Endpoint Cluster Identification for End-to-End Distance Estimation

Matthias Scheidegger, Torsten Braun, and Florian Baumgartner
Institute of Computer Science and Applied Mathematics
University of Bern, Neubrückstrasse 10, 3012 Bern, Switzerland
Email: (mscheid|braun|baumgart)@iam.unibe.ch

***Abstract*— Distributed systems such as peer-to-peer networks and distributed servers can optimize their performance by adapting to the underlying network. End-to-end measurements are an important basis for such adaptivity. Although most applications measure similar properties of the network, the measurements are mostly done in application-specific ways. In this paper we propose a general peer-to-peer measurement service based on clusters of endpoints that show virtually identical QoS properties when observed from outside the cluster. We discuss the clustering concept as well as its use in the measurement service, and we present a measurement-based method for the remote identification of clusters. This method allows for detecting clusters that are not part of the peer-to-peer network. Our evaluation shows that the presented method is able to reliably detect clusters using measurements of round-trip time or of available bandwidth.**

I. Introduction

During recent years the focus of many developers has moved away from the standard client-server model towards more scalable and robust distributed designs. Many client-server applications now support redundant and distributed servers, and many peer-to-peer systems and overlay networks have been proposed for distributed information storage (e.g. Chord [1]), file serving for large communities (e.g. BitTorrent [2] or Slurpie [3]) and multicast [4], [5], [6], [7], [8]. The QoS between endpoints greatly impacts the performance of a distributed application and the choice of topology is often a determining factor for its efficiency and robustness. It is therefore important to dynamically adapt the system to the properties of the underlying network.

End-to-end measurements are a prerequisite for such adaptivity. Most distributed applications measure end-to-end latency, delay jitter, or available bandwidth. Delegating measurements to a more general, application-independent measurement service would have many benefits. First, coordinating the measurement activities of several applications could reduce the measurement overhead. Second, storing the combined measurements would allow for predicting QoS and identifying longer-term developments. Suitable algorithms typically require a certain ‘critical mass’ of data, which is provided by the combined measurements of several applications. Third, a common API would facilitate the development of distributed applications.

Some actual networks (e.g. Géant-2 [9]) provide a similar service using dedicated meters and measurement servers inside the network. However, global deployment of this kind of architecture is unlikely to succeed. All ISPs would have to agree on a common design and publish details about the structure and performance of their networks, which they are usually reluctant to do because they fear to lose competitive advantage. Therefore, we believe that a peer-to-peer design is preferable.

Performing and storing end-to-end QoS measurements for every pair of Internet endpoints clearly does not scale. Therefore, we propose an architecture based on grouping endpoints into clusters based on their distance in the network topology. We effectively replace endpoint-to-endpoint measurements with cluster-to-cluster measurements, which greatly reduces the problem’s complexity. For example, if two endpoints are very close together, measurements from a distant observer to both endpoints will be very similar and can be merged. This approach requires a mechanism to identify suitable clusters of endpoints. In this paper we describe an architecture for a measurement service based on clusters, and we present and evaluate a method to remotely identify clusters based on measurements of available bandwidth and round-trip time. The proposed method does not require cooperation from the remote measurement peers and thus facilitates local and incremental deployment.

The remainder of this document is structured as follows: Related work is presented in Section II. Section III discusses the clustering concept and presents our measurement-based approach to remote cluster identification. Evaluation of the approach is shown in Section IV, and Section V concludes the paper.

II. Related Work

Several global Internet end-to-end measurement services and distance estimation services have been proposed. SONAR [10] defines an API for distance estimation services. Clients send a list of IP addresses to the SONAR server, which returns the estimated distances for each address. IDMaps [11] is a more recent proposal for a host distance estimation service. It is based on so called *tracers* inside ISP networks measuring latency between each other. The distance between two endpoints is then approximated by the distances to their tracers, plus the distance between those tracers. mOverlay [12] aims at constructing a locality-aware overlay network. Endpoints are organized into groups based on proximity using a dynamic landmark procedure for joining endpoints. The endpoint groups maintain information about their distance to other groups and so provide a distance estimation service. QRON [13] is a more general framework for QoS-aware overlay routing, aimed at providing an overlay service network based on overlay brokers (OBs). The overlay provides paths with given QoS properties between OBs. MULTI+ [6] is a topology-aware overlay multicast protocol based on hierarchical grouping of peers according to their IP network prefixes. The approach constructs optimal overlay multicast trees using Skitter [14].

