 The definitive version of this paper has been published in The First International Workshop on Dependable and Sustainable Peer-to-Peer Systems in conjunction with The First International Conference on Availability, Reliability and Security, 2006, 10.1109\/ares.2006.58.

Efficient Link Failure Detection and Localization using P2P-Overlay Networks

Barbara Emmert *
Telecommunications Research Center Vienna (ftw.)
Donaucitystraße 1, 1220 Wien, Austria
emmert@ftw.at

Andreas Binzenhöfer
Department of Distributed Systems
University of Würzburg
Am Hubland, 97074 Würzburg, Germany
binzenhoefer@informatik.uni-wuerzburg.de

Abstract

Peer-to-Peer (P2P) networks offer a great potential that goes well beyond simple file-sharing. We present a novel approach for using P2P-overlay networks to ensure a sustainable operation of a distributed system. In particular, we show how to detect and localize the causes of physical link failures using the maintenance traffic of a P2P overlay network. The network monitoring architecture can be set up autonomously thereby reducing both the installation costs and the traffic overhead.

1. Introduction

Current computer networks consist of hundreds, thousands or even millions of nodes and links. It is a very difficult or even impossible task to guarantee a sustainable operation of such systems surveying the health of all physical links with only one single central network monitoring entity. The concept of *distributed network monitoring* involving several smaller distributed entities can be seen as a first step toward autonomic networks [7], which are able to diagnose and eliminate a number of failures autonomously. However, most existing distributed network monitoring proposals represent no great progress toward an autonomic network as all information is still gathered in a central place and has to be manually evaluated by an operator. In this context, the concept of P2P overlay network monitoring promises an improved level of autonomy. If peers in a network interact in order to combine their individual knowledge about small local areas of the system to a complete image of the whole network, the central monitoring entity can be greatly supported or may even no longer be needed.

A first proposition of a less centralistic network monitoring solution uses *probes* or test transactions to verify that all components of the system work properly [2]. A similar concept is presented in [3] where a minimum number of paths used to route packets between two nodes of a network is used to calculate the loss rate of every single routing path. In [5] it is shown that if a fraction of all nodes in the networks is used as *beacons*, i.e. as nodes that are able to survey all of their outgoing paths, this is sufficient for detecting all link failures. However, all those solutions suffer from the problem that they require additional configuration overhead and more administration work than does a single network monitoring station. Furthermore, they need to produce additional traffic to survey the network.

In our work, we propose to utilize the self-configuring properties of a P2P network to avoid unnecessary configuration overhead, while exploiting its maintenance overhead to detect and localize possible link failures. In general, any structured P2P overlay requires some maintenance traffic in order to stabilize the structure of the overlay. If, e.g., a distributed hash table (DHT) algorithm is deployed, all peers regularly contact about $O(\log_2(n))$ other peers in the overlay to guarantee the functionality of the system. In this paper we will analyze the potential of highly meshed overlay networks to support a provider or an operator in maintaining a sustainable operation of their system. Such overlay networks were, e.g., presented in [4], where every node maintained a complete routing table to enable lookups in $O(1)$.

The remainder of this work is organized as follows: In Section 2 we define the problems of detecting and localizing link failures using P2P overlays. Section 3 describes the concept of overlay network monitoring and explains how to detect link failures efficiently. We demonstrate how existing overlays can be extended, in order to pinpoint the root cause of a detected failure in Section 4, before we present results in Section 5 and conclude our work in Section 6.

2. Fundamentals

2.1. Theoretical Foundations

We represent an IP network by a connected graph $G = (V, E)$, where V is the set of nodes or vertices, E is the set of links or edges and $E \subseteq V \times V$. A *path* in G

*Research was done while the author was at the University of Würzburg

is defined by the set of its physical links. We assume, that through an OSPF-like routing policy the *shortest path* $p_{u,v}$ between two nodes u, v of G is uniquely determined. P denotes the *set of all shortest paths* in the network, $P := \{p_{u,v} : u, v \in V\}$. We assume also, that all links in the network are assigned equal or similar bandwidths and that every link is thus contained in at least one routing path.

In the following, we consider the situation, where only one link failure occurs at a specific time and we hence distinguish $L = |E|$ different failures. To detect and localize those link failures, we use a probe base approach inspired by the work of Brodie et al. [2]. For a network with a well ordered set of links $E = \{e_1, \ldots, e_L\}$, we define the *set of failures* as $F = \{f_1, \ldots, f_L\}$. If the physical link e_i is down, failure f_i *occurs*. In this context, a *probe* is a method of obtaining information about objects in the system. It is *affected by failure* f_i if its execution fails as soon as f_i occurs. Note that we can easily make this scheme suitable for the case of simultaneous failures, if we add additional failures f to F, whose occurrences indicate the concurrent failure of several links $E_S \subseteq E$.

