

Combining Active and Passive Network Measurements to Build Scalable Monitoring Systems on the Grid

Bruce B. Lowekamp¹
Computer Science Department
College of William and Mary
Williamsburg, VA 23187-8795
lowekamp@cs.wm.edu

Abstract

Because the network provides the wires that connect a grid, understanding the performance provided by a network is crucial to achieving satisfactory performance from many grid applications. Monitoring the network to predict its performance for applications is an effective solution, but the costs and scalability challenges of actively injecting measurement traffic, as well as the information access and accuracy challenges of using passively collected measurements, complicate the problem of developing a monitoring solution for a global grid. This paper is a preliminary report on the Wren project, which is focused on developing scalable solutions for network performance monitoring. By combining active and passive monitoring techniques, Wren is able to reduce the need for invasive measurements of the network without sacrificing measurement accuracy on either the WAN or LAN levels. Specifically, we present topology-based steering, which dramatically reduces the number of measurements taken for a system by using passively acquired topology and utilization to select the bottleneck links that require active bandwidth probing. Furthermore, by using passive measurements while an application is running and active measurements when none is running, we preserve our ability to offer accurate, timely predictions of network performance, while eliminating additional invasive measurements.

1 Introduction

Measuring and predicting the performance of a network is critical to the performance of grid applications, as well as many other uses of distributed computing. The importance of this field is indicated by the tremendous number and range of measurement tools that have been developed [6], including active TCP-based probes, SNMP queries, and packet dispersion techniques. The number of options makes selecting and administering the proper tool a daunting task for an end-user, and furthermore, the wide variety of grid-based applications

and endless combinations of machines on which they can run dictate that no single solution, and no small set of measurements, will adequately encompass the needs of all applications and users.

To simplify the process of taking measurements, and to address the challenge of providing measurements as a standard service of a grid computing environment, several monitoring systems have been developed. NWS combines host and network measurements, primarily focused on WAN measurements, into a system with advanced features for prediction [27, 28]. Remos, which we participated in developing, has similar goals but instead focuses on using SNMP queries to obtain measurements of LAN and campus networks [8, 17]. Nimi was developed to run a variety of tools used by network administrators [1]. These systems, and several others, do a good job of making network measurement manageable. But none of them does a complete job of scaling to the diversity and complexity of grid environments and applications.

1.1 Application Requirements

We will discuss three application classes: bulk data transfer, interactive visualization, and optimistic computation. Each has a different set of requirements from the network, requiring slightly different information to select resources and adapt their behavior effectively. The applications include long-term TCP data streams, high-bandwidth latency sensitive messages, and latency-tolerant single-packet messages.

As diverse as the application requirements are, they require equally diverse network and computing resources. In particular, the topology of the networks can be quite different and can play an important factor in their performance. Most network measurements taken today ignore topology—they simply measure the characteristics of the path between two endpoints. This approach works well in many cases, but in others it may fail to provide sufficient information. For example, a quick glance at the three applications would seem to imply that bulk data transfer is least likely to be affected by topology, being only a TCP connection between two endpoints itself, but that the other applications may be affected by contention among their own messages depending on network topology. However, even the simple bulk-transfer may

¹This research was supported in part by the Advanced Research Projects Agency and Rome Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-96-1-0287; by The College of William and Mary; and by the National Science Foundation under award ACI-0203974.

require some knowledge of topology. For example, if the measurement between two sites on the Internet is taken from different machines at those sites than those performing the data transfer, there may be a bottleneck on the LAN that is not measured by the measurement across the Internet (not unlikely in today's world of 100Mb to the desktop and 10Gb research backbones). Therefore, even such a simple application as bulk data transfer may require topology knowledge of the LANs on which the two endpoints are located.

To meet the challenges of widespread grid computing, a monitoring system must be able to respond to the diverse needs of grid applications, both in terms of the range of traffic they will generate and the variety of systems they may run on. Meeting these challenges requires the measurement tools and supporting software to meet several requirements:

Portability Measurements of the same basic characteristic, such as available bandwidth, must be interchangeable. For example, it must be possible to compare 128K TCP probes with 16K TCP probes.

Topology The measurement system must be aware of topology to detect local bottlenecks, in the case of WAN connections, and to predict contention, in the case of parallel connections.

Scalability The measurement system must scale, both for LAN and WAN connections.

