N only contains these values of $X_N^{(j)}$ for each j, we have that $\Pr\left[L_N\right]=1$, else it is zero (because then L_N is not consistent with a k-ary tree). Summarizing, for the k-arry tree with $N=\frac{k^{D+1}-1}{k-1}$ and D levels, the distribution of the minimum hopcount to the anycast group is

$$\Pr\left[h_N(m)=j\right]=\frac{\binom{N-\frac{k^j-1}{k-1}}{m}-\binom{N-\frac{k^{j+1}-1}{k-1}}{m}}{\binom{N}{m}} \tag{11}$$

Extension of the integer k to real numbers in the formula (11) is expected to be of value as suggested by previous work [17] where we have computed the gain in the number of used links in multicast as compared to unicast. When a k-ary tree was used to fit corresponding Internet multicast measurements, we found that a remarkably accurate agreement was obtained for the value $k \approx 3.2$, which is about the average degree of the Internet graph. Hence, if we were to use the k-ary tree as model for the hopcount to an anycast group, we expect that $k \approx 3.2$ is the best value for Internet shortest path trees. However, we feel we ought to mention that the hopcount distribution of the shortest path between two arbitrary nodes is definitely not a k-ary tree, because $\Pr\left[h_N(1) = j\right]$ increases with the hopcount j which is in conflict with Internet trace-route measurements (see e.g. [8]).

Figure 2 displays $\Pr\left[h(m) \leq j\right]$ for a k-ary with outdegree $k = 3$ possessing a number of nodes equal to $N = 500$. This type of plot allows us to solve the 'server placement problem'. For example, assuming that the k-ary tree is a good model and the network consists of $N = 500$ nodes, Figure 2 shows that at least $m = 10$ servers are needed to assure that any user is not more than four hops separated from an arbitrary server of the anycast group with a probability of 93%. More precisely, the equation $\Pr[h_{500}(m) > 4] < 0.07$ is obeyed if $m \geq 10$.

Figure 3 gives an idea how the performance measure η decreases with the size of the anycast group in k-ary trees (all with outdegree $k = 3$), but with different size N. For small m, we observe that η decreases logarithmically in m, which is in agreement with the law $\eta \approx 1 - a \log m$.

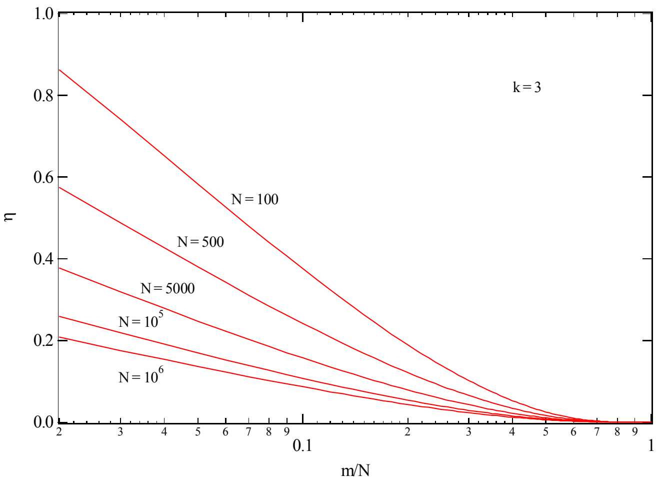

Figure 3: The performance measure η for several size of k-ary trees (with $k = 3$) as a function of the ratio of anycast nodes over the total number of nodes.

5 The uniform recursive tree (URT).

A uniform recursive tree (URT) of size N is a random tree that starts from the root A and where at each stage a new node is attached uniformly to one of the existing nodes until the total number of nodes is equal to N.

5.1 Motivation.

The interest in URTs is four-fold. First, as mentioned above, the URT is the prototype of an irregular tree. Second, we have demonstrated earlier [15] that the shortest path tree in a connected random graph $G_p(N)$ (and also in the Waxman graph [19]) with independent and uniformly or exponentially distributed link weights is a URT. As mentioned in [18],[15],[16] the law of the hopcount $h_N = h_N(1)$ of the shortest path between two arbitrary nodes is, for $0 \leq k \leq N-1$,

$$\Pr[h_N = k] = \frac{E\left[X_N^{(k)}\right]}{N} = \frac{(-1)^{N-1-k} S_N^{(k+1)}}{N!}, \tag{12}$$

where $S_N^{(k)}$ denote the Stirling numbers of the first kind [1, 24.1.3] with corresponding generating function

$$\varphi_N(z) = \sum_{k=0}^{N-1} \Pr[h_N = k]\, z^k = \frac{\Gamma(N+z)}{\Gamma(N+1)\Gamma(z+1)} \tag{13}$$

