Critical-Mass of a Distributed End-System Monitoring Service

Michael Finkenzeller, Gerald Kunzmann, Andreas Kirstädter, Rüdiger Schollmeier

***Abstract*— The Peer-to-Peer paradigm to utilize resources at the edge of a network offers a wide area of application scenarios based on services like file sharing and GRID computing. In this paper, we employ the P2P paradigm to provide a monitoring service for network statistics to end-users, network operators or interested applications. The basic principle is that end-user devices monitor local observable data e.g. bandwidth consumption and packet loss. This data is then organized in a P2P-based distributed directory.**

In this paper we estimate the potential of such a service by applying it to a simple topology discovery problem. The question we answer is the number of peers and connections needed to discover a certain percentage of network topology and how long this will take.

To answer that question, a simulator was implemented that is composed of different abstraction layers. It allows estimating a critical mass of peers that is necessary to cover a given percentage of random network topologies.

***Index Terms*— Peer-to-Peer, End-System, Monitoring, Topology Discovery, Network Management**

I. Introduction

Network status information is essential to network operators. Numerous solutions exist for either active probing or passive monitoring. The CAIDA [1] and NLANR [2] websites give a good overview on the different approaches. The main drawback of most solutions is the amount of monitored data being limited by storage capacity of central entities that collect and process that data.

Another limiting factor is the fact that the monitoring data has to be collected somehow and therefore has to be transported via the network being monitored. Thus, additional overhead is generated even if there is no active probing involved.

M. Finkenzeller and A. Kirstädter are with Corporate Technology Information & Communication, Siemens AG, Munich, Germany email: michael.finkenzeller / andreas.kirstaedter@siemens.com

G. Kunzmann is with the Institute of Communication Networks, Munich University of Technology, Munich, Germany email: gerald.kunzmann@tum.de

R. Schollmeier is with BMW AG, Munich Germany email: ruediger.schollmeier@bmw.de

Network operators always query their network equipment for performance statistics on throughput and dropped packets [3]. However, obtaining true end-to-end statistics spanning multiple autonomous systems is difficult. In a customer-centric market however it is essential for network operators and service providers to know the actual performance of a network connection end-to-end as it is experienced by their customers. End-to-end statistics can give greater and more up-to-date insight into the current network usage by the customers - an information resource of great value to network operators wanting to optimize the network performance.

To gather end-to-end performance statistics companies like Keynote [4] offer a measurement service built on top of a global infrastructure of geographically distributed probes.

The end-to-end statistics are getting more and more important also for end-users to decide which network operator to choose.

The idea of a Distributed End-System Monitoring Services (DEMS) is rather simple: Take locally observable data, analyze them, and publish the results via a Peer-to-Peer (P2P) framework.

Locally observable data may be everything from packet statistics like throughput and jitter to detailed information extracted via deep-packet inspection as used in personal firewalls.

The main difference of such a monitoring service organizing the collected data in a P2P information space compared to server-based approaches is its almost infinite storage and processing capacity. This huge distributed information space offers the possibility to collect monitored data over long periods of time facilitating the identification of trends and predictions like traffic growth or the early identification of bottlenecks. Looking at file-sharing networks like KaZaA [5] or eDonkey [6] with terabytes of stored files gives an idea on that.

As such it is expected that queries on performance data need to be answered with up-to-date information. Using end-system network monitoring, popular servers and content is automatically monitored more often and from different access networks, distributed worldwide.

To aggregate and analyze locally monitored data,

mechanisms used for network tomography can be utilized. One major challenge of a DEMS is to combine correlation methods from network tomography with the distributed nature of the P2P organized data storage.

Another major challenge is the deployment of such a system. A DEMS can offer the user an added value like always finding the best source or route for a desired content as complementary component in content delivery networks (CDN)s like Akamai [7]. A DEMS can also be deployed as an IT policy in a controlled end-system environment like company intranets to reduce network management costs.

One of the first questions to ask however is how accurate and up to date is the information offered by such a system. To start with, we chose the case of topology discovery. For very large scale networks like the internet, this is an almost impossible task. In intranet scenarios however our simulation results give a rough estimation what to expect from a DEMS.