Several of these approaches group endpoints according to their closeness in the network, either based on IP addresses or based on measurements. While comparing IP addresses is simple, similar IP addresses do not necessarily correspond to closeness in the topology. The measurement-based methods are more robust in this respect. However, they can only estimate distances between members of the overlay and therefore require large scale deployment to be useful. The measurement-based remote cluster identification method presented in this paper overcomes this constraint, which makes even local deployment of the distance estimation service a viable option.

III. Remote Cluster Identification

A. Clusters

We propose a measurement service that provides a common measurement API to client applications and optimizes the measurement process by coordinating their measurement activities. The service is based on creating groups of endpoints that are sufficiently close together in the topology to be able to directly use each other's end-to-end measurements. Furthermore, if the remote measurement targets are sufficiently close to each other the measurements shall be combined. This requires a criterion to decide whether two given endpoints are close enough to be in the same group, which is given by the following definition of a *cluster*.

Let N be the set of endpoints in the network. $d_t : N \times N \mapsto \mathbb{R}_0^+$ is a time dependent *distance function* if and only if, for all $n \in N$, $d_t(n,n) = 0$. Note that, in contrast to Euclidean geometry, neither symmetry $\big(d(n,m) = d(m,n)\big)$ nor the triangle inequation $\big(d(n,m) + d(m,o) \geq d(n,o)\big)$ are required. Given a distance function d we call $C \subseteq N$ a *cluster with respect to* d if and only if, for all points t in time

$$|d_t(o,n) - d_t(o,m)| < \varepsilon, \quad \forall\, n,m \in C,\ \forall\, o \in N \setminus C \qquad (1)$$

where ε is a threshold that depends on the statistical error. The endpoint o is called the *observing node*. Two endpoints from the same cluster are called *neighbors*. The sets $\{n\}$ and N are trivial cases of clusters. If $C \subseteq D$, we call cluster C a *subcluster* of cluster D and D a *supercluster* of C.

In order to detect neighbors we define distance difference functions $\delta_o(n,m)$, which take values close to 0 if $d_t(o,n) - d_t(o,m) \approx 0$ (the positive case), and greater values otherwise. We require these functions to be commutative, i.e. $\delta_o(m,n) = \delta_o(n,m)$. In Section III-C we present a combined distance difference function that can identify neighbors from a single observing node using time series of measurements.

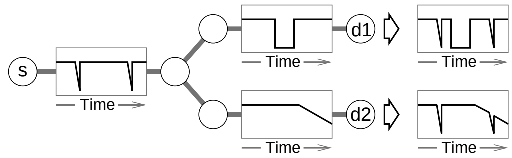

Fig. 1. Impact of common path segments on end-to-end measurements

This restriction to a single observation node enables local deployment of the service.

Clusters have several purposes in the measurement service. First, endpoints may exchange measurements if they belong to the same local cluster. Second, exchanged measurements may be combined if the targets belong to the same remote cluster. Together, this greatly reduces the measurement and storage overhead of the service. There is usually a 'natural' local cluster for a given endpoint node n, either $\{n\}$ or its first supercluster (e.g. its LAN). The local cluster can be informally determined by choosing a set of widely distributed reference endpoints R and finding the maximum cluster C that satisfies $R \cap C = \emptyset$. The resulting C is the local cluster.

These 'natural' clusters are the basic building blocks of the measurement service. Within the cluster the *cluster leader* (CL) coordinates the measurements of the *subordinate nodes* (SN) and collects the results in a database. Before starting a measurement a subordinate node first announces the upcoming measurement to the CL. The CL may either tell the SN to proceed, or it may return an appropriate prediction together with a confidence value. Recent measurements result in higher confidence values. The SN may choose to accept the prediction. Otherwise, it performs a real measurement and sends the results to the CL. Passive measurements (e.g. from monitoring an RTP session) may also be a data source. CLs may also exchange data between clusters to complement and refine available data, or to request distance estimates between remote endpoints. While this design has a single point of failure (the CL), it can easily be adapted to use the more robust local host cache approach used in mOverlay [12].

