About the impact of randomly failing nodes and links in shortest path trees

Nora Berg

October 24, 2015

Contents

1. Introduction

Change within computer networks is introduced by reconfiguration methods of network algorithms such as routing. This change is necessary to reestablish the functionality of the network in case of failures. While interacting between nodes, network layers and algorithms, the reconfiguration methods react to a number of different observables, and impact other components as well. The increasing complexity of the system can result in unpredictable behavior and unstable networks. To optimize the stability and resilience of networks it is necessary to investigate why, when and which change is introduced by a network algorithm itself and how much additional change is inherent to the underlying network structure. While the former can be optimized by configuring the algorithm, the latter is given by the problem and need to be dealt with.

Many network algorithms are based on shortest path trees (SPT). If a link or node of a shortest path tree fails, a number of nodes get separated from the tree and need to be reconnected. In this context the number of nodes which get separated if a random link or node fails is a key observable. It is the damage in an SPT many network algorithms have to repair and therefore the common denominator. Therefore, the resilience and stability of different network algorithms can be compared by the effort of reconnecting nodes into the network.

In this work, we calculate the damage caused by a randomly failing link in an SPT. We investigate the scope of the failure as well as the number of nodes which get separated.

This work is structured as follows. The related work is presendet in Section 2. In Section 3, some basic notations are established. First calculations are made on k-ary trees in Section 4 to try out basic calculation methods. Thereafter, we outline the basics of uniform recursive trees (URTs) in Section 5. In Section 6, we use calculations on URTs to get approximations for the damage in SPTs. A comparison of the different mathematical approaches and their results follows in Section 7. We summarize and give an outlook in Sections 8 and 9.

2. Related Work

The formal quantification of reorganizing networks is an ongoing effort in several disciplines. A mathematical quantification of the number of new links in a SPT is given in [13]. An overview on quantification approaches of network behavior from other disciplines as biology, sociology, physics and economics is given in [10]. It lists theoretical approaches to describe emergent behavior by, e.g., statistical physics, agent systems with autonomous decision making, and graph theoretic approaches.

Van Mieghem et al. measured in [14] the statistical properties of shortest path trees and elaborate that uniform recursive trees (URTs) can be used to model shortest path trees. Furthermore in [13, 15], they deduce for multicast distribution trees modelled as URTs the distribution of the link changes if a node joins or leaves the multicast group.

From the mathematical point of view, there exist several results on URTs that were used in this work. An overview and many results about URTs can be found in [4]. In Section 6.1, we use results from [5, 12, 9], e.g., the degree of the root vertex in URTs. To calculate the sizes of random branches in URTs, we use results from [6] and the Polya-Eggenberger Distribution from [7]. Cohen and Havlin [3] provide more general information about complex networks. Additional information about k-ary trees can be found in [14]. Furthermore, networks and their complexity are researched to obtain information about emergent and unpredictable behavior. Some basic information about the effects of network complexity are obtained by the Network Complexity Research Group of the IETF [2].

3. Basic Definitions

In this section, we recapture properties of trees which will be used in the following sections. Some of the properties are shown in Figure 1. A graph G is defined over its sets of edges and vertices with $G = (V, E)$. The *size* of a graph is defined as the number of its vertices with $|G| = |V|$. A tree T is defined as a connected, loop free graph with one selected vertex that is the root v_r. In this work, only undirected trees are considered. The number of nodes in a tree is as usually denoted

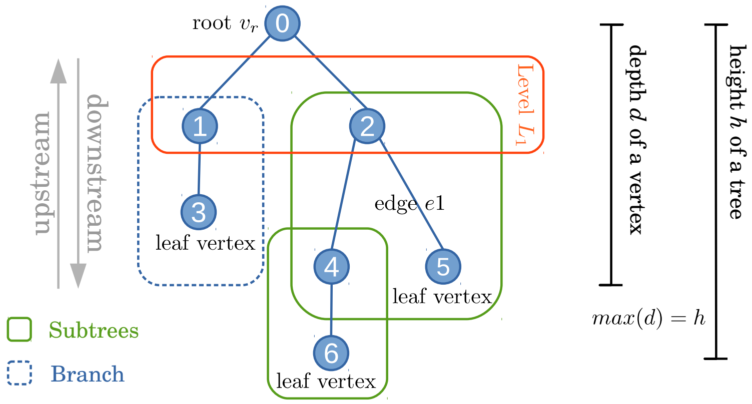


Figure 1: basic properties of a tree

by N. Let N_l be the size of a separated subtree after a link l failed, and N_n be the size of the separated subtree after the adjacent child vertex n of link l failed, then $N_l = N_n + 1$. Because of this, it suffices to only consider failing links. We assume, that each link can fail with equal probability.

In a tree there exist always an *upstream* and *downstream* direction. The former points towards v_r from an edge or vertex while the latter points towards a leaf vertex. Thereby, a child vertex v is always an adjacent vertex in downstream direction from parent vertex w. The *depth* of a vertex v is the number of edges between v and v_r. Therefore, the root vertex has always depth zero. The *height* of a tree is the maximal depth of a vertex in the tree. A *depth* of a link is the depth of the incident child vertex. A *level* L_d of a tree is the set of vertices or links of a specific depth d. The node degree of the root vertex equals the size of the set $|L_1|$. The *profile* of a tree is defined as the number of vertices per depth d. A *subtree* is a subgraph of a tree, which is also a valid tree. A branch is the downstream subtree that is separated if an edge is removed. In the following work, the short form *log* stands for the natural logarithm.