Third, from the hopcount distribution of paths in the Internet deduced from trace-route measurements, we found [8] that this distribution is reasonably well modeled by that of the URT given by (12). Fourth and last motivation, a more striking agreement with the URT is shown by the degree law [8]: for small multicast groups (m around about 50) from a root to uniformly spread users the measured multicast tree possesses a degree distribution close to the $\Pr[\deg = k] \sim 2^{-k}$ of the URT (for large N) [10]. (At the time of writing, this correspondence with the URT is further studied in order to understand the transition from an exponential degree law (small m) towards a power law degree law observed in Internet (see e.g. [5]) if m increases)

These arguments motivate that the URT is believed to provide a reasonable, first order estimate for the hopcount problem to an anycast and multicast group in Internet.

5.2 Recursion for $\Pr[h(m) = j]$

Usually, a combinatorial approach such as (10) is seldom successful for URTs while structural properties often lead to results. Previously, we have proved in [18] that,

Lemma 3 *Let $\{Y_N^{(k)}\}_{k,N\geq 0}$ and $\{Z_N^{(k)}\}_{k,N\geq 0}$ be two independent copies of the vector of level sets of two sequences of independent URTs. Then*

$$\{X_N^{(k)}\}_{k\geq 0} \overset{d}{=} \{Y_{N_1}^{(k-1)} + Z_{N-N_1}^{(k)}\}_{k\geq 0}, \tag{14}$$

where on the right-hand side the random variable N_1 is uniformly distributed over the set $\{1, 2, \ldots, N-1\}$.

This Lemma 3, applied to the anycast minimum hop problem, is illustrated in Figure 4.

Figure 4 shows that any URT can be separated into two subtrees T_1 and T_2 with size k and $N-k$ respectively. Moreover, Lemma 3 states that each subtree is independent of the other and again an URT. Consider now a specific separation of an URT T into $T_1 = t_1$ and $T_2 = t_2$, where the tree t_1 contains k nodes and i of the m anycast members and t_2 possesses $N-k$ nodes and the remaining $m-i$ anycast members. The event $\{h_T(m) = j\}$ equals the union of all possible

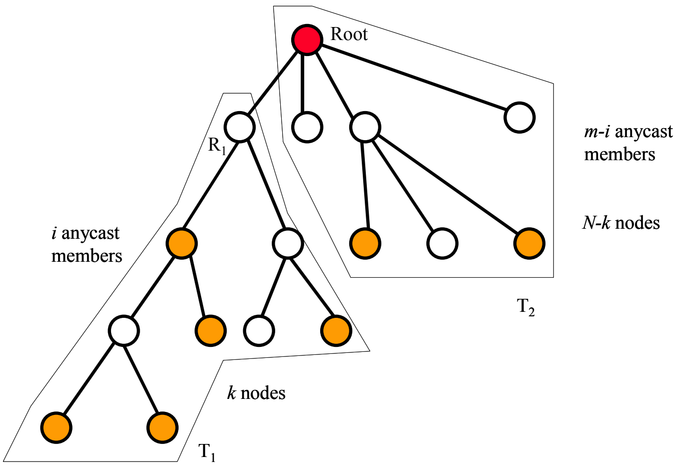

Figure 4: A uniform recursive tree consisting of two subtrees T_1 and T_2 with k and $N-k$ nodes respectively. The first clusters contains i anycast members while the cluster with $N-k$ nodes contains $m-i$ anycast members.

sizes $N_1 = k$ and subgroups $m_1 = i$ of the event $\{h_{t_1}(i) = j-1\} \cap \{h_{t_2}(m-i) \geq j\}$ and the event $\{h_{t_1}(i) > j-1\} \cap \{h_{t_2}(m-i) = j\}$,

$$\begin{aligned} \{h_T(m) = j\} &= \cup_k \cup_i \{\{h_{t_1}(i) = j-1\} \cap \{h_{t_2}(m-i) \geq j\}\} \\ &\quad \cup \{\{h_{t_1}(i) > j-1\} \cap \{h_{t_2}(m-i) = j\}\} \end{aligned}$$

