The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior

Andrew Adams, Pittsburgh Super Computing Center

Tian Bu, Timur Friedman, Joseph Horowitz, and Don Towsley, University of Massachusetts

Ramón Cáceres, Nick Duffield, and Francesco Lo Presti, AT&T Labs–Research

Sue B. Moon, Sprint ATL

Vern Paxson, ACIRI

ABSTRACT

We present a novel methodology for identifying internal network performance characteristics based on end-to-end multicast measurements. The methodology, solidly grounded on statistical estimation theory, can be used to characterize the internal loss and delay behavior of a network. Measurements on the MBone have been used to validate the approach in the case of losses. Extensive simulation experiments provide further validation of the approach, not only for losses, but also for delays. We also describe our strategy for deploying the methodology on the Internet. This includes the continued development of the National Internet Measurement Infrastructure to support RTP-based end-to-end multicast measurements and the development of software tools to analyze the traces. Once complete, this combined software/hardware infrastructure will provide a service for understanding and forecasting the performance of the Internet.

This work was sponsored in part by DARPA and the Air Force Research Laboratory under agreement F30602-98-2-0238, by DARPA award #AOG205, and by the National Science Foundation under Cooperative Agreement No. ANI-9720674. The Government has certain rights in this material.

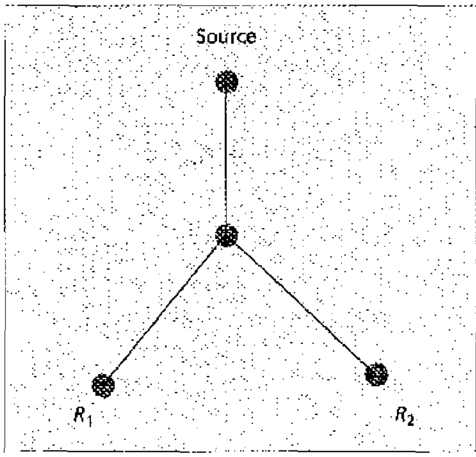
INTRODUCTION

As the Internet grows in size and diversity, its internal performance becomes ever more difficult to measure. Any one organization has administrative access to only a small fraction of the network's internal nodes, whereas commercial factors often prevent organizations from sharing internal performance data. End-to-end measurements using unicast traffic do not rely on administrative access privileges, but it is difficult to infer link-level performance from them, and they require large amounts of traffic to cover multiple paths. Consequently, there is a need for practical and efficient procedures that can take an internal snapshot of a significant portion of the network.

We have developed a measurement technique that addresses these problems. *Multicast inference of network characteristics* (MINC) uses end-to-end multicast measurements to infer link-level loss rates and delay statistics by exploiting the inherent correlation in performance observed by multicast receivers. These measurements do not rely on administrative access to internal nodes since they are done between end hosts. In addition, they scale to large networks because of the bandwidth efficiency of multicast traffic.

Focusing on loss for the moment, the intuition behind packet loss inference is that the arrival of a packet at a given internal node in the tree can be inferred from the packet's arrival at one or more receivers descended from that node. Conditioning on this latter event, we can determine the probability of successful transmission to and beyond the given node. Consider, for example (Fig. 1), a simple multicast tree with a root node (the source), two leaf nodes (receivers R_1 and R_2), a link from the source to a branch point (the shared link), and a link from the branch point to each of the receivers (the left and right links). The source sends a stream of sequenced multicast packets through the tree to the two receivers. If a packet reaches either receiver, we can infer that the packet reached the branch point. Thus, the ratio of the number of packets that reach both receivers to the total number that reach only the right receiver gives an estimate of the probability of successful transmission on the left link. The probability of successful transmission on the other links can be found by similar reasoning.

This technique extends to general trees [1], and it can be shown that the resulting loss rate estimates converge to the true loss rates as the number of probes grows indefinitely large. This and related approaches can be used to estimate path delay distributions [2], path delay variances [3], and the logical multicast topology itself [4]. We have validated the accuracy of the loss rate inference techniques against measurements on the MBone. Further validation of both the loss rate and delay statistics inference techniques has been made through simulation experiments.



■ **Figure 1.** *A tree connecting a sender to two receivers.*

In this article we describe the MINC methodology and the results of the network measurements and simulation experiments . Following this, we describe our efforts to deploy this methodology. These include the further development of the National Internet Measurement Infrastructure (NIMI) [5] to support the required multicast measurements, the extension of the Real-Time Transfer Protocol (RTP) control protocol (RTCP) to include detailed loss reports, and the development of the Multicast Inference Network Tool (MINT) to visualize and manipulate the multicast-based inferred internal network performance.

