# Stability of a Multicast Tree

P. Van Mieghem and M. Janic

***Abstract*— Most of the currently deployed multicast protocols (e.g. DVMRP, PIM, MOSPF) build one shortest path multicast tree per sender, the tree being rooted at the sender's subnetwork. This paper examines the stability of such a tree, specifically, "How does the number of links change as the number of multicast users in a group changes?" Two modelling assumptions are made. We assume that (a) packets are delivered along the shortest path tree (which is a realistic assumption as indicated above) and that (b) the $m$ multicast group member nodes are chosen uniformly out of the total number of nodes $N$. The probability density function for the number of changed edges $\Delta_N(m)$ when one multicast user joins or leaves the group is studied. For random graphs of the class $G_p(N)$ with $N$ nodes, link density $p$ and with uniformly (or exponentially) distributed link weights, the probability density function $\Pr[\Delta_N(m) = k]$ is proved to tend to a Poisson distribution for large $N$. The proof of this theorem enables a generalization to an arbitrary topology. Simulations, mainly conducted to quantify the validity of the asymptotic regime, reveal that the Poisson law seems more widely valid than just in the asymptotic regime where $N \to \infty$. In addition, the effect of the link weight distribution on the stability of the multicast tree is investigated. Finally, the stability of a Steiner tree connecting $m$ multicast users is compared to the shortest path tree via simulations.**

## I. Introduction

The demand for multimedia which combines audio, video and data streams over a network, is rapidly increasing. Some of the more popular real-time interactive applications are desktop video/audio conferencing, shared white boards, software updates, tele-classing, interactive games and animated simulations. Even when data compression is used, multimedia applications require in general a considerable amount of bandwidth. IP multicast is regarded as a promising network service for group multimedia applications.

One of the major points of interest in IP multicasting is the efficient multicast routing. The goal of multicast routing is to find a loopless (acyclic) tree of links that connects all the members of the multicast group. Multicast packets are then forwarded along this tree from the sender to all multicast group members. Several approaches have been adopted for determining the multicast spanning tree. The simplest way to build a spanning tree is to add one participant at a time, using a shortest path algorithm (e.g. Dijkstra's [16]). New participants are connected along a shortest path to the nearest node in the existing spanning tree. Improved versions of this source-specific tree principle are implemented in DVMRP [4], MOSPF [5] and PIM-Dense Mode [7]. While the shortest path tree between the source node and each destination node guarantees that multicast packets will be delivered as fast as possible, it does not necessarily result in a tree that economizes on network resources. The second approach is to construct a single tree to distribute the traffic from all senders in the group, regardless of the sender's location, and to minimize the total weight of the tree. Hence, it optimizes the use of network resources. The problem of finding a minimum weight tree that spans all multicast users is known as the Steiner tree problem [20]. However, due to the high computational effort and the less stable dynamic behavior, a Steiner tree is not implemented in multicast routing protocols. Instead, a group-shared tree used in protocols like CBT [3] and PIM-SM [6] is based on defining a center node (or rendez-vous point) in the routing tree. Finally, a third approach is the recently proposed explicit multicast [21] for small multicast groups. Explicit multicast essentially forwards packets, with in the header all the (unicast) IP addresses of the multicast group members, along the shortest paths. At branches of the shortest path tree, the packet is multiplicated on the outgoing links and the list of IP addresses in the header is splitted properly.

Apart from the dynamics of topology updates, IP multicast offers the possibility of joining and leaving a group at any time. This activity requires the multicast tree to be dynamically updated (e.g. branches without multicast members must be discarded.) These changes in the shortest path tree imply that the forwarding of IP multicast packets may change dramatically, resulting in undesirable transient routing effects. The goal of this article is to investigate and quantify multicast stability, in particular, to determine the probability density function for the number of branches that change if one user joins or leaves the group. In addition, we quantify the common belief (see e.g. the book of Huitema [11]) that Steiner trees are more instable than shortest path trees.

The paper is organized as follows. In Section II, we present the theoretical and analytic results, while Section III presents simulation results for both the shortest path (SPT) and the Steiner tree (MST). Finally, we conclude in Section IV.

## II. On the efficiency of multicast.

In this section, previous theoretical results are first briefly reviewed. For the proofs, discussion and the nice agreement with Internet measurements, we refer to [14]. In the second part, the stability of the multicast tree is defined and basic theoretical results are deduced.

### *A. Theory: A review.*

We focus on the efficiency or gain of multicast in terms of network resource consumption compared to unicast. Specifically, we concentrate on a one-to-many communication, where a source distributes messages (packets) to $m$ different, uniformly distributed destinations along the shortest path. In

Delft University of Technology, Faculty of Information Technology and Systems, P.O Box 5031, 2600 GA Delft, The Netherlands. Email: {P.VanMieghem, M.Janic}@its.tudelft.nl

unicast, these messages are sent $m$ times from the source to each destination. Hence, unicast uses on average $f_N(m) = mE[H_N]$ link-traversals or hops, where $E[H_N]$ is the average number of hops of a message to a uniform location in the graph under consideration containing $N$ nodes. One of the main properties of multicast is that it economizes on the number of link-traversals. If we define for multicast $g_N(m)$ to be the average number of hops in the shortest path tree rooted at a source to $m$ randomly chosen distinct destinations, then, of course, $g_N(m) \leq f_N(m)$. For the extreme sizes of the multicast group, we have simple expressions: $g_N(1) = f_N(1) = E[H_N]$ while $g_N(N-1) = N-1$ reflecting the number of links in a (complete) spanning tree. Below we merely list the more important results obtained previously [14].

*Theorem 1:* For any connected graph with $N$ nodes,

$$m \leq g_N(m) \leq \frac{Nm}{m+1}. \tag{1}$$

*Theorem 2:* For any connected graph with $N$ nodes, the map $m \mapsto g_N(m)$ is concave and the map $m \mapsto \frac{g_N(m)}{f_N(m)}$ is decreasing.

Next, we need the following definition. Let $X_i$ be the number of joint hops that *all* $i$ uniformly chosen and different group members have in common. Then we have the identity:

*Theorem 3:* For any connected graph with $N$ nodes,

$$g_N(m) = \sum_{i=1}^{m} \binom{m}{i} (-1)^{i-1} E[X_i]. \tag{2}$$

*Corollary 4:* For any connected graph, the multicast efficiency $g_N(m)$ is bounded by

$$\frac{f_N(m)}{g_N(m)} \leq E[H_N], \tag{3}$$

where $E[H_N]$ is the average number of hops in unicast.

This Corollary 4 means that the maximum savings in resources an operator can gain by using multicast (over unicast) never exceeds $E[H_N]$, which is roughly about 15 in Internet today.

Remark the generality of these theorems: they hold for any graph, including the graph of the Internet. The remaining two Theorems only apply to a specific type of graph. The class of the random graphs $G_p(N)$ with $N$ nodes, with independently chosen links with probability $p$ (studied in detail by Bollobas [2] and later by Janson *et al.* [10]) and with uniformly on [0,1] (or exponentially) distributed link metrics $w$ is further referred to as *RGU*.

*Theorem 5:* For the class RGU,

$$g_N(m) = mN \left( \frac{\psi(N) - \psi(m)}{N-m} \right) - 1, \tag{4}$$

where $\psi(x)$ is the digamma function. For large $N$, we have the accurate asymptotic,

$$g_N(m) \sim \frac{mN}{N-m} \log\left(\frac{N}{m}\right) - \frac{1}{2} \tag{5}$$

We have also considered the regular $k$-ary tree of depth[1] $D$ with the source at the root of the tree and $m$ receivers at randomly chosen nodes. In a $k$-ary tree the total number of nodes satisfies

$$N = 1 + k + k^2 + \ldots + k^D = \frac{k^{D+1}-1}{k-1}, \tag{6}$$

so that $N \sim k^D$.