To some extent, these requirements are met by various existing systems, but they have not been integrated into a single system for providing information to grid applications. First, significant results have already been achieved that address the *portability* requirement. These results serve as a proof-of-concept, but more work is necessary to ensure full portability across multiple systems. Similarly, *topology* information can be captured for both LAN and WAN systems. This information must be integrated into measurement systems so that it is available to applications [7, 17]. Furthermore, topology information is vital to solving the *scalability* problem, because in complex systems, such as most campus networks and a global grid with thousands of sites, it allows measurements to be concentrated on bottleneck links, rather than blindly measuring N^2 pairs.

1.2 The Wren System

The principal feature of Wren's approach to providing network measurements that scale from clusters to WANs is *topology-based steering*. Wren uses the topology information we can gather to identify potential bottlenecks in the network. Knowing where the bottlenecks are, and what traffic will be affected by them, we then steer active measurement techniques to measure only those links that may cause bottlenecks between machines. Appropriate application of this technique reduces the complexity of measuring the connections between nodes from N^2 to approximately $\log N$.

Wren combines topology-based steering with techniques to further reduce the measurement load by combining passive and active measurements into one system. The concept is simple—when a grid application is running, all measurements are made passively. However, when no application is running, or when the grid application is not sending sufficient traffic between the desired pairs of machines, we use active bandwidth probes. This capability hinges on the conversion from the passive measurements to active measurements—it must look like a continuous series of identical measurements for most prediction systems to operate.

Section 2 will discuss the networking requirements of the three application classes mentioned above. Sections 3 and 4 discuss the approach we are implementing in Wren to provide scalable network measurements. We finish the paper by discussing important directions for deploying scalable measurement services across grid environments.

2 Application Requirements

Before discussing the appropriate way to take network measurements, we must discuss application requirements. There are two components to this discussion—first the operations that the application performs, and second, the environment in which that application is generally run. These two factors dictate the measurements that must be made to schedule or adapt the application.

2.1 Bulk Data Transfer

Bulk data transfers, most notably for high-energy physics, but also required for remote instrumentation and database collections such as medical, remote microscopy, and genetics research, require moving large amounts of data between points on a network. If there is no choice of data source or destination, then there is no purpose in measurement, except for tuning the protocol parameters. However, when there is the option of either retrieving the data from multiple sources or processing it at different destinations, there must be quantitatively comparable measurements available so the application can select the best sites for the transfer. Among the requirements for measurements in this situation are:

- The measurements must all be of the same type, or if they are taken using different techniques, they must be quantitatively comparable.
- Any shared bottlenecks must be reflected. For example, retrieving in parallel one-quarter of the data from each of four sites measured at 5Mbps will not be helpful if that 5Mbps was a bottleneck at the destination site's connection to the Internet.
- The measurements must be valid for the machines performing the transfer—if the machine taking the measurement is in a university computing center, but the

machine retrieving the data is in a separate department, there must not be a more restrictive bottleneck along the path to the retrieving machine than to the measuring machine.

2.2 Interactive Visualization

Interactive visualization imposes a more demanding set of requirements on the network. First, the location of the data and the user dictate the sites to which the traffic must be directed. However, measurements are still important because there many parameters of such a visualization are flexible, including video quality and frame rate. Additionally, the computation itself may be parallel and relocatable, with only the source and destination fixed, but the network and computational path used along the way open to adaptation.

Interactive visualization shares the measurement requirements of bulk data transfer, listed above, and adds additional requirements:

- A parallel interactive visualization must deal with both the connections between users as well as the performance within whatever parallel resources are being used for the computation.
- Short-term bandwidth is much more important to application performance and adaptation than long-term bandwidth.

2.3 Optimistic Computation

Conventional wisdom has parallel computations moving to dedicated clusters, while naturally distributed applications, where users or resources are already distributed, are run on grids. However, new algorithms or ways of approaching old problems sometimes open up new possibilities. We are currently exploring the use of optimistic computation for parallel mesh generation, previously regarded as a fine-grained tightly-coupled application [4]. The optimistic approach to the computation allows the application to tolerate latencies and run efficiently on a variety of platforms, ranging from well-designed clusters to resources distributed across the Internet.