In our work, we consider each packet routed via the unique path from a node u to another node v as a probe: Let p' be the route a packet with source u and destination v actually takes. The probe $p = p_{u,v} \in P$ *fails* as soon as p' differs from p. Thus, to "evaluate probe p" we need to compare the actual way a packet routed on p has taken to the predefined one in the routing table. The *dependency matrix* for P, $D_P \in \{0,1\}^{N^2 \times L}$, describes which physical links are used by which path or which failure affects which probe. For $p_j \in P$ and $1 \leq k \leq L$, we set

$$D_P(i,j) = \begin{cases} 1 & \text{if } e_j \in p_i \\ 0 & \text{otherwise } . \end{cases} \quad (1)$$

Using P as the set of probes, the relationship between the outcome of every probe and all possible failures is thus given by D_P, as $p_i \in P$ is affected by f_j if and only if $D_P(i,j) = 1$. If more than one failure can occur at a time, this can be modelled using a dependency matrix, where additional columns represent the occurrence of several simultaneous link failures. Those additional columns are computed by OR-ing the columns representing the individual link failures.

We focus on scenarios like campus or company LANs where the topology of the monitored network is known. Given the topology and the deployed routing algorithm, we have full knowledge of the routing table and can thus derive the set of shortest paths P which we use as the set of available probes.

2.2. MINIMUM SUCCESSFUL OVERLAY

We define the number of links that can be monitored using a given probe set as follows: $P' \subseteq P$ *covers* F, if for every $f \in F$ at least one probe in P' is affected, i.e. if

$$E = \bigcup_{p \in P'} p \Leftrightarrow \sum_{k=1}^{|P'|} D_{P'}(k,i) \geq 1 \text{ for } 1 \leq i \leq L. \quad (2)$$

The problems of determining the minimal probe set which is able to detect and localize all possible failures were introduced as the FAULT DETECTION and the FAULT LOCALIZATION decision problem in [2]. Both problems are NP-complete, but the authors presented heuristics for setting up appropriate minimal probe sets. Applied to the case of monitoring IP networks, those heuristics produce a very small set of probes, but require a quite large number of beacons. Moreover, every evaluation of a probe requires a packet to be sent through the network.

To minimize the additional load imposed on the system, we propose to exploit the signaling overhead of a fully meshed P2P overlay for monitoring purposes. In a network $G = (V, E)$ with the shortest paths given by P, we denote any $O \subseteq P$ as an *overlay*. The sources and destinations of paths in O are called *the base* V' of O, which *spans* the overlay: $V' = \{v \in V : \exists_{u \in V}\, p_{u,v} \in O \vee p_{v,u} \in O\}$.

For $u, v \in V'$, v is called *a neighbor* of u in O if $p_{u,v} \in O$, a neighborhood-relation that is not necessarily bidirectional. We define the *coverage* of an overlay O as

$$c_O = \frac{|\bigcup_{p \in O} p|}{|E|}. \quad (3)$$

An overlay $O \subseteq P$ is denoted as *successful* if it covers F, i.e. if $c_O = 1$. In case each node of the overlay is a neighbor of all other overlay nodes, we speak of a *fully connected* or *fully meshed overlay*, $O^f_{V'} = \{p_{u,v} \in P : u, v \in V'\}$. These definitions lead to the following decision problem:

Problem MINIMUM SUCCESSFUL OVERLAY
INSTANCE: *A network $G = (V, E)$ with the set of shortest paths P and a positive integer $n \leq |V|$.*
QUESTION: *Does there exist a $V' \subseteq V$ with $|V'| \leq n$ so that $O^f_{V'}$ is successful?*

Although this problem is not exactly equal to FAULT DETECTION it is in the same range of complexity. In Section 3.3, we therefore present heuristics to find small bases for successful overlays.

The probe set represented by a positive instance of the MINIMUM SUCCESSFUL OVERLAY problem bears many redundancies, since it is possible that $p_{u,v} = p_{v,u}$ or $p_{u,v} \subseteq p_{u,w}$. The problem of choosing a minimum number of probes out of a given $O^f_{V'}$ whose evaluation still allows the surveillance of the physical links is again NP-complete. In Section 3.4 we will introduce algorithms which greatly reduce the number of probes to evaluate and establish an *improved successful overlay* $O^i_{V'} \subset O^f_{V'}$.

2.3. MINIMUM EXTENDED PINPOINT OVERLAY

Using the probe set given by a successful overlay, we are able to *detect* that a failure has occurred, but in general we can not *pinpoint* the cause of the failure. To tackle this problem, we exploit the interdependency of the different probes and, following [2] we define: Given a set of failures $F = \{f_1, \ldots, f_L\}$, a probe set $P = \{p_1, \ldots, p_K\}$ and the dependency matrix D_P, the *signal vector* or *signal*, $s_i \in \mathbb{F}(2)^K$ of fault f_i is given by the ith column of D_P, where $\mathbb{F}(2) = (\{0,1\}, +, \cdot)$ denotes the Galois field. We call a failure $f_i \in F$ *identifiable* or *locatable* by P, if s_i is *unique*, i.e. if s_i is linear independent from all other signal vectors of faults in F. Two failures $f_i, f_j \in F$ are called *indistinguishable*, noted $f_i \sim f_j$, if their fault signals are equal. Two signals are linear independent if they differ in at least one entry, two failures are thus *distinguishable* if at least one probe of the probe set is affected by one of the failures but not by the other one.