*Theorem 6:* For the $k$-ary tree,

$$g_{N,k}(m) = N - 1 - \sum_{j=0}^{D-1} k^{D-j} \frac{\binom{N-1-\frac{k^{j+1}-1}{k-1}}{m}}{\binom{N-1}{m}}. \tag{7}$$

Figure 1 illustrates the behavior of $g_m(m)$ for the random graph and several $k$-values of the $k$-ary tree together with the extreme values given by (1). Finally, when fitting Internet mea-

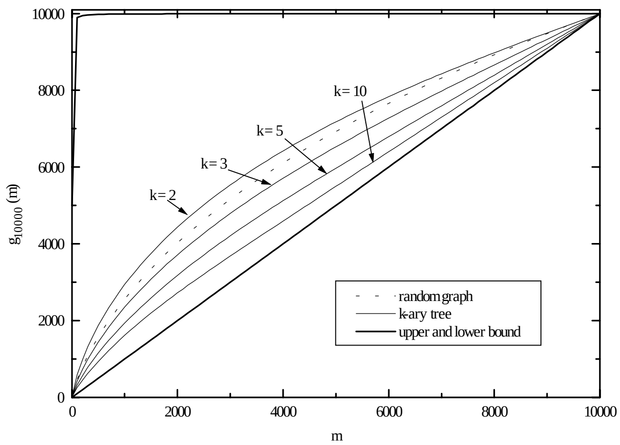


Fig. 1. The multicast efficiency $g_N(m)$ versus group member size $m$ in a network with $N = 10^4$ nodes and various topologies

surements with (7) and treating $1 \leq k \leq N-1$ as a real positive number (instead of an integer), an accurate fit has been obtained with $k_{Internet} = 3.2$. Since the corresponding $k$-value for the class RGU is $k_{RGU} = e = 2.718\ldots$, the simple analytic model (5) is well suited to provide first order estimates of $g_N(m)$ in (a subgraph of) the Internet.

### B. Stability.

We now turn to the problem of quantifying the stability in a multicast tree and present new theoretical results which will be verified by simulations in the next section. Inspired by Poisson arrival processes, at a single instant of time, we assume that either no or one group member can leave. In the sequel, we do not make any further assumption about the time-dependent process of leaving/joining a multicast group and refrain from dependencies on time. As measure for the stability of the multicast tree, the number of links in the tree that change after one multicast group member leaves the group has been chosen. If we denote this quantity by $\Delta_N(m)$, then, by definition of $g_N(m)$, the average number of changes equals

$$E[\Delta_N(m)] = g_N(m) - g_N(m-1) \tag{8}$$

[1]The depth $D$ is equal to the number of hops from the root to a node at the leaves.

Since $g_N(m)$ is concave (Theorem 2), $E[\Delta_N(m)]$ is always positive and decreasing in $m$. If the scope of $m$ is extended to real numbers, $E[\Delta_N(m)] \approx g'_N(m)$ which simplifies further estimates.

The situation where on average less than 1 link changes if one multicast group member leaves may be regarded as a stable regime. Since $E[\Delta_N(m)]$ is always positive and decreasing in $m$, this regime is reached when the group size $m$ exceeds $m_1$, which satisfies $E[\Delta_N(m_1)] = 1$. For example, for the recursive tree which is the shortest path tree (as shown in [13]) for the class RGU, this condition approximately follows from (5) as

$$E[\Delta_N(m)] \sim \frac{mN}{N-m}\log\left(\frac{N}{m}\right) - \frac{(m-1)N}{N-m+1}\log\left(\frac{N}{m-1}\right) \tag{9}$$

Let $x = \frac{m}{N}$, then $0 < x < 1$ and

$$\frac{E[\Delta_N(m)]}{N} \sim \frac{-x}{1-x}\log x + \frac{(x-1/N)}{1-(x-1/N)}\log\left(x - \frac{1}{N}\right)$$

After expanding the second term in a Taylor series around $x$ to first order in $\frac{1}{N}$,

$$E[\Delta_N(xN)] \sim \frac{x-1-\log x}{(1-x)^2} + O\left(\frac{1}{N}\right)$$

Hence, for large $N$, $E[\Delta_N(x_1 N)] \sim 1$ occurs when $x_1 = 0.3161$, which is the solution in $x$ of $\frac{x-1-\log x}{(1-x)^2} = 1$. For the class RGU, a stable tree as defined above is obtained when the multicast group size $m$ is larger than $m_1 = 0.3161N \approx \frac{N}{3}$. In the sequel, since $m_1$ is high and of less practical interest, we will focus on multicast group sizes smaller than $m_1$. The computation of $m_1$ for other graph types turns out to be difficult. Since, as mentioned above, the comparison with Internet measurement (see [14]) shows that formula (5) provides a fairly good estimate, we expect that $m_1 \approx \frac{N}{3}$ also approximates the stable regime in Internet well.

The following Theorem quantifies the stability in the class RGU.

*Theorem 7:* For sufficiently large $N$ and fixed $m$, the number of changed edges $\Delta_N(m)$ in a random graph $G_p(N)$ with uniformly distributed link weights tends to a Poisson distribution,

$$\Pr[\Delta_N(m) = k] \sim e^{-E[\Delta_N(m)]}\frac{(E[\Delta_N(m)])^k}{k!} \tag{10}$$

where $E[\Delta_N(m)] = g_N(m) - g_N(m-1)$ and $g_N(m)$ is given by (4) or approximately by (5).

*Proof:* Previously [12],[13] we have shown that the shortest path tree from a source to an arbitrary node in the random graph $G_p(N)$ with uniformly (or exponentially) distributed link weights, is a uniform recursive tree for large $N$. In addition, the random variable for the number of hops (the hopcount $H_N$) from that source to an arbitrary node tends, for large $N$, to a Poisson random variable with mean $E[H_N] \sim \log N + \gamma - 1$, where $\gamma$ is Euler's constant ($\gamma = 0.5772156\ldots$). Hence, $\Delta_N(m)$ is the random variable that counts the absolute value of the difference between the hopcount $H_N(m)$ from the source to user $m$ and the hopcount $H_N(m-1)$ from the source to the user closest in the tree to $m$, which we label by $m-1$. Both users $m$ and $m-1$ are not independent, nor the two random variables $H_N(m)$ and $H_N(m-1)$ are independent in general due to possible overlap in their paths. If the shortest paths from the

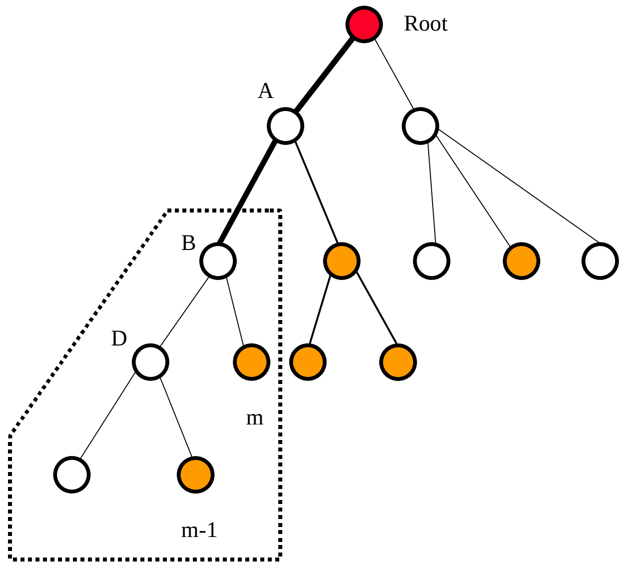


Fig. 2. A sketch of a uniform recursive tree, where $H_N(m) = 3$ and $H_N(m-1) = 4$ and the number of links in common is two (shown in bold Root-A-B).