Although the application is designed to tolerate latency, it still requires bandwidth and latency predictions about the environment in which it is running. Each message is very short, but the rate at which messages can be sent is still determined by bandwidth. Furthermore, the optimistic computation relies on knowing when it can expect responses to previous messages to come in. Those prediction are dependent on the latency predictions available.

The nature of an optimistic, latency-tolerant application requires, again, a different set of network measurements:

- The messages exchanged by this application are very short (typically one packet), but frequent.

- It requires bandwidth and latency information to adapt its optimism thresholds.
- For best performance, a different algorithm can be run within each tightly-coupled cluster than is run between the clusters. Such adaptation requires precise information about each cluster it uses.

In summary, these three grid applications have messages varying from gigabytes to one packet. The timescales they care about vary from minutes to microseconds. The systems vary from an Internet path to detailed information on the communication within each cluster. *Network monitoring systems for grid applications must scale in all three dimensions just as the applications do.*

3 Combining Active and Passive Measurements

Our goal is to provide accurate, up-to-date measurements between any pair of machines on the network, while maintaining the scalability of the system and without introducing unnecessary load on the system. Wren seeks to reduce the load imposed by measurements by passively utilizing existing grid traffic to obtain measurements that would otherwise require invasive probes to be used. Other projects have also looked at passive monitoring of application data [2, 14, 15, 19, 21]. In the development of Wren, we are looking to expand on these approaches by using both active and passive measurements, as well as relying on measurement portability to allow different measurement techniques to be chosen according to available traffic, while preserving what appears to be a series of the same probe to other applications and services.

There are two challenges to implementing Wren's approach. The first is the conversion from one set of measurements to another, while preserving enough accuracy that the numbers can be used in the same time-series. The second challenge is instrumenting the system such that we obtain enough information to implement this approach, without compromising the efficiency of the application while running.

3.1 Measurement Portability

With the large number of projects and techniques currently being used for taking measurements, sharing information, and ensuring the portability of that information has become a significant concern for many researchers. The Network Measurements Working Group (NMWG) of the GGF was started with the goal of bringing together experts in the field to work on ensuring that measurements are labeled in such a way that the exchange and conversion of network measurements is possible. Its first work has been a taxonomy of measurements [18]. Although simple in principle, the first step towards converting between measurements is agreeing on definitions of what is being measured.

The culmination of that work is still in the future. In the meantime there have been several proof-of-concept works that have indicated that it is possible to obtain useful quantitative predictions of one form of a network measurement by using another measurement, and possibly a scaling function to map between them. As part of Remos, we have previously implemented system that predicted the performance of bulk TCP data transfers by observing bottleneck link utilizations through SNMP queries [16]. In those experiments, we used previous history correlating SNMP-observed utilization to the bandwidth achieved by TCP bulk data transfers. We found that even when the mapping function was built using data obtained significantly in the past, the SNMP-observed utilization was still able to accurately predict the future performance of TCP bulk transfers.

More recently, Swamy and Wolski have studied using short TCP probes to predict the performance of longer TCP probes [24]. They take the approach of using a previous history of correlated short and long TCP probes to build a mapping function to be used in the future. This work is quite important, because there are significant differences in the performance and behavior of short-term versus long-term data transfers. In particular, Swamy and Wolski observe that even when the short TCP probes never fully open their congestion window, their performance is still capable of providing accurate predictions of future long-term TCP data transfers using their correlation function.

As part of the Wren project, we have been exploring using instrumented kernels on the machines performing data transfers to observe their behavior at packet level. In particular, our work has focused on using not only the long-term performance of TCP transfers, which can be highly influenced by how the application uses the network, but also on observing packet pairs to analyze network performance based on packet dispersion techniques. Figure 1 shows results comparing these techniques on the same network connection. In this example, two hosts across a WAN are exchanging data at full rate. We see a high correlation between the performance obtained by the TCP transfer and that calculated by each of the other techniques. This correlation indicates that we should be able to transparently instrument the machines being used in a computational cluster to obtain information while the grid application is running. When the application terminates, or if it is not sending sufficient data between the pairs of machines we are interested in, we can use active probes to supplement those data.

We are currently extending these experiments to environments where the host is not consistently sending full-bandwidth communication. We expect that these results will indicate how often we can take advantage of the natural traffic sent by the application to provide essentially zero-cost measurements of network performance while applications are running.