Considering the required management overhead, we use the stabilization overhead in an improved overlay network for both detecting *and* localizing failures. In a network, where the shortest paths are given by P, the *exactness*, e_O, of an overlay $O \subseteq P$ in respect to the set of failures $F = \{f_1, f_2, \ldots, f_L\}$ is given by

$$e_O = \frac{|\{1 \leq i \leq L : f_i \text{ is identifyable}\}|}{L}. \qquad (4)$$

O is called a *pinpoint overlay* if it is successful and distinguishes all failures, i.e. if $c_O = e_O = 1$.

To set up an efficient solution for network monitoring, we extend existing successful overlay to a minimal failure localizing overlay by adding a minimal number of peers and paths to the overlay until we are able to pinpoint the cause of all failures. Therefore, we define the *extended overlay* $\text{Œ}^{\tilde{V}}_{V'}$ of an improved overlay $O^i_{V'}$ as the overlay which results from adding the nodes $v \in \tilde{V} \subseteq V \setminus V'$ to V' and paths $p \in \{O^f_{\tilde{V}} \cap P \setminus O^i_{V'}\}$ to $O^i_{V'}$. The corresponding decision problem reads as follows:

Problem MINIMUM EXTENDED PINPOINT OVERLAY
INSTANCE: *A network $G = (V, E)$ with the set of shortest paths P, an overlay $O^i_{V'}$ with base $V' \subseteq V$ which is not a pinpoint overlay and a positive integer $n \leq |V| - |V'|$.*
QUESTION: *Does there exist a $\tilde{V} \subseteq V \setminus V'$ with $|\tilde{V}| \leq n$ so that the extended overlay $\text{Œ}^{\tilde{V}}_{V'}$ is a pinpoint overlay?*

Intuitively, MINIMUM EXTENDED PINPOINT OVERLAY resembles FAULT LOCALIZATION introduced in the last subsection. We expect it to be in the same class of complexity and thus develop heuristics to find small extended pinpoint overlays in Section 4.2.

3. Detecting Link Failures

3.1. Random Overlays

In this subsection, we examine the coverage of *generalized overlays* $O^q_{V'}$, where a random set of peers $V' \subseteq V$ of size n is taken and q random neighbors out of V' are selected for each peer $v \in V'$. To obtain credible results, we repeat the experiment 2000 times for $1 \leq q < n \leq N$ and compute the coverage $c_{O^q_{V'}}$ for each resulting overlay.

During our studies, we analyzed different sample networks with $N \in \{6, 7, \ldots, 25\}$ nodes. Fig. 1 illustrates the results for the AT&T backbone network, which consists of 25 nodes. The upper surface in the figure visualizes the mean values of the coverage for all investigated overlay sizes and the corresponding number of neighbors. The lower surface shows the coefficient of variation of the coverage. The 95%-confidence intervals for all combinations of n and q range between 0.01% and 0.3% and are thus not shown.

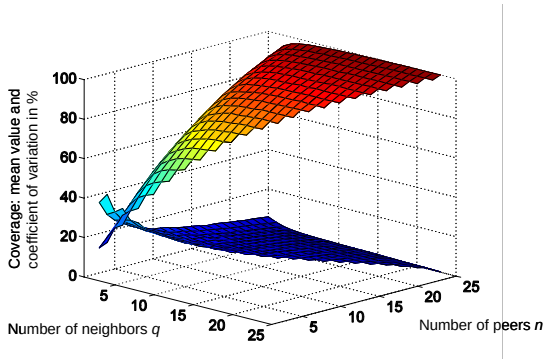

Figure 1. Analysis of random overlays

Generally it can be said, that we obtained different mean values of the coverage for the same relative size of the base $\frac{n}{N}$ in different networks. Obviously, the coverage depends on the particular structure of the network and we can hence not make a qualitative statement concerning the interdependence of $c_{O^q_{V'}}$ and $\frac{n}{N}$. However, the curves for the different networks roughly showed the same shape: The number of detectable failures is increasing with the size of the base as well as with the number of neighbors in the overlay.

The coefficient of variation in Fig. 1 increases with decreasing values of n and q, i.e. with smaller and less meshed overlays. That is, small and little meshed successful overlays do exist, but their percentage is too small to be found randomly. We develop therefore more sophisticated methods in the remainder of this section.