A survey of related work is included, and the last section offers some conclusions.

Statistical Methodology

MINC works on *logical* multicast trees, that is, those whose nodes are identified as branch points of the physical multicast tree. A single logical link between nodes of the logical multicast tree may comprise more than one physical link. MINC infers composite properties of the logical links. Henceforth, when we speak of trees we will be speaking of logical multicast trees.

Loss Inference

We model packet loss as independent across different links of the tree, and independent between different probes. Thus, the loss model associates with each link k in the tree, the probability α_k that a packet reaches the terminating node of the link, also denoted by k, given that it reaches the parent node of k. The link loss probability is then $(1-\alpha_k)$. Each receiver records the outcome of each probe sent by the source (i.e., whether or not it is received). The α_k can be expressed directly as a function of the probabilities of all possible outcomes of success and loss of a probe at each receiver. An experiment consists of a series of probes transmitted from the source. The outcome of each probe at each receiver is recorded, and the link probabilities are inferred by the estimators $\widehat{\alpha}_k$ obtained by using the actual frequencies of the outcomes. Reference [1] contains a detailed description and analysis of the inference algorithm.

The estimators $\widehat{\alpha}_k$ exhibit several desirable statistical properties. It was shown in [1] that $\widehat{\alpha}_k$ is the maximum likelihood estimator (MLE) of α_k when sufficient probes are used. The MLE is defined as the set of link probabilities that maximizes the probability of obtaining the observed outcome frequencies. The MLE property in turn implies two further properties of $\widehat{\alpha}$:

- *Consistency*: $\widehat{\alpha}_k$ converges to the true value α_k almost surely as the number of probes n grows to infinity.
- *Asymptotic normality*: The distribution of the quantity $\sqrt{n}\,(\widehat{\alpha}_k - \alpha_k)$ converges to a normal distribution as n grows to infinity.

The latter property implies that the probability of an error of a given size in estimating a link probability goes to zero exponentially fast in the number of probes.

The computation of the $\widehat{\alpha}_k$ is performed recursively on the tree; the computational cost is linear in the number of probes and number of nodes in the tree.

Delay Distribution Inference

A generalization of the loss inference methodology allows one to infer per link delay distributions. More precisely, we infer the distribution of the variable portion of the packet delay: what remains once the link propagation delay and packet transmission time are removed. Packet link delays are modeled as discrete random variables that can take one of a finite number of values, independent between different packets and links. The model is specified by a finite set of probabilities $\alpha_k(t)$ that a packet experiences delay t while traversing the link terminating at node k, with infinite delay interpreted as loss.

When a probe is transmitted from the source, we record either the time taken by a probe to reach each receiver or the loss of the probe. As with loss inference, a probabilistic analysis enables us to relate the $\alpha_k(t)$ to the probabilities of the outcomes at the receivers. We infer the link delay probabilities by the estimators $\widehat{\alpha}_k(t)$ obtained by using instead the actual frequencies of the outcomes arising from the dispatch of a number of probes. In [2] it was shown that the corresponding estimator $\widehat{\alpha}(\cdot)$ of the link delay distributions is strongly consistent and asymptotically normal.

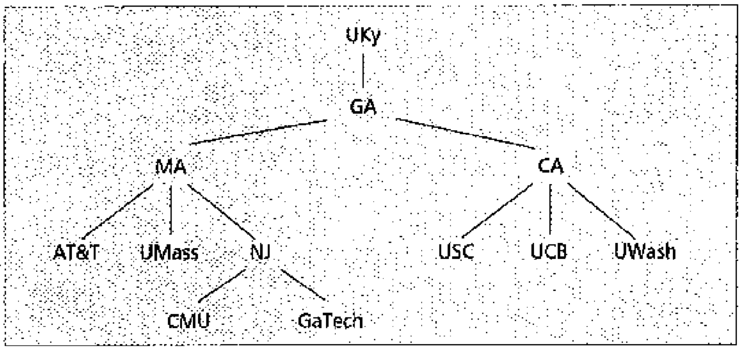
Delay Variance Inference

The delay variance can be directly estimated. Consider the binary topology of Fig. 1. Let D_0 be the packet delay on the link emanating from the source, and D_i, $i = 1, 2$, the delay on the link terminating at receiver i. The end-to-end delays from the source to leaf node $i = 1, 2$ is expressed as $X_i = D_0 + D_i$. A short calculation shows that, under the assumption that the D_i are independent, $\mathrm{Var}(D_0) = \mathrm{Cov}(X_1, X_2)$. Thus, the variance of the delay D_0 can be estimated from the measured end-to-end delays from the source to the leaves. This approach has been generalized to estimate link delay variances in arbitrary trees [3].

