AQUILA Distributed QoS Measurement

Felix Strohmeier
Advanced Network Center (ANC)
Salzburg Research Forschungsgesellschaft mbH
Jakob-Haringer-Str. 5/III, 5020 Salzburg, Austria
felix.strohmeier@salzburgresearch.at

Heinz Dörken
Multimedia-Transportsysteme, E513b
T-Nova Deutsche Telekom Innovationsgesellschaft mbH
Am Kavalleriesand 3, 64295 Darmstadt, Germany
heinz.doerken@t-systems.de

Bernhard Hechenleitner
Advanced Network Center (ANC)
Salzburg Research Forschungsgesellschaft mbH
Jakob-Haringer-Str. 5/III, 5020 Salzburg, Austria
bernhard.hechenleitner@salzburgresearch.at

30th August 2001

Abstract

The AQUILA project defines and implements a QoS architecture for dynamic end-to-end service provisioning in IP networks. This paper presents the distributed QoS measurement architecture (DMA) designed as a part of AQUILA. The scope of QoS measurement in AQUILA is the evaluation and validation of the AQUILA architecture and the support of the AQUILA resource control mechanisms. The measurement methodology in AQUILA comprises active and passive measurement. Active measurement is performed by synthetic application-like flows and by probing flows. While application-like flows are emulating real end-user applications, probing flows are thin measurement flows for monitoring the network behavior. Passive measurement in AQUILA relies on data gathered from different network elements. To implement the different measurement methods, three tools were developed: Measurement agents with application-like load generators for the measurement of end-to-end QoS of single and aggregated flow loads, measurement agents for active network probing to monitor the path performance characteristics and a router QoS monitor collecting management information from the routers for workload analyses. For the test scenarios and the measurement results, a measurement database was designed, which facilitates performance analysis and the correlation of the measurement results gathered from the different tools.

1 Introduction

The AQUILA architecture [5, 11] guarantees QoS parameters for end-user applications, like low delay, low packet loss and a specific amount of bandwidth. To evaluate and validate QoS architectures, active as well as passive measurement approaches can be used. While active measurements inject defined measurement packets into the network, passive measurements rely on data collected from the network elements (e.g. links,

routers). In AQUILA, measurements are also used to support QoS mechanisms like resource control and admission control. Therefore an integrated measurement environment for both measurement methods has been developed. Within the integrated environment, the results of the different measurements are correlated. This can express reasons why the targeted QoS parameters are reached or not. To carry out both active and passive measurements, several tools have been developed [10].
To avoid ambiguity of measurement results from different tools, the QoS parameters are measured with respect to standardized metrics [19]. Currently the following QoS parameters are measured:

- **One-way Delay:** The time required by a measurement packet from the source to the destination [1].
- **IP Packet Delay Variation:** The difference between the one-way delay between two selected (usually consecutive) measurement packets [7].
- **One-way Packet Loss:** The number of lost packets in a stream of packets from a source to a destination [2].
- **One-way Packet Loss Patterns:** The patterns of packet losses in a stream of packets, e.g. the number of consecutive lost packets, the number of loss periods, etc. [14]
- **Goodput/Throughput:** The rate perceived by a flow in bit/s. In case of TCP flows, the goodput of the payload is measured.

The document is structured as follows. The next section describes the general approach for QoS measurements in AQUILA and the derived measurement requirements. Section 3 shows the measurement architecture designed for AQUILA and its distributed characteristics. In section 4 the implementation components and their interaction is described. Finally, section 5 ends up with the future plans and conclusions.