B. Time Series

With a single point of observation a single measurement to two endpoints is not enough for cluster detection. However, similarity in several successive measurements may indicate a neighborship. We therefore use time series of measurements for distance difference estimation. From the field of network tomography [15], [16] we know that time series of end-to-end delay measurements can in fact indicate whether two nodes are close to each other in the network topology. This can be used to detect the routing tree between a measuring node and a group of peer nodes [17]. The basic idea is illustrated in Fig. 1. The three boxes on the left show the evolution of a given link property (e.g. queuing delay) over time. The two boxes on the right show the impact on the end-to-end measurements. Note that the influence of the common link (the two negative peaks) is observed by both peers, whereas the influence of the other links can only be observed by one peer each. Such similarities can be used to estimate the length of the common path segment. The closer the peers, the more similarities can be observed. In the following we use these observations to define a measurement-based cluster identification approach.

Time series methods require uniform intervals between observations. However, measurements often come in irregular intervals. They can be described as tuples (t, v, p) of a value v observed at time t for peer p. We define a common base time T_{min} and a time step Δt. The step size is given by the minimum time interval between two consecutive observations of the same peer, i.e. $\Delta t = \min_{p \in P} \big(\min_i (t_{p,i+1} - t_{p,i}) \big)$ where P is the set of all peers. Then, we compute the normalized time series by linear interpolation of the input series at the instants $T_{min} + i \cdot \Delta t$.

Measurements we have performed on the Internet indicate that the standard deviation of available bandwidth grows approximately linear with the mean. We transform this multiplicative relationship between deviation and underlying trend into an additive one by applying the natural logarithm to every element of the normalized time series. Note that this only applies to measurements of available bandwidth.

C. Distance Difference Calculation

In order to detect whether two measurement targets are neighbors we compute distance differences between their respective time series. Distance difference values close to 0 indicate a neighborship. We combine two separate distance difference functions to make the cluster identification process more robust. These distance difference functions are applied to both, time series of round-trip time and available bandwidth, although with different parameters to account for the different characteristics.

If two remote endpoints belong to the same cluster, the distance measured to one endpoint should be a good estimate for the distance to the other. Accordingly, the first distance difference function identifies pairs of time series x, y where this is the case. We consider y_t to be a good estimate of x_t if y_t lies within an n-percent band around x_t. We detect violations of this rule using the following function

$$o_b(x_t, y_t) = \begin{cases} 0, & (1-b)x_t \le y_t \le x_t/(1-b) \\ 1, & otherwise \end{cases} \tag{2}$$

Parameter b denotes the size of the band. E.g., $b = 0.02$ denotes a 2% band. We chose the definition of o_b from (2) to ensure that $o_b(x_t, y_t) = o_b(y_t, x_t)$. Based on o_b we define the distance difference function O_b, which calculates the ratio of pairs x_t, y_t outside a band of size b of each other.

$$O_b(x, y) = \frac{\sum_{t=0}^{N-1} o_b(x_t, y_t)}{N} \tag{3}$$

where N is the length of the times series. A value of $O_b(x, y) = 0$ means that 100% of both time series are inside each others band. Unfortunately, O_b is sensitive to outliers. In order to make the approach more robust to outliers we introduce a threshold parameter $t_O \in [0, 1]$. The criterion for neighbor detection thus becomes $O_b(x, y) \le t_O$.

The O_b distance difference function in (3) detects similar measurement values, but it does not detect systematic differences between time series. Accordingly, we define a second function to check if the time series are unbiased estimators of each other. The relative bias between two time series x and y with mean values $\overline{x}$ and $\overline{y}$ is calculated with

$$B(x, y) = \begin{cases} 1 - \overline{x}/\overline{y}, & x \le y \\ 1 - \overline{y}/\overline{x}, & x > y \end{cases} \tag{4}$$

Again, we have chosen the definition such that $B(x, y) = B(y, x)$. Using the threshold parameter t_B the second criterion becomes $B(x, y) \le t_B$. This definition of B is closely related to the parameter b of O_b in (3): If $O_b(x, y) = 0$ for time series x and y, it follows that $B(x, y) \le b$. Accordingly, values for threshold t_B should be chosen in the range $[0, t_O]$.

We combine both criteria for cluster identification. Two endpoints are considered neighbors if they satisfy $O_b(x, y) \le t_O \wedge B(x, y) \le t_B$. The algorithm therefore depends on the three parameters b, t_O, and t_B. We will investigate the influence of both functions in Section IV.

We cannot give definite values for the parameters because we do not have any *a priori* knowledge about the statistical error in the input time series. However, the parameters can be approximated by applying cluster identification to a set of known reference clusters. The parameters should be chosen such as to maximize the number of detected neighborships while keeping the number of false positives close to zero.