Topology Inference

In the loss inference methodology described above, the logical multicast tree was assumed to be known in advance. However, extensions of the method enable inference of an unknown

A generalization of the loss inference methodology allows one to infer per link delay distributions. More precisely, we infer the distribution of the variable portion of the packet delay: what remains once the link propagation delay and packet transmission time are removed.



■ **Figure 2.** *The multicast routing tree during our representative MBone experiment.*

multicast topology from end-to-end measurements. We briefly describe three approaches.

Loss-Based Grouping — An approach to topology inference was suggested in [6], in the context of grouping multicast receivers that share the same set of network bottlenecks from the source. The loss estimator of an earlier section estimates the shared loss to a pair of receivers, that is, the composite loss rate on the common portion of the paths from the source, irrespective of the underlying topology. Since this loss rate is larger the longer the common path in question, the actual shared loss rate is maximized when the two receivers are siblings.

A binary tree can be reconstructed iteratively using this approach. Starting with the set of receiver nodes R, select the pair of nodes j, k in R that maximizes the estimated shared loss, and group them together as the composite node. Iterate on this and the set of remaining nodes from R until all are grouped. The algorithm is consistent: the probability of correct identification converges to one as the number of probes grows [4]. General (i.e., nonbinary) trees can be inferred using this algorithm and then transforming the resulting binary tree by pruning links with inferred loss rates less than some threshold $\varepsilon > 0$.

General Grouping Algorithms — The above approach can be extended by replacing shared loss with any function on the nodes:
- That increases on moving further from the source
- Whose value at a given node can be consistently estimated from measurements at receivers descended from that node

The mean and variance of the cumulative delay from the source to a given node exhibit these properties. Hence, multicast end-to-end delay measurements can also be used to infer the multicast topology.

Direct Maximum Likelihood Classification — The direct ML approach calculates the maximum likelihood of the measured outcomes over all possible α_k. The topology that maximizes this quantity is chosen to be our estimate. This classifier is consistent [4].

Accuracy and Comparison — Experiments show similar accuracy for all the approaches described above. However, computational costs differ widely. The cost of the direct ML classifier grows rapidly with the number of receivers. The grouping methods avoid this since each grouping narrows the set of viable topologies; the binary grouping + pruning approach has near optimal accuracy and is simplest to implement.

Experimental Results

In this section we briefly describe our efforts to validate the MINC methodology. The next section contains a description of the results of a measurement study in which we collected end-to-end loss traces from the MBone and validated the results from inferences of loss rates collected using the Internet tool `mtrace`. Another section contains a description of the results from more detailed simulation studies of both loss and delay.

Measurement Experiments

To validate MINC under real network conditions, we performed a number of measurement experiments on the MBone, the multicast-capable subset of the Internet. Across our experiments we varied the multicast sources and receivers, the time of day, and the day of the week. We compared inferred loss rates to directly measured loss rates for all links in the resulting multicast trees. The two sets of quantities agreed closely throughout.

During each experiment, a source sent a stream of 40-byte sequenced packets every 100 ms to a multicast group consisting of a collection of receivers over the course of one hour. The resulting traffic stream placed less than 4 kb/s of load on any one MBone link. At each receiver, we made two sets of measurements on this traffic stream using the `mtrace` (see [7] for a description) and `mbat` software tools.

We used `mtrace` to determine the topology of the multicast tree. `mtrace` traces the *reverse* path from a multicast source to a receiver. It runs at the receiver and issues trace queries that travel hop by hop along the multicast tree toward the source. Each router along the path responds to these queries with its own IP address. We determined the tree topology by combining this path information for all receivers.

We also used `mtrace` to measure per-link packet losses. Routers also respond to `mtrace` queries with a count of how many packets they have seen directed to the specified multicast group. `mtrace` calculates packet losses on a link by comparing the packet counts returned by the two routers at either end of the link. We ran `mtrace` every 2 min during each 1-hr experiment. These `mtrace` queries were also used to verify that the topology remained constant during each experiment.

It is important to note that `mtrace` does not scale to measurements of large multicast groups if used in parallel at all receivers, as we describe here. Parallel `mtrace` queries converge as they travel up the tree. Enough such queries will overload routers and links with measurement traffic. We used `mtrace` in this way only to validate MINC on relatively small multicast groups.

We used `mbat` to collect traces of end-to-end packet losses. `mbat` runs at a receiver, subscribes to a specified multicast group, and records the sequence number and arrival time of each incoming packet. We ran mbat at each receiver for the duration of each 1-hr experiment.

We then segmented the mbat traces into 2-min subtraces corresponding to the 2-min intervals on which we collected mtrace measurements. Finally, we ran our loss inference algorithm on each 2-min interval and compared the inferred loss rates with the directly measured loss rates.