root to each of the two users $m$ and $m-1$ overlap, there always exists a node in the shortest path tree, say node $B$ as illustrated in Figure 2, that sees the partial shortest paths from itself to $m$ and $m-1$ as non-overlapping and independent. Since the shortest path tree is a uniform recursive tree, the subtree rooted at that node $B$ (shown in dotted line in Figure 2) is again a uniform recursive tree[2]. With respect to $B$, the nodes $m$ and $m-1$ are uniformly chosen. We denote the unknown number of nodes in that subtree rooted at $B$ by $\nu(m) \le N$. We have that $\nu(m) \le \nu(m-1)$ because by adding a group member, the size of the subtree can only decrease. For large $N$ and small $m$, $\nu(m)$ is large such that the above mentioned asymptotic law of the hopcount applies. If both $m$ and $N$ are large, $\nu(m)$ will become too small for the asymptotic law to apply (a fact illustrated by the simulations in sec. III). Thus, for fixed $m$ and large $N$, this implies that $\Delta_N(m)$ tends to Poisson random variables with mean $E[\Delta_N(m)]$. For any graph and any $m$ and $N$ applies relation (8). Since $E[\Delta_N(m)]$ can be explicitly computed as (9), this completes the proof. ∎

Remark that the proof can be extended to a general topology. Assume for a certain class of graphs that the pdf of the hopcount $\Pr[H_N = k]$ and the multicast efficiency $g_N(m)$ can be computed for all sizes $N$. The subtree rooted at $B$ is again a shortest path tree in a subcluster of size $\nu(m)$, which is an unknown random variable. The argument similar as the one in the proof above shows that

$$\Pr[\Delta_N(m) = k] = \Pr\left[H_{\nu(m)} = k\right]$$

[2] Recall that a uniform recursive tree possesses the property that any new node $N$ has equal probability to be attached to any of the $N-1$ node already in the tree.

This argument implicitly assumes that all multicast users are uniformly distributed over the graph. By the law of total probability,

$$\Pr\left[H_{\nu(m)} = k\right] = \sum_{n=1} \Pr\left[H_{\nu(m)} = k|\nu(m) = n\right] \Pr\left[\nu(m) = n\right]$$
$$= \sum_{n=1} \Pr\left[H_n = k\right] \Pr\left[\nu(m) = n\right]$$

which, unfortunately shows that the pdf of $\nu(m)$ is required to specify $\Pr\left[\Delta_N(m) = k\right]$. However, we can proceed further in an approximate way by replacing the unknown random variable $\nu(m)$ by its best estimate, $E\left[\nu(m)\right]$. In that approximation, the average size $E\left[\nu(m)\right]$ of the shortest path subtree rooted at $B$ can be specified, at least in principle, with the use of (8). Indeed, since $E\left[H_{E[\nu(m)]}\right] = \sum_{k=1}^{E[\nu(m)]-1} k \Pr\left[H_{E[\nu(m)]} = k\right]$, by equating

$$E\left[H_{E[\nu(m)]}\right] = g_N(m) - g_N(m-1)$$

a relation in one unknown $E\left[\nu(m)\right]$ is found and can be solved for $E\left[\nu(m)\right]$. In conclusion, we end up with the approximation

$$\Pr\left[\Delta_N(m) = k\right] \approx \Pr\left[H_{E[\nu(m)]} = k\right]$$

which roughly demonstrates that, in general, $\Pr\left[\Delta_N(m) = k\right]$ is likely related to the hopcount distribution in that certain class of graphs.

Unfortunately, for very few types of graphs, both the pdf $\Pr\left[H_N = k\right]$ and the multicast gain $g_N(m)$ can be computed. This fact augments the value of Theorem 7, although the class RGU is *not* a good model for the *graph* of the Internet. Fortunately, the *shortest path tree* deduced from that class seems a reasonable approximation (as shown in [13]) and sufficient to provide first order estimates. In any case, we believe its value outweighs simulation results. Moreover, its relatively simple analytic character is desirable in modeling problems.

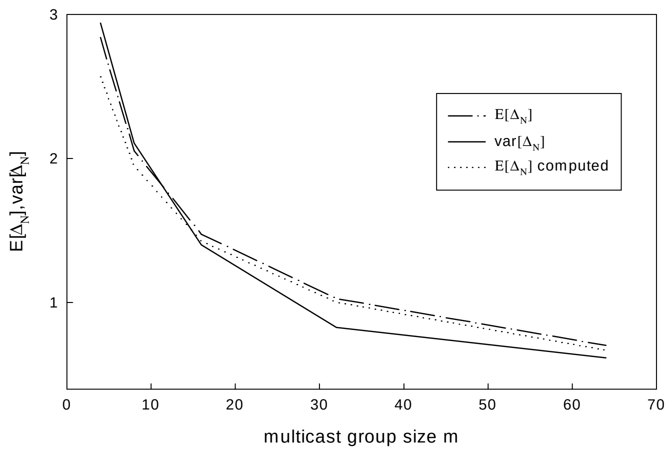


Fig. 3. SPT:The mean and variance of $\Delta_{100}$.

## III. Simulation Results.

The main goal of the simulations is to verify the quality of the asymptotic result in Theorem 7. In particular, section III-A is devoted entirely to that purpose. In section III-B, results for the Steiner tree on the same type of graphs for the class RGU are presented and compared to those of the corresponding shortest path tree.

In order to anticipate frequently received criticism about the class RGU, the value of the results only applies to this class RGU and no attempt is made to correlate these results to the current Internet, although the previous section did so. The main reasons are as follows:

1) The topology of the Internet is currently not sufficiently known to categorize the Internet as a type or an instance of a class of graphs. The Internet is most likely best seen as an organism changing over time; there does not exist a fixed Internet topology and, hence, a class specification is desirable, in particular for simulations. There are measurements (on a part) of the Internet that show that the Internet graph is sparse (low link density $p$) and that the distribution of the degrees (number of links per node/router) is likely polynomially distributed with exponent close to -2.2 (see. e.g. [9]). Unfortunately, these measurements only reveal a part of what we need to know (e.g. Are there large subgraphs in the Internet that are planar? Is the Internet clearly hierarchically structured? Is there a relation to the structure of the autonomous domains (when collapsed in a single point) and the structure inside an autonomous domain?) And many more of such questions can be posed.
2) For any routing problem, in addition to the network topology, we need also knowledge about the link weight distribution. Older systems are more likely to define all links with unit weight ($w = 1$). More recently, it makes sense to distinguish between a satellite link, a large bandwidth link and a smaller, or legacy link. Hence, not all link weights will be equal to $w = 1$.

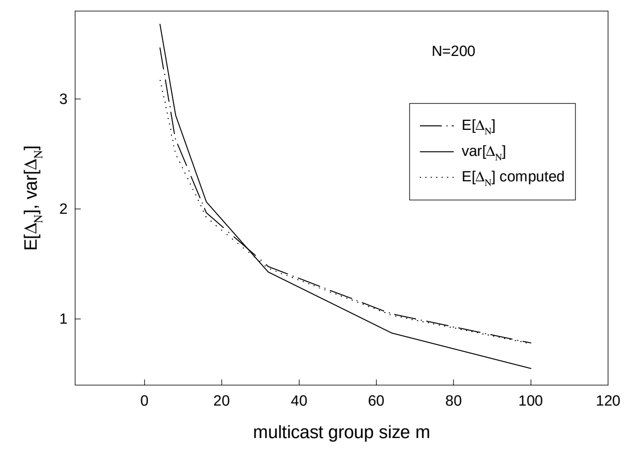


Fig. 4. SPT: The mean and the variance of $\Delta_{200}$

Even if more realistic topology generators (such as e.g. gt-itm [17]) are used, the second problem of the link distributions will be debatable. Moreover, the link weight distribution is equally important as the topology of the graph itself. Although it is believed that Waxman graphs [15] represent communication networks in a more realistic way, it has been demonstrated in [13] and [18] that there is no significant difference in the hopcount of the shortest path in these two families of graphs, provided the link weight distribution is uniformly or exponentially distributed. It has been shown in [13] that for $N$ large enough (in practice $N > 50$), the dependency of the hopcount of the shortest path on the link density $p$ (i.e. the number of links in the graph) becomes insignificantly small. Hence, by attaching a certain weight to a link, the specific details of the underlying topology may be shielded (or become irrelevant) in a routing problem.