2 QoS Measurement Approach and Measurement Requirements

QoS measurement is a central component for a dynamically operating QoS provisioning architecture like AQUILA, as the efficiency of resource allocation and the grade of the experienced QoS is directly connected to a reliable traffic management. Traffic management comprises short-term traffic control and long-term traffic engineering. Traffic control in the sense of AQUILA means all kinds of automatic mechanisms for effective resource allocation and QoS provisioning (e.g. admission control) and traffic engineering means all kinds of longer-term valid concepts (e.g. for the most effective distribution of several kinds of traffic into separate classes or for the most effective distribution of resources among several resource pools). The studies of traffic engineering concepts lead to specific traffic control mechanisms. For a validation of the concepts of AQUILA's traffic engineering [9], the architecture has to be tested and verified to get important feedback and if necessary improve the concepts. Feedback can be provided by means of QoS measurements. For carrying out AQUILA's traffic control mechanisms, feedback in the sense of measurements could also be very helpful, so that the traffic control mechanisms are not decoupled from the actual network status.
Hence, the deployment of QoS measurements serves for reaching two goals. On one side, there must be a mechanism to validate the traffic engineering concepts of the AQUILA architecture, which are - in corporation - responsible for the difficult task of dynamic resource allocation. On the other side the execution of dynamic resource allocation seems to be impracticable without having feedback information about the current status of the network.
Generally, measurement methods can be classified in various ways. One kind of classification is the distinction between direct and indirect measurements [8]. Indirect measurement methods rely on network models and assumptions, e.g. measurements are done only at network ingress points and further evaluation is done

by calculation with respect to models of the actually used mechanisms throughout the architecture. Obviously, this kind of evaluation has inherent uncertainties. Direct measurement methods do not rely on any models or expected behaviors but only on direct traffic observation at several points within the architecture. With respect to the measurement requirements of AQUILA, only direct measurement methods will be considered.

Another kind of classification of measurement methods is the distinction between passive and active measurement methods. Passive measurement methods collect information without disturbing network operation or interfering with operational network traffic[1]. Examples of measurement systems which facilitate passive measurements are SNMP-based network management tools, tcpdump [15], NetFlow [4] or DAG-card based systems [21]. Active measurement methods inject measurement traffic into the network and therefore interfere with operational traffic. Active measurement systems include e.g. NIMI [18], Surveyor [13] and AMP [17]. The AQUILA QoS measurement approach integrates both active and passive measurement methods.

A third kind of classification of measurement methods is the distinction between aggregation-based measurement and sampling-based measurement [8]. Aggregation-based measurement methods collect and process data before providing results (e.g. NetFlow). Such measurement methods obviously have the disadvantage of loss of information. Sampling-based measurement methods provide detailed measurement results of all or a subset of possible observations providing e.g. deep insight in the progression of a measured QoS metric. Both methods are used by the proposed measurement approach.

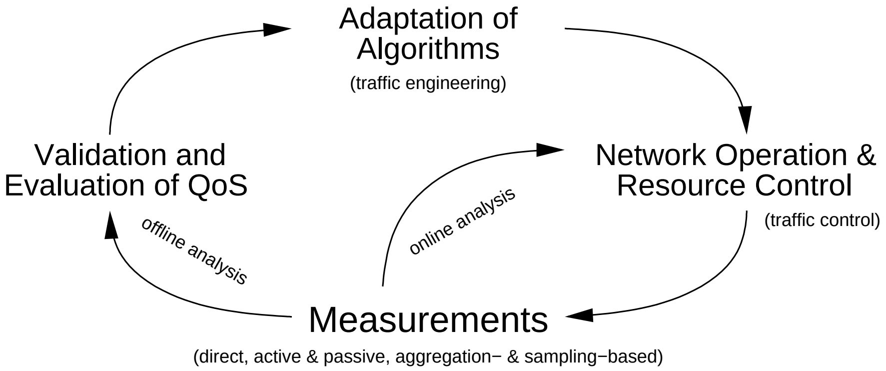

Figure 1: QoS Measurement Approach

The twofold approach for the measurements used within AQUILA is depicted in Figure 1. On the one hand, measurements are done to evaluate and validate the overall QoS architecture. Therefore application-like measurement flows are sent to the different traffic classes of the network and their resulting QoS parameters are measured and evaluated. With respect to the different classifications of measurement methods, this method is direct, active and sampling-based. If the validation fails, the algorithms and parameters of the resource control can be adapted. The evaluation and validation is termed as *offline result analysis*.