Here we highlight results from a representative experiment on August 26, 1998. Figure 2 shows the multicast routing tree in effect during the experiment. The source was at the University of Kentucky, and the receivers were at AT&T Laboratories, the University of Massachusetts, Carnegie Mellon University, Georgia Tech, the University of Southern California, the University of California at Berkeley, and the University of Washington. The four branch routers were in California, Georgia, Massachusetts, and New Jersey.

Figure 3 shows that inferred and directly measured loss rates agreed closely despite a link experiencing a wide range of loss rates over the course of a 1-hr experiment. Each short horizontal segment in the graph represents one 2-min 1200-probe measurement interval. As shown, loss rates on the link between the University of Kentucky and Georgia Tech varied between 4 and 30 percent. Nevertheless, differences between inferred and directly measured loss rates remained below 1.5 percent.

In summary, our MBone experiments showed that inferred and directly measured loss rates agreed closely under a variety of real network conditions:
- Across a wide range of loss rates (4–30 percent) on the same link
- Across links with very low (< 1 percent) and very high (> 30 percent) loss rates
- Across all links in a multicast tree regardless of their position in the tree
- Across different multicast trees
- Across time of day and day of the week

Furthermore, in all cases the inference algorithm converged to the desired loss rates well within each 2-min 1200-probe measurement interval.

Simulation Experiments

We have performed more extensive validations of our inference techniques through simulation in two different settings: the simulation of the model with Bernoulli losses and simulations of networks with realistic traffic. In the model simulations, probe loss and delay obey the independence assumption of the model. We applied the inference algorithm to the end-to-end measurements, and compared the inferred and actual model parameters for a large set of topologies and parameter values. We found that loss rates, mean delay, and variance estimates converged to close to their actual values with 2000 probes. The number of probes required to accurately compute the entire delay distributions is higher. In our experiments we found good agreement with 10,000 probes.

The second type of experiment is based on the ns simulator. Here delay and loss correspond to queuing delay and queue overflow at network nodes as multicast probes compete with traffic generated by TCP/UDP traffic sources. Multicast probes are generated by the source with fixed mean interarrival times; we used constant bit rate (CBR) or Poisson probes. We simulated different topologies with different background traffic mixes comprising infinite FTP sessions over TCP and exponential or Pareto on-off UDP sources. We considered both Drop Tail and Random Early Detection (RED) buffer discard methods [8].

We compared the inferred loss and delay with actual probe loss and delay. We found rapid convergence of the estimates, although with small persistent differences. We attribute this to the presence of spatial dependence (i.e., dependence between probe losses and delays on different links). This can arise through correlations in the background traffic due to correlation arising from TCP dynamics, such as synchronization between flows as a result of slow start after packet loss. We have shown in [1] that small deviations from the spatial independence assumption lead to only small errors in inference.

We also found that background traffic introduces temporal dependence in probe behavior (e.g., its burstiness can cause back-to-back probe losses). We have shown that while temporal dependence can decrease the rate of convergence of the estimators, consistency is unaffected. In the experiments the inferred values converged within 2000 probes despite the presence of temporal dependence.

While there is understanding of mechanisms by which temporal and spatial dependence can occur, as far as we know there are no experimental results concerning its magnitude. We believe that large or long-lasting dependence is unlikely in the Internet because of traffic and link diversity. Moreover, we expect loss correlation to be reduced by the introduction of RED.

We also compared the inferred probe loss rates with the background loss rates. The experiments showed these to be quite close, although not as close as inferred and actual probe loss rates. We attribute this to the inherent difference in the statistical properties of probe traffic and background traffic.