4. Calculations of branch sizes on k-ary trees

In this section, we observe the number of nodes in a perfect k-ary tree that gets separated if a random link breaks. For that, we need the number of vertices in a perfect k-ary tree of height h.

$$N(h) = \frac{k^{h+1} - 1}{k - 1} \tag{1}$$

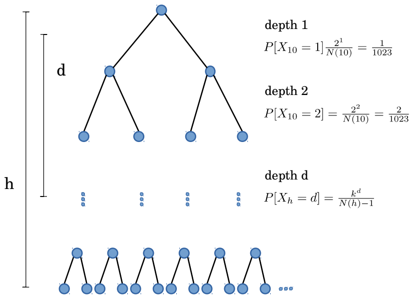
A well known property of all trees is that the number of vertices of a tree $|V|$ is related to the number of edges $|E|$ by $|V| = |E| + 1$. The tree with only the root vertex has $max(l) = 0$. Any tree with more vertices has the maximum depth of the height of the tree $max(l) = h$.

Let the random variable X_h map from a randomly chosen edge to its depth d, with $1 \leq d \leq h$:

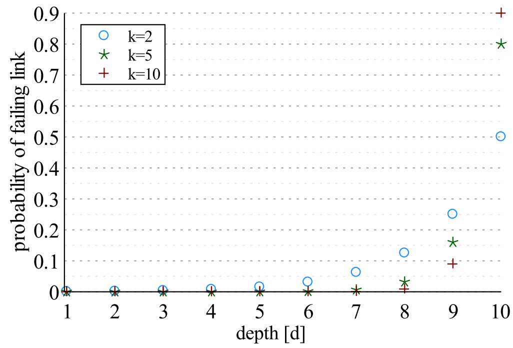
$$\begin{aligned} P(X_h = d) &= \frac{k^d}{N(h) - 1} \\ &= \frac{(k-1)}{k^h - 1} \cdot k^{d-1} \end{aligned} \tag{2}$$

Therein, k^d is the number of edges per depth d and the edge is chosen from the number of all edges $N(h) - 1$. Figure 2a illustrates a tree with $k = 2$.

Equation (2) is an exponential function with a static coefficient due to a fixed k and h. In Fig. 2b Eq. (2) is plotted as a function of depth d. The symbols denote different k. We see that the probability of choosing a lower link increases exponentially, and this effect is intensified with increasing k.



(a) The probability choosing an edge at level d is calculated by counting and comparing the edges per level with the total number of edges.



(b) Probability of choosing an edge at a depth d at a tree with $h = 10$

Figure 2: The probability of choosing an edge at a specific depth depends on k and h

X_h is directly related to the height h_s of a separated branch with $h_s = h - l$. So let random variable Y_h map from a randomly removed edge of a perfect k-ary tree with height h, to the height h_s of the separated branch. Y_h and X_h have the same probabilities. Thus with $0 \leq h_s < h$:

$$P(Y_h = h_s) = \frac{k^{h-h_s}}{N(h_s) - 1} = P(X_h = (h - l)) \tag{3}$$

For obtaining the expected number of nodes that are separated if a random link or node fails, we can use $N(h)$ from Eq. (1). Let the random variable D map from a randomly chosen edge to the number of descendent nodes in the separated branch. The expected value of D follows as:

$$E[D] = E[N(Y_h)] = \sum_{h_s=0}^{h-1} N(h_s) \cdot P(Y_h = h_s) = \sum_{d=1}^{h} N(h - d) \cdot P(X_h = d) \tag{4}$$

So the depth of the failing link d determines the size of the seperated subtree as $N(h_s) = N(h-d)$.

An example is given in Figure 3, in which a link is removed on level 2. The probability that a link is removed on this level is expressed as $P(X_{2,3} = 2) = \frac{2}{7}$. Therefore, the probability that a subtree of height 1 is seperated as $P(Y_{2,3} = 1) = \frac{2}{7}$. The probability variable D is more interesting for the problem of damage in shortest path trees, since it describes directly how many nodes need to be reconnected to the SPT.

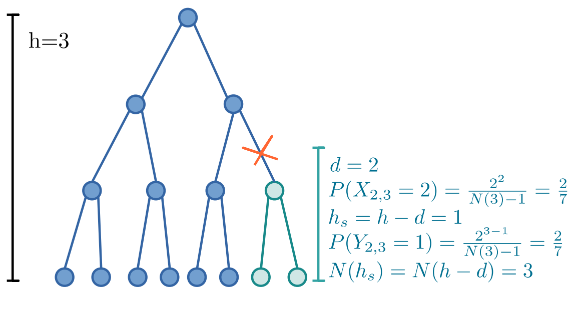
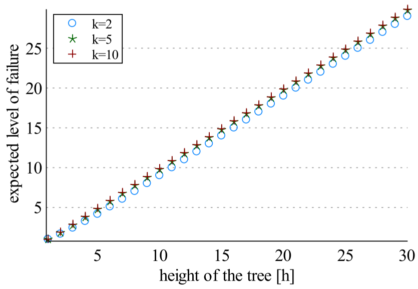
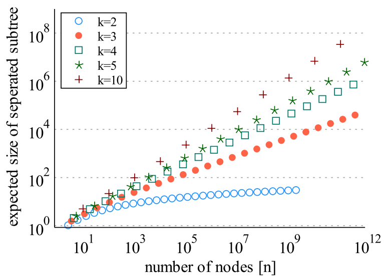


Figure 3: The expected size of a seperated branch depends on the probabilty $P(X_{k,h} = d)$ and the respective height of the tree $h - d$

In Figure 4a and 4b the expected size of the separated subtree $E[D]$ is shown with several values of k. The former graph depends on the height of the tree h and shows linear behavior. Furthermore the expected level of failure is independent from k. The latter depends on the total number of nodes in the tree and shows also nearly a linear increasing size of the expected subtree. Over several ks a threshold seem to exist, after which the size can be approximated linearly. The threshold converges towards zero with increasing k.



(a) The expected depth of failure $E[X]$ as a function of the height of the original tree



(b) The expected number of nodes in the seperated subtree $E[D]$ as a function of the number of nodes in the original tree. Since we use perfect k-ary trees not all numbers are possible x-values.