On the other hand, the measurement results support the network operation and the resource control mechanisms of the QoS network. Direct active and passive measurements can detect, whether the network is overloaded or overprovisioned. In such cases, the resource control can reject/admit further flows from/to

---

[1] In practice, some measurement tools do in fact interfere with operational network traffic for the sake of transporting gathered information to another place in the network or they disturb network operation in the sense that collecting and processing measurement data consumes processing time and other resources of network elements. Nevertheless, these interferences should be kept to a minimum and such tools are also called passive measurement tools.

the network during network operation. An automatic feedback loop can be established. This part is termed as *online result analysis*.
As measurements within AQUILA are used for both, the support of resource control as well as for the evaluation of the QoS architecture, different measurement requirements have to be met.
For the evaluation of the architecture we have to verify, whether the measured QoS parameters are at least equal to the targeted QoS parameters and whether the service provisioning was appropriate among the traffic classes. This is necessary to assure, that the available resources of bandwidth were utilized to an optimum.
Different kinds of measurements are used to support the resource control within AQUILA. These measurements must not influence the network operation significantly, but have to be done often enough to retrieve a actual view of the network situation. This can be reached by monitoring the network performance with passive measurements (passive monitoring) and low bandwidth consuming active measurements (active probing).

3 AQUILA Distributed QoS Measurement Architecture (DMA)

3.1 Architecture Overview

The distributed QoS measurement architecture designed for AQUILA consists of three separate tools.

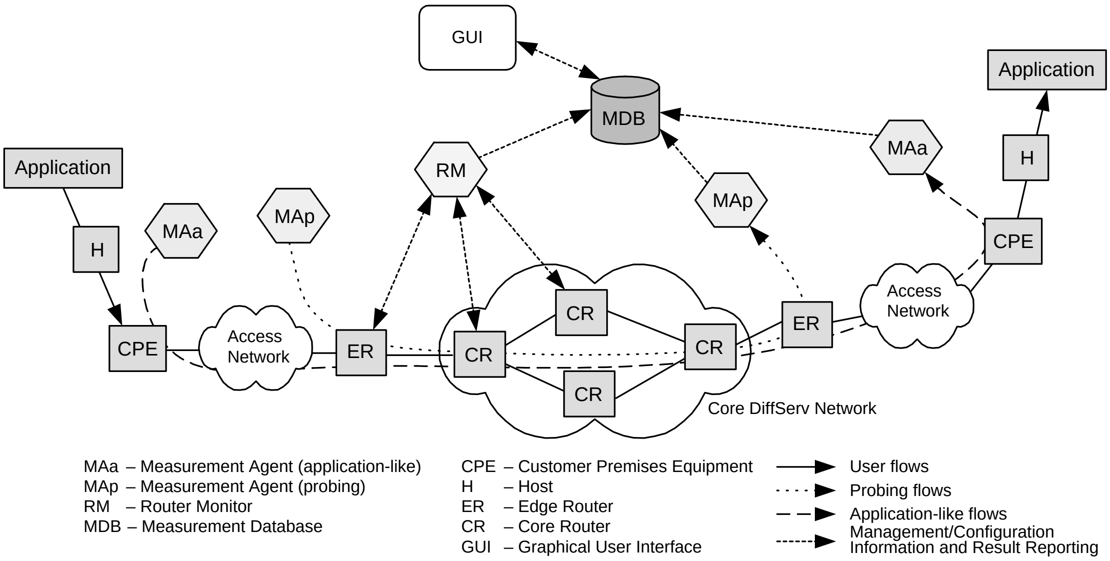

Figure 2: AQUILA Distributed Measurement Architecture Components

- **Application-like Measurement Agents (MAa):** These agents produce traffic that follows different Internet applications, like FTP, WEB, VoIP and audio-/video streaming. As the application-like measurement agents are emulating end-user traffic, they are located near the users' end-hosts.

- **Probing Measurement Agents (MAp):** These agents inject probing packets into the network, to evaluate the path performance characteristics of the network. As they are designed to support the network operation for ISPs, they are located at the providers network edges.