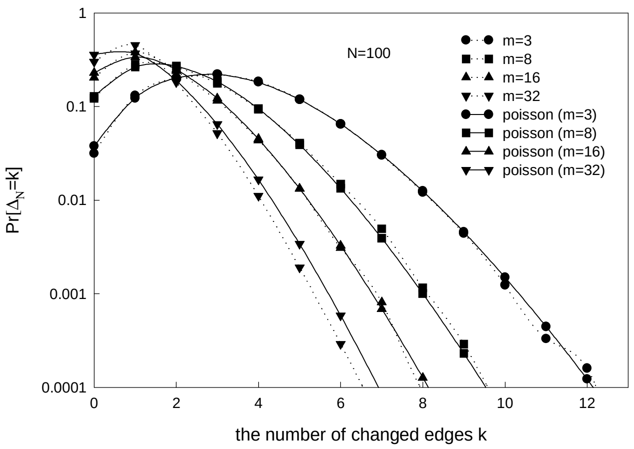


Fig. 5. SPT: Pdf $\Pr[\Delta_N = k]$ for $N = 100$ and $m < N/3$

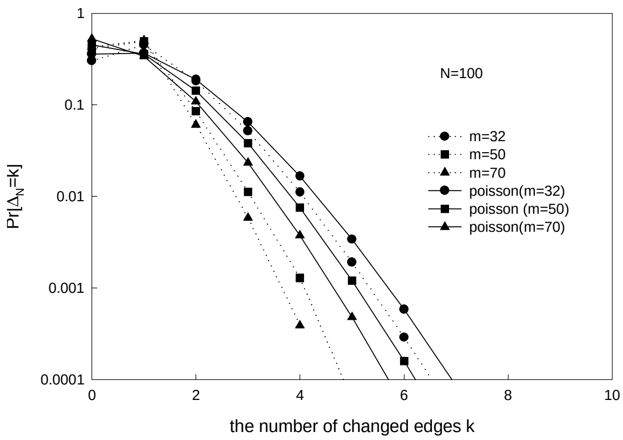


Fig. 6. SPT: Pdf $\Pr[\Delta_N = k]$ for $N = 100$ and $m > N/3$

## A. The shortest path tree

We confine ourselves to graphs of the class RGU with $N \geq 100$ and with link density $p = 0.2$. For each graph of $N$ nodes, we define the number of multicast users in the network, and the source node. For each $N$ and $p$, $10^5$ topologies are generated randomly. The connectivity is tested using the Prim's minimum spanning tree algorithm [16]. Only if the generated topology is connected, $m$ nodes out of $N - 1$ (the node number one was defined as a source node) are uniformly chosen, and the shortest path tree is computed using a modification of Dijkstra's algorithm. The number of edges in the tree was computed as well as the number of edges in the tree that interconnects one (uniformly chosen) multicast user less. The difference of those two values was stored in a histogram, from which the probability density function was deduced, and simultaneously also the mean $E[\Delta_N]$ and the variance $var[\Delta_N]$ of the number of changed edges. These two variables ($E[\Delta_N]$ and $var[\Delta_N]$) are plotted as a function of the multicast group size $m$, for three different values of $N$ $(100, 200$ respectively) on Figures 3 and 4.

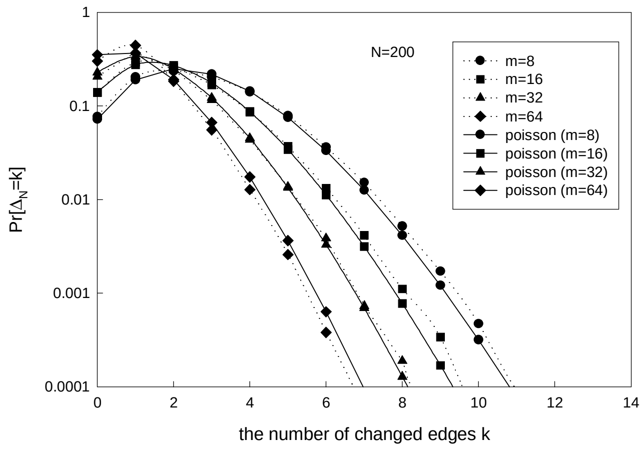


Fig. 7. SPT: Pdf $\Pr[\Delta_N = k]$ for $N = 200$ and $m < N/3$

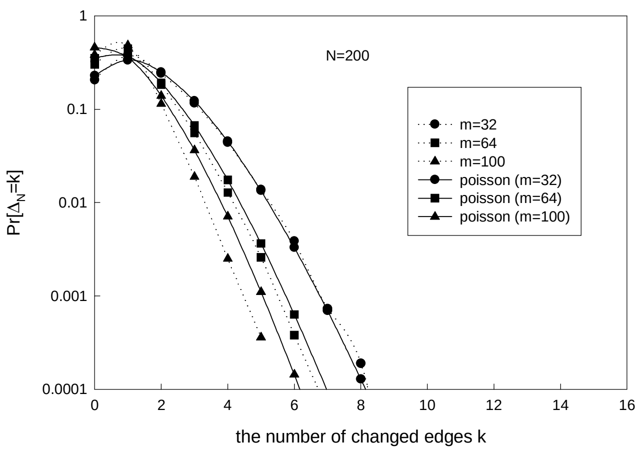


Fig. 8. SPT: Pdf $\Pr[\Delta_N = k]$ for $N = 200$ and $m > N/3$

However, for larger values of $N$, this simulation process is time consuming, and not efficient. Therefore, for $N$ larger than 500, we used a Markov discovery process to find the shortest paths from the source node to the other multicast group members. The Markov discovery process has been explained in detail in [13]. The Markov discovery process allows us to compute the shortest path tree very efficiently in large graphs (even up to $10^5$ nodes) of the class RGU.

We observe that the mean $E[\Delta_N]$ determined via the simulations, and the mean $E[\Delta_N]$ computed by (9) are almost identical. Another important observation is that there is an area where the mean $E[\Delta_N]$ and the variance $var[\Delta_N]$ tend to each other. Since this is a property of the well-known Poisson distribution, we are led to the conclusion that the probability distribution function of the number of changed edges $\Delta_N$ is very likely a Poisson distribution. In Figures 5 to 8, simulation results together with the Poisson law (10) are plotted in the dotted and the solid line respectively, as a function of the number of changed edges, with the multicast group size as a parameter.