A simulator, which places DEMS peers around the edges of random network topologies, was developed as the basis for our studies. For the first step, we investigate the critical mass for simple topology discovery, i.e. the minimum number of clients/peers, necessary to discover more than 90% of the network.

The document is structured as follows. Starting with related work, an overview on the architecture of DEMS is presented in section III. Section IV introduces the simulation environment to investigate the critical mass for topology discovery. Simulation results are presented and interpreted in Section V. An outlook on next steps planed for the simulation tool follows in section VI. We conclude with some remarks on application scenarios and deployment strategies of a DEMS.

II. Related Work

DEMS is closely related to M-coop [8] [9]. M-coop estimates distance metrics like latency, hop-count etc. These distance metrics are published on a P2P overlay. Most concepts of hierarchical organization of responsibilities, trust and accuracy are also applicable to DEMS. In contrast to M-coop DEMS is more focused on passive measurements. Also DEMS does not assume prior topology knowledge but collects this information on-demand. M-coop is a general measurement and monitoring infrastructure whereas DEMS at the moment is more content oriented. This means that DEMS wants to find the best source or route for a desired content under varying network conditions. In conjunction with dynamically constructed virtual private networks (VPN)s or proxy services at end-systems this information allows an indirect access to desired data.

The process of inferring the internal network information from end-to-end measurements on the borders is commonly denoted as network tomography. Interesting information detected by network tomography methods comprises not only the network topology itself but also the congestion status of network links and the current availability and load of the nodes in the network. Network tomography is of special interest in cases where information about internal network elements is either hidden by the network operator or simply not available to customers running traffic over the considered networks. Generally spoken, network tomography consists of two main steps: Data collection and data correlation to extract the requested network information. Castro and Coates give an overview on network tomography techniques in their papers [10][11]. It is planned to utilize these network tomography methods in future versions of DEMS for topology and congestion discovery.

III. DEMS-ARCHITECTURE

This section gives a brief overview on the architecture of a DEMS. Although our simulation results deal with discovery of network topology, DEMS is planned to collect and correlate statistics of different types depending on the application scenarios.

The first information to get is the network topology itself. Based on this topology information, loss and delay characteristics of single network elements can be determined by correlating subsequent local measurements with measurements from remote DEMS peers.

To avoid overhead, the only active monitoring we plan to use are well known route discovery methods like traceroute or the ICMP record route option.

All other information is gained by passively monitoring packets. Passively observing the end-to-end mechanics in the network has the advantages of no additional bandwidth consumption. Also, the network operation is not disturbed by probe packets. This normally pays off the less accurate data collection process resulting from the fact that we are now limited to the experiences of the ongoing data traffic. Only aggregated statistics can be collected from these operations and a lot of data has to be gathered to be able to calculate useful data. Another important point is the fact that passive measurements are always strongly correlated with the directions of the traffic flow. This implies that we will get more monitoring information on those network regions that are used more heavily: The measurement matches the usage of the network.

The DEMS architecture consists of the four components shown in Figure 1. Based on this architecture, a prototype for demonstrating the DEMS principles was already implemented.

A *local monitoring module* sniffs the network traffic

to and from the localhost in a way similar to personal firewall tools. The level of inspection may vary: It reaches from simply parsing pairs of source and destination IP addresses for calculating the throughput up to protocol data at application level.

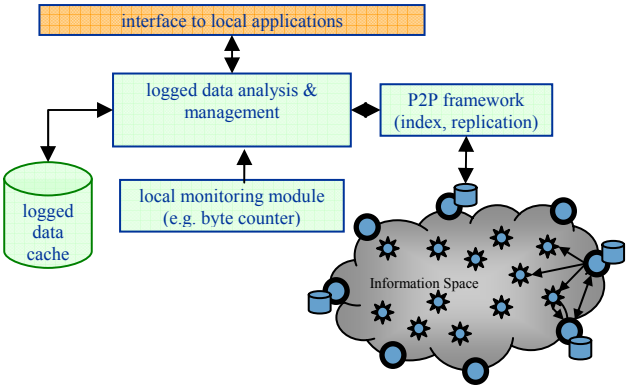

Figure 1 DEMS components

DEMS tries to avoid as far as possible active measurements that cause overhead by inserting probe packets periodically. The only overhead that is introduced by DEMS is a traceroute to discover as far as possible the topology of the monitored route. At a minimum in a very restrictive environment, the topology information is reduced to the server address of an established connection and a destination IP. This address is either the DEMS hosts IP or it is extracted from the local network configuration e.g. default gateway or web proxy.