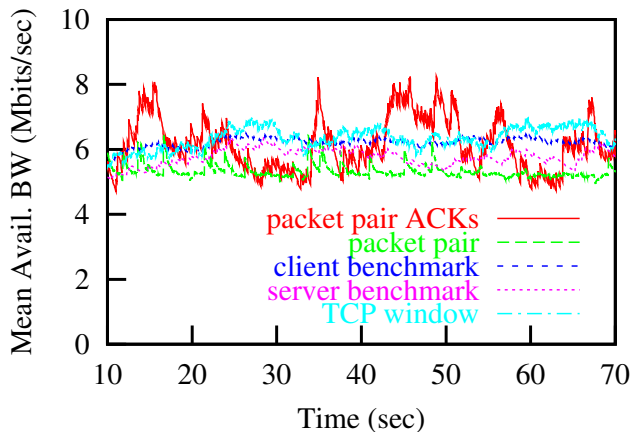


Figure 1: Available bandwidth calculated several different ways. Packet pair ACKs calculates dispersion of ACK arrivals on the sender’s side compared to data transmission. Packet pair performs dispersion calculations using instrumentation on the receiver’s side. Client and server benchmarks calculate data over time for the receiver and sender, respectively. TCP window calculates the expected rate based on the congestion window size on the sender’s side.

3.2 Kernel Instrumentation

The information necessary for our technique is gathered by instrumenting the network stack within the 2.4 series of Linux kernels. The connection, sequence numbers, and flags of each packet are recorded with a timestamp as the packet is transferred to or from the network interface. This information is then made available to the user level through an interface in the /proc filesystem. Our implementation minimizes the overhead imposed at kernel level during program execution, and hopefully allows analysis to be scheduled when the application is waiting for data. The overhead imposed by the user-level code is minimal, and the data can be transferred to another machine for processing, as necessary.

We have implemented several different techniques for calculating available and achievable bandwidth using this instrumentation. The simplest is the bulk-transfer benchmark, where we observe the amount of data TCP successfully sends over a given interval of time, duplicating the functionality of traditional user-level probes within the kernel. These techniques are accurate as long as the application is sending sufficient data. Our other techniques are based on variants of the packet dispersion techniques. We have implemented techniques using either instrumentation at one or both ends, using data packets or ACK packets depending on which ends instrumentation is available.

Figure 1 illustrates the correlation in the numbers produced by the different techniques in our implementation. We have measured the overhead imposed by our instrumentation and compared it with an uninstrumented kernel and the Web-100 instrumentation of the same kernel [26]. Our implementation is in between the two versions, adding a small amount

of overhead necessary to achieve its specific purpose, but less than the more general-purpose Web-100 code.

4 Topology-Based Steering

Enhancing measurement portability allows us to reduce the number of measurements taken on a system by utilizing traffic already on that network, but it does not address the fundamental issues of reducing the complexity of the measurements taken from N^2 . As discussed in Section 2, grid applications depend on measurements of both WAN communication and cluster communication. We believe the best way to provide this information in a scalable manner is to use topology information to determine when a measurement between one pair of machines can be applied to another pair.

Consider a network such as the one shown in Figure 2. For the purposes of our discussion, this network could be either a LAN or WAN. To meet our requirement of obtaining measurements for all pairs of machines, there are two naive approaches to this network:

- A completely naive approach is to take all N^2 measurement possible. Assuming that the measurements are synchronized to prevent contention, this approach could take measurements continually, consuming a significant portion of the network, and as the number of processors grows, still not perform measurements frequently enough to provide up-to-date information.
- A somewhat more intelligent approach is to partition the measurement by choosing a node within each cluster to perform measurements across the WAN, and only perform pairs of measurement across the LAN. NWS uses this approach by organizing hosts into cliques [11]. Cliques are effective, but naively selecting the clique representative can be problematic. If there are significant differences in how the clique representative is attached to the backbone and how the other nodes are connected to the backbone, the measurements between cliques will not be useful for the other machines.

Wren’s topology-based steering approach addresses these concerns by using knowledge of the network’s topology to determine the best machines to use to take measurements. By combining the information available through various measurement approaches, we can effectively reduce the number of measurements that need to be taken without sacrificing accuracy or frequency of measurement.