3.2. Brute Force Search

An examination of all existing overlays is another possibility to find efficient successful overlays, but does not

scale to larger networks. Therefore we use a Linear Programm (LP) to realize an efficient as possible brute force search for successful overlays. In particular, we consider a network $G = (V, E)$ with the set of shortest paths P and assume the set of vertices and the set of edges to be ordered in a well defined way. That is, $V = \{v_1, v_2, \ldots, v_N\}$ and $E = \{e_1, e_2, \ldots, e_L\}$. The LP to find a set of peers V' of size n which spans a successful fully connected overlay $O^f_{V'}$ can be described using the following three integer variables:

$\hat{e} \in \{0,1\}^L$ representing the physical links

$$\hat{e}(i) = \begin{cases} 1 & \text{if } e_i \in \bigcup_{p \in O^f_{V'}} p \\ 0 & \text{otherwise} \end{cases},$$

$\hat{v} \in \{0,1\}^N$ representing the nodes

$$\hat{v}(i) = \begin{cases} 1 & \text{if } v_i \in V' \\ 0 & \text{otherwise} \end{cases},$$

$\hat{n} \in \{0,1\}^{N \times N}$ describing the overlay structure

$$\hat{n}(i,j) = \begin{cases} 1 & \text{if } v_i \text{ is a neighbor of } v_j \\ 0 & \text{otherwise} \end{cases}.$$

Our target is to maximize the number of covered links. The objective function f is therefore given by

$$f(\hat{e}, \hat{v}, \hat{n}) = \sum_{i=1}^{L} \hat{e}(i). \tag{5}$$

The corresponding constraints must hold for $1 \leq i, j \leq N$:

At most n nodes may be chosen to be peers

$$\sum_{i=1}^{N} \hat{v}(i) \leq n. \tag{6}$$

Neighbors have to be peers in the overlay

$$\hat{n}(i,j) \leq \hat{v}(i) \text{ and } \hat{n}(i,j) \leq \hat{v}(j). \tag{7}$$

Overlay links cover physical links

$$\hat{e}(i) \leq \sum_{x,y: e_i \in p_{x,y}} \hat{n}(x,y). \tag{8}$$

The computation time and the overhead involved in finding a solution are too large to make the LP suitable for real-life networks, but CPLEX, enabled us to get results for all our sample networks with 6 to 25 nodes. Those results showed that efficient overlays offering full coverage do exist and can be obtained with a peer set of size $n \ll N$. Moreover, efficient heuristics can derived using the specific features of the obtained overlays, as will be described shortly.

3.3. Heuristics for Successful Overlays

For all sample networks G, we define the *importance* I_v of a node v in G as the percentage of the successful overlays found by a brute force search in which v occurred. If $I_v = 1$, v is a member of all successful bases found for this network. Any reasonable heuristic must therefore incorporate v in the base of an overlay. During our analysis we identified the following node characteristics to be correlated with I:

- T_v, the number of *transit flows* traversing v:

  $T_v = |\{p_{u,w} \in P : u, w \in V \backslash \{v\} \wedge \exists_{x \in V}\, vx \in p_{u,w}\}|$

- T^L_v, the number of *long transit flows* traversing v:

  $T^L_v = |\{p_{u,w} \in P : u, w \in V \backslash \{v\} \wedge \exists_{x \in V \backslash \{u,w\}}\, vx \in p_{u,w}\}|$

- M_v, the *marginality* of v: $M_v = \sum_{u \in V \backslash \{v\}} l_{p_{v,u}}$

- d_v the *degree* of v: $d_v = |\{u \in V : uv \in E\}|$,

where uv denotes the edge between the node u and v and l_p gives the length, i.e. the number of physical links of path p. Note that these characteristics allow to differentiate both the type and the topological position of a node.

To analyze which features are common for nodes with an I_v close to 1, we calculated the *correlation* $\rho_{I,X}$ for each nodal characteristic X introduced above. Since none of the correlations was equal to 0, all characteristics depend on I to some extent. In detail, T, T_L and d are positively and M is negatively correlated to I, while the correlation between T, T_L and I is stronger than the correlation between M, d and I. However, the results strongly depend on the network structure. The introduced parameters can therefore be used as indicators, but are not able to guarantee that a specific node is indeed necessary to span a successful overlay.

We utilize these correlations to derive four different heuristics, which are able to find a base for a fully meshed successful overlay. Each heuristics starts with $V' = V$, while for the heuristics h_T, h_{T^L} and h_d, V' is brought in a descending order according to T_v, T^L_v and d_v respectively. For the heuristics h_M, the nodes of V' are brought in an ascending order according to M_v. We also examine a heuristics h_R, that simply randomizes the order of the nodes. Once the set is arranged, the heuristics runs through the set of nodes and removes each considered node from the overlay, if the resulting overlay is still successful. We compared the results of all heuristics to the theoretical minimum base size determined by the LP for the small sample topologies. All heuristics performed nearly identical and matched the theoretical minimum in almost all cases.