Every monitored data is aggregated and stored in a *logged data cache*. A unique data set is identified by a source / destination IP pair. This is also the key for publishing the availability of a data set over a P2P framework. Other key combinations have to be examined with respect to the type of analysis intended and optimized inter-working with the P2P framework. For example it is also possible to use content-related keys like file-names or URLs and combine performance statistics. The granularity of observation is adjustable by the *local monitoring module*.

For the demonstrator, we currently use the P2P framework JXTA [12]. It is planed to integrate DEMS with advanced (structured) P2P frameworks like Chord [13] or Kademlia [14]. It will also be necessary to introduce different levels of data aggregation to implement a hierarchy. For example a first level of aggregation is typically a domain of hosts using the same default gateway. A next level could be the internet gateway of a company and so on. As a further requirement the P2P framework has to support an implicit replication mechanism to cope with the dynamic nature of distributed directories. The Siemens Resource Management Framework (RMF) [15] supports DHTs as well as replication and is therefore an interesting candidate.

The core component for *logged data analysis and management* conducts the correlation of monitored data and requests additional data sets from the P2P information space if needed. It also offers an *interface to local applications* that want to use network status information to adapt to changing network conditions.

As an example, a video-streaming application encounters low network performance and queries the local instance of DEMS for a better connection. DEMS will ask via the P2P framework for peers that perceive a better connection. By using the servers IP address as index, a structured P2P framework will route the request to peers which had recently used the same server with better download statistics. In the case of a positive answer, the better route of a remote peer could then be used by end-system based routing or by using a different access network or gateway.

IV. SIMULATION ENVIRONMENT AND IMPLEMENTATION

We evaluate the performance of the DEMS architecture described above via simulation. Our major goal is to determine the number of links used by the clients establishing their connections to a server at random instants. Thus, we can evaluate true client-server scenarios, where the number of servers is significantly smaller than the number of peers. Additionally, we can also evaluate P2P scenarios where the number of servers simply equals the number of clients - as in a P2P network every peer acts as a server as well as a client. Therefore, we implemented a dedicated simulator as existing solutions like e.g. the ns-2 [16] do not scale well to large network sizes. Basically, we introduce a 4-layer model to reduce the complexity of the simulation. As indicated by Figure 2 it consists of four different layers of abstraction, i.e. the physical layer L_{phys}, the routing layer L_{OSPF}, the layer describing the maximum number of detectable links L_{max}, and the layer describing the number of detected links, L_{detect}.

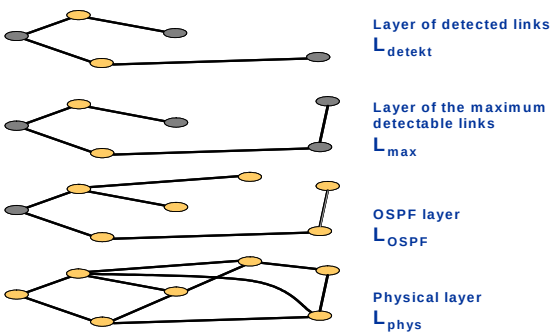

Figure 2 4-layer-simulation model

L_{phys} describes the fundament of every network, i.e. the physical structure and the properties of the network. Within this work we employ the topology generator Brite [17] based on a hierarchical Waxman model [18] to generate realistic physical scenarios. Thus, we are able to imitate closely the basic properties of real networks such as the hierarchical layout of different autonomous systems and the heavy-tailed connectivity distribution of the physical nodes.