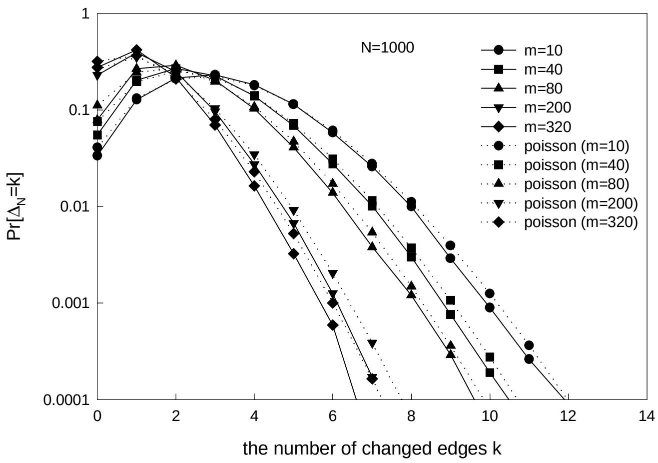


Fig. 9. SPT: Pdf $\Pr[\Delta_N = k]$ for $N = 1000$ and $m < N/3$

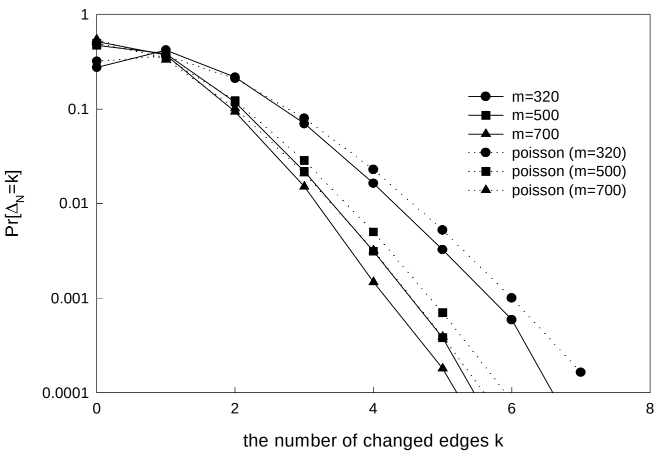


Fig. 10. SPT: Pdf $\Pr[\Delta_N = k]$ for $N = 1000$ and $m > N/3$

Figures 5 to 8 show that for $m < \frac{N}{3}$ (equivalent to $E[\Delta_N] > 1$), the probability distribution function is remarkably well described by the Poisson distribution. For $m > \frac{N}{3}$, the noticeable differences between the mean $E[\Delta_N]$ and the variance $var[\Delta_N]$ appear, and there are significant deviations of the probability distribution function from the Poisson distribution. The explanation is that the size $\nu(m)$ of the subtree rooted at $B$ as illustrated in Figure 2, becomes too small to justify a Poisson law for the hopcount in that subtree. But, as we have already

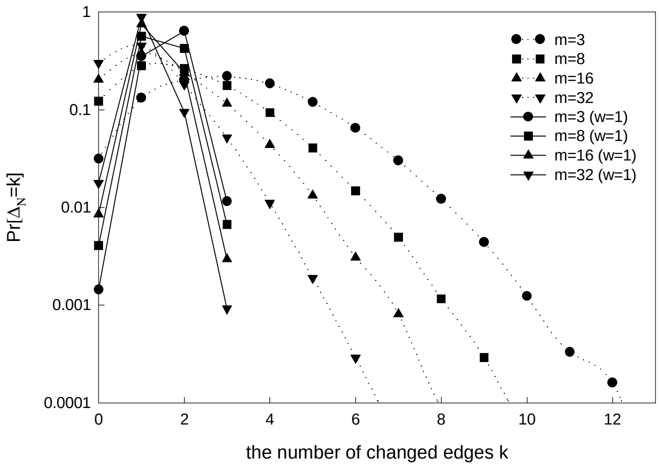


Fig. 11. SPT: Comparison of the pdf $\Pr[\Delta_{100} = k]$ in the class $G_{0.2}(100)$ with uniformly distributed link weights (dotted line) and all link weights $w = 1$ (full line).

explained in the section II-B, if the average number of changed links is less than one, the multicast tree can be considered as stable.

Figure 9 and 10 represent results obtained from the Markov discovery process, for $N = 1000$. These Figures show that, for $N = 1000$, the probability distribution function matches the Poisson distribution (10) even for larger values of $m$.

Finally, the effect of the link weight distribution on the number of changed branches $\Delta_N$ in the shortest path tree is illustrated in Figure 11. For graphs of the class $G_p(N)$, this Figure 11 compares the pdf $\Pr[\Delta_N = k]$ obtained with uniformly (or exponentially) distributed and with constant ($w = 1$) link weights. Earlier in [13], it is shown that, for all link weights equal in $G_p(N)$, the probability that the hopcount exceeds 2 hops precisely equals

$$\Pr[H_N > 2] = (1-p)\left[1-p^2\right]^{N-2}$$

and very rapidly decreases with $N$ for all link densities $p > \frac{1}{\sqrt{N}}$. This phenomenon is also observed in the behavior of $\Delta_N$ in Figure 11 and supports the generalization of the Poisson law (10) - which is deduced for uniformly (or exponentially) distributed link weights - that $\Pr[\Delta_N(m) = k]$ is reasonably well approximated by $\Pr\left[H_{E[\nu(m)]} = k\right]$.

Figure 11 also seems to indicate that less variability in the link weight distribution amounts to a higher stability of the shortest path multicast tree. Although concluded from the class $G_p(N)$, similar simulations with more realistic topologies generated by gt-itm [17] (transit-stub method) confirm this stable shortest path tree behavior.

In conclusion, the simulation results indicate that, in spite of the applicability of Theorem 7 to an asymptotic regime (large $N$ and fixed $m$), the law (10) seems to have a wider validity region. This feature, also previously observed in [19], reflects a robustness property of the Poisson law, that may be associated with almost sure behavior.

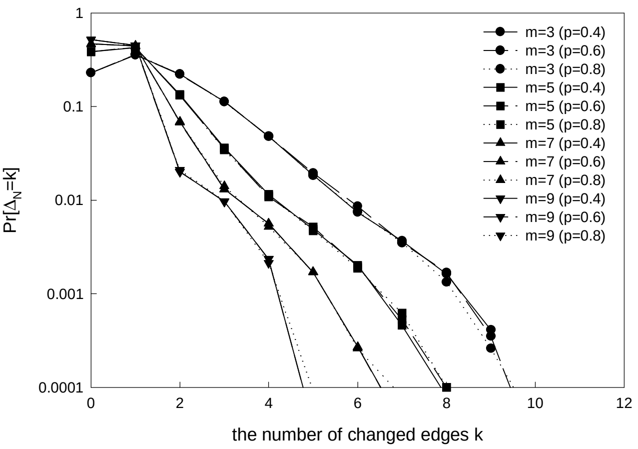


Fig. 12. MST: Pdf $\Pr\left[\Delta_N = k\right]$ for $N = 10$ $(\alpha = 0.2)$

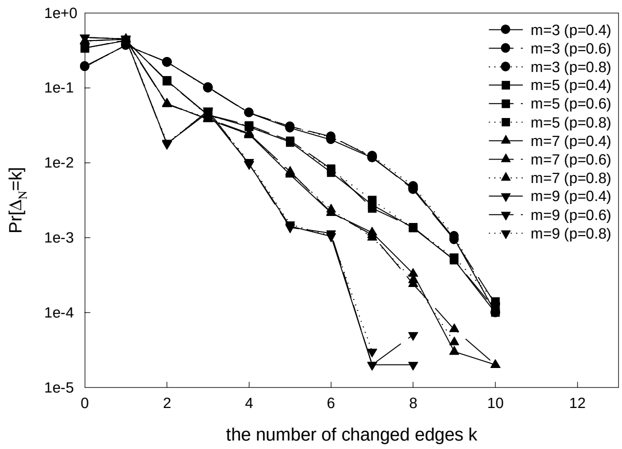


Fig. 13. MST: Pdf $\Pr\left[\Delta_N = k\right]$ for $N = 10$ $(\alpha = 0.5)$

## B. The Steiner tree

In this subsection we will present corresponding results obtained for the Steiner trees. The simulation process is similar to the one used for generating the shortest path tree. Again we performed simulations in the class RGU. We generated $10^5$ random graphs of that class RGU. In each graph, $m$ multicast group members are chosen uniformly out of the $N$ possible nodes. Depending on $m$, the Steiner tree [20] is generated using different algorithms. For $m = 2$, the minimum Steiner tree (MST) problem reduces to the computation of the shortest path between those two users. If $m = N$, the MST is actually the (complete) minimum spanning tree, and is computed with the Prim algorithm. For $2 < m < N$, the MST problem belongs to the class of hard *NP-complete* problems. Certain reductions [20] in the topology decrease the number of nodes and links to a reduced graph, and increase the speed of simulations. In spite of the implemented reductions, the simulation process is extremely time consuming for large $N$. Therefore, we confine ourselves to graphs where $N$ is not larger than 20. In each graph, the MST is computed for $m$ an $m-1$ members of the multicast group. The difference $\Delta_N$ in the number of the links forming these trees was stored in a histogram, from which the probability density function was deduced.