Fig. 2 compares the sizes of the bases of successful overlays found by the different heuristics. It shows the results

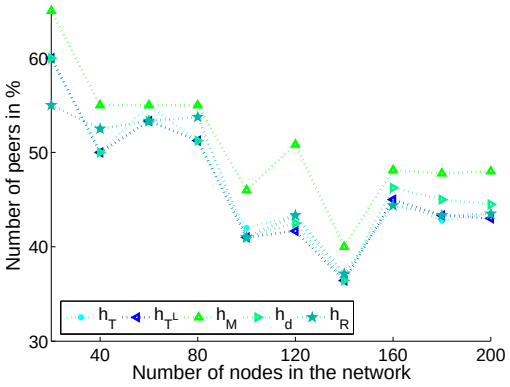

Figure 2. Comparison of successful overlays

for 10 different random topologies with $20, 40, \ldots, 200$ nodes (cf. Section 5 for details on the topologies). Due to the influence of topological characteristics, the percentage of the nodes in the network that need to be peers in the base, i.e. are either the source or the destination of a path used as probe, is varying quite strongly. In fact, we will show in Section 5 that the network topology has an even higher influence on the efficiency than the type of heuristics.

The running time of the heuristics is dominated by the time needed to initialize V'. h_R, h_d and h_M can be therefore performed in $O(N)$. The heuristics h_T and h_{T_L} arrange the nodes according to their occurrence in routing paths and need therefore a time in $O(N^2)$. A MATLAB implementation of all heuristics required a reasonable amount of time for a random topology with 200 nodes on a Pentium 1.7 Ghz CPU: The computations took less than 8 seconds for heuristics with time complexity of $O(N)$ and around 18 seconds for heuristics with a time complexity of $O(N^2)$.

3.4. Heuristics for Reducing the Overhead

As outlined earlier, the use all paths that are in a fully meshed overlay for failure detection, leads to a highly redundant probe set. In this subsection we develop four heuristics to minimize the number of paths needed to detect all possible failures. The heuristics can be summarized as follows:

- a_u (*add most useful*):
  In each step the *most useful* $p_i \in O^f_{V'} \setminus O^i_{V'}$ is determined and added to an initially empty $O^i_{V'}$, until this overlay is successful. The usefulness of a probe $u^+_{p_i}$ is defined as the number of links contained in the path p_i that are not yet covered by $O^i_{V'}$.

- a_r (*add random*):
  The initial empty set $O^i_{V'}$ is extended by randomly choosing a $p \in O^f_{V'} \setminus O^i_{V'}$. The probe p is only added to the overlay if it improves the coverage, i.e. if $u^+_p > 0$. This step is repeated until a successful overlay is found.

- d_u (*drop most useless*):
  We initialize $O^i_{V'} = O^f_{V'}$. Then at each step the *most useless* probe $p \in O^i_{V'}$ is dropped from the probe set, as long as the overlay remains successful. The more redundancy there is in a probe and the shorter it is, the more it is considered to be useless. We calculate the uselessness of a probe by $u^-(p_i) = \lambda_{p_i} \cdot e^{-l_{p_i}}$, where λ_{p_i} represents the *weighted length* of p_i for $O^i_{V'}$. Thereby for each link of a probe p_i the number of other probes in $O^i_{V'}$ which do also cover this link is calculated. All these numbers are then added up to obtain λ_{p_i}.

- d_r (*drop random*):
  We initialize $O^i_{V'} = O^f_{V'}$ and choose a random $p \in O^i_{V'}$. If $O^i_{V'} \setminus \{p\}$ is still successful, the probe is dropped from the overlay. This step is repeated until all probes have been considered or if each column of $O^i_{V'}$ has weight 1.

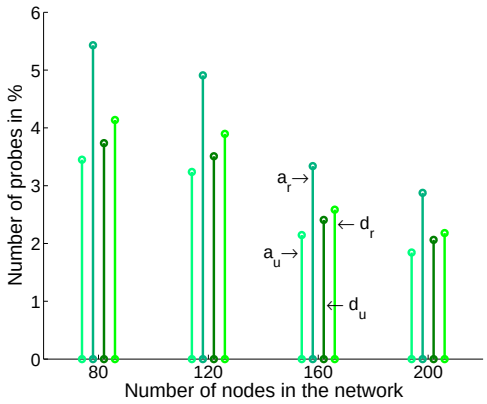

Figure 3. Probes needed for failure detection

In Fig. 3 we show the number of probes that are determined as necessary for failure detection by the different heuristics for some of the successful overlays analyzed in Fig. 2. All values are mean values of the five different fully meshed successful overlays in relation to the number of probes in the fully meshed overlay. Obviously less than 5 % of all paths from the fully meshed overlays are required as probes, thus, the computational effort can be cut down significantly.