In all of our simulations, the physical network consists of 1000 routers connected via the model described above. A certain percentage of these routers act as access routers to which a varying number of clients and servers are connected. The minimum number of routers and servers connected to one access router is at least one.

For the distribution of servers and clients to the different autonomous systems we employ three different distributions, namely a uniform distribution, and two Pareto distributions, the Pareto distribution being defined as follows:

$$p(x) = \alpha k^{\alpha} x^{-(\alpha+1)}$$

The two distributions are thus determined by the values of α and k $\left((\alpha_1 = 1.2; k_1 = 2), (\alpha_2 = 5; k_2 = 1)\right)$.

The percentage of DEMS clients located in the autonomous system with the ID ASID is accordingly computed to

$$D_{DEMS}(ASID) = \frac{p(ASID + k)}{\sum_{ASID} p(ASID + k)}$$

And the number of servers is computed to

$$D_{server}(ASID) = D_{DEMS}(Max\_number\_AS - ASID)$$

With the Pareto distribution described above we thus are able to simulate autonomous systems with an above average number of servers and a small number of DEMS clients and vice versa.

All in all up to 5027 links were established between the routers leading to a rather complex topology, which is abstracted further on by the OSPF layer described below. To derive meaningful results, we executed each simulation at least 100 times with different randomly generated physical topologies and randomly distributed clients and servers.

In our simulation, we assume that every router has the complete knowledge about the network topology right from the start, to avoid transient periods in the physical abstraction layer of our simulation. Thus a DEMS-client can detect within this simulation the path of the packets through the network in order to link the measurements to this route. Therefore, it must determine actively the route with a traceroute technique before it can execute its passive measurements (e.g., bandwidth) with adequate methods.

L_{OSPF} describes the routing layer of the analyzed network. Here we have to take into account that not all physical links are used to transmit packets. Routing protocols like OSPF (Open Shortest Path First) regulate and manage the flow of traffic. Some become real highways whereas others stay simply unused. Even though different routing protocols are used in real networks, we simplify matters in our simulations and narrow upon the OSPF protocol.

The routing layer is thus determined by all shortest path trees of every router in the network. Some links – mainly links with high costs, i.e. low bandwidth - will not be part of this layer and therefore will not be used for packet routing. These links thus can be neglected in the further abstraction layers as we expect route and cost changes to occur only very rarely within our simulation time[19][20][21].

Instead of configuring a single routing table for every node, we build up a routing matrix for every autonomous system. To send a packet from A to B we have to lookup the entry (A; B) in this matrix to identify the next hop for the packet. The complete algorithm is as follows:

1. Lookup the corresponding access routers (AR) for the source and target peer. (s: AR source; t: AR target)
2. Lookup next hop h := entry at matrix position (s; t)
3. If h = t then routing is finished.
4. If h ≠ t then lookup next hop h := (h; t). Go on with 3.

If source and target peer are located in different autonomous systems then the routing algorithm is getting a bit more complicated. But it is still basically executed in a similar manner. However, establishing the routing matrix results in a further abstracted view of the network where the number of links to be evaluated is smaller than the number of links in the physical layer. In average the OSPF layer only described 88.4% of all the links in the physical layer in our simulations.

To execute measurements in the network some or all nodes in the network must understand the measurement protocol and run the DEMS measurement service. The DEMS-service operates within the clients at the edge of the networks and not within routers and servers within the network. Therefore, it is only possible to measure

connections ending at a DEMS-client.

To evaluate the maximum number of detectable links, we use an algorithm, which is very similar to the routing algorithm described above. We systematically establish all possible connections from every DEMS client to every server in the network. According to the routes used (being defined by the OSPF and the network layer) we thus mark every used link in a L_{max} matrix describing all possible links from the OSPF layer. As a result, the more DEMS-clients and the more servers exist in the network the more links can be detected.

L_{max} represents the maximum value of detectable links for a certain arrangement of clients and peers. We need this value for the interpretation of our simulation results. L_{max}(100% DEMS) means, that all clients are DEMS compatible. The value for the maximum number of detectable links thus equals the number of the OSPF links if all clients in the network are DEMS clients. It therefore represents the maximum number of detectable links for a certain arrangement of clients and peers.