*1) Influence of the link weight distribution.:* For the class of $G_p(N)$ with various polynomial link weight distributions specified by the power exponent $\alpha$,

$$\Pr\left[w \le x\right] = x^{\alpha} 1_{0 \le x \le 1} + 1_{x \ge 1}$$

where $1_x$ is the indicator function[3], we have simulated the pdf $\Pr\left[\Delta_{10} = k\right]$ as shown in Figure 12 to 17 for $a = 0.2, 0, 5, 1, 2, 5, \infty$. The class RGU corresponds to $\alpha = 1$ and the last case ($\alpha = \infty$) corresponds to $w = 1$ everywhere.

The first observation from these Figures is that the pdf $\Pr\left[\Delta_{10} = k\right]$ appears to be independent of the link probability $p$ for $\alpha \le 1$. Second, the larger $x = \frac{m}{N}$, the more correlation there is in the Steiner tree which is reflected by oscillatory

[3]The indicator function $1_x$ equals 1 if the condition $x$ is true, otherwise it is zero.

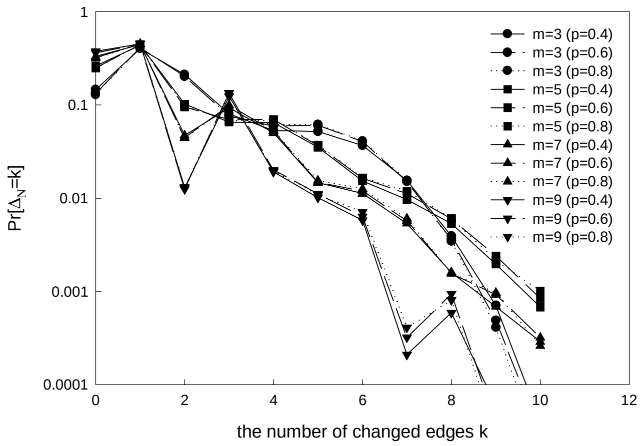


Fig. 14. MST: Pdf $\Pr\left[\Delta_N = k\right]$for $N = 10$ $(\alpha = 1)$

behavior of the probability density function. Third, these oscillations are more pronounced for increasing power exponents $\alpha$.

If the power exponent $\alpha$ is small (but $\alpha > 0$), $var\left[w\right] = \frac{\alpha}{(\alpha+2)(\alpha+1)^2}$ is relatively large (with a maximum for $\alpha = \frac{\sqrt{5}-1}{2}$ which is the 'golden number') while $E\left[w\right] = \frac{\alpha}{\alpha+1}$ is small. This variation implies the existence of smaller link weights that will play a dominant role in the Steiner tree. Since the Steiner tree is a minimum link weight tree, the links with smaller weights will more likely be included in both the Steiner tree with $m$ and $m-1$ multicast users. This will lead to a reasonable stable situation which is similar to the shortest path tree dynamics. The larger part of the tree will not change if a multicast user leaves or joins. The number of changed branches $\Delta_N$ in the Steiner tree is very unlikely to be smaller than in the corresponding shortest path tree because by choosing a longer hop path, it may be possible to achieve a lower total weight of the tree. As a second implication of small $\alpha$, the link weights have a thinning effect on the topology and overshadow the influence of the link density $p$: even if there is a link, it is the link weight that determines the importance of that link especially in shortest link weight problems. This explains the negligible effect of

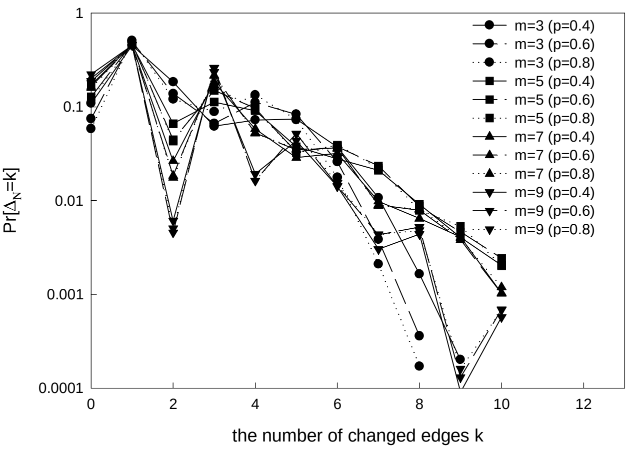


Fig. 15. MST: Pdf $\Pr[\Delta_N = k]$ for $N = 10$ ($\alpha = 2$)

$p$ as observed in Figure 12, 13 and 14.

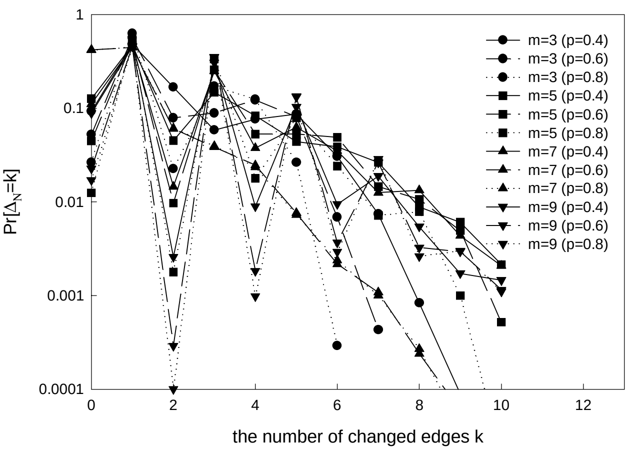


Fig. 16. MST: Pdf $\Pr[\Delta_N = k]$ for $N = 10$ ($\alpha = 5$)

When $\alpha$ is large, $var[w] \to 0$ and $E[w] \to 1$. Let us consider the limit case of $\alpha \to \infty$. All links are equally important and, hence, the effect of the topology quantified by the link density $p$ is important. If $p \to 1$, then $G_p(N) \to K_N$ and the behavior of $\Delta_N$ in the complete graph $K_N$ with $w = 1$ is readily analyzed. Any Steiner tree $s(m)$ in $K_N$ connecting $m$ multicast users consists of precisely $m - 1$ links while the total link weight of that tree also equals $m - 1$. Moreover, there exists a large number of different Steiner trees. In particular, the number of different minimum spanning trees or $s(N)$ trees in $K_N$ is precisely $(N-1)!$. The number of changed branches $\Delta_N$ consists of the total number of branches in $s(m)$ and $s(m-1)$ minus the 2 times the number $L_c$ of links in common. Hence, $\Delta_N = 2m - 3 - 2L_c$ or $\Delta_N$ is always odd, which explains the oscillatory behavior between odd and even values for $\Delta_N$ in Figures 16 and 17, especially for $p$ high. The stability of these Steiner trees is as worse as can be: the Steiner tree $s(m)$ in $K_N$ may consist of entirely different branches from those of the Steiner $s(m-1)$ as exhibited by the wild oscillations in Figure 17.