Because $h_N(0)$ is meaningless, the relation must be modified for the case $i = 0$ to

$$\{h_T(m) = j\} = \{h_{t_2}(m) = j\}$$

and for the case $i = m$ to

$$\{h_T(m) = j\} = \{h_{t_1}(m) = j-1\}$$

This decomposition holds for any URT T_1 and T_2, not only for the specific ones t_1 and t_2. The transition towards probabilities becomes

$$\begin{aligned} \Pr[h_T(m) = j] &= \sum_{\text{all } t_1, t_2, k, i} (\Pr[h_{t_1}(i) = j-1] \Pr[h_{t_2}(m-i) \geq j] \\ &\quad + \Pr[h_{t_1}(i) \geq j-1] \Pr[h_{t_2}(m-i) = j]) \Pr[T_1 = t_1, T_2 = t_2, N_1 = k, m_1 = i] \end{aligned}$$

Since T_1 and T_2 and also m_1 are independent given N_1, the last probability l simplifies to

$$\begin{aligned} l &= \Pr[T_1 = t_1, T_2 = t_2, N_1 = k, m_1 = i] \\ &= \Pr[T_1 = t_1 | N_1 = k] \Pr[T_2 = t_2 | N_1 = k] \Pr[m_1 = i | N_1 = k] \Pr[N_1 = k] \end{aligned}$$

Lemma 3 states that N_1 is uniformly distributed over the set with $N-1$ nodes such that $\Pr[N_1 = k] = \frac{1}{N-1}$. The fact that i out of the m anycast members, uniformly chosen out of N nodes, belong to the recursive subtree T_1 implies that $m-i$ remaining anycast members belong to T_2. Hence, analogous to a combinatorial problem outlined by Feller [6, pp. 43] that lead to the hypergeometric distribution, we have

$$\Pr\left[m_1 = i | N_1 = k\right] = \frac{\binom{k}{i}\binom{N-k}{m-i}}{\binom{N}{m}}$$

because all favorable combinations are those $\binom{k}{i}$ to distribute i anycast members in T_1 with size k multiplied by all favorable $\binom{N-k}{m-i}$ to distribute the remaining $m-i$ in T_2 containing $N-k$ nodes. The total way to distribute m anycast members over N nodes is $\binom{N}{m}$. At last, we remark that the hopcount of the shortest path to m anycast members in a recursive tree (or random graph) only depends on its size. This means that the sum over all t_1 of $\Pr\left[T_1 = t_1 | N_1 = k\right]$, which equals 1, disappears and likewise also the sum over all t_2. Combining the above leads to

$$\begin{aligned}
\Pr\left[h_N(m) = j\right] \quad = \quad & \sum_{k=1}^{N-1}\sum_{i=1}^{m-1}(\Pr\left[h_k(i) = j-1\right]\Pr\left[h_{N-k}(m-i) \geq j\right] \\
& + \Pr\left[h_k(i) > j-1\right]\Pr\left[h_{N-k}(m-i) = j\right])\frac{\binom{k}{i}\binom{N-k}{m-i}}{(N-1)\binom{N}{m}} + \\
& \sum_{k=1}^{N-1}\Pr\left[h_{N-k}(m) = j\right]\frac{\binom{N-k}{m}}{(N-1)\binom{N}{m}} + \Pr\left[h_k(m) = j-1\right]\frac{\binom{k}{m}}{(N-1)\binom{N}{m}}
\end{aligned}$$

By substitution of $k' = N-k$ and $m' = m-i$, this expression simplifies to a recursion for $\Pr\left[h_N(m) = j\right]$ in the URT,

$$\begin{aligned}
\Pr\left[h_N(m) = j\right] \quad = \quad & \sum_{k=1}^{N-1}\sum_{i=1}^{m-1}\left(\Pr\left[h_k(i) = j-1\right] + \Pr\left[h_k(i) = j\right]\right)\frac{\binom{k}{i}\binom{N-k}{m-i}}{(N-1)\binom{N}{m}}\sum_{q=j}^{N-k-1}\Pr\left[h_{N-k}(m-i) = q\right] \\
& + \sum_{k=1}^{N-1}\left(\Pr\left[h_k(m) = j\right] + \Pr\left[h_k(m) = j-1\right]\right)\frac{\binom{k}{m}}{(N-1)\binom{N}{m}} \qquad (15)
\end{aligned}$$