- **Router Monitor (RM):** Monitors QoS related parameters from core and from edge routers to get a view of the network situation and to detect possible bottlenecks.

Figure 2 shows an overview of the interaction of the components. The integration of the three tools is established through the introduction of the following additional components:

- **Measurement Database (MDB):** Is the central entity, which stores the measurement scenarios and results of the distributed QoS measurements.

- **Graphical User Interface (GUI):** Provides the interface for interaction between the user and the measurement system.

3.2 Description of Architecture Components

3.2.1 Application-like Measurement Agents (MAa)

The intention of the application-like measurement agents is to emulate emerging Internet applications like Voice over IP (VoIP) or audio/video streaming, but also traditional Internet applications like web surfing and ftp downloading. The load produced by such applications is measured and analyzed in a separate task. The collected data is either investigated to find traffic models for this applications or the traces (e.g. retrieved with tcpdump) are used to feed the load generators of the measurement agents to emulate the traffic. This can be done for both, single flow loads and aggregated flow loads. The end-to-end QoS of this emulated traffic is measured and evaluated. To implement the traffic models, currently three different distributions (constant, exponential and uniform) can be parameterized for the packet size as well as for the inter-departure time of the packets. With the possibility to start the load generators trace-driven, new traffic types can be introduced easily. To simplify the generation of many equal traffic flows at the same time, a multiplex functionality is implemented, which starts several equal flows at the same time.

As depicted in Figure 2 the sender MAa generates a synthetic traffic flow and the corresponding receiver MAa processes the received packets with regard to performance metrics like packet delay, delay variation, loss rate, etc. The measured performance metrics are sent back to the measurement database (see section 3.2.4) to allow an *offline result analysis* of the achieved and a comparison with the expected performance. The measurement database distinguishes between two kinds of results. Raw data contains the sending and receiving timestamps, a sequence number and a packet state information of every single measurement packet. To save space in the measurement database, the raw data can be aggregated to statistical values for defined measurement periods. These calculations are performed by the receiving agent, which then reports only the compressed aggregated results.

3.2.2 Probing Measurement Agents (MAp)

The probing agents measure end-to-end QoS parameters between defined pairs of senders and receivers on the IP layer. The idea of this active measurement is to inject small independent measurement packets into a network to get online results of the achieved IP performance metrics like end-to-end packet delay, packet delay variation, packet loss rate. The measurement results are stored in the measurement database.

The probing part of the AQUILA DMA allows the online monitoring of the mentioned parameters of a complete IP network. Therefore the measurement agents have to be ”fully meshed” which means that one measurement flow per traffic class is established from one agent to every other agent in the network. Measurement agents are situated near each of the edge devices of the network to be monitored.

The measurement results can be used for:

- online analysis of the path performance parameter (one-way delay, delay variation and packet loss) in a network under test,
- online analysis of the achieved and a comparison with the expected performance,
- detection of ”hot spots” in a monitored network,
- studying the asymmetry of end-to-end delay and packet loss.

3.2.3 Router Monitoring (RM)

The router monitor uses passive measurements to obtain QoS-related statistics from the router. It distinguishes between two types of routers, the core and the edge routers. The routers in the core network have usually a limited set of functions since they are supposed to handle great amounts of traffic as efficiently as possible. On the other hand, the edge routers between administrative domains or between the network provider and the customer have to make sure, that there is no misuse of network resources and that the processing load of the core routers is minimized. In the core routers of the network the statistics about traffic classes into which the flows in AQUILA are mapped, are monitored. At the edge routers of the network each single flow reservation can be monitored. The routers are polled in configurable time intervals. Examples of monitored parameters are the queue lengths, number of conforming/exceeding packets and bytes, number of dropped packets and the average CPU utilization.

3.2.4 Measurement Database (MDB)

The measurement database is the central entity of the distributed measurement architecture. It stores the test scenarios and the according measurement results. To enable multi-user functionality, each test scenario is assigned to the user who has specified the scenario. A test scenario consists of one or more flows between arbitrary measurement clients in arbitrary directions. Each flow is specified through its sender and receiver and a traffic model, which will be applied to the flow. The measurement results of the flow are then stored in relation to the flows.