The average time consumed by the heuristics to set up the improved overlays ranges between $O(N^2)$ and $O(N^4)$. However, we are able to reduce the time needed for optimization dramatically, if we start with an efficient fully meshed successful overlay. An evaluation of the combination of the heuristics on larger topologies generated according to different models can be found in Section 5.

4. Localizing Link Failures

Our experiments showed, that successful overlays already allow to localize the causes of a number of failures. In this section, we therefore examine how existing successful overlays can be extended to overlays that allow the localization of link failures as well. This two-level approach of network monitoring is also motivated by intuition: In a highly utilized network, it is important to detect any failure as fast as possible. Once a failure has been detected, additional effort can be used to pinpoint the cause of the failure.

We also noticed that, even if the signal of a failure is not unique, the set of possible causes is quite small: On average, only two or three physical link failures affect the same set of probes. This is especially interesting for the case of large networks where thousands of links are to be monitored. To manage medium sized networks, we show how to pinpoint the root cause of a failure exactly in the following.

4.1. The Localization Quality of a Probe Set

In Eq. (4) we introduced the exactness, e_O, of a given overlay O as a measure for the number of failures, which can be distinguished by O. In the following, we will present a more exact characterization to estimate the quality of a probe set, again based on the ideas of Brodie et al. [2].

For any $f \in F$ and in respect to $\sim$, the equivalence class $[f] := \{g \in F : g \sim f\}$ represents the set of failures which are not distinguishable from the specific failure f. The corresponding partition of F is called the *localization decomposition* S_P of P and is given by $S_P := F/_\sim = \{[f] : f \in F\}$. Note that for a pinpoint overlay O, S_O consists of singleton sets. To rate the quality of a probe set P, the *localization quality* Q_P with respect to a set of failures F is defined as the expected minimum number of additional probes required to localize all faults. If all failures are independent and occur with equal probability, Q_P is given by

$$Q_P = \sum_{S \in S_P} \frac{|S|}{|F|} \log_2 |S|. \tag{9}$$

Smaller values of Q_P denote better probe sets, as for each $S \in S_P$, $\log_2 |S|$ gives the minimum number of additional probes needed to distinguish all failures in S.

4.2. Extending Successful Overlays

We consider a network $G = (V, E)$ and a successful overlay $O^i_{V'}$, which is already able to pinpoint some, but not all failures. We investigate two approaches to find a failure localizing extension of this successful overlay: The first possibility is to determine a minimal set of probes $P' \subseteq P \setminus O^i_{V'}$, the second possibility is to determine a minimal set of peers $\tilde{V} \subseteq V \setminus V'$, which allows to extend the successful overlay to a pinpoint overlay.

Some more details:

- c^p_i (*choose most informative probe*):
  We initialize $P' = O^i_{V'}$ and add the *most informative* $p \in P \setminus P'$ to P' until the corresponding extended overlay is a pinpoint overlay. Then $\tilde{V} \subseteq V \setminus V'$ is derived as the base of $\mathfrak{E}^{\tilde{V}}_{V'}$. A probe p is considered the more informative, the more it improves the localization quality of P'. Consequently, we define the *entropy* of a probe as $j_p = Q_{P'} - Q_{P' \cup \{p\}}$. According to Eq. (9) lower values of Q_P correspond to better probe sets. In each step, we therefore add the probe $p \in P$ with the highest j_p.

- c^p_r (*choose random probe*):
  We start with $P' = O^f_{V'}$, choose a random probe out of $P \setminus P'$ and add it to P' if it improves the localization quality. That is, we add it to P' if its entropy is not zero. This step is repeated until P' allows to detect and localize all failures and $\tilde{V}$ is constructed as the union of all endpoints of added probes.

- c^b_i (*choose most informative beacon*):
  We initialize $\tilde{V} = \emptyset$ and iteratively add the *most informative* beacon $v \in V \setminus \{V' \cap \tilde{V}\}$ to $\tilde{V}$ until $\mathfrak{E}^{\tilde{V}}_{V'}$ is a pinpoint overlay. The quality of v is rated in dependence of the entropy of all paths with source v: $j_v = Q_{P'} - Q_{P' \cup T_P(v)}$ with the shortest paths spanning tree rooted at v, $T_P(v)$. Again, larger values of j_v indicate more informative nodes. Thus, at each step we add the node v with the highest j_v. After adding a node v to $\tilde{V}$, we only add those probes $p \in T_P(v)$ to the overlay which improve the localization quality, i.e. those with $j_p > 0$.

- c^b_r (*choose random beacon*):
  We start with $\tilde{V} = \emptyset$ and add random nodes $v \in V \setminus V'$ to $\tilde{V}$ as long as $\mathfrak{E}^{\tilde{V}}_{V'}$ is not a pinpoint overlay. In analogy to c^b_i, we include $p \in T_P(v)$ with $j_p > 0$ in $\mathfrak{E}^{\tilde{V}}_{V'}$ as soon as v is added to $\tilde{V}$.