For $m = 1$, the recursion (15) reduces to

$$\Pr\left[h_N = j\right] = \frac{1}{N-1}\sum_{k=1}^{N-1}\left(\Pr[h_k = j] + \Pr\left[h_k = j-1\right]\right)\frac{k}{N}$$

Multiplying both sides by z^j, summing over all j leads to the recursion for the generating function (13)

$$(N+1)\varphi_{N+1}(z) = (z+N)\varphi_N(z)$$

from which (13) and (12) follows.

For $m = 2$, the recursion (15) is solved in [20] resulting in

$$\begin{aligned}
\Pr\left[h_N(2) = j\right] \quad = \quad & \frac{2(-1)^{N-1-j}}{N!}S_N^{(j+1)} + \frac{2(-1)^{N-1}}{N!(N-1)}\sum_{k=1}^{j}(-1)^{j+k+1}\binom{2k-1}{k}S_N^{(k+j+1)} \\
& + \frac{2(-1)^{N-1}}{N!(N-1)}\sum_{k=0}^{j-1}\left[\binom{j+k+1}{j} - \binom{2k+1}{k}\right](-1)^{k+j+1}S_N^{k+j+1} \qquad (16)
\end{aligned}$$

Since there are $(N-1)!$ possible URTs [18] and there is only one line tree with $N-1$ hops where each node has precisely one child node, the probability to have precisely $N-1$ hops from the root is

$\frac{1}{(N-1)!}$ (which also is $\Pr[h_N = N-1]$ given in (12)). The longest possible hopcount from a root to m anycast members occurs in the line tree where all m anycast members occupy the last m positions. Hence, the probability for the longest possible hopcount equals

$$\Pr[h_N(m) = N - m] = \frac{m!}{(N-1)!\binom{N}{m}} \tag{17}$$

because there are $m!$ possible ways to distribute the m anycast members at the m last positions in the line tree while there are $\binom{N}{m}$ possibilities to distribute m anycast members at arbitrary places in the line tree.

In [20], we have studied the recursion (15) in detail. For other values of m than those listed above, we did not succeed to solve the recursion (15) exactly but we have presented tight approximations. The recursion (15) is solved numerically for $N = 20$. The result is shown in Figure 5 which demonstrates that $\Pr[h_N(m) > N - m] = 0$ or, the path with the longest hopcount to an anycast group of m members consists of $N - m$ links. Figure 5 allows us to solve the 'server placement problem'. For example, consider the scenario in which a network operator announces that any user request will reach a server of the anycast group in no more than $j = 4$ hops in 99.9% of the cases. Assuming his network has $N = 20$ routers and the shortest path tree is an URT, the network operator has to compute the number of anycast servers m he has to place uniformly spread over the $N = 20$ routers by solving $\Pr[h_{20}(m) > 4] < 10^{-3}$. Figure 5 shows that the crosspoint of the line $j = 4$ and the line $\Pr[h_{20}(m) = 4] = 10^{-3}$ is the curve for $m = 7$. Since the curves for $m \geq 7$ are exponentially decreasing, $\Pr[h_{20}(m) > 4]$ is safely[1] approximated by $\Pr[h_{20}(m) = 4]$ which leads to the placing of $m = 7$ servers. When following the line $j = 4$, we also observe that the curves for $m = 5, 6, 7, 8$ lie near to that of $m = 7$. This means that placing a server more does not considerably change the situation. It is a manifestation of the law $\eta \approx 1 - a \log m$ which tells us that by placing m servers the