4.1 Detecting Topology

Although the discussion above assumed knowledge of the topology in Figure 2, acquiring that information isn’t necessarily trivial. There are two approaches to characterizing network topology: physical and functional. The physical approach determines the physical links that connect the

network together. By determining the connections between links, along with their capacities, queuing algorithms, and traffic load, the network can be modeled and its behavior analyzed or predicted. Physical topology can be determined for both LANs [3, 17] and WANs [12, 20].

The functional approach differs in that it makes use of end-to-end information, under the assumption that such observations are more readily available and usable than modeling low-level network behavior. Functional topology representations attempt to group and arrange network sites according to their perceived closeness determined by traffic performance, rather than according to the actual connections of physical links. This approach may be taken across a variety of sites distributed around the Internet, or using a single-source tree [5, 13, 22, 25].

If the functional topology is accurate, it provide the same information that a physical topology does in terms of steering what pairs of machines to take measurements between. The accuracy of the functional topology will, of course, impact the accuracy of the measurements taken, but a reasonable functional topology should be good enough to significantly reduce the needed number of measurements while maintaining most of the accuracy.

4.2 Topology with Utilization

The SNMP queries that Remos uses cannot measure the achievable bandwidth of a TCP connection directly. However, they can measure the capacity and utilization of each link. Our goal is to use the utilization information to determine which links have sufficient competing traffic that active probes must be used to measure the achievable bandwidth against that traffic, and which are essentially free, and will therefore not present a bottleneck.

While utilization is a good guide to achievable bandwidth, it is certainly not the only factor. In particular, the type of the competing traffic and the queuing discipline used on the switches can have a substantial impact on the bandwidth an application actually achieves. These properties make utilization most interesting as a way to identify bottlenecks that must be measured, rather than as the sole measurement themselves. For example, links that average below 1 percent utilization and never are fully utilized are not nearly as important as links that average 10 percent utilization and are frequently fully utilized for short time periods. Identifying the portions of the network where there is a potential for congestion and using active probes to measure only those links is one advantage of obtaining utilization information.

The LAN topology obtained through Remos makes it easy to form accurate cliques by grouping the hosts attached to each edge switch. After grouping the cliques, the next step is determining the best representative for each clique. For this discussion, we will define the best representative as the one with the highest achievable bandwidth to the edge switch connecting the clique. A better technique would be to determine the

capacity of the WAN link to be measured and select any node (or multiple nodes if necessary to saturate a high-bandwidth link) with as much available bandwidth as the capacity of the upstream link from the edge switch forming the clique. For our initial implementation, however, we are simply working to select the node with the highest achievable bandwidth.

Low utilization can be a guide to which host to select as the clique representative, but the best confirmation is an active measurement among the hosts in the clique, or to other nodes outside the clique. We are working on refining algorithms to select the optimal representative by applying active probes, but currently select the node with the highest unutilized bandwidth.

Once clique representatives have been determined, the pairs of measurements needed to probe the backbone can be determined. In Figure 2, each of the links A, B, and C needs only one measurement. Capacity and utilization may provide enough information to know which pair of cliques 1–4 will be able to effectively probe links B and C.

4.3 Topology without Utilization

Remos is designed to provide topology, capacity, and utilization information for a LAN. Capacity and utilization may simplify the process of identifying backbone links that need to be measured, as well as identifying which nodes should be representatives of each clique for measurements across the larger network.

However, there are still many cases where bandwidth information is not available through SNMP. It is rarely available (especially in real-time) for WANs, and although Remos is designed to acquire it from bridges and routers in LANs, implementations of VLANs and trunking frequently prevent utilization numbers from being collected effectively. Furthermore, in some networks no layer-2 topology can be determined, but the network subnets provide some basic topology information that should be useful in partitioning the hosts. Therefore, we must be able to use topology information to jumpstart the process of identifying the appropriate links to measure bandwidth on.

For small groups of leaf nodes, a straightforward approach is to perform the full N^2 pairs measurements, and select the node with the highest bandwidth. By performing these measurements continually, as done by a monitoring system, the representative selection can be based on the node most likely to have the highest available bandwidth, and can change over time.