IV. Evaluation

The cluster identification procedure was evaluated in three separate experiments. In the first experiment we investigate the impact of the three parameters b, t_O, and t_B based on a large set of averaged round-trip time measurements, and we show that the approach is able to reliably identify clusters. The second experiment demonstrates that the approach also works with non-averaged round-trip time measurements. In the third experiment we use measurements of available bandwidth to identify clusters.

A. Cluster Identification with Averaged Round-Trip Times

In the first experiment we have used data from PlanetLab [18] consisting of time series of round-trip time measured every 15 minutes during three days. Each value in the time series represents the mean of 10 RTT probes. The measurements were done between 77 PlanetLab nodes. Every endpoint measured the round-trip time to each of the other 76 endpoints, resulting in $77 \cdot 76 = 5852$ time series of round-trip time. The original data from PlanetLab included measurements from more than 77 endpoints, however with gaps. Therefore, we have used the maximum subset of endpoints that provided a full mesh of complete measurements. Cluster identification was done using the measurements from the first 1.5 days. The measurements from the second 1.5 days were used to verify the results.

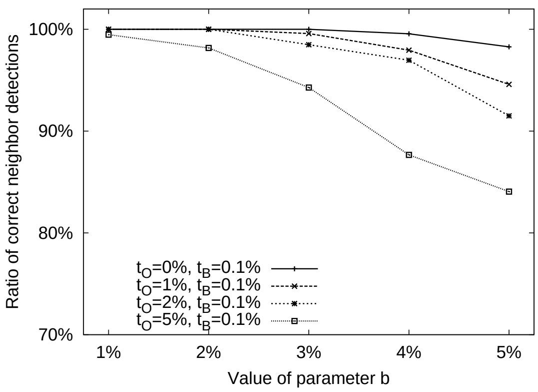

Fig. 2. Ratio of confirmed neighbor detections with $t_B = 0.1\%$

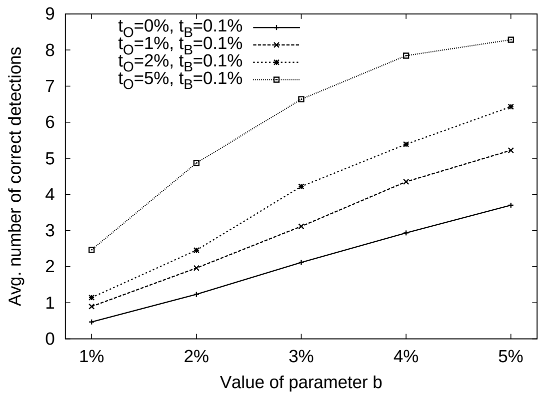

Fig. 4. Number of confirmed neighbor detections per observation point

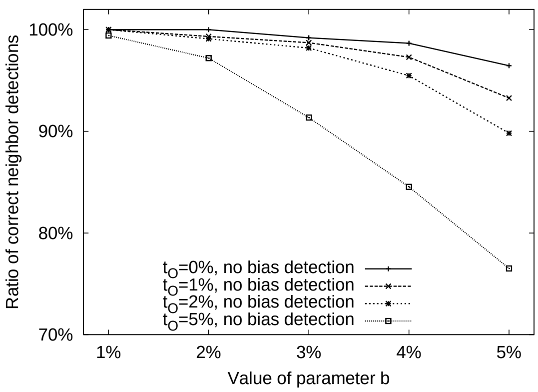

Fig. 3. Ratio of confirmed neighbor detections without bias detection

Since measurements were available from each of the 77 endpoints to all other endpoints we have performed 77 cluster identification procedures, one for each endpoint as an observation point. Each time, we computed the distance difference functions for every pair of the 76 other endpoints. Depending on the thresholds t_O and t_B we then decided whether a given pair of endpoints are neighbors (belong to the same cluster). These computations were repeated with several different values for parameters b, t_O, and t_B. In order to examine the influence of bias detection we have also performed this experiment without using the distance difference function B.

Two criteria were used to verify the results of the experiment. First, if two given endpoints were identified as neighbors from a given observation point, then the second 1.5 days of the measurements should confirm this. The second halves x', y' of the respective time series should still be good estimates of each other and thus satisfy $O_b(x', y') \leq t_O$ for suitable values of b and t_O. Second, the average round-trip time between neighbors should be small compared to the average round-trip times between other pairs in the data set. The full-mesh structure of the available measurement data allowed us to verify this easily. We discuss the results of applying both criteria in the following two sections.