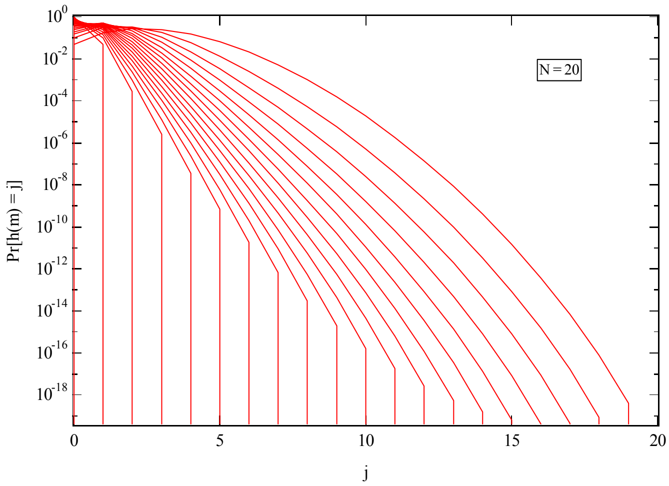

Figure 5: The pdf of $h_N(m)$ in a recursive tree with $N = 20$ nodes for all possible m. Observe that $\Pr[h_N(m) > N - m] = 0$. This relation connects the various curves to the value for m.

[1] If more precision is desired, $\Pr[h_{20}(m) > j]$ curves are easily computed from Figure 5.

gain measured in hops with respect to the single server case is slowly, more precisely logarithmically, increasing. The performance measure η for the URT is drawn for several sizes N in Figure 6.

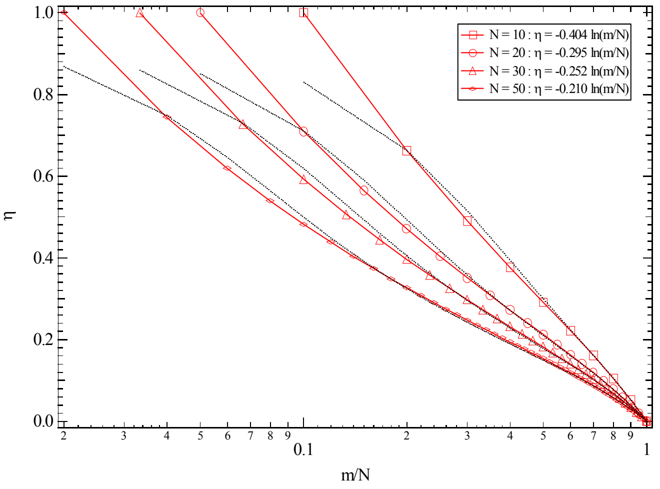

Figure 6: The performance measure η for several sizes N of recursive trees as a function of the ratio m/N. The dotted line present the approximation for η where $E[h_N(m)]$ is computed as $E[\widetilde{h}_N(m)]$ but $E[h_N(1)]$ by its exact value.

5.3 Approximate Analysis.

Since the general solution (10) is in many cases difficult to compute as shown for the URT in section 5, we consider a simplified version of the above problem where each node in the tree has equal probability $p = \frac{m}{N}$ to be a server. Instead of having precisely m servers, the simplified version considers on average m servers and the probability that there are precisely m servers is $\binom{N}{m}p^m(1-p)^{N-m}$. In the simplified version, the associated equations to (5) and (4) are

$$\begin{aligned}
\Pr\left[\left\{m^{(j)}=0\right\}|e_{j-1}\right] &= \Pr\left[\left\{m^{(j)}=0\right\}\right]=(1-p)^{X_N^{(j)}} \\
\Pr\left[e_j\right] &= \prod_{l=0}^{j}\Pr\left[\left\{m^{(j)}=0\right\}\right]=(1-p)^{\sum_{l=0}^{j-1}X_N^{(l)}}
\end{aligned}$$

which implies that the probability that there are no servers in the tree is $(1-p)^N$. Since in that case, the hopcount is meaningless, we consider the conditional probability (3) of the hopcount given the level set contains at least one server (which is denoted by $\widetilde{h}_N(m)$) is

$$\Pr\left[\widetilde{h}_N(m)=j|L_N\right]=\frac{\left(1-(1-p)^{X_N^{(j)}}\right)(1-p)^{\sum_{l=0}^{j-1}X_N^{(l)}}}{1-(1-p)^N}$$

Thus,

$$\Pr\left[\widetilde{h}_N(m)\leq n|L_N\right]=\frac{1-(1-p)^{\sum_{l=0}^{n}X_N^{(l)}}}{1-(1-p)^N}$$