Figure 4: Exemplary calculations on a perfect k-ary tree with $k = 2$

5. Fundamentals about Uniform Recursive Trees

Uniform recursive trees (URT) are graph theoretic structures that can approximate shortest path trees [15]. Since many protocols on the routing layer create shortest path trees, we can use the widely studied properties of URTs for our approximations. Similar to the previous section, our goal is to quantify the number of nodes which get separated if a random link or node fails.

Uniform recursive trees are created by the following generating process [5]. We start with an empty graph. The first vertex (the root) with the label 1 is added to the graph. In each step, the new node with an increasing label is connected to one of the existing nodes with equal probability. Thus, the second vertex with label 2 is connected to the root, the third vertex is connected either to vertex 1 or to vertex 2 each with probability of $\frac{1}{2}$ and so on. The probability in the k^{th} step, to which of the existing nodes with labels $i = 1, 2, ..., k$ the new edge for vertex $k + 1$ is created, is independent of previous attachments. Therefore

$$p_{k,i} = \frac{1}{k}, \;\; i = 1, 2, ..., k, \;\; k = 1, 2, ..., n - 1$$

is the probability within the generating process at step k to connect vertex $k + 1$ to vertex i [5]. This is called a uniform recursive tree (URT).

The so called *recursive* property of URT means that every branch of an URT is a smaller URT itself. This property helps us to derive properties for the root vertex, and apply them to any vertex v in the tree.

6. Branch Sizes of Uniform Recursive Trees

In this section, we calculate the branch sizes of URTs. The goal is to get more accurate approximations than with k-ary trees for the damage in SPTs introduced by random link or node failures. To calculate the branch sizes we follow three approaches which will be introduced in the following sections. At the beginning of each section, the mathematical basics are introduced on which the approach is build on. After that, the approach itself is introduced.

In the first approach, the expected branch sizes of URTs are roughly approximated on basis of the root degree. Afterwards, we take a look at the profile of URTs and the related distribution of path lengths. This yields information about the depth of randomly failing links. In the last step, the distribution of nodes that are separated from a failing link or node is calculated with the Polya-Eggenberger distribution.

6.1. Rough approximation of Subtree Sizes based on the Root Degree

In this approach, we use the expected root degree $E[U_N]$ for an URT with N vertices to approximate the size of a branch rooted at depth d. A branch of this size would be separated if an edge of level L_d was removed.

Mathematical Fundamentals

Let U_N be the random variable that maps to the degree of the root vertex in the URT with N nodes.

The number U_N can be derived from the random variable S_j that describes that the vertex $j+1$ is connected to the root vertex in the j^{th} step of the generating process [5]. Since each choice of the parent node in the generating proces is independent of the previous one we obtain

$$P(S_j \in L_1) = \frac{1}{j} \tag{5}$$

Thereby, L_1 is the set of nodes of depth 1. Furthermore, we obtain

$$E\left[S_j\right] = \frac{1}{j} \quad \text{with } j = 1, ...N-1 \tag{6}$$

According to [5], in a graph with large N the expected number of branches can be calculated as

$$E\left[U_N\right] = \sum_{j=1}^{N-1} E\left[S_j\right] = \sum_{j=1}^{N-1} \frac{1}{j} \tag{7}$$

which is the $(N-1)^{th}$ harmonic number.

Approximation of Subtree Sizes

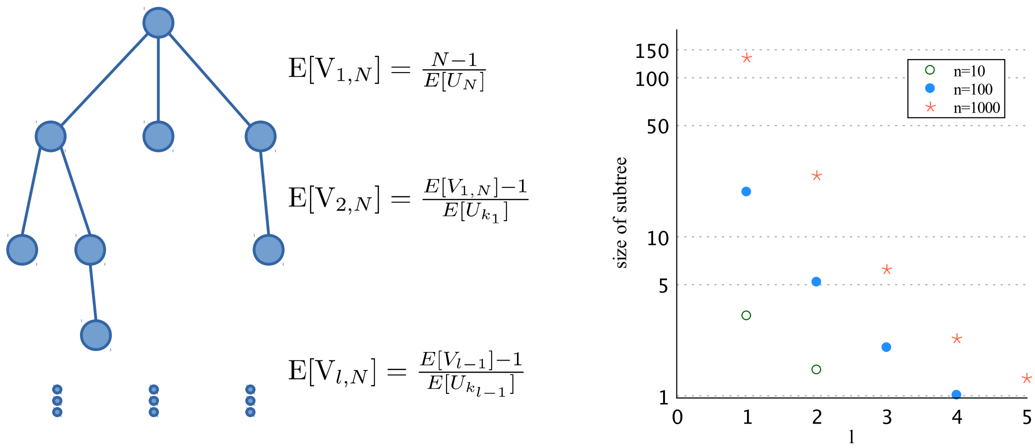
The size of an URT T has is the number of the sum of the branches sizes of the root $i = U_N$, so $|T| \approx |T_1| + ... + |T_i| + 1$. Since the expected value is additive, we get also $E[|T|] \approx E[|T_1|] + ... + E[|T_i|]$. Therefore the expected size of branches rooted at depth one can be calculated by $E[V_1, N] = (N-1)/T_i$. In the following approach, we assume that i is always the expected node degree of the root with $i = E[U_N]$.

Let $V_{d,N}$ denote the random variable that maps from a randomly chosen URT (out of all possible URTs) to the size of a branch that is separated by removal of an edge in depth d. Even within a single URT $V_{d,N}$ can take several values. In this approach, we assume that the size of a separated branch rooted at depth d is $k_d = E[V_{d,N}]$.