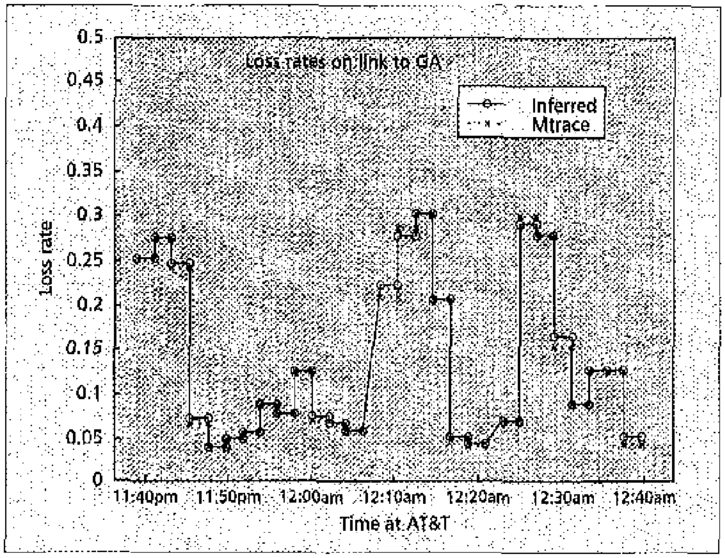
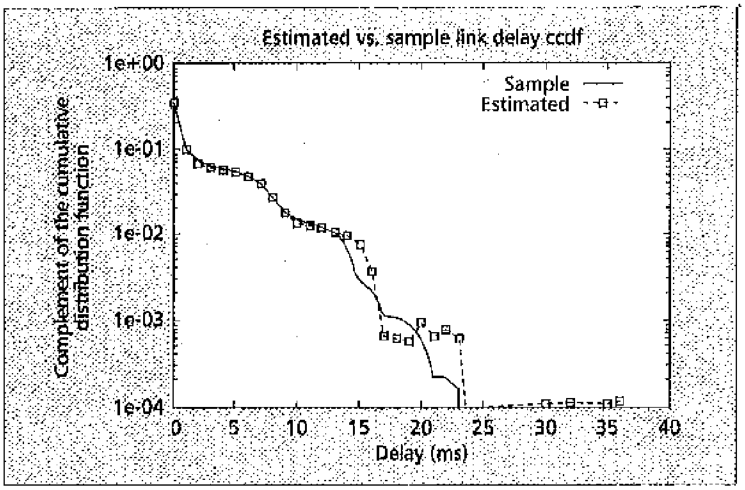


Figure 3. *Inferred vs. actual loss rates on the link between the University of Kentucky and Georgia Tech.*



■ **Figure 4.** *Inferred and sample delay ccdf for a leaf link in the topology of Fig. 2.*

To illustrate the distribution of delay inference results, we simulated the topology of the multicast routing tree shown in Fig. 2. In order to capture the heterogeneity between the edges and core of a network, interior links have higher capacity (5 Mb/s) and propagation delay (50 ms) than those at the edge (1 Mb/s and 10 ms). Background traffic comprises infinite FTP sessions and exponential on-off UDP sources. Each link is modeled as a FIFO queue with 4-packet capacity. Real buffers are usually much larger; the capacity of 4 is used to reduce the time required to simulate the network. The discard policy is Drop Tail. In Fig. 4 we plot the inferred vs. sample complementary cumulative distribution function (discretized in 1-ms bins) for one of the leaf links, using about 18,000 Poisson probes. The estimated distribution closely follows the sample distribution and is quite accurate for tail probabilities greater than 10^{-2}. Note that the estimated distribution is not always monotonically decreasing. This is because negative probabilities are occasionally estimated in the tail due to an insufficient number of samples. It is worth pointing out that, given the irregular shape of the sample distribution, the same level of accuracy would not be possible using a parametric model.

Deployment Efforts

It was observed in the previous section that MINC is a very promising methodology for providing detailed internal network performance characteristics. In this section we describe our efforts in deploying this methodology and making it available on the Internet. Our efforts are threefold. First, we are continuing the development of NIMI to support multicast-based measurement experiments. This is described next. Second, we have identified RTP and its associated control protocol, RTCP, as promising mechanisms for generating and collecting end-to-end multicast measurement traces. Our efforts to develop an RTP-based tool are described later. A description of an analysis and visualization tool, MINT, currently under development is included.

Deployment on NIMI

A major difficulty with characterizing Internet dynamics comes from the network's immense heterogeneity [9]. Load patterns, congestion levels, link bandwidths, loss rates, protocol mixes, the patterns of use of particular protocols — all of these exhibit great variation both at different points in the network, and over time as the network evolves. Accordingly, the sound characterization of Internet behavior requires measuring a diverse collection of network paths. It is not adequate to measure between just a few points, regardless of how carefully done.

The same problem arises in assessing the accuracy of measurement techniques such as MINC. To address this concern, we are deploying MINC measurement utilities within NIMI [5]. NIMI consists of a number of measurement "platforms" deployed at various locations around the Internet. Each platform is capable of sourcing and sinking active measurement traffic, and recording the timing of the traffic at both sender and receiver. Measurement "clients" that wish to use the infrastructure make authenticated requests to the platforms to schedule future measurement activity.

A key property of such an infrastructure is its N^2 scaling: if the infrastructure consists of N platforms, they together can measure network traffic along $O(N^2)$ distinct paths through the network. Consequently, with a fairly modest N, one can obtain a wide cross-section of the network's diverse behavior. (The NIMI infrastructure currently consists of 31 sites.)