The top layer L_{detect}, depicted in Figure 2 describes the actual number of links detected in a time interval *t*. In the ideal case it is identical to the maximum number of detectable links L_{max}. As the clients establish their connections not all at once, but along the time, the value L_{detect}, depends on the duration of the simulation. The longer the simulation runs, the closer we get to the maximal limit L_{max}:

$$L_{\det ect}(t) \xrightarrow[t\to\infty]{} L_{\max}$$

To estimate the duration of the topology discovery, we have to take into account a transient phase. We want to know how long it will take, when the service is started initially to cover a certain percentage of the topology In this transient period definitively a significantly longer interval is needed to cover a great percentage of the topology, than in the stable case where our DEMS architecture has to cope with leaving and new joining nodes. Assuming a reliable P2P framework with a implicit replication strategy, the fluctuation of nodes in a fixed network scenario only causes minor difficulties. Their measurement results are available until they expire. An appropriate expiration strategy in alignment with route stability in fixed networks has to be validated.

To simulate a more realistic access scheme of clients to servers we execute the same algorithm as above. However, this time we do not systematically establish all possible connections but employ a random generator. Of course, we have to use the same topology and arrangement of the peers to compare our simulation results afterwards.

V. RESULTS

First, we investigate the maximum number L_{max} of detectable links in general networks. Assuming that every access router possesses at least one DEMS enabled server and one DEMS enabled client we achieve the same result by marking all possible routes between the access routers ($L_{max} = L_{OSPF}$).

A network with less than two access routers will not have any detectable links (we consider only links between two routers and not between a client/server and its access router). The more access routers exist the more links in the core network can be detected.

As depicted in Figure 3, L_{max} depends greatly on the number of available access routers (ARs). This is not surprising as DEMS clients can only be attached to ARs. It means that in the case where only a small percentage of all routers are ARs also only a small part of the network can be monitored by DEMS peers. However such a network with only a small amount of access routers and a comparably high number of core routers is usually over dimensioned.

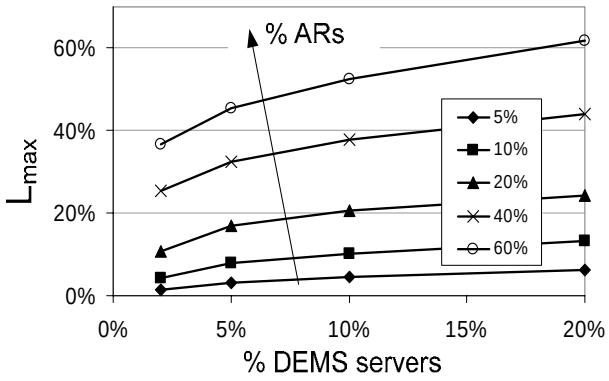

Figure 3 Devlopment of the number of maximal detectable links for different access router occurence rates against the occurrence rate of servers at one AR (occurrence rate of DEMS clients: 100%)

Figure 4 shows the results obtained by leaving the number of ARs constant at e.g. a level of 30% of all routers while at the same time varying the number of DEMS clients from 0% to 100% and the number of servers from 0% to 100%. This represents the scenario where the network is over dimensioned, i.e. that not all available links at the OSPF layer are used. Further on, it also proves the fact that a P2P network where every node acts as a client as well as a server is best suited to detect the highest number of links within the network. This is indicated by the point where the percentage of servers equals 100% and the number of clients equals 100%. In any other typical client server scenario, e.g. where 10% of the nodes represent a server, a significantly lower amount of links can be detected for any percentage of clients participating in the network.

Thus our approach, to employ a P2P network to monitor and to distribute the measurement data is reasonable from our point, as it shows the best performance according to the number of detected links.

This result is underlined additionally by the graphs depicted in Figure 5. Here the percentage of the maximum number of detectable links is given for a varying number of servers against the percentage of available access routers. The DEMS client percentage in this figure is set to 20%. The low occurrence rate of clients is the reason why the DEMS clients cannot detect all available links in the OSPF layer. Thus, we can conclude that it is more important to increase the number of client peers than the number of serving peers as only the client peers are able to monitor the network passively.