Assuming that $E[U_N]$ is the degree of the root of an URT with N vertices, we try to approximate the size of the URTs rooted in depth one as $E[V_1, N] = (N-1)/E[U_N]$. Therefore, we approximate the expected value $E[V_{d,N}]$ recursively as follows:

$$E[V_{d,N}] \approx \begin{cases} \frac{E[V_{d-1,N}]-1}{E[U_{k_{d-1}}]} & \text{if } d > 1 \text{ and } E[U_{k_{d-1}}] > 0 \\ \frac{N-1}{E[U_N]} & \text{if } d = 1 \text{ and } E[U_N] > 0 \end{cases} \tag{8}$$

Equation (8) is a rough approximation which assumes that every vertex in the same depth has the same number of child vertices and all of them are the root of branches of the same size. Therefore the $E[V_{d,N}]$ is not really the expected value of $V_{d,N}$ but describes a k-ary tree in which all leafs have the same depth. In that tree, the number of child vertices follows the expected values of an URT. In Figure 5b we see, that the sizes of branches decrease very fast, even in graphs with large N. This is the result of the assumption on the branching structure and branch sizes in URTs. A low height is a typical characteristic of shortest path trees and URTs. An inaccuracy in this approach is that expected values of the recursive steps are not necessary integers, so we need to round in such cases.



(a) $E\left[V_{l,N}\right]$ recursively approximated over levels

(b) The expected subtree size depending on level l for URTs with 10, 100, and 1000 vertices

Figure 5: The approximated expected value for the size of separated subtrees if a link fails in level l is $E\left[V_{d,N}\right]$

6.2. Depth of randomly failing links and nodes

A profile of a tree is defined as the number of nodes over all depths d. URTs mostly become wide and have a low height (compare Fig. 6). We see that most nodes are located at a medium depth. Nevertheless, the depth of the branches can vary for single branches.

Therefore, the size of randomly separated branches is closely related to the profile of a tree. Let the random variable $X_{d,N}$ map to the number of vertices at depth d of an URT. The expected profile of a tree with N nodes is given (from [4]) by

$$E[X_{d,N}] = \frac{|s_{d+1,N}|}{(N-1)!} \tag{9}$$

with $s_{d,N}$ as the first sterling number. It can be approximated with $\alpha = \frac{d}{log\ N}$ as:

$$E[X_{d,N}] \sim \frac{N^{\alpha\cdot(1-log\ \alpha)}}{\Gamma(\alpha+1)\cdot\sqrt{2\pi\alpha\ log\ N}} \tag{10}$$

The profile gives us the information about the distribution of depth of all nodes. In Fig. 7a, the profile is approximated with Eq. (10) for URTs of the sizes 10, 100, and 1000. It shows that the number of a nodes peaks around depth $d = log\ N$. Furthermore, nearly all nodes are located around this level [4]. Accordingly, a higher probability of a node or link failure in this depth is also expected.

The expected height of an URT is about $e\ log\ N$ [4]. This means, nodes with depth $d < log\ N$ have a high out degree so that the peak around level $log\ N$ is reached. In addition, the fast decreasing curve (Fig. 7a) for $d > log\ N$ shows that only a small amount of nodes lie deeper. Therefore, we can expect that the probability of a node failing around level $log\ N$ is high, as well as we can expect that only a small number of nodes will be separated by that.

We can deduce the expected depth at which a randomly chosen link or node is removed. This distribution is shown in Fig. 7b and derived by renormalizing Eq. (10) with N. So the x-axis shows the depth and the y-axis fraction of all nodes that are of depth d. This distribution shows the node depths for shortest path trees. One can see that the number of disconnected nodes in case of a link or node failures is expected to stay relatively small. The reason is that the link will break with a high probability around $d = log\ N$ and there is only a small fraction of nodes nodes beneath level d.

Based on that observation, we can expect that the number of initially started reconfiguration methodes to reconnect the nodes to the network is relatively small. Nevertheless, the calculation

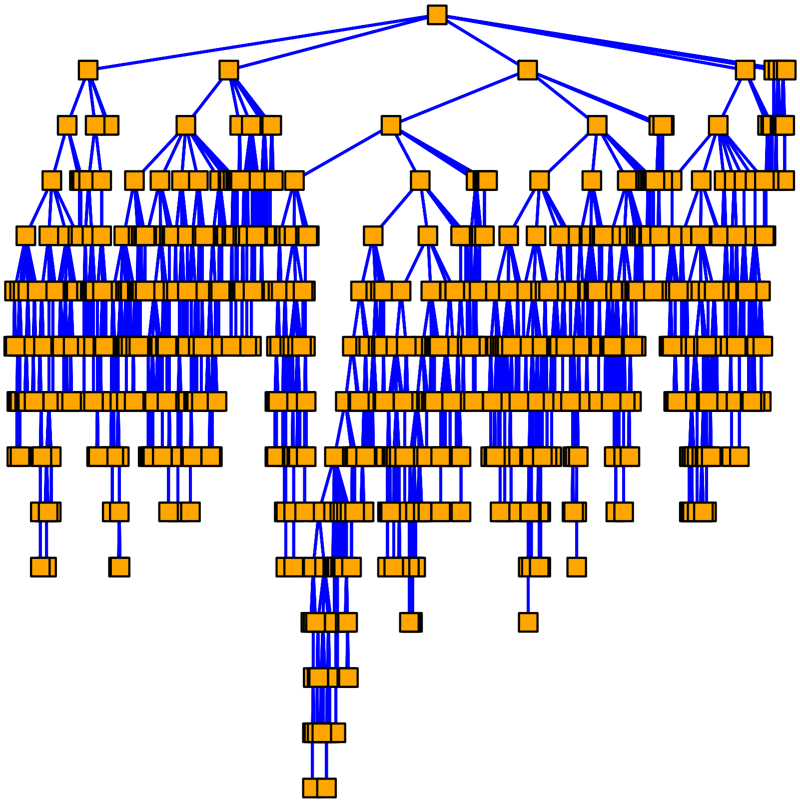
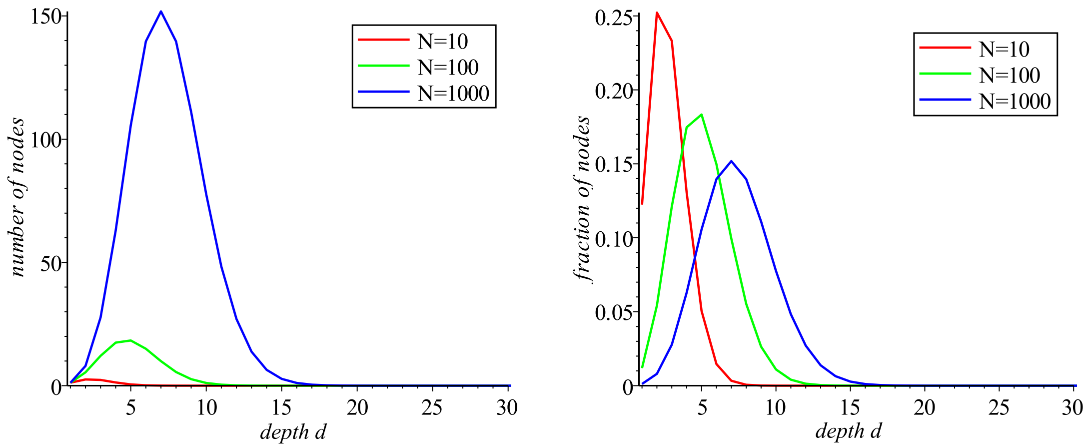