The measurement processes (MAa, MAp, RM) are interacting using the measurement database. The correlation between the stored information (collected by the different tools of the DMA) is possible because all tools are storing timestamp information to the measured parameters.

The measurement database distinguishes between raw data and aggregated data. While raw data includes the sending and receiving timestamp of each single packet, the aggregated data is already compressed data over a period of time, including the results of several measurement packets.

3.2.5 Graphical User Interface (GUI)

To enable the management of the test scenarios, a platform independent graphical user interface for user interaction with the measurement system is provided. The GUI supports functions for the administration of users, configuration of load generators and their parameters, configuration of test scenarios, starting, stopping and monitoring of measurement flows and the offline browsing of the measurement results. For the configuration of a test scenario, the user specifies one or more flows. A flow is mainly specified through its source and destination IP address, the transport protocol to be used and the distributions and parameters of the traffic model. In addition, the user can choose a port number for the flow (otherwise it is assigned automatically), a time interval for the aggregation of results and the result options to specify the parameters to be measured. For application-like measurement flows a start time and a duration or end time or the number of packets to be sent can be specified. For probing flows, the flows will be started and stopped manually by the network administrator.

To evaluate the measurement results, the GUI provides different types of output. With the flow monitor, the measured aggregated results of flows can be monitored. For a detailed offline analysis of the measurement results (including raw data), simple graphs for a fast analysis are available, but also the raw data itself can be downloaded for further analysis with mathematic and statistic tools.

4 Current Implementation

The implementation of the distributed QoS measurement architecture is currently based on standard PCs (Intel architecture, Pentium II or better) with the Linux operating system (Kernel 2.2.x and higher). As depicted in Figure 3, the implementation is divided into two major parts. The measurement management station, which hosts the database, the management processes, the router monitor and the web-server for the GUI and several measurement client stations, which host the measurement agents for active measurements.

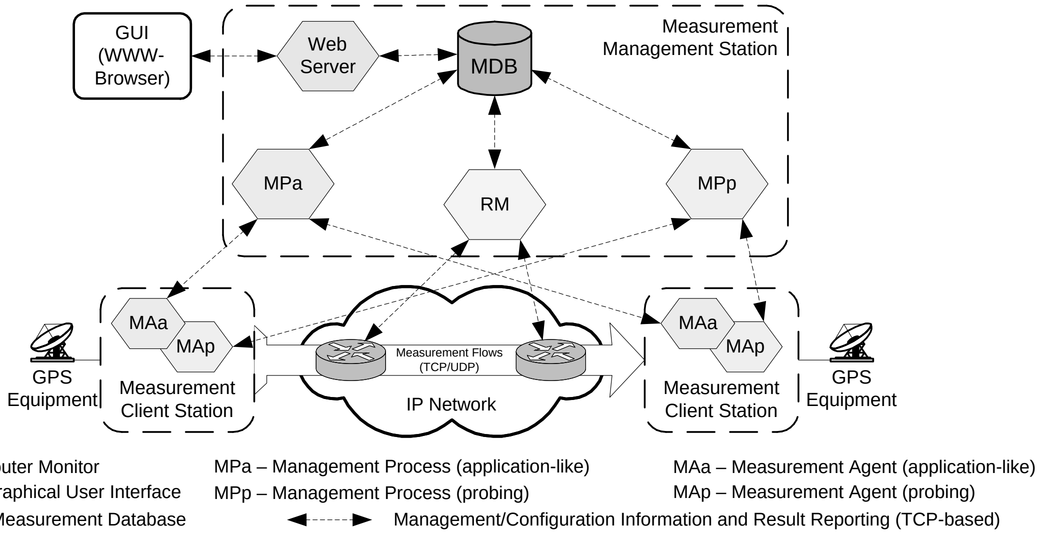

Figure 3: Implementation components and their interaction

4.1 Measurement Management Station

The measurement management station is responsible for distributing the test scenarios among the measurement client stations, storing the scenarios and measurement results, querying the router statistics and hosting the web server for the GUI. The different tasks are managed by separate processes. In principle, the tasks could also be distributed to several machines, but to reduce the network traffic (produced by data exchanges between the processes) the integration to one machine is recommended.