Using NIMI for MINC measurements required several extensions to NIMI. The first was modifying the standard NIMI packet generator, *zing*, to send and receive multicast traffic, and the corresponding analysis program, *natalie*, to incorporate the notion that a single packet might arrive at several places (and fail to arrive at others). MINC also required the generalization of NIMI control mechanisms in order to allow for a single measurement run spanning multiple senders and receivers. A possible future change will be to use multicast itself for both scheduling measurements and disseminating the results.

Our experiences with using NIMI to date have been quite frustrating, not due to the infrastructure itself, but because of the poor quality of multicast connectivity between the different platforms. Until recently, at best only 1/3 of the NIMI platforms had multicast connectivity between them. We gather anecdotally that problems with poor interdomain multicast connectivity have been endemic to the Internet. Recently, connectivity has begun to improve, and it appears likely that over the next several years it will continue to do so, as agreement is reached on the proper set of intradomain and interdomain routing protocols and the interoperation between them. We are also attempting to address this problem in two ways:

- To grow the NIMI infrastructure by adding sites with high-quality multicast connectivity
- To investigate theoretical work on inferring network characteristics using correlated unicast traffic, where, instead of exploiting the perfect correlations inherent in multicast packet reception, we send back-to-back

unicast packets and attempt to exploit the considerably weaker correlations in their loss and delay patterns

INTEGRATION WITH RTCP

We are developing tools to apply MINC in real time so that MINC can be used by applications to respond to changing network conditions in new and more sophisticated ways. For example, a management program might adaptively adjust its probes to home in on a problem router.

Our tools transmit network information using RTCP, the control protocol for RTP [10]. By sharing their traces using RTCP, they benefit from RTCP's built-in scaling mechanisms.

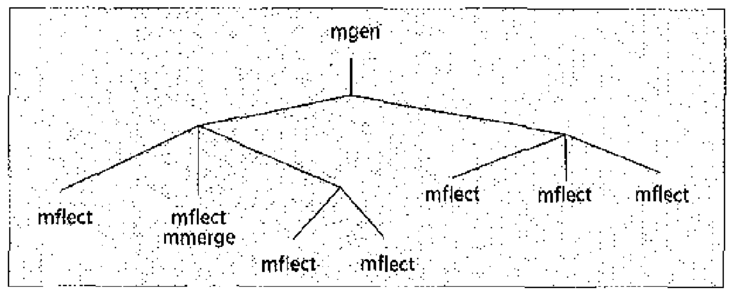
The approach is based on three tools: `mgen`, `mflect`, and `mmerge` (Fig. 5). `mgen` generates a stream of data (and may be replaced by any other application that multicasts data over RTP). A copy of `mflect` at each receiver maintains traces of the packets it does and does not receive from `mgen`. It periodically multicasts these (in a sense reflecting the data stream; hence, "`mflect`"). `mmerge` collects the traces sent by mflect, collates those from the different data receivers, and makes them available to a tool such as MINT for inference.

`mflect` and `mmerge` are designed so that they may be incorporated directly into existing and future multicast applications. Their joint functionality is available as an extension to the RTP common code library from University College London, called Extended Reporting (RTPXR). An application using RTPXR would be in a position to respond adaptively to information on the topology of its data distribution tree.

Ongoing research related to these tools concerns the scalability of trace sharing. For example, a raw bit vector loss trace for 3000 packets would consume 375 octets, far more than the four octets allocated for summary loss information in a standard RTCP packet. To limit the traces to an acceptable intermediate size we are investigating the use of compression techniques such as run length encoding, as well as distributed methods by which all copies of `mflect` in a single session agree on which portions of the trace to share in place of the whole trace.

MULTICAST INFERENCE NETWORK TOOL

MINT is intended to facilitate multicast-based inference. It takes as inputs all the traces collected from the end hosts. These traces may or may not include `mtrace` outputs. Currently, MINT comprises three components: a Web-based user interface, a topology discovery algorithm, and an inference engine. Users interact with MINT to manipulate the inference, such as by choosing number of samples, visualizing the multicast tree with losses, or showing the performance evolution over specific links. Depending on the availability of `mtrace` output, MINT discovers the topology by either parsing `mtrace` inputs or inferring the multicast tree from the loss traces. The inference engine takes topology information and loss traces to infer the network internal loss and then provides this to the user. The user can then view the results in one of several ways. One way is to lay out the logical multicast tree and display the links in different colors to distinguish different average loss rates (Fig. 6). The user can also focus on a single link and observe how the loss rate evolves over time for that link.



■ **Figure 5.** *An RTCP-based tool deployment example on the same topology as shown in Fig. 2, with inference performed at UMass.*

Our future plans for MINT are to include support for delay inference and to test it thoroughly by feeding it with daily traces collected from NIMI.