Finally, to avoid the knowledge of the entire level set L_N, we use $E\left[X_N^{(l)}\right] = N\Pr\left[h_N\left(1\right) = l\right]$ as the best estimate for each $X_N^{(l)}$ and obtain the approximate formula

$$\Pr\left[\widetilde{h}_N(m) = j\right] = \frac{\left(1-(1-p)^{E\left[X_N^{(j)}\right]}\right)(1-p)^{\sum_{l=0}^{j-1} E\left[X_N^{(l)}\right]}}{1-(1-p)^N} \tag{18}$$

In dotted lines in Figure 6, we have added the approximate result for the URT where $E\left[h_N(m)\right]$ is computed based on (18), but where $E[h_N(1)]$ is computed exact. For $m = 1$, the approximate analysis (18) is not wel suited: Figure 6 illustrates this deviation in the fact that $\eta_{appr}(1) = E\left[\widetilde{h}_N(1)\right]/E\left[h_N(1)\right] < 1$. For higher values of m we observe a fairly good correspondence. We found that the probability (18) reasonably approximates the exact result plotted on a linear scale. Only the tail behavior (on log-scale) and the case for $m = 1$ deviate significantly. In summary for the URT, the approximation (18) for $\Pr\left[h_N(m) = j\right]$ is much faster to compute than the exact recursion and it seems appropriate for the computation of η for $m > 1$. However, it is less adequate to solve the server placement problem that requires the tail values $\Pr\left[h_N(m) > j\right]$.

6 The performance measure η in the exponentially growing trees.

In this section, we present an order estimate that supports our claimed law $\eta \approx 1 - a\log m$ for a much larger class of trees, namely the class of exponentially growing trees to which both the k-ary tree and the URT belong. Also most trees in the Internet are exponentially growing trees. A tree is said to grow exponentially in the number of nodes N with degree κ if $\lim_{j\to\infty}\left(X_N^{(j)}\right)^{1/j} = \kappa$ or, equivalently, $X_N^{(j)} \sim \kappa^j$, for large j. As explained earlier [17], the fundamental problem with this definition is that it only holds for infinite graphs $N = \infty$. For real (finite) graphs, there must exists some level $j = l$ for which the sequence $X_N^{(l+1)}, X_N^{(l+2)}, \cdots, X_N^{(N-1)}$ ceases to grow because $\sum_{j=0}^{N-1} X_N^{(j)} = N < \infty$. This boundary effect complicates the definition of exponential growth in finite graphs. The second complication is that even in the finite set $X_N^{(0)}, X_N^{(1)}, \cdots, X_N^{(l)}$ not necessary all $X_N^{(j)}$ with $0 \leq j \leq l$ need to obey $X_N^{(j)} \sim \kappa^j$, but 'enough' should. Without the limit concept, we cannot specify the precise conditions of exponential growth in a finite shortest path tree. If we assume in finite graphs that $X_N^{(j)} \sim \kappa^j$ for $j \leq l$, then $\sum_{j=0}^{l} X_N^{(j)} = \alpha N$ with $0 < \alpha < 1$. Indeed, for $\kappa > 1$, the highest hopcount level l possesses by far the most nodes since $\frac{\kappa^{l+1}-1}{\kappa-1} \approx \kappa^l$ which cannot be larger than a fraction αN of the total number of nodes.

We now present an order calculus to estimate η for exponentially growing trees based on relation (9). Let us denote

$$y = \frac{\binom{N-x}{m}}{\binom{N}{m}} = \prod_{j=0}^{m-1}\left(1 - \frac{x}{N-j}\right)$$

For large N and fixed m,

$$y = \exp\left(-\frac{xm}{N}\right)\left(1 + o\left(1\right)\right)$$

In case the tree is exponentially growing for $j \leq l$ as $X_N^{(j)} = \beta_j\kappa^j$ with β_j some slowly varying sequence, only very few levels Δl (bounded by a fixed number) around l holds that $\sum_{k=0}^{n} X_N^{(k)} = O(N)$ where

$n \in [l - \Delta l, l]$, while for all $j > l$, we have $\sum_{k=0}^{n} X_N^{(k)} = \mu_n N$ with some sequence $\mu_n < \mu_{n+1} < \mu_{\max n} = 1$. Applied to (9) where $x = \sum_{k=0}^{n} X_N^{(k)} < N$,

$$E\left[h_N(m)|L_N\right] \approx (1 + o\,(1)) \sum_{n=0}^{l} \exp\left(-\frac{m}{N}\beta_n\kappa^n\right) + \sum_{n=l+1}^{N-2} \frac{\binom{(1-\mu_n)N}{m}}{\binom{N}{m}}$$