4.1.1 Management Processes (MPa, MPp)

Two management processes are running on the measurement management station. One for the probing measurement agents and one for the application-like measurement agents. The management processes request the measurement database in regular intervals for new test scenarios and distribute these scenarios to the specified measurement agents. Furthermore, they retrieve the result reports from the measurement agents and write them into the measurement database. If measurement agents are not reachable by the management process, a flag is set in the database.

In the case of monitoring large networks with active network probing, the MPp can be distributed into different network regions. In this case, the measurement agents (MAp) will report the results to the nearest management process, which then reports aggregated results back to the measurement database.

4.1.2 Router monitor (RM)

The router monitor queries the routers located along the path of a measurement flow. The communication between the router monitor and the routers is done via a telnet connection to the command-line interface (CLI) of the routers. For different types of routers, different commands are necessary for requesting the QoS related parameters. For each of the supported router types a configuration file is available. Another possibility to query the routers would be the use of the simple network management protocol (SNMP). While the CLI provides all available parameters related to the DiffServ implementations, SNMP currently only provides a limited set of parameters in the vendor specific parts of the management information bases (MIBs).

4.1.3 Measurement Database (MDB)

The measurement database stores the test scenarios as well as the appendant measurement results in a relational manner. In addition, the information about the measurement agents with their current state and the path information of the measurement flows is stored. The measurement database consists of 13 related tables and is implemented on the relational database management system *MySQL* [16].

4.1.4 Graphical User Interface (GUI)

The graphical user interface is a web-based application accessible from several client PCs using a JavaScript enabled standard web browser. This also implies platform independency of the GUI. To serve the web-pages, the *Apache* HTTP server [3] with the *PHP4* extension [20] is used. To allow multi-user access, the users are authenticated.

The specification of test scenarios is done via HTML forms. The display of the results uses tables for textual representation and *Gnuplot* for a graphical representation of the results. Also the output as plain text (comma separated values) is provided to enable the processing of measurement data with mathematic and statistic tools.

4.2 Measurement Client Station

The measurement agents situated on dedicated measurement client stations are responsible for the execution of the active measurements and reporting the results back to the management station. For the different measurement tasks (application-like, probing) separate measurement agents (MAa, MAp) exist.

To be able to measure exact one-way delays of the measurement packets, the measurement agents use the time information provided by the Global Positioning System (GPS). Therefore the hosts need to be equipped with GPS cards, which are connected to GPS antennas. For the AQUILA testbeds GPS equipment from *Meinberg* is used. The GPS equipment is accessed by a network time protocol (NTP) daemon to synchronize the internal system clocks of the measurement client stations. The accuracy reached using this approach is about 60 μs.

The measurement agents are multi-threaded daemon programs implemented in C++, listening for new measurement tasks from the corresponding management processes. After retrieving the measurement task with certain parameters, the measurement agent starts itself either as sending or receiving agent. When starting as sending agent, it establishes the measurement connection to the specified receiving agent and starts sending data. The receiving agent gets the enumerated and timestamped packets, adds another timestamp, processes the packets and stores the results until the reporting period ends. When the reporting period has ended, the collected data is sent to the corresponding management process.

5 Conclusions and Future Plans

The described implementation of the distributed QoS measurement architecture was successfully used in the first trial of the AQUILA project for the evaluation and validation of the AQUILA architecture. The measurement results showed that the AQUILA network services generally met the requirements [6, 12].

Valuable feedback was gained from the trial sites which will be taken into account for the development of the next versions of the measurement tools. The following improvements are projected:

Enhancement of Load Generators: Future load generators will support a wider range of distributions to enable a more flexible traffic modeling of single flow and aggregated flow loads. Also some predefined traffic models will be provided for already investigated applications.