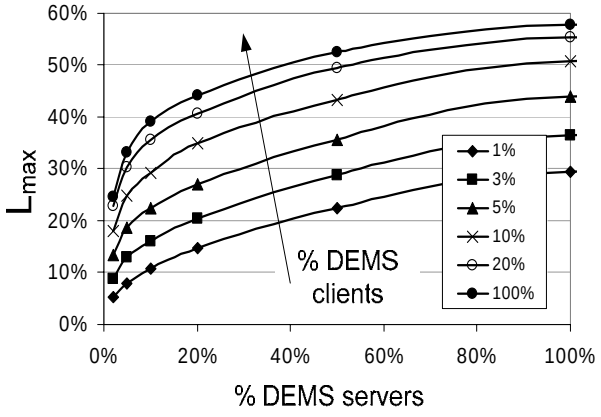

Figure 4 Development of the number of maximal detectable links for different DEMS client occurence rates against the occurrence rate of servers at one AR (occurrence rate of an AR: 30%)

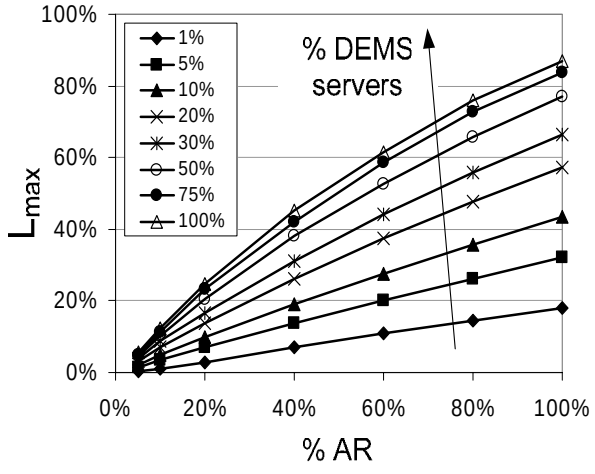

Figure 5 Development of the number of maximal detectable links for different DEMS server occurence rates against the occurrence rate of ARs (client occurrence rate: 20%)

In all of the figures above the clients and servers are distributed uniformly among the access routers. To evaluate our approach also in another scenario, we distribute the clients and the servers according to a Pareto distribution, as introduced in section IV. The distribution within one autonomous system is still equal, but we thus obtain some autonomous systems with an above average number of servers or clients.

We can observe in Figure 6 that the number of detectable links significantly decreases with an increasing accumulation of clients and servers on certain but different autonomous systems. As the number of access routers is constant in this simulation scenario, the clients and servers are far more distributed across the network if we apply a uniform distribution. Thus, we can reach a better detect-ability of the OSPF links.

If the servers are accumulated in a few autonomous systems then some access routers have to carry more servers than others but the number of access routers decreases to which servers are connected. The same is true for the clients, as the same Pareto distribution is applied to them. As a consequence less different client-server connections exist resulting in a decreasing size of the L_{max} layer. However, in this case we can again conclude that the considered network is simply over dimensioned as most of the links are not used at all by any user traffic.

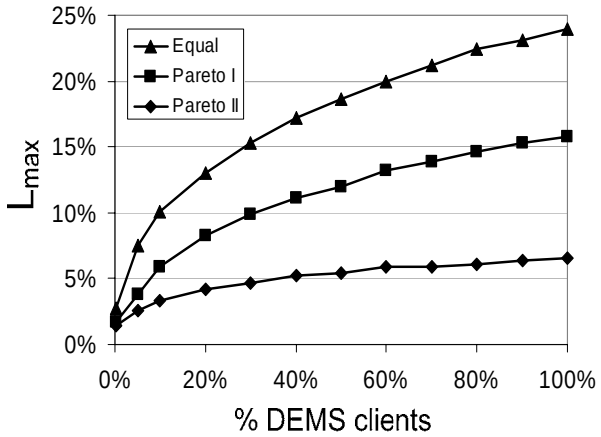

Figure 6 Development of the number of maximal detectable links for different server and client distributions against the occurrence rate of clientss (server occurrence rate 5%)

With above figures we are able to show how the number of access routers, DEMS clients, and servers affects the best case performance as we only investigated the maximum number of detectable links. However, to prove the possibilities of our approach in realistic scenarios we have to simulate the setup of connections between the clients and the servers. Therefore, we cannot assume any further a systematic setup of the connections between the clients and the servers. Rather, we have to assume a random behaviour of the clients.