1) Verification Using the Second Half of the Measurements: For every pair of endpoints identified as neighbors we verified whether the second halves of the respective time series show similar behavior. We considered the detection confirmed if the time series' values were within a 5% band of each other 95% of the time. Using the O_b function from (3) we can formulate this as $O_{5\%}(x', y') \leq 5\%$.

We have compared the number of confirmed neighbor detections with the number of total detections for several parameter sets. Fig. 2 shows the results for several values of b (the size of the relative band in O_b) and of t_O (the maximum ratio of values outside the band). The bias detection parameter t_B was constantly 0.1%. We can see from Fig. 2 that 100% of the neighbor detections were confirmed for small values of t_O and b. This shows that the approach is able to correctly determine whether two given endpoints belong to the same cluster. Values of b greater than 3 resulted in smaller ratios of confirmed detections. The choice of parameter t_O also significantly influences the quality of the detections. The greater t_O, the lower the ratio of confirmed detections. Nonetheless, only $t_O = 5\%$ and values of b greater than 3 resulted in less than 90% confirmed detections.

In order to evaluate the contribution of bias detection to the cluster identification method we have performed the same experiment using only the out-of-band criterion from (3). If bias detection really has a positive influence we should observe a significantly smaller ratio of confirmed neighbor detections. Fig. 3 shows the results of this experiment. While the ratio of confirmed neighbor detections only slightly decreases for small values of t_O and b, the effect is significant for greater values. For example, the ratio of confirmed detections with $t_O = 5\%$ and $b = 5\%$ decreases by 7.5%, from 84% with bias detection (Fig. 2) to 76.5% without bias detection (Fig. 3).

For all parameters, small values lead to better ratios of confirmed neighbor detections. However, there is always a trade-off between the number of confirmed neighbor detections and the number of false negatives, i.e. the number of endpoint pairs that would be confirmed as neighbors but are not detected as such. Fig. 4 illustrates this. The average number of confirmed neighbor detections per observation point is very small for $b = 1\%$ and rises considerably with higher values of b. This trade-off must be considered when choosing

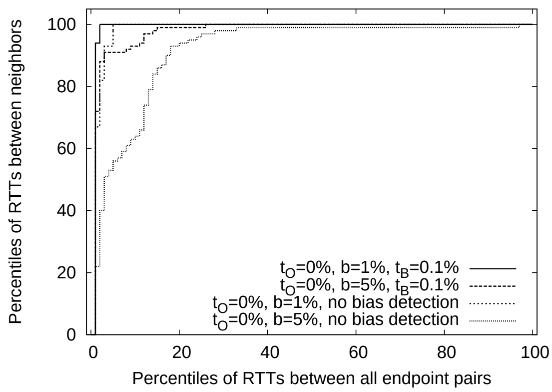

Fig. 5. Percentile-percentile plot of round-trip times between identified neighbors versus the round-trip times between all endpoint pairs

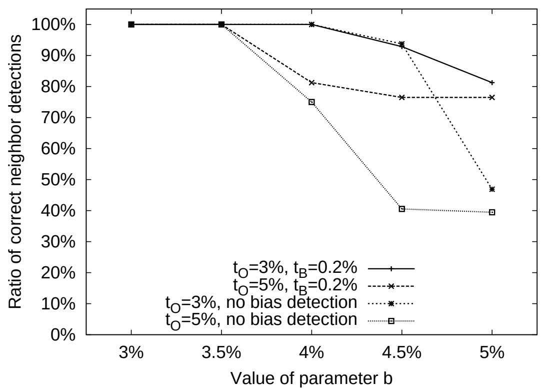

Fig. 6. Ratio of confirmed neighbor detections for different parameter sets, non-averaged RTT values

parameters for cluster identification. The criteria should not be more restrictive than necessary for a given application. Note that the number of identified neighbor pairs per observation point is rather small because PlanetLab [19] nodes are widely dispersed throughout the Internet, with only a few nodes per site. At the time of writing, PlanetLab consisted of 606 nodes distributed over 286 sites. Other measurement scenarios may lead to much higher numbers of identified neighbor pairs.