Figure 6: Example of URT with 1000 nodes

of the URT profiles gives no further information about the overall change in the network which may follow in a cascading manner. The reason is that the latter depends on the specific network algorithm.



(a) Expectation of URT profiles

(b) The renomalized expected profile of node depths

Figure 7: Profile of URTs

6.3. Average Path Length

In the next step, we want to calculate the change that is introduced into the rest of the SPT after a branch is separated. For that we take a look at the depth on which a randomly chosen link is located. This depth also equals the number of nodes which can be affected through cascading change towards the root. In this section, the relation between the path length and the insertion depth of a node n is shown. The insertion depth of a node in n is the depth of which the node at the n^{th} step in the generating process is put into an URT. Let the probability variable Q_n map from the inserted node n to its insertion depth d the distribution (from [4]) reads

$$P(Q_n = d) = \frac{E[X_{l-d,n-1}]}{n-1} \tag{11}$$

Thereby is X the profile of the tree as shown in Section 6.2. The relation to the overall depth of the node is that the width of the levels in an URT is the probability distribution that the new node is added on the level beneath.

An example is given in Fig 8. The probability that node $n = 6$ is inserted in the level $d = 2$ is $2/(n-1)$ since we have exactly two nodes on level $d - 1 = 1$.

Let random variable R map to the depth of a random node in an URT of N nodes. We obtain the relation to the insertion depth as:

$$P(R_N = d) = P(Q_{N+1} = d+1) = \frac{E[X_{d,N}]}{N} \tag{12}$$

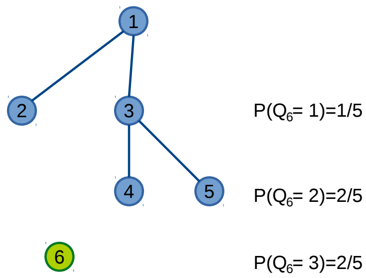
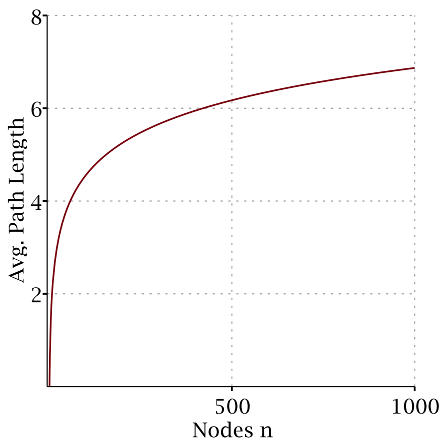


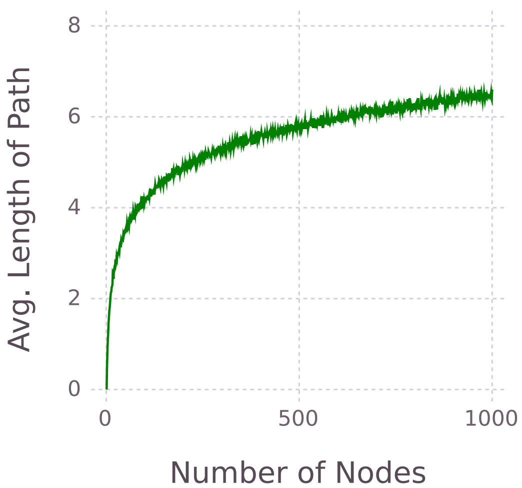
Figure 8: The distribution of node depths determines the distribution of insertion depth

In Fig. 9a, the expected value of Eq. (12) is shown. We see that it complies to the measured average of node depths in Fig. 9b. The measurements are done for trees of size 1 to 1000. For each tree size the average path is tested 100 times. The expected path grows very slow with large N. This complies with the profile of URTs that has nearly all nodes around the depth $log\ N$ [4].

The average path length also reveals information about the change in the network which is introduced by randomly failing links or nodes. The depth shows the effect a separation of nodes can have to the rest of the tree. For that we assume that only the path to the root is affected by the separated branch.



(a) expected path length over Eq. (12)



(b) measured avg. path length

Figure 9: The distribution and average path length of URTs

6.4. Distribution of Branch Sizes

This section derives the distribution of branch sizes in uniform recursive trees. At first, we derive the distribution of branch sizes in URT dependent on the label n of the node, as in [6, 7]. Thereafter, we derive the total probability of branch sizes in an URT with N nodes independent from the label. At last we show the comparison between the calculated and tested distributions.