The time complexities of the different heuristics lie between $O(N^2)$ and $O(N^4)$. Similar to the optimization of fully meshed overlays, we achieve smaller running times due to the preparatory work of setting up successful overlays. In the next section, we will evaluate our heuristics more thoroughly.

5. Evaluation and Results

Our experiments showed that the network topology has a strong influence on the quantity of nodes and probes needed to detect and localize link failures. We therefore investigate the performance of the heuristics on different topology

types and used the topology generator BRITE [6] to create several random graphs with $20, 40, \ldots, 200$ nodes. We created the topologies according to two different models: One group of graphs was built as flat router-level Waxman topologies [9], the other one following the ideas of Barabási and Albert [1]. The first type of graphs is produced by randomly placing nodes in a given area, then connecting any two nodes with a probability that is inversely proportional to the Euclidean distance between them. The second model aims at creating Internet like topologies. It suggest to simulate the evolution of the Internet by adding nodes to an initially empty topology until the desired number of nodes is reached. The probability that a newly added node is connected to an already present node v, is directly proportional to the degree of v. This is based on the observation, that nodes joining a network tend to connect to nodes which already have a good connectivity, a behavior that leads to a power-law degree distribution.

The most striking difference between those topologies lies thus in the degree distributions. For Barabási-Albert topologies, it is significantly more heavy tailed than for Waxman topologies: Considering topologies with 200 nodes, we observed maximal degrees of 45 and 18 respectively for nearly the same average nodal degree of around 4. Therefore, the routing paths in the Barabási-Albert topologies are on average shorter than those of the Waxman topologies. This has of course an influence on the number of probes needed for network monitoring: if the average path length is shorter, more paths will be needed to detect all possible failures.

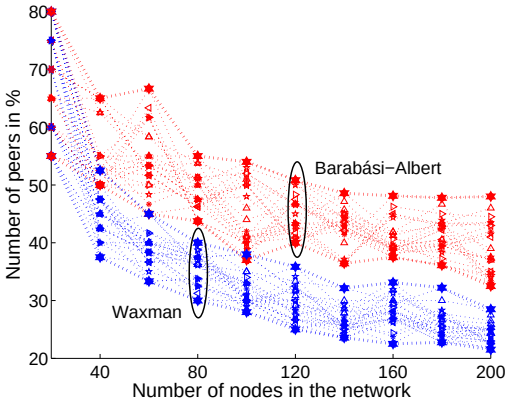

Figure 4. Failure detection - peers

But even for two topologies emerging from the same model the distributions are different. This can be seen in Figure 4 where we show the number of peers needed to span a successful overlay found by the different heuristics, distinguished by the markers introduced in Fig. 2. We show results for 100 different topologies, i.e. 5 different topologies for every type and every considered network size. We indicate the convex hulls of all results for one topology type too, to illustrate that the topology structure has a very strong influence on the outcome of the heuristics. As no specific heuristics is better suited for one type of topology than for another one, we use therefore ten different random topologies generated as Barabási-Albert graphs in the remainder of this section.

In Fig. 5 we illustrate how many probes are needed to detect one link failure if the redundancies in the successful fully meshed overlays are reduced by the heuristics introduced in Section 3.4. We show mean values of the five different fully meshed successful overlays determined by the heuristics presented in Section 3.3 and indicate the 95%-confidence intervals. If we used all probes that are in the fully meshed overlays for failure detection, this would result in an average relation of 19 probes per possible failure for the topology with 200 nodes. The reduction of the overhead allows to cut down this relation dramatically and the use of one probe enables on average the detection of 3 failures in the best case. Furthermore, we can also reduce the administration costs, as a significant number of peers in the base is no longer required to evaluate the outcome of probes.

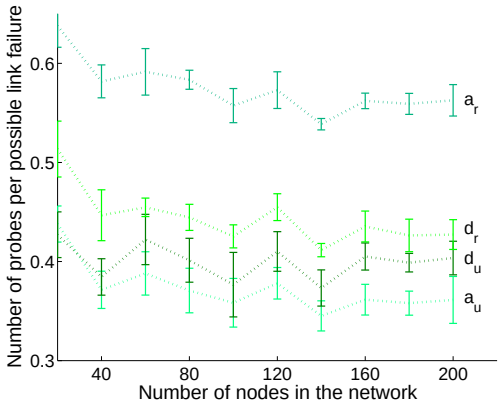

Figure 5. Failure detection - probes

In Fig. 6 we compare some pinpoint overlays obtained by the extension of successful overlays with the heuristics discussed in Section 4.2. At first we examine the number of required probes by looking at the pinpoint overlays emerged from the successful overlays with the smallest number of probes, that is indicated by the curve labelled "$O^i_{V'}$". The number of probes needed for failure detection and localization using the most efficient heuristics proposed in [2] is labelled "pure c^p_i", the other curves represent the number of probes in the pinpoint overlays resulting from the extension of the successful overlay. We see that on the one hand, an extension using c^p_i results in a way smaller number of probes than the use of the other heuristics. On the other hand, our two-level approach needs more probes for failure localization than the pure probe based approach. However,

we have to consider that using this method, *all* probes in the resulting probe set have to be checked regularly. Our method allows to cut the periodic evaluation of probes down to the set of probes represented by the paths in the improved overlay and to evaluate the additional probes only in the case of a failure.