If there are only a few levels more than l, the last series is much smaller than 1 and can be omitted. Since the slowly varying sequence β_n is unknown, we approximate $\beta_n = \beta$ and

$$\begin{aligned}
\sum_{n=0}^{l} \exp\left(-\frac{m}{N}\beta_n\kappa^n\right) &\approx \int_0^l \exp\left(-\frac{m}{N}\beta\kappa^n\right) dn = \frac{1}{\log\kappa}\int_{\frac{m}{N}\beta}^{\frac{m}{N}\beta\kappa^l} \frac{e^{-u}}{u} du \\
&\approx \frac{1}{\log\kappa}\int_{\frac{m}{N}\beta}^{\infty} \frac{e^{-u}}{u} du - \frac{e^{-m}}{m\log\kappa} \\
&= \frac{1}{\log\kappa}\left(-\gamma - \frac{e^{-m}}{m} - \log\frac{m}{N}\beta + O\left(\frac{m}{N}\right)\right)
\end{aligned}$$

where in the last step a series [1, 5.1.11] for the exponential integral is used. Thus,

$$\begin{aligned}
\eta &\approx (1 + o\,(1)) \frac{\left(1 + \frac{-\gamma - \frac{e^{-m}}{m} - \log m - \log\beta}{\log N} + O\left(\frac{m}{N}\right)\right)}{\left(1 - \frac{\gamma + e^{-1} + \log\beta}{\log N} + O\left(\frac{1}{N}\right)\right)} \\
&= (1 + o\,(1))\left(1 - \frac{\log m}{\log N} - \frac{e^{-1} - \frac{e^{-m}}{m}}{\log N} + O\left(\frac{1}{\log^2 N}\right)\right)
\end{aligned}$$

Since by definition $\eta = 1$ for $m = 1$, we finally arrive at

$$\eta \approx 1 - \frac{\log m}{\log N} - \frac{e^{-1} - \frac{e^{-m}}{m}}{\log N} + O\left(\frac{1}{\log^2 N}\right)$$

which supplies evidence for the conjecture $\eta \approx 1 - a\log m$ that exponentially growing graphs (such as the Internet) possess a performance measure η that logarithmically decreases in m, which is rather slow.

Measurement data in Internet seem to support this $\log m$-scaling law. Apart from the correspondence with Figures in the work of [7], Figure 6 in Krishnan *et al.* [9] shows that the relative measured traffic flow reduction decreases logarithmically in the number of caches m.

7 Discussion and conclusions.

The probability density function of the hopcount to the nearest member of an anycast group has been analysed. Two types of trees, the k-ary tree and the URT have been computed. The exact and simple results for the k-ary tree enable the computation of $\Pr[h_N(m) = j]$ for any reasonable size N. The computation of $\Pr[h_N(m) = j]$ for the URT is, unfortunately, much more complex.

Our results may shed some quantitative insight in the performance of anycast. At least two applications, the efficiency of anycast over unicast and the server placement problem have been targeted. In both types of trees, the performance measure $\eta = \frac{E[h_N(m)]}{E[h_N(1)]}$ decreases proportional with $\log m$.

More generally, an order calculus on exponentially growing graphs such as the Internet supports the conjecture that $\eta \approx 1 - a \log m$ for small m. The law $\eta \approx 1 - a \log m$ means that, if an anycast group consists of $m > 1$ servers, adding a few more servers does not significantly improve the performance measure expressed in the number of hops. Other studies seem to indicate that this anycast law also holds for performance measures such as delay and traffic reduction.

Computations of $\Pr[h_N(m) > j] < \epsilon$ for given strigency ϵ and hop j, allow to determine the minimum number m of servers. The solution of this server placement problem may be regarded as an instance of the general quality of service (QoS) portofolio of an network operator. When the number of servers for a major application offered by the service provider are properly computed, the service provider may announce levels ϵ of QoS (e.g. via $\Pr[h_N(m) > j] < \epsilon$) and accordingly price the use of the application. More potential applications are envisaged as anycast is still in an embryonic state.