In conclusion, the simulations have shown that the link

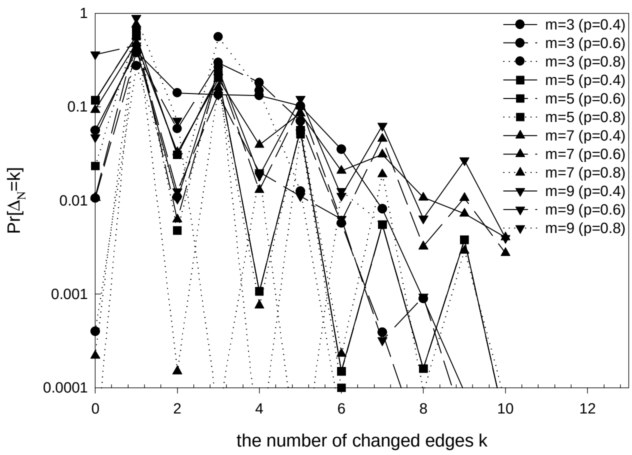


Fig. 17. MST: Pdf $\Pr[\Delta_N = k]$ for $N = 10$ ($\alpha = \infty$)

weight distribution has profound influence on the stability of the Steiner tree. The more links are equal (equivalent to $\alpha$ large), the higher the instability. If more links have different link weights, the more stable the Steiner tree is. Whereas the underlying topology is decisive in the former, it plays hardly a role in the latter situation. Thus, the more the link weight structure of a network is heterogeneous, the more healthy for the stability of the Steiner trees. Recall the opposite behavior for the shortest path tree as illustrated in Figure 11.

*2) Influence of the size $N$ of the graph.:* If we compare the results for the pdf obtained for $N = 10$ and $N = 20$ in the class RGU as illustrated in Figure 18, we observe that the probability density function for $N = 10$ and $N = 20$ match each other well for $x = \frac{m}{N} > 0.7$.

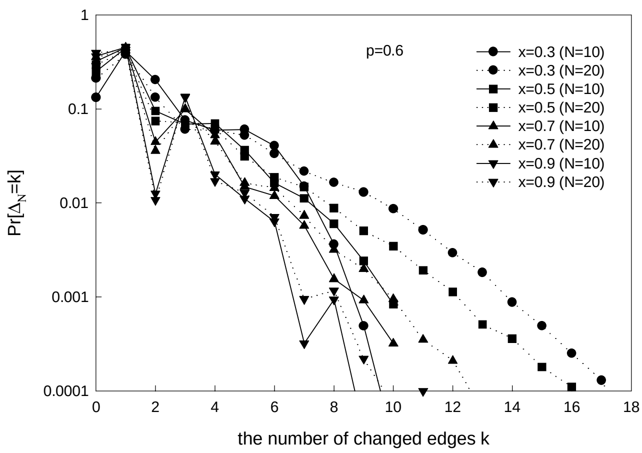


Fig. 18. MST: Pdf $\Pr[\Delta_N = k]$ for $N = 10$ and $N = 20$

The mean $E[\Delta_N]$ and the variance $var[\Delta_N]$ were also computed and plotted as a function of the ratio $x = \frac{m}{N}$ in Figure 19. We observe for the class RGU ($\alpha = 1$) that the mean value seems independent of the number of nodes in the network, although the variances differ.

*3) Comparison of Steiner and shortest path tree.:* In order to compare the stability of the Shortest path tree (SPT) and the

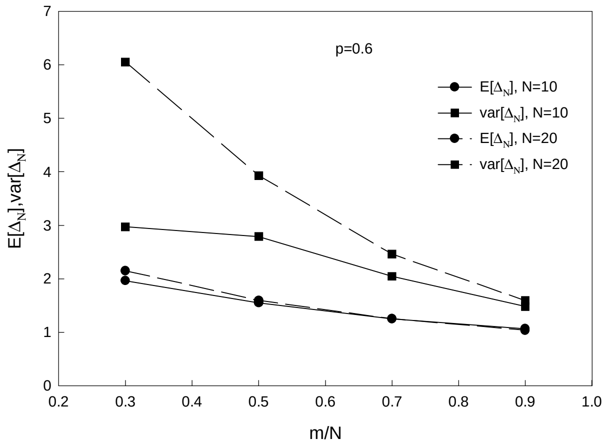


Fig. 19. MST: Mean and variance of $\Delta_N$ for $N = 10$ and $N = 20$)

Minimum Steiner tree (MST) in the class RGU, we have plotted in Figures 20 and 21, the probability density functions of changed number of edges $\Delta_N$ for $N = 10$ and $N = 20$ nodes, and in Figures 22 and 23 the mean value and the variance of these pdfs. From these Figures, the following observations can be made: (A) The maximum number of changed edges $\Delta_N$ in

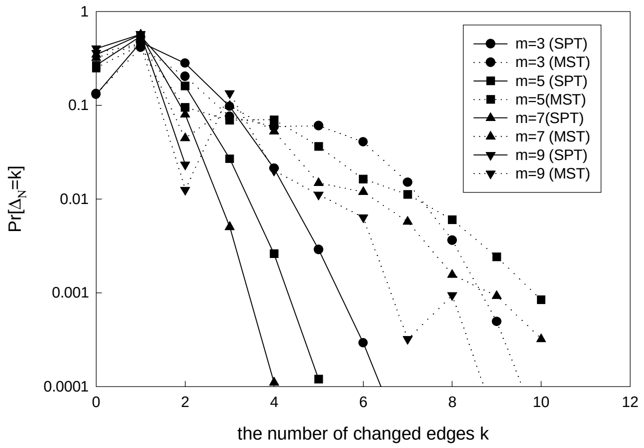


Fig. 20. Comparison SPT and MST ($N = 10$)

SPT does not increase with the increase of $N$ as fast as for the Steiner tree (MST). This phenomenon has been explained previously: the minimization of the weight of the total tree forces the Steiner tree to include longer hop paths if the sum of their link weights is smaller. (B) The pdf of $\Delta_N$ for the Steiner tree possesses a larger tail which agrees with the common intuition that Steiner trees are less stable than shortest path trees. (C) The larger tail for the Steiner tree also causes that the mean $E[\Delta_N]$ of MST is larger than that of the SPT and similarly for the variance. (D) The more remarkable observation is that the mean $E[\Delta_N]$ for $N = 10$ and $N = 20$ in both MST and SPT, hardly changes with $N$ for nearly all value of $x = \frac{m}{N}$. Most likely, for RGU or $\alpha = 1$, the dynamics of the Steiner tree resembles those of the SPT as argued above. The equality of $E[\Delta_N]$ and

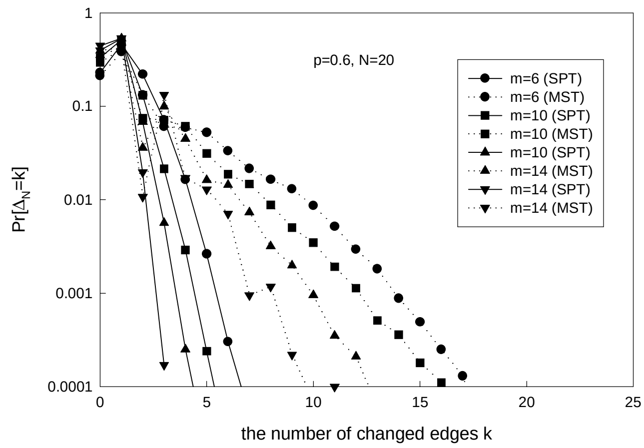


Fig. 21. Comparison SPT and MST ($N = 20$)

$var[\Delta_N]$ in SPT follows from the Poisson law (10).