Mathematical Fundamentals

To calculate the distribution of URT-Branchsizes two mathematical fundamentals are needed. The rising factorials and the Polya-Eggenberger distribution [7] are summarized in this section.

The value $n^{\overline{k}}$ is called *rising factorial* [1]. It is defined as

$$n^{\overline{k}} := n \cdot (n+1) \cdot ... \cdot (n+k-1) \tag{13}$$

As in [6], we get the distribution of branch sizes in URTs from the Polya-Eggenberger distribution [7]. In the Polya urn analogy, we have an urn with balls of two colors. The number of white and black balls is given as w resp. b. At each draw, the ball as well as a number of s balls of the same color of the chosen ball is returned to the urn. So the total number of balls increases and the probability which color is drawn changes at each draw. Within n steps of drawing, the random variable B maps to the number of times a black ball is chosen. B's distribution is given by the Polya-Eggenberger distribution

$$P\left(B=j\right) = \binom{n}{j} \cdot \frac{\alpha^{\overline{j}} \cdot \beta^{\overline{n-j}}}{(\alpha+\beta)^{\overline{n}}} \tag{14}$$

whereby $x^{\overline{y}}$ is the rising factorial, $\alpha = b/s$ and $\beta = w/s$. If $s = 1$, the formula can be simplified to

$$P\left(B=j\right) = \binom{n}{j} \cdot \frac{b^{\overline{j}} \cdot w^{\overline{n-j}}}{(b+w)^{\overline{n}}} \tag{15}$$

Application to URTs

The Polya-Eggenberger distribution gives us the distribution of branchsizs in an URT. The reason is that the adding of bowls fits to the growth process of the branches as follows. The vertex with label k is added in the k^{th} step to the tree generating process. In the Polya urn analogy, the label k can be thought of black color while all the other vertices are white. During the URT grows, the added vertices become the same color as their respective parent vertex, analog to returning one additional black or white ball into the urn ($s = 1$). So the number of black nodes that are chosen(= drawn) as parent vertices in the generating process equals the number of descendents of the original black colored ball k. Hence, the number of k's descendents can be described with the Polya-Eggenberger distribution with $s = 1$.

Let the random variable $C_{k,n}$ map from an URT of size k nodes and n successively added nodes, to the number of descendents of the node with label k. Thus, the probability distribution of $C_{k,n}$ is the Polya-Eggenberger distribution with $s = 1$ and $b = 1$ and follows as:

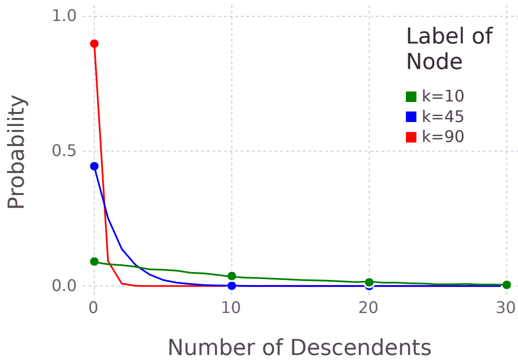
$$P(C_{k,n} = j) = \binom{n}{j} \cdot \frac{1^{\overline{j}} \cdot (k-1)^{\overline{n-j}}}{k^{\overline{n}}} \tag{16}$$

Furthermore, we get the number of descendents of node k in an URT with N nodes by setting $N = n - k$. Let this be described by the random variable $D_{k,N}$ with

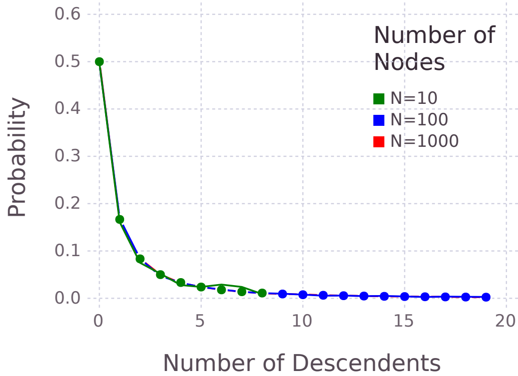
$$\begin{aligned} P(D_{k,N} = j) &= \binom{N-k}{j} \cdot \frac{1^{\overline{j}} \cdot (k-1)^{\overline{N-k-j}}}{k^{\overline{N-k}}} \\ &= \frac{(k-1) \cdot (N-k-j+1)^{\overline{j}}}{(N-j-1) \cdot (N-1)^{\overline{j}}} \end{aligned} \tag{17}$$

$D_{k,N}$ yields the distribution of the of the number of descendents under the condition, that k is already chosen.

In Figure 10a the distribution of Eq. (4) is shown in comparison to a measured distribution in an URT with 100 nodes in 10000 tests for each k. The lines show the measured distribution while the points are plottet from Eq. (17). We see that the values coincide.



(a) The distribution of the number of descendents of node k in an URT of 100 Nodes



(b) The total distribution of branch sizes in an URT. The lines are measured branch sizes and the points are given by Eq. (19)

Figure 10: Lines are plotted from the measured distribution. The points are the equivalent values from the theoretical derived distribution.

In the next step, we can use the law of total probability to get the overall distribution of branch sizes. Let the random variabel L_N map from a randomly chosen node in an URT with N nodes to its label k. Furthermore, let random variable D_N map from an URT with N nodes to the number of descendents for a randomly chosen node.