Assuming that accurate topology is available, the number of measurements taken could possibly be reduced by taking advantage of the fact that there are only two links, plus the switch, involved in each exchange. This is, however, a complex statistical question that we have not yet attempted to solve, preferring instead to acquire utilization information from the network. However, even in the absence of any

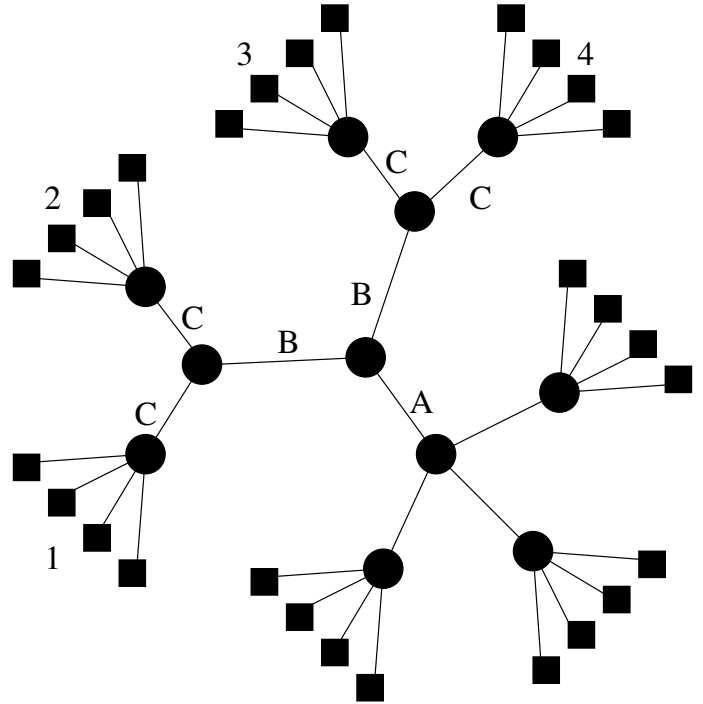


Figure 2: A hypothetical network topology to consider for topology-based steering.

other information, an examination of the topology in Figure 2 shows that link A can be handled by only one probe (although possibly several hosts may need to participate simultaneously to saturate it). With this knowledge, especially if combined with packet dispersion techniques designed to determine capacity and availability [9, 10], we can still utilize the topology information to scale the number of measurements required to monitor the system.

5 Future Directions

Our assertion that grid environments require network monitoring at all levels of their architecture, from the WAN links to intra-cluster links, is somewhat controversial—for years the assumption has been that the bottleneck of a wide-area distributed application is in the WAN portion. However, several factors are coming together to support our position. WAN bandwidths have increased dramatically over the past few years, to the point that multi-gigabit ISP connections are widely available. In that same time, other than those used by network specialists, servers, and custom-designed clusters, few nodes are receiving more than 100Mb Ethernet connections. As grid computing becomes more widespread, we anticipate that the number of machines participating in a grid, but not on a network designed to give them the full WAN capabilities at their institution will grow. Here at William and Mary, we have three computational clusters in different departments, only one of which was given any more than a regular 100Mb to-the-desktop network connection.

The Wren project is focused on using techniques of both passive and active measurements to provide scalable measurement services capable of providing information at all levels of grid architectures. It harnesses passive techniques for topology discovery, utilization monitoring, and traffic measurement for application in steering and selecting the best active probes to provide the information needed by applications. By maximizing the information collected by a small number of active probes, Wren will be able to achieve the goal of scalable measurements. This paper has described the preliminary results and goals of two components of the Wren system. Upon completion it will be usable standalone, or as a component to provide information to other systems such as NWS or systems supporting standardized interchange of monitoring information.

Figure 1 was contributed by Marcia Zangrilli, who is implementing the active-passive monitoring techniques. Section 4 includes contributions from Sam Small, who is working on using Remos' topology information for clique selection [23].