As mentioned in section IV we simply have to

execute the same algorithm as above to simulate the detected links. However, this time we do not systematically establish all possible connections but employ a random generator. Of course, we have to use the same topology and arrangement of the peers to compare our simulation results afterwards. We also want to have a look at the setup time of such a system, i.e. assume that at the beginning of the simulation no measurement data is available and no links are detected so far. Thus, with a growing number of connections established by the different clients over the time the number of detected links first grows very fast. With a growing number of detected links, the clients will use more often already detected links. Both effects are depicted in Figure 7. First, the slope is notably larger than one but decreases with an increasing number of connections.

Additionally, we can observe that approximately 1000 connections by all clients are necessary to detect more than 90% of all detectable links. These 1000 connections may also include connections to the same server as all connections are chosen randomly. Only if the client occurrence rate is too small (as in this case indicated by the 1% graph) it takes significantly longer to detect large parts of the network. As soon as the client occurrence rate is greater than 10% the clients at the edge of the network can monitor passively a large fraction of the considered network.

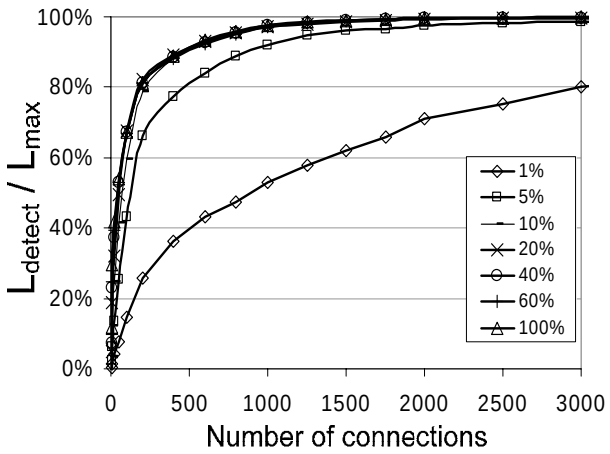

Figure 7 Development of the number of detected links for different DEMS client occurence rates against the number of connections established by the clients (server occurrence rate: 5%, occurrence rate access routers: 20%)

In the scenario depicted in Figure 7 each of the 200 clients has to establish on average 5 connections to arbitrary servers so that more than 90% of all links are detected by the clients. Considering a Peer-to-Peer environment this number is reached very fast if we assume three signalling connections and two downloads within a user session, which lasts on average about 900 seconds [22]. An additional advantage of this system is certainly that the most important links are detected and measured first by the nodes monitoring the network from the edge. Thus, the most interesting information for the network provider is available very early.

As indicated by Figure 4 the performance of the overall monitoring system can additionally be improved if the number of servers is increased to the level of clients as it is typical for a P2P network. In Figure 7 we only considered a server occurrence rate of 5%. If this value is increased too, the number of connections necessary to monitor and detect a large fraction of the network should decrease significantly. This results in a notably smaller transient period.

Having passed this transient period, the system stays at the high number of detected links as we assume that the network changes only at the access (as clients/servers leave and join the network). The core of the network being monitored by the clients does not change frequently [19][20][21]. When monitored data is stored in a reliable P2P system as described in section III, data is deleted only when it's too old (time-out) or updated. No data is lost if one of the peers storing network performance statistics leaves the network as the data is replicated on other peers and thus still available.

VI. Conclusion and Next Steps

In this paper we investigated a novel passive approach to network tomography, which relays only on information collected from payload data and stored in a P2P style. As demonstrated we can achieve a high coverage of the network topology already with a rather small number of peers. Our approach enables those areas of the network currently having a high load situation (and therefore being of high interest) to be covered by these passive measurements more intensely than the rest.