The probability for each k is $P(L_N = k) = \frac{1}{N}$. Since Eq. (16) describes the number of descendents in case a node k is already chosen, it can be rewritten as conditional probability dependent on L_N:

$$P(D_N = j|L_N = k) = P(D_{k,N} = j) \tag{18}$$

From this we can apply the law of total probability [8] to retrieve $P(D_N = j)$ with

$$\begin{aligned} P(D_N = j) &= \sum_{k=1}^{N} P(D_{k,N} = j) \cdot P(L_N = k) \\ &= \sum_{k=1}^{N} P(D_{k,N} = j) \cdot \frac{1}{N} \end{aligned} \tag{19}$$

This distribution describes the probability of a branch size j when a random node is chosen. In the case of the vulnerability of multicast trees, it shows the distribution of how many nodes are separated from the multicast tree, after a random node fails.

In Figure 10b, the distribution of Eq. (19) (points) is plotted in comparison to counted branch sizes (lines). The tests include the counting of all branch sizes in URTs with 10, 100 and 1000 nodes. The measured branch sizes comply with the probability distribution of Eq. (19). We also see, that the distribution is independent of the actual number of nodes.

The distribution of branch sizes show directly how many nodes are disconnected if a random node or link fails. It supports the impression from Section 6.2 that most branches are relatively small and therefore the number of initialized rejoin procedures is relatively low. Furthermore, the number of initialized reconfiguration methods, e.g. rejoin, is independent from the total number of nodes within the shortest path trees. This means, that if a network experiences a higher churn by random failures with a higher number of nodes, the reason for that lies in the global processing and forwarding behavior but is not just given by the pure number of nodes. Furthermore, this means the churn of a network can be influenced by optimizing the forwarding behavior of the network while the network size stays the same.

7. Comparison of the Calculation of Branch Sizes

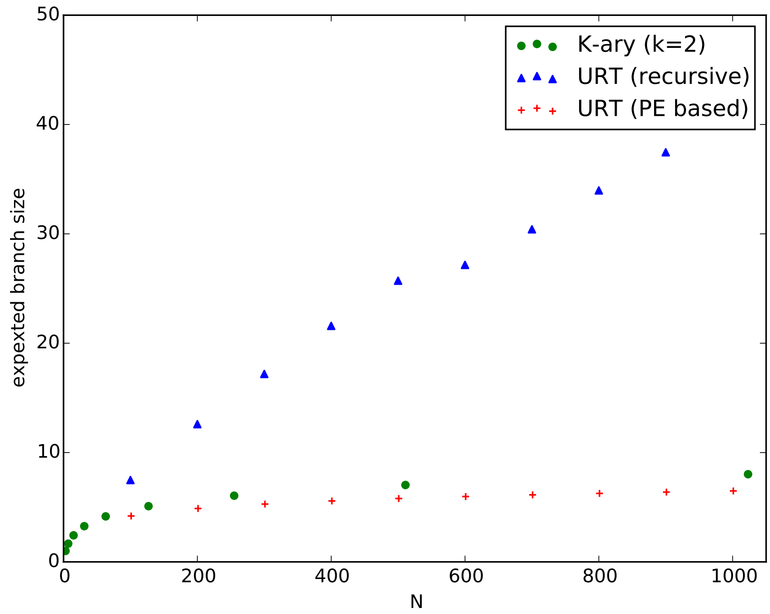


Figure 11: Comparison of the three approaches for expected branchsize as a function of N. K-ary trees (Sec. 4), the recursive URT approach (Sec. 6.1), and the Polya-Eggenberger based approach (Sec. 6.4)

In the last sections, we showed three approaches to calculate the expected branch size in case a random link breaks. Figure 11 shows the expected branch size as a function of N for the three approaches. The first approach is the expected branchsize from k-ary trees $E\left[V_{d,N}\right]$ as described in detail in Section 4. The URT based approaches are the recursive approach from section 6.1 and the Polya-Eggenberger based approach of Section 6.4. In the last case, the data of Fig. 11 is the expected value of the random variable $P[D_N = j]$.

In the first approach we used perfect k-ary trees to observe which properties of trees are of interest for the damage in SPTs. The predictable structure of k-ary trees is useful to test basic calculation ideas. Furthermore, it gives us an immediate probability distribution where a link is removed and which size the branches are. As shown in Fig. 11, we find that the expectation for the number of nodes in the separated subtree increases nearly linear, or logarithmic in case k=2.

In the second step we use uniform recursive trees (URTs) which are known to model shortest path trees. URTs have the advantage of resulting in reliable information about shortest path trees but introduce probability into the calculations. We see, that the expected branchsizes from the recursive appraoch increases much faster than the other URT based calculation. Therefore the approaches need to be verified with measurements. The approach based on the Polya-Eggenberger distribution coincides with the measurements of the average branchsizes. The reason is that the successively added vertices during the generation process result in branch sizes that follow this distribution. In comparison, the approach based on the recursively applied expected root degree does not result in a reliable approximation.

The most accurate expected size of a randomly separated branch is therefore given by the Polya-Eggenberger approach.

8. Summary

This work is about the impact of random failures in distribution trees within the network. The overall goal is to obtain reliable information where within the network algorithm change, churn and cascading failures arise, and how much reconfiguration effort is needed from the underlying network structures. While the former can be optimized by changing the network algorithm, the latter is just given and provides the limits for optimization efforts.

In this work several approaches for the approximation of damage in shortest path trees are studied. The profile of the URTs shows that these trees are more wide than high. Furthermore, the profile shows that the number of nodes that are separated by random failures is relatively small. This means for SPT algorithms that the number of individually initialized reconfiguration behavior is also relatively small in case of a random failure. With the Polya-Eggenberger distribution we calculated the distribution of a randomly separated branch in a URT with N nodes. This distribution was verified with measurements. It turns out that this distribution is independent from N. This supports the idea that the initially needed reconfiguration within the network is not necessarily crucial for the churn in the network. If the churn rises with the number of nodes, this can be optimized by focusing on the forwarding and processing of these information within the network because it is not automatically given by the pure network size. Furthermore, in many network algorithms the reconfiguration information is only forwarded to the root vertex. This intensifies this idea because the absolute height of these trees stay relatively small.