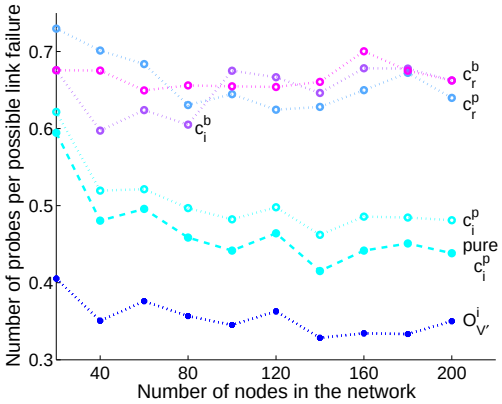

Figure 6. Failure localization - probes

We compare the number of beacons needed for the different pinpoint overlays developed from the successful overlays with the smallest number of beacons in Fig. 7. We omit the results for the pinpoint overlays created by c_r^p as the number of required beacons is unreasonably high, but the trade-off between the required number of probes and beacons needed for the localization of link failures is nevertheless evident: the overlays resulting from the use of c_i^p represent an efficient *probe* set, but require a high number of beacons. Thus, if we want to reduce the administration overhead and are willing to accept a higher number of probes, the use of c_r^b or c_i^b leads to efficient solutions, as in general less than half the nodes in the network have to be set up as beacons and an even smaller number of nodes has to perform frequent checks to enable failure detection.

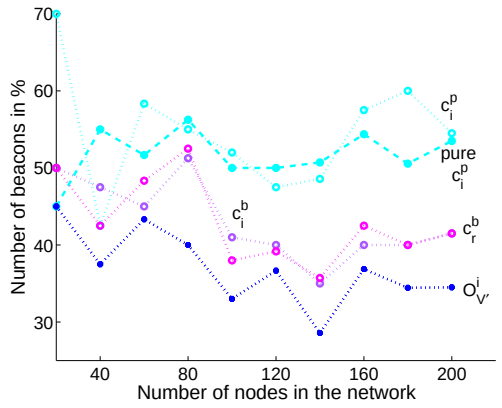

Figure 7. Failure localization - beacons

6. Conclusion and Outlook

In this paper we investigated the potential P2P overlay networks may contribute to the sustainable operation and maintenance of IP networks. We analyzed to what extent fully meshed P2P overlays can be used for efficient link failure detection and localization. Using a Linear Program for smaller networks it was shown that an autonomous surveillance can already be obtained with relatively few overlay peers. We derived different heuristics which scale to larger networks and achieved results close to the theoretical optimum. We believe that our results can be regarded as a further step toward the next level of autonomy on our journey to autonomic networks.

The P2P architecture has become ubiquitous in current computer networks, we will therefore examine in how far the stabilization overhead caused by specific DHT-based P2P networks like Chord [8] can be exploited to detect and localize not only single, but also simultaneous link failures.

Acknowledgments

The authors would like to thank Dirk Staehle and Phuoc Tran-Gia for their much appreciated input. This work has been partially supported by the Austrian Kplus funding programme.

References

[1] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, Volume 286, October 1999.

[2] M. Brodie, I. Rish, S. Ma, A. Beygelzimer, and N. Odintsova. Strategies for Problem Determination using Probing. Technical report, IBM T.J. Watson Research Center, 2002.

[3] Y. Chen, D. Bindel, H. Song, and R. Katz. An Algebraic Approach to Practical and Scalable Overlay Network Monitoring. *ACM SIGCOMM Computer Communication Review*, 34(4), 2004.

[4] A. Gupta, B. Liskov, and R. Rodrigues. One Hop Lookups for Peer-to-Peer Overlays. In *HotOS-IX 2003*, Lihue, Hawaii, 2003.

[5] J. Horton and A. López-Ortiz. On the Number of Distributed Measurement Points for Network Tomography. In *IMC 2003*, Miami Beach, Florida, USA, October 2003.

[6] A. Medina, I. Matta, and J. Byers. BRITE: An Approach to Universal Topology Generation. In *MASCOTS 2001*, Cinncinnati, Ohio, USA, August 2001.

[7] R. Murch. *Autonomic Computing*. Prentice Hall, Upper Saddle River, New Jersey, USA, 2004.

[8] I. Stoica, R. Morris, D. Karger, M. F. Kasshoek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *SIGCOMM 2001*, San Diego, California, USA, August 2001.

[9] B. Waxman. Routing of Multipoint Connections. *IEEE Journal of Selected Areas in Communication*, 6(9), December 1988.