The next steps in our work will be the introduction and evaluation of distributed data analysis methods for the determination of the topology of the networks and the delay and loss situation found on the network links. This task comprises the selection of suitable correlation methods and their adaptation to the P2P environment. In this case we are on the one hand especially looking into pre-aggregation methods that will allow the determination of single data transfer trees already from the data being present in single peers or very few peers closely neighbouring. On the other hand, we are surveying methods for efficient data storage and incremental backups.

Another important aspect is the deployment of DEMS peers. In enterprise internal networks they could be deployed by the network service provider as part of an IT policy. For public networks we follow an "appetizer"

approach by integrating the measurement software into a screensaver always showing the user the current view onto the detected network topology and events (like node failures) in the network. Better links could be distinguished from links with high delay by using different colours e.g. red means “high” delay green means “low” delay. In general, evaluation and development of deployment strategies with respect to value add for end users and/or network operators will be another focus of our future work.

VII. 7. Acknowledgements

The work presented in this paper was supported by the European Commission. We would like to gratitude for the final support and sponsoring of the mCDN project, which is part of the 6th IST Framework Program.

References

[1] CAIDA, the Cooperative Association for Internet Data Analysis, provides tools and analyses promoting the engineering and maintenance of a robust, scalable global Internet infrastructure, http://www.caida.org, 2006

[2] National Laboratory for Applied Network Research (NLANR), Network Performance and Measurement Tools, http://dast.nlanr.net/NPMT/, 2006

[3] Cisco Systems. Netflow services solutions guide. white paper, http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/nfwhite.pdf, July 2001

[4] Keynote Systems, http://www.keynote.com, 2006

[5] KaZaA, www.kazaa.com, 2006

[6] eDonkey, www.edonkey2000.com, 2006

[7] Akamai, www.akamai.com, 2006

[8] Srinivasan, S., Zegura, E. “Network Measurement as a Cooperative Enterprise”. Proc. IPTPS02, Cambridge, MA, USA

[9] Srinivasan, S., Zegura E. “M-coop:A Scalable Infrastructure for Network Measurement”. Third IEEE Workshop on Internet Applications (WIAPP '03).

[10] Coates, M. J., Hero, et al. “Internet Tomography”. IEEE Signal Processing Magazine, May 2002.

[11] Castro, R., Coates, M. J., et al. “Internet Tomography: Recent Developments”. Statistical Science, vol. 19, no. 3, Aug. 2004, pp. 499–517.

[12] JXTA P2P Framework, www.jxta.org, 2006

[13] Stoica, I. Morris, R., et al. “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications”. ACM SIGCOMM 2001, San Diego, CA, August 2001

[14] Maymounkov, P., Mazieres., D. “Kademlia: A peer-to-peer information system based on the xor metric.”. Proc. of IPTPS02, Cambridge, USA, March 2002

[15] Rusitschka, S., Southall, A. “The Resource Management Framework”. AP2PC 2002

[16] Fall, K., Varadhan, K., editors. "ns notes and documentation. The VINT Project". UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 1997

[17] Medina, A., Lakhina, A., Matta, I., Byers J. "Brite: Boston University Representative Internet Topology generator". http://www.cs.by.edu/brite, 2002

[18] Wei L., Estrin D. "The trade-offs of multicast trees and algorithms". In Proc. of ICCCN'94. 1994

[19] Govindan R. and Reddy A. "An Analysis of InterDomain Topology and Route Stability". Proc. IEEE INFOCOM '97, Kobe, pp. 850--857,Apr. 1997

[20] Chinoy, B. "Dynamics of Internet Routing Information". In proc. of the ACM SIGCOMM Symposium on Communication Architectures and protocols. 1993

[21] Paxson, V. "End-to-End Routing behavior in the Internet". In Proc. of the ACM SIGCOMM. 1996

[22] Schollmeier R., Dumanois A. "Peer-to-Peer Traffic Characteristics". EUNICE 2003. Budapest, Hungary. September 2003