RELATED WORK

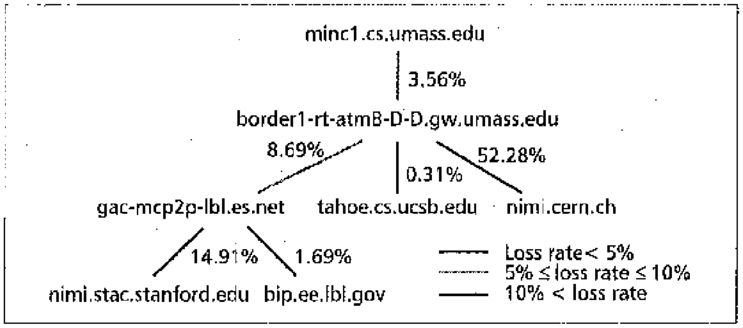
A growing number of measurement infrastructure projects (e.g., AMP, Felix, IPMA, NIMI, Surveyor, and Test Traffic [11]) aim to collect and analyze end-to-end performance data for a mesh of unicast paths between a set of participating hosts. We believe our multicast-based inference techniques would be a valuable addition to these measurement platforms. We are continuing to work on incorporating MINC capabilities into NIMI.

Recent experimental work has sought to understand internal network behavior from endpoint performance measurements (e.g., TReno [12]). In particular, `pathchar` [13] is under evaluation as a tool for inferring link-level statistics from end-to-end unicast measurements. Much work remains to be done in this area; MINC contributes a novel multicast-based methodology.

Regarding multicast-based measurements, we have already described `mtrace`. This forms the basis for several tools for performing topology discovery (`tracer` [14]) and visualizing loss on the multicast distribution tree of an application (MHealth [7]). However, `mtrace` suffers from performance and applicability problems in the context of large-scale Internet measurements. First, `mtrace` needs to run once for each receiver in order to cover a complete multicast tree, which does not scale well to large numbers of receivers. In contrast, MINC covers the complete tree in a single pass. Second, `mtrace` relies on multicast routers to respond to explicit measurement queries. Although current routers support these queries, providers may choose to disable this feature since it gives anyone access to detailed delay and loss information about paths inside their networks. In contrast, MINC does not rely on cooperation from any internal network elements.

CONCLUSIONS

We describe a new approach to identifying internal network characteristics based on the use of end-to-end multicast measurements. This methodology is rigorously based in estimation



■ **Figure 6.** *The MINT view of the logical multicast tree with losses.*

theory. A preliminary evaluation for identifying loss rates based on measurements made over the MBone indicates that it is accurate and readily able to track dynamic fluctuations that occur over time. More detailed investigations based on simulation further corroborate this conclusion, not only for the case of losses, but for delays as well. Finally, we describe our current efforts to deploy this methodology on the Internet and make it available to the community at large.

We believe MINC is an important new methodology for network measurement, particularly Internet measurement. It does not rely on network cooperation and should scale to very large networks. MINC is firmly grounded in statistical analysis backed up by packet-level simulations, and now experiments under real network conditions. We are continuing to extend MINC along both analytical and experimental fronts.

Acknowledgments

Mark Handley suggested using RTCP receiver reports to carry MINC loss traces.

References

[1] R. Cáceres *et al.*, "Multicast-Based Inference of Network-Internal Loss Characteristics," *IEEE Trans. Info. Theory*, Nov. 1999.

[2] F. Lo Presti *et al.*, "Multicast-Based Inference of Network-Internal Delay Distributions," preprint, AT&T Labs and Univ. MA, 1999.

[3] N. G. Duffield and F. LoPresti, "Multicast Inference of Packet Delay Variance at Interior Networks Links," *Proc. INFOCOM '00*, Tel Aviv, Israel, Mar. 26–30, 2000.

[4] R. Cáceres *et al.*, "Loss-Based Inference of Multicast Network Topology," *Proc. 1999 IEEE Conf. Dec. and Cont.*, Phoeniz, AZ, Dec. 1999.

[5] V. Paxson *et al.*, "An Architecture for Large-Scale Internet Measurement," *IEEE Commun. Mag.*, vol. 36, no. 8, Aug. 1998, pp. 48–54.

[6] S. Ratnasamy and S. McCanne, "Inference of Multicast Routing Tree Topologies and Bottleneck Bandwidths Using End-to-End Measurements," *Proc. IEEE INFOCOM '99*, Mar. 1999.

[7] D. Makofske and K. Almeroth, "MHealth: A Real-Time Multicast Tree Visualization and Monitoring Tool," *Proc. NOSSDAV '99*, Basking Ridge, NJ, June 1999.