2) Verification Using Average End-to-end Round-Trip Time: A second way of verifying the results of cluster identification is to compare the average round-trip times between identified neighbors to the average round-trip times between other endpoint pairs. We have used the set of average round-trip times between each pair of endpoints as a reference. If the cluster identification method performs well the round-trip time between identified neighbors should be among the smallest in the reference set. We verify this using a percentile-percentile plot of the set of average round-trip times between identified neighbors and the reference set (Fig. 5).

We can clearly see that the round-trip times between identified neighbors are very small compared to the round-trip times between other pairs of endpoints. For $t_O = 0\%$, $b = 1\%$, and $t_B = 0.1\%$, 94% of the round-trip times between identified neighbors are smaller than the first percentile of the reference set. 100% are smaller than the second percentile. Even with $b = 5\%$, 88% of the identified neighbors have average round-trip times smaller than the second percentile of the reference set. For comparison, we have also included the plots for cluster identification without bias detection. The parameters were otherwise the same. Again we can see a positive impact of the bias detection function B on cluster identification. In the case $b = 5\%$, only 40% of the identified neighbors had average round-trip times smaller than the second percentile of the reference set, as compared to 88% with bias detection. Nevertheless, the results for $b = 1\%$ without bias detection are still rather good. 100% of the round-trip times between identified neighbors were smaller than the fifth percentile of the reference set.

B. Cluster Identification with Non-averaged Round-Trip Times

The results from Section IV-A show that the presented method is able to identify clusters based on end-to-end round-trip time measurements performed from a single point of observation. However, the measurement values in the data represent the mean of ten measurements each, which significantly reduces their variance. Consequently, we have also evaluated the approach with non-averaged measurements.

We have gathered 25 two-day time series of round-trip time, measured using `ping`. The first halves of the time series were again used for cluster identification while the second halves were used for verification. We have selected 25 distinct endpoints from the sites of five universities as measurement peers. The round-trip times were measured every five seconds for 48 hours from a single endpoint at the University of Bern.

As in Section IV-A we have applied the cluster identification procedure several times with different parameter sets. Verification was also done similarly. For each detected neighbor pair we have compared the second halves of the time series. If the values were within a 5% band of each other 95% of the time we would consider the detection confirmed. The results are shown in Fig. 6.

We observe that the ratio of confirmed neighbor detections also reaches 100% with non-averaged round-trip time values. However, the algorithm becomes more sensitive to changes in all three parameters. Fig. 6 shows that the ratio of confirmed neighbor detections rapidly decreases with rising values of b and t_O. On the other hand, the algorithm rejects all endpoint pairs with $b < 3\%$. Since outliers are much more frequent with non-averaged values, we also had to choose t_O greater than 0%. The increased variance of non-averaged measurement values thus effectively reduces the range of useful choices of parameters. The same effect can be observed for bias detection. Fig. 6 shows that cluster identification without bias detection results in significantly more errors. However, compared to the experiment with averaged round-trip values, the difference between both cases is much bigger. We conclude that the presented cluster identification approach is also useful for non-averaged measurements of round-trip time.

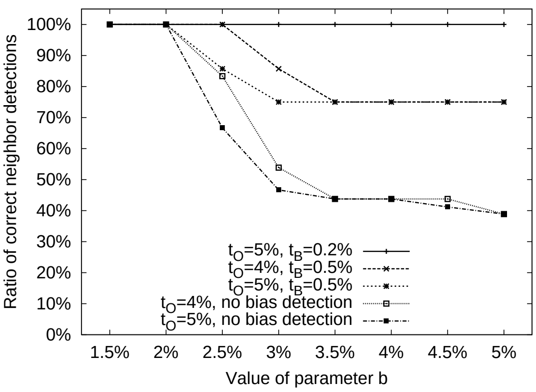

Fig. 7. Ratio of confirmed neighbor detections with available bandwidth

C. Cluster Identification with Available Bandwidth

In Sections IV-A and IV-B we have evaluated the presented cluster identification method using measurements of round-trip time. Nevertheless, the method should also be able to detect clusters based on measurements of available bandwidth. We investigate this using a similar experiment as in Section IV-B. We have gathered sixteen 24-hour time series of available bandwidth between a single observation point at the University of Bern and various endpoints in other universities' sites. The available bandwidth was estimated by downloading a sufficiently large file via HTTP, which was repeated every five minutes during 24 hours. We have used TCP throughput as an estimate for available bandwidth since it does not require any special software on the measurement peers. The results presented in [20] show that TCP throughput is a good estimate of available bandwidth.