Finally, the focus on the URT as reasonable model for realistic trees can be motivated. Only very few classes of trees, including the URT, make analytic computations possible. Apart from this computational argument, it [8] was found that measured trees in the Internet are fairly well modeled by a URT. In ad-hoc networks with uniformly distributed mobile users, the URT may also model the tree from one mobile user to the others. Recent work on peer-to-peer networks indicates that some of these networks (such as Gnutella) possess properties well described by random graphs. Present results on the URT may also have value in the context of peer-to-peer networking. For example, if the RIPE measurement boxes [8] are considered as peers, the distribution of the peer-to-peer delay seems roughly exponentially distributed. In the complete graph (each peer has a connection to any other) with exponentially distributed weights, the shortest path tree is a URT [15, 16].

References

[1] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions*, Dover, 1968.

[2] P. Barford, J.-Y. Cai and J. Gast, "Cache Placement Methods Based on Client Demand Clustering", IEEE INFOCOM 2002.

[3] B. Bollobas, *Random Graphs*, Cambridge University Press, second edition, 2001.

[4] R. C. Chalmers and K. C. Almeroth, " On the Topology of Multicast Trees, IEEE/ACM Trans. on Networking, vol. 11, No. 1, pp. 153-165, 2003.

[5] M. Faloutsos, P. Faloutsos and C. Faloutsos , "On power-law relationships of the Internet Topology ", Proceedings of ACM SIGCOMM'99, Cambridge, Massachusetts, pp. 251-262, 1999.

[6] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 1, John Wiley & Sons, 1970.

[7] S. Jamin, C. Jin, A. R. Kurc, D. Raz and Y. Shavitt, "Constrained Mirror Placement on the Internet", IEEE INFOCOM 2001.

[8] M. Janic, F. Kuipers, X. Zhou and P. Van Mieghem, "Implications for QoS provisioning based on traceroute measurements", Proceedings of 3nd International Workshop on Quality of Future

Internet Services, QofIS2002 [edited by B. Stiller et al. in Springer Verlag LNCS 2511], Zurich, Switzerland, October 16-18, pp. 3-14.

[9] P. Krishnan, D. Raz and Y. Shavitt, ”The Cache Location Problem”, IEEE/ACM Transactions on Networking, vol. 8, No. 5, pp. 586-582, 2000.

[10] H. S. Na and A. Rapoport, ”Distribution of Nodes of a Tree by Degree”, Mathematical Biosciences, vol. 6, pp. 313-329, 1970.

[11] L. Qiu, V. N. Padmanabhan and G. M. Voelker, ”On the Placement of Web Server Replicas”, IEEE INFOCOM2001.

[12] G. Phillips, S. Shenker and H. Tangmunarunkit, ”Scaling of Multicast Trees: Comments on the Chuang-Sirbu scaling law.”, Proc. ACM Sigcomm 1999.

[13] P. Rodriguez, C. Spanner and E. W. Biersack, ”Analysis of Web Caching Architectures: Hierarchical and Distributed Caching”, IEEE/ACM Transactions on Networking, vol. 9, No. 4, pp. 404-418, 2001.

[14] E.T. Titchmarsh, *The Theory of Functions*, Oxford University Press, 1964.

[15] R. van der Hofstad, G. Hooghiemstra and P. Van Mieghem, "First Passage Percolation on the Random Graph", Probability in the Engineering and Informational Sciences (PEIS), vol. 15, pp. 225-237, 2001.

[16] P. Van Mieghem, G. Hooghiemstra and R. van der Hofstad, ”A Scaling Law for the Hopcount in Internet”, report 2000125 (http://www.nas.ewi.tudelft.nl/people/piet/teleconference.html)

[17] P. Van Mieghem, G. Hooghiemstra and R. van der Hofstad, ”On the efficiency of multicast”, IEEE/ACM Transactions on Networking, vol. 9, No. 6, pp. 719-732. 2001.

[18] R. van der Hofstad, G. Hooghiemstra and P. Van Mieghem , ”On the covariance of the level sizes in random recursive trees”, to be published in Random Structures and Algorithms, 2002.

[19] P. Van Mieghem, ”Paths in the simple random graph and the Waxman Graph”, Probability in the Engineering and Informational Sciences (PEIS), vol. 15, pp. 535-555, 2001.

[20] P. Van Mieghem, "The Probability Distribution of the Hopcount to an Anycast Group ", report 2003605 (http://www.nas.ewi.tudelft.nl/people/piet/teleconference.html)