References

- [1] A. Adams, J. Mahdavi, M. Mathis, and V. Paxson. Creating a scalable architecture for Internet monitoring. In *Proceedings of the 8th Annual Internet Society Conference (INET'98)*, Geneva, Switzerland, July 1998.
- [2] J. Bolliger, T. Gross, and U. Hengartner. Bandwidth modelling for network-aware applications. In *Proceedings of Infocomm'99*, 1999.
- [3] Y. Breitbart, M. Garofalakis, C. Martin, R. Rastogi, S. Seshadri, and A. Silberschatz. Topology discovery in heterogeneous IP networks. In *Proceedings of INFOCOM 2000*, March 2000.
- [4] N. Chrisochoides, C. Lee, and B. B. Lowekamp. Mesh generation and optimistic computation on the grid. In *Proceedings of the Workshop on Performance Analysis and Distributed Computing, Performance Analysis and Grid Computing*, Schloss Dagstuhl, August 2002.
- [5] M. Coates, A. O. H. III, R. Nowak, and B. Yu. Internet tomography. *IEEE Signal Processing Magazine*, 19(3):47–65, May 2002.
- [6] L. Cottrell. Network monitoring tools. <http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html>.
- [7] M. den Burger, T. Kielmann, and H. E. Bal. TOPOMON: A monitoring tool for grid network topology. In *International Conference on Computational Science (2)*, pages 558–567, 2002.
- [8] P. Dinda, T. Gross, R. Karrer, B. B. Lowekamp, N. Miller, P. Steenkiste, and D. Sutherland. The architecture of the Remos system. In *Proceedings of the Tenth IEEE International Symposium on High Performance Distributed Computing (HPDC 10)*, pages 252–265, August 2001.
- [9] C. Dovrolis and M. Jain. Pathload: A measurement tool for end-to-end available bandwidth. In *Passive and Active Measurements Workshop*, 2002.
- [10] A. B. Downey. Using pathchar to estimate Internet link characteristics. In *Proceedings of ACM SIGCOMM 1999*, 1999.
- [11] B. Gaidioz, R. Wolski, and B. Tourancheau. Synchronizing network probes to avoid measurement intrusiveness with the network weather service. In *Proceedings of the Ninth IEEE International Symposium on High Performance Distributed Computing (HPDC 9)*, pages 147–154, Pittsburgh, PA, August 2000.
- [12] R. Govindan and H. Tangmunarunkit. Heuristics for Internet map discovery. In *IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000.
- [13] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. On the placement of Internet instrumentation. In *IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000.
- [14] G. Kin, G. Yang, B. R. Crowley, and D. A. Agarwal. Network characterization server (NCS). In *HPDC11*. IEEE, August 2001.
- [15] K. Lai and M. Baker. Nettimer: A tool for measuring bottleneck link bandwidth. In *Proceedings of USENIX Symposium on Internet Technologies and Systems*, 2001.
- [16] B. B. Lowekamp, D. O'Hallaron, and T. Gross. Direct queries for discovering network resource properties in a distributed environment. In *Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computing (HPDC)*, pages 38–46. IEEE Computer Society, August 1999.
- [17] B. B. Lowekamp, D. R. O'Hallaron, and T. Gross. Topology discovery for large Ethernet networks. In *Proceedings of SIGCOMM 2001*, pages 237–248. ACM, August 2001.
- [18] B. B. Lowekamp, B. Tierney, L. Cottrell, R. Hughes-Jones, T. Kielmann, and M. Swamy. A hierarchy of network measurements for grid applications and services. draft at <http://www-didc.lbl.gov/NMWG/>.
- [19] M. Mathis, J. Semke, and J. Mahdavi. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communications Review*, 27(3), 1997.
- [20] V. N. Padmanabhan and L. Subramanian. An investigation of geographic mapping techniques for Internet hosts. In *Proceedings of ACM SIGCOMM 2001*, pages 173–185, 2001.
- [21] S. Seshan, M. Stemm, and R. H. Katz. SPAND: Shared passing network performance discovery. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, pages 135–46, December 1997.
- [22] G. Shao, F. Berman, and R. Wolski. Using effective network views to promote distributed application performance. In *Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)*, 1999.

- [23] S. Small and B. B. Lowekamp. Topology-based clique organization for distributed resource monitoring. In *Proceedings of the IEEE/ACM SC2002 Conference*, Baltimore, MD, November 2002. Poster.
- [24] M. Swany and R. Wolski. Multivariate resource performance forecasting in the network weather service. In *Proceedings of the IEEE/ACM SC2002 Conference*, Baltimore, MD, November 2002.
- [25] W. Theilmann and K. Rothermel. Dynamic distance maps of the Internet. In *IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000.
- [26] Web100. <http://www.web100.org/>.
- [27] R. Wolski. Forecasting network performance to support dynamic scheduling using the network weather service. In *Proceedings of the 6th High Performance Distributed Computing Conference (HPDC)*, pages 316–25, August 1997.
- [28] R. Wolski. Dynamically forecasting network performance using the network weather service. *Cluster Computing*, 1(1):119–132, 1998.