As in the previous experiments we have used the first halves of the time series for cluster identification and the second halves for verification of the results. However, we have used different parameter values than with round-trip time measurements for the verification because of the high variance of the measurements. We have reduced the threshold for outliers to 2% but have increased the band b to 10%. Note that cluster detection used transformed values as described in Section III-B while the verification was done using the original values.

Fig. 7 shows the results of the experiment. Again, the method is able to reach a 100% ratio of confirmed neighbor detections. However, the parameter t_B (bias threshold) has much more impact than with round-trip times. With $t_B = 0.2\%$ the ratio of confirmed detections was constantly 100%. Increasing it to $t_B = 0.5\%$ resulted in a 25% smaller ratio for $b > 3.5\%$. Without any bias detection the ratio of confirmed neighbor detections even dropped below 50% in some cases. Another difference to the previous experiments is the effect of parameter b, which decreases for values greater than 3%. With round-trip times it grew with higher values. The choice of parameter t_O has an influence similar to the one for non-averaged measurements of round-trip time.

V. Conclusion

A general end-to-end measurement service for distributed Internet applications has advantages over proprietary measurement mechanisms built into the applications. Several related proposals already exist. In this paper we have proposed a peer-to-peer-based measurement service for distributed applications based on the concept of clustering nodes that, when observed from outside the cluster, show virtually identical quality of service properties over time. We have presented a method to identify such clusters remotely using time series of round-trip time and of available bandwidth, which enables clustering of endpoints outside the peer-to-peer network. Therefore, the approach does not require global deployment to be practicable. Evaluation of this cluster identification method using measurements from the Internet has shown promising results.

Acknowledgments

This work was carried out within the Swiss National Fonds project "eXperience Based Admission Control" (SNF No. 200021-101679/1) and the EU project EuQoS (IST-2003-004503).

References

[1] I. Stoica et al, "Chord: A scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM ToN*, vol. 11, no. 1, February 2003.
[2] M. Izal et al, "Dissecting bittorrent: Five months in a torrent's lifetime," in *PAM2004*, April 2004.
[3] R. Sherwood, R. Braud, and B. Bhattacharjee, "Slurpie: A cooperative bulk data transfer protocol," in *IEEE INFOCOM*, March 2004.
[4] L. Cherkasova and J. Lee, "Fastreplica: Efficient large file distribution within content delivery networks," in *4th USENIX Symposium on Internet Technologies and Systems*, March 2003.
[5] J. Janotti et al, "Overcast: Reliable multicasting with an overlay network," in *4th USENIX Symposium on Operating System Design and Implementation*, October 2000, pp. 197–212.
[6] L. Garcés-Erice, E. W. Biersack, and P. A. Felber, "MULTI+: Building topology-aware overlay multicast trees," in *QofIS'04*, September 2004.
[7] M. Castro et al, "Splitstream: High-bandwidth multicast in a cooperative environment," in *ACM SOSP'03*, October 2003.
[8] D. A. Tran, K. A. Hua, and T. T. Do, "ZIGZAG: An efficient peer-to-peer scheme for media streaming," in *IEEE INFOCOM*, 2003.
[9] "GÉANT2," http://www.geant2.net.
[10] "SONAR: A network proximity service," http://www.netlib.org/utk/projects/sonar.
[11] P. Francis, S. Jamin, J. Cheng, Y. Jin, D. Raz, and Y. S, "IDMaps: a global internet host distance estimation service," *IEEE/ACM ToN*, vol. 9, no. 5, pp. 525–540, October 2001.
[12] X. Y. Zhang et al, "A construction of locality-aware overlay network: mOverlay and its performance," *IEEE JSAC*, vol. 22, no. 1, pp. 18–28, January 2004.
[13] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," *IEEE JSAC*, vol. 22, no. 1, pp. 29–40, January 2004.
[14] "CAIDA," http://www.caida.org.
[15] M. J. Coates, A. Hero, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal Processing Magazine*, May 2002.
[16] N. G. Duffield and F. L. Presti, "Network tomography from measured end-to-end delay covariance," *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, pp. 978–992, December 2004.
[17] M. Coates et al, "Maximum likelihood network topology identification from edge-based unicast measurements," *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 1, pp. 11–20, June 2002.
[18] J. Stribling, "All-pairs-pings for Planetlab," http://pdos.csail.mit.edu/~strib/pl_app.
[19] "Planetlab home page," http://www.planet-lab.org.
[20] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 537–549, August 2003.