[8] S. Floyd and V. Jacobson, "Reandom Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. Net.*, vol. 1, no. 4, Aug. 1993, pp. 397–413.

[9] V. Paxson and S. Floyd, "Why We Don't Know How to Simulate the Internet," *Proc. 1997 Winter Simulation Conf.*, Atlanta, GA, Dec. 1997.

[10] H. Schulzrinne *et al.*, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, Jan. 1996.

[11] Cooperative Association for Internet Data Analysis, "Internet Measurement Efforts," http://www.caida.org/Tools/taxonomy.html/InternetMeasurement, 1999.

[12] M. Mathis and J. Mahdavi, "Diagnosing Internet Congestion with a Transport Layer Performance Tool," *Proc. INET '96*, June 1996.

[13] A. Downey, "Using pathchar to Estimate Internet Link Characteristics," *Proc. ACM SIGCOMM '99*, Sept. 1999.

[14] B. N. Levine, S. Paul, and J. J. Garcia-Luna-Aceves, "Organizing Multicast Receivers Deterministically According to Packet-Loss Correlation," *Proc. ACM Multimedia '98*, Sept. 1998.

Biographies

ANDREW ADAMS is a network engineer in the Networking Group of the Pittsburgh Supercomputing Center at Carnegie Mellon University. The group is part of the National Laboratory of Applied Network Research. Projects he is currently working on include NIMI, Webl00, and researching network software tools, drivers, and hardware interfaces, as well as holding production networking responsibilities.

TIAN BU received his B.S. in computer science at Qingdao University and his M.S. at the University of Massachusetts, Amherst. He is currently a Ph.D. student in the Department of Computer Science at the University of Massachusetts, Amherst.

RAMON CACERES is a principal technical staff member at AT&T Labs-Research. He received a Ph.D. in computer science from the University of California at Berkeley in 1992. Most of his work since 1985 has been in the context of the Internet and Internet protocols.

NICK DUFFIELD (duffield@research.att.com) received his B.A. in natural sciences from Cambridge University, United Kingdom, in 1982, and his Ph.D. in mathematical physics from the University of London in 1987. He subsequently held post-doctoral and faculty positions in Heidelberg, Germany, and Dublin, Ireland. He is a technology consultant in the Internet and Networking Research group at AT&T Labs-Research, Florham Park, New Jersey.

TIMUR FRIEDMAN is a Ph.D. student in networking at the Computer Science Department, University of Massachusetts Amherst. He received an A.B. in philosophy in 1987, an M.S. in technology management in 1991, and an M.S. in computer science in 1995. He is presently a visiting researcher at the Laboratoire d'Informatique, Université Pierre et Marie Curie, Paris.

JOSEPH HOROWITZ received his B.S. degree from MIT in 1962, and his M.S. and Ph.D. degrees from the University of Michigan in 1963 and 1967, all in mathematics. He joined the Department of Mathematics and Statistics at the University of Massachusetts (Amherst) in 1969, where he has been a professor since 1980. He has been a visiting faculty member at Stanford University, the University of Strasbourg, ETH-Zurich, and the Technion (Haifa), and a Fulbright Research Fellow (1988, 1992) at the Indian Statistical Institute, New Delhi.

FRANCESCO LO PRESTI received his "Laurea" in electrical engineering and his Ph.D. in computer science from the University of Rome "for Vergata," Rome, Italy, in 1993 and 1997, respectively. He is currently with the Computer Science Department, University of Massachusetts at Amherst.

SUE B. MOON (sbmoon@sprintlabs.com) received her B.S. and M.S. from Seoul National University, Korea, in 1988 and 1990, respectively, both in computer engineering. From 1990 to 1991 she worked for IMIGE Systems, Inc., Seoul, Korea. She received a Ph.D. degree in computer science from the University of Massachusetts at Amherst. Since 1999 she has been with Sprint Advanced Technology Labs, Burlingame, California.

VERN PAXSON is a senior scientist with the AT&T Center for Internet Research at the International Computer Science Institute, Berkeley, California, and a staff scientist at the Lawrence Berkeley National Laboratory. He also serves on the Internet Engineering Steering Group as one of two area directors for the Transport area, and co-chairs the Internet Engineering Task Force's working groups on TCP Implementation and Endpoint Congestion Management.

DON TOWSLEY [F] (towsley@cs.umass.edu) is a distinguished professor in the Department of Computer Science at the University of Massachusetts. His research focuses on Internet modeling and analysis and network support for streaming media applications. He is a Fellow of the ACM and IEEE and has won several paper awards, including the 1998 IEEE Communications Society William Bennett award.