Furthermore we studied the average path length of URTs. These turn out to be relatively low. If we consider reconfiguration information that is distributed upwards in the tree, these information may often not travel very wide. This is also a well known property of small world networks [3]. The distribution of branch sizes turned out to be independent of the number of nodes in the network.

In summary, we find that shortest path trees do not require much reconfigurational change by themselves because of their low height and small branch sizes. So the complexity arises from the forwarding and processing behavior within the network.

9. Outlook

In the ongoing work we investigate how much change within the network is introduced by damage from randomly failing links. From the change inherent to the network, we try to find a lower and an upper bound for the necessary link state changes. These two bounds are expected to happen in two cases. In the best case only one node needs to rejoin to reconnected the whole branch leading to a minimum of state changes. In the worst case, all nodes rejoin on distinct paths and would lead to a maximum of state changes.

From the algorithmic point of view further steps are to compare the reconfigurational effort (e.g. forwarding of reconfigurational messages) of different network algorithms with the lower bound of mathematical necessary link changes. The goal is to check whether emerging phenomenons can be predicted with a measure between reconfigurational effort and necessary link changes inhereted from the underlying network and approximated with URTs.

A further application for branchsizes and depths of URTs comes with Information Centric Networking (ICN) [11]. In this context, the depth of the URT and its branches yields information about the path lengths on which interest messages are forwarded. The branchsizes yield information about the impact of flooding data requests in an ICN network. So the connection between properties in ICN networks and damage in URTs is a further step to investigate.

A. Notations

Notation	Meaning	Context
$P(\cdot)$	probability of an event $(\cdot)$	any
$E[\cdot]$	expected value of a random variable $[\cdot]$	any
N	number of vertices in a tree	any
h	max. height of a tree	any
d	depth of an edge or vertex	any
$X_h = d$	random variable X_h maps from a randomly chosen edge to its level d in a perfect k-ary tree of height h	k-ary tree
D, D_N	random variable representing the number of descendents for a randomly chosen node or edge in a tree with N nodes	any
$0 \leq h_s < h$	height of a seperated branch h_s of a tree of height h	k-ary tree
$Y_h = h_s$	random variable Y_h maps from a randomly removed edge to the height of the separated subtree in a perfect k-ary tree of height h	k-ary tree
$p_{k,i}$	probability, that in the k^{th} step of the generating process of an URT, the new vertex $k+1$ is connected to node i. The vertices i are the already existing nodes, so that $1 \leq i \leq k$	URT
L_d	set of all vertices of depth d (vertices of "Level d")	URT
U_N	random variable representing the degree of the root vertex in an URT of size N . Their number equals the size of set L_1	URT
$V_{d,N}$	random variable representing the number of nodes which are separated if an edge is removed in depth d in an URT of size N	URT
B	random variable representing the Polya-Eggenberger Distribution	URT
$C_{k,n}$	random variable representing the Polya-Eggenberger Distribution with $s = 1$	URT
$D_{k,N}$	random variable representing the number of descendents of node k in an URT with N nodes	URT
$X_{d,N}$	random variable representing the size of level d in an URT with N nodes	URT
Q_n	random variable representing the insertion depth of node n in an URT	URT
R_N	random variable representing the depth of a random node in an URT with N nodes	URT

References

[1] Martin Aigner. *Diskrete Mathematik.* Springer-Verlag, 2006.

[2] Michael Behringer and Geoff Huston. A Framework for Defining Network Complexity. Internet-Draft – work in progress 01, IETF, November 2013.

[3] Reuven Cohen and Shlomo Havlin. *Complex Networks: Structure, Robustness and Function.* Cambridge University Press, 2010.

[4] Michael Drmota. *Random Trees: An Interplay between Combinatorics and Probability.* Springer Science & Business Media, 2010.

[5] Qunqiang Feng, Chun Su, and Zhishui Hu. Branching Structure of Uniform Recursive Trees. *Science in China Series A: Mathematics*, 48(6):769–784, 2005.

[6] JL Gastwirth and PK Bhattacharya. Two Probability Models of Pyramid or Chain Letter schemes demonstrating that their promotional claims are unreliable. *Operations Research*, 32(3):527–536, 1984.

[7] NL Johnson and S Kotz. Urn Models and Their Application: An Approach to Modern Discrete Probability Theory, 1977.

[8] Srinivasan Keshav. *Mathematical Foundations of Computer Networking.* Addison-Wesley, 2012.

[9] Alois Panholzer and Helmut Prodinger. Level of Nodes in Increasing Trees Revisited. *Random Structures & Algorithms*, 31(2):203–226, 2007.

[10] D. Papadimitriou, P. Van Mieghem, K. Salamatian, and F. Griffths. D1. 5: Roadmap on developing the fundamental basis of autonomous and dynamic complex networks. 2014.

[11] Thomas C. Schmidt, Sebastian Wölke, Nora Berg, and Matthias Wählisch. Partial Adaptive Name Information in ICN: PANINI Routing Limits FIB Table Sizes. In *2nd ACM Conference on Information-Centric Networking (ICN 2015), Poster Session*, pages 193–194, New York, Oct. 2015. ACM.

[12] Chun Su, Qunqiang Feng, and Zhishui Hu. Uniform Recursive Trees: Branching Structure and Simple Random Downward Walk. *Journal of mathematical analysis and applications*, 315(1):225–243, 2006.

[13] Piet Van Mieghem. *Performance Analysis of Communications Networks and Systems.* Cambridge University Press, Cambridge, New York, 2006.

[14] Piet Van Mieghem, Gerard Hooghiemstra, and Remco van der Hofstad. A Scaling Law for the Hopcount in Internet. Technical report, Delft University of Technology, 2000.

[15] Piet Van Mieghem and Milena Janic. Stability of a Multicast Tree. In *21st Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM'02)*, volume 2, pages 1099–1108. IEEE, 2002.