Improvement of GUI: The GUI will be improved in user friendliness and will provide more functionalities. Planned enhancements are some macros for the flow definition and a search function for measurement flows and results.

Automatic Reservation Requests for Measurement Flows: Measurement flows will automatically request a flow reservation before they start. This is possible through an API provided by the AQUILA architecture.

Provision of a measurement interface: To enable the AQUILA resource control layer (RCL) or other applications to request the current state of the QoS network, an interface to the measurement results will be provided.

Storage of GPS coordinates: As the measurement agents are equipped with GPS cards, they will store their coordinates in the measurement database. This enables the spatial localization of the measurement agents within the providers network.

Acknowledgments

This work was performed in the framework of the IST Project AQUILA (Adaptive Resource Control of QoS Using an IP-based Layered Architecture - IST-1999-10077) funded in part by the EU. The authors wish to express their gratitude to the other members of the AQUILA Consortium for valuable discussions.

References

[1] G. Almes, S. Kalidindi, and M. Zekauskas. A one-way delay metric for IPPM. RFC 2679, IETF, September 1999.

[2] G. Almes, S. Kalidindi, and M. Zekauskas. A one-way packet loss metric for IPPM. RFC 2680, IETF, September 1999.

[3] Apache Software Foundation. The Apache HTTP Server Project. Available at: http://httpd.apache.org.

[4] Cisco Systems. White Paper: Netflow Services and Applications, June 2000. Available at: http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm.

[5] AQUILA consortium. Adaptive Resource Control for QoS Using an IP-based Layered Architecture. AQUILA homepage: http://www-st.inf.tu-dresden.de/aquila/.

[6] M. Dabrowski, A. Bak, A. Beben, W. Burakowski, Z. Kopertowski, and H. Tarasiuk. On handling streaming and elastic traffic in IP-based AQUILA network: measurement results. In *Proceedings of COMCON8, Crete, Greece*, June 2001.

[7] C. Demichelis and P. Chimento. IP packet delay variation metric for IPPM. Internet Draft, IETF, March 2001. Work in progress.

[8] N.G. Duffield and M. Grossglauser. Trajectory Sampling for Direct Traffic Observation. In *Proceedings of ACM SIGCOMM 2000. Stockholm, Sweden*, August 2000.

[9] F. Ricciato et al. Specification of traffic handling for the first trial. AQUILA deliverable D1301, July 2000.

[10] F. Strohmeier et al. Report on the development of measurement utilities for the first trial. AQUILA deliverable D2301, September 2000.

[11] M. Winter et al. System architecture and specification for first trial. AQUILA deliverable D1201, June 2000.

[12] Z. Kopertowski et al. First Trial Report. AQUILA deliverable D3201, September 2000.

[13] S. Kalidindi and M.J. Zekauskas. Surveyor: An infrastructure for internet performance measurements. In *Proceedings of INET'99, San Jose, CA, USA*, June 1999.

[14] R. Koodli and R. Ravikanth. One-way loss pattern sample metrics. Internet Draft, IETF, November 2000. Work in progress.

[15] LBNL's Network Research Group. Tcpdump: a protocol packet capture and dumper program. Available at: http://ee.lbl.gov.

[16] MySQL AB. MySQL Database Management System. Available at: http://www.mysql.com.

[17] National Laboratory for Applied Network Research (NLANR). Active Measurement Project (AMP), August 2001. Information available at: http://watt.nlanr.net.

[18] V. Paxson, A. Adams, and M. Mathis. Experiences with NIMI. In *Proceedings of PAM2000. Hamilton, New Zealand*, April 2000.

[19] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. Framework for IP performance metrics. RFC 2330, IETF, May 1998.

[20] PHP Group. PHP: Hypertext Preprocessor. Available at: http://www.php.net.

[21] Waikato University. DAG-Project: DAG4 architecture, February 2001. Information available at: http://dag.cs.waikato.ac.nz/dag/dag4-arch.html.