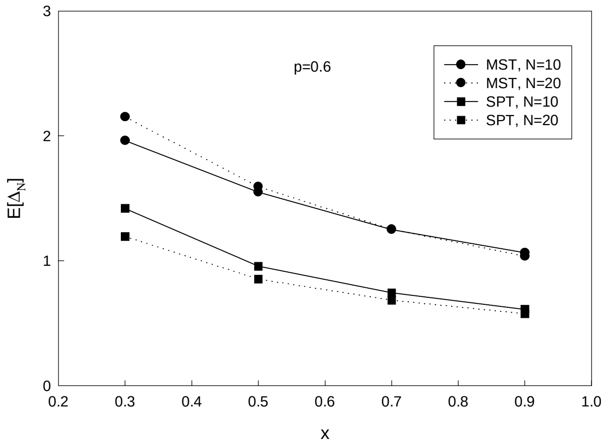


Fig. 22. Mean of $\Delta_N$ in MST and SPT ($N = 10, 20$)

## IV. Conclusion.

The stability of both the shortest path tree (SPT) and the Steiner tree (MST) has been quantified for the class RGU. The Poisson law (10) for the number of changed edges $\Delta_N$ in SPT, has been proven mathematically for the class RGU, while simulation results point towards a larger applicability of the Poisson law than the asymptotic regime. In addition, we have argued that similar laws as the Poisson law for the class RGU can be obtained for a general topology (including that of the Internet), provided both the hopcount distribution $\Pr[H_N = k]$ and the multicast efficiency $g_N(m)$ are known. Hence, the stability (in our setting) of the shortest path tree problem may be regarded in principle as approximately solved.

The behavior of the Steiner trees is not entirely understood and requires further analysis. Especially, for large $N$, it would be interesting to find the scaling laws of the Steiner tree as well as the tail behavior. Apart from large network sizes $N$, the simulations show that the link weight distribution determines the

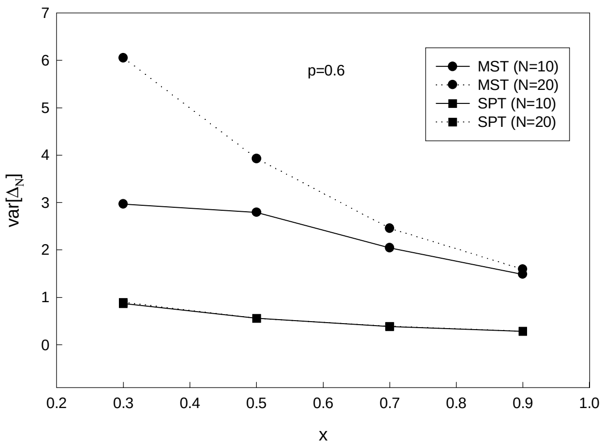


Fig. 23. Variance of $\Delta_N$ in MST and SPT ($N = 10, 20$)

stability of the Steiner tree problem. If the majority of the links is differently weighted, the stability of the Steiner tree resembles that of the shortest path tree. The other extreme, where most link weights are equal, leads to large instabilities reflected by wild oscillations in the corresponding pdf $\Pr[\Delta_N = k]$. At last, the stability of the Steiner tree is in most situations worse than that of the corresponding shortest path tree. Mainly because the departure or arrival of a multicast member may cause other branches to be included in the Steiner tree (to achieve an overall minimum in the sum of the weights) than just the branches of the shortest path towards the subtree rooted at $B$ (as defined in Figure 2).

## REFERENCES


[1] M. ABRAMOWITZ AND I.A. STEGUN, *Handbook of Mathematical Functions*, Dover, 1968.

[2] B. BOLLOBAS, *Random Graphs*, Academic Press, 1985.

[3] A. BALLARDIE, P. FRANCIS AND J. CROWCROFT, "Core Based Trees (CBT)", Proc. ACM SIGCOMM '93, San Francisco, CA, 1993., Sept. 1997.

[4] D. WAITZMAN, S. DEERING AND C. PARTRIDGE, "Distance Vector Multicast Routing Protocol", RFC 1075, November 1988

[5] J. MOY, "Multicast Extensions to OSPF (MOSPF)", RFC 1584, 1994.

[6] D. ESTRIN, D. FARINACCI, A. HELMY, D. THALER, S. DEERING, M. HANDLEY, V. JACOBSON, C. LIU, P. SHARMA AND L. WEI, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification," RFC 2362, June 1998.

[7] S. DEERING, D. ESTRIN, D. FARINACCI, V. JACOBSON, A. HELMY, D. MEYER, AND L. WEI, "Protocol independent multicast version 2 dense mode specification," Internet Draft, Internet Engineering Task Force, June 1999. Work in progress.

[8] F. BEGTASEVIC AND P. VAN MIEGHEM, "Measurements of the Hopcount in Internet", Proceedings of Passive and Active Measurement (PAM2001), Amsterdam, The Netherlands, April 23-24, pp. 183-190, 2001.

[9] M. FALOUTSOS, P. FALOUTSOS, C. FALOUTSOS, "On Power-Law Relationships of the Internet Topology", Proc. SIGCOMM '99, pp. 251-262, Aug. 1999.

[10] S. JANSON, T. LUCZAK AND A. RUCINSKI, *Random Graphs*, Wiley & Sons, New York, 2000.

[11] C. Huitema, *Routing in the Internet*, Prentice Hall PTR, New Jersey, 1995.

[12] R. VAN DER HOFSTAD, G. HOOGHIEMSTRA AND P. VAN MIEGHEM, "First Passage Percolation in the Random Graph", Probability in the Engineering and Informational Sciences (PEIS), vol. 15, 225-237, 2001.

[13] P. VAN MIEGHEM, G. HOOGHIEMSTRA AND R. VAN DER HOFSTAD, "A Scaling Law for the Hopcount in Internet", report 2000125 (http://wwwtvs.et.tudelft.nl/people/piet/teleconference.html).

[14] P. VAN MIEGHEM, G. HOOGHIEMSTRA AND R. VAN DER HOFSTAD, "On the efficiency of multicast", IEEE/ACM Transactions on Networking, 9(6), dec. 2001.

[15] B.M. WAXMAN, "Routing of Multipoint Connections", IEEE Journal on Selected Areas in Communications, 6(9):1617-1622, 1988.

[16] T.H. CORMEN, C.E. LEISERSON AND R.L. RIVEST, *Introduction to Algorithms*, The MIT Press, McGraw-Hill, 1995.

[17] E.W. ZEGURA, K.L. CALVERT AND M.J. DONAHOO, "A Quantitative Comparison of Graph-based Models for Internet Topology", IEEE/ACM Transactions on Networking, 5(6), December 1997.

[18] 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.

[19] P. VAN MIEGHEM, G. HOOGHIEMSTRA AND R. W. VAN DER HOFSTAD, "Stochastic Model for the Number of Traversed Routers in Internet", Proceedings of Passive and Active Measurement (PAM2001), Amsterdam, The Netherlands, April 23-24, pp. 190-194, 2001.

[20] F. HWANG, D. RICHARDS, P. WINTER, The Steiner Tree Problem (Annals of Discrete Mathematics, Vol 53), North Holland, pp. 103, 1992.

[21] R. BOIVIE, N. FELDMAN, Y. IMAI, W. LIVENS, D. OOMS AND O. PARIDIAENS, "Explicit Multicast (Xcast) basic Specification", Internet Draft <draft-ooms-xcast-basic-spec-01.txt>, 2001.