Evaluation of architectures for QoS analysis of applications in Internet environment

Ulrich Hofmann
University of Applied Sciences and Technologies
Schillerstraße 30
A-5020 Salzburg, Austria
ulrich.hofmann@fh-sbg.ac.at

Ilka Miloucheva
Advanced Network Center, Salzburg Research
Jakob-Haringer-Strasse 5/III
A-5020 Salzburg
ilka.miloucheva@salzburgresearch.at

Thomas Pfeiffenberger
Advanced Network Center, Salzburg Research
Jakob-Haringer-Strasse 5/III
A-5020 Salzburg
thomas.pfeiffenberger@salzburgresearch.at

Felix Strohmeier
Advanced Network Center, Salzburg Research
Jakob-Haringer-Strasse 5/III
A-5020 Salzburg
felix.strohmeier@salzburgresearch.at

Abstract

Advanced architectures for performance analysis of QoS enabled applications and services need to integrate different components which are not only concerned with the end-to-end QoS measurement and monitoring, but also with network path performance analysis and QoS modeling. Additionally, the performance analysis for QoS enabled applications and services requires the integration of further components for topology and autonomous system discovery as well as QoS/SLA verification dependent on the Internet environment (intra- and inter- domain).

To illustrate novel concepts for integrated performance analysis in an intra- and inter-domain environment, the Distributed Measurement Architecture (DMA) together with its component - CMToolset - developed in the framework of the European IST project AQUILA [1] is presented. CMToolset is used in AQUILA for end-to-end QoS measurement and verification considering traffic aggregation, multiplexing and resource reservation in DiffServ/MPLS environment. Usage scenarios for QoS analysis based on practical experiences with CMToolset are addressed.
The novel concepts of Inter-domain Distributed Measurement Architecture (IDMA) based on the enhancements of AQUILA DMA in the framework of INTERMON project [21] are discussed.
AQUILA DMA is compared with current state of the art of integrated toolkits for performance analysis of QoS enabled application and services.

1. Introduction

For performance analysis and optimization of QoS enabled applications in intra- and inter-domain environment QoS monitoring architectures based on integration of different kinds of QoS measurement components and techniques are required.

Different architectures for QoS monitoring are proposed in international research activities [4], [5], [10], [11], [12], [13]. Considering the current state-of-the art, following features are proposed for current advanced QoS monitoring architectures:

- Topology and router information discovery
- Event detection for performance problems
- Active probing for path performance characterization
- Passive monitoring of links for detection of performance problems and QoS verification
- QoS and traffic modeling toolkits
- Internet structure discovery (topologies, paths)
- Measurement databases with user interface for integration of the components.

This paper discusses the Distributed Measurement Architecture (DMA) of AQUILA [2] which integrates passive and active measurement toolkits with different functionality by using a common database and a common graphical user interface. The AQUILA DMA has been developed for the measurement and validation of QoS based (DiffServ/MPLS) Internet infrastructures[1].

CMToolset as a part of AQUILA DMA is focused on facilities for emulated traffic generation, QoS measurement and verification. CMToolset include functions to obtain the QoS parameter on transport level without considering the influence of routing path QoS. For performance analysis of applications in an inter-domain environment, the QoS of the inter-domain connections (between border routers) and routing parts within autonomous systems is to be taken into account.

In order to determine the impact of the inter-domain QoS on the end-to-end services and applications, additional facilities are proposed which are integrated in the Inter-domain Distributed Measurement Architecture (IDMA).

IDMA is based on novel concepts such as:

- integration of toolkits for end-to-end QoS monitoring and modeling, inter-domain path discovery and performance of routing paths on AS and IP level, as well as topology QoS and performance modeling on AS and IP level, using a common relational database

- configuration of integrated toolkits, data base access functions, and interaction control mechanisms, for instance inter-domain performance analysis of routing paths on AS and/or IP level

- measurement based modeling of application traffic and QoS on end-to-end and inter-domain (border router) connection level.

This paper is structured in the following chapters. Section 2 describes the QoS measurement methods, especially those found in the framework of AQUILA IST project. Section 3 discusses the distributed QoS measurement architecture – AQUILA DMA with special focus on CMToolset component, an active measurement toolkit for QoS analysis with integrated data base for QoS measurement results. Scenarios for CMToolset usage for QoS measurement and analysis are discussed. Section 4 focuses on the novel features of the IDMA required for QoS analysis in an inter-domain environment and measurement based QoS modeling. Section 5 summarizes the state-of-the-art of architectures for QoS measurement and analysis. The paper is concluded by section 6 with the further work on QoS data mining in the framework of INTERMON project.

2. Distributed QoS Measurement

2.1. QoS Measurement Requirements

QoS measurement is a central component for a dynamically operating QoS provisioning network like an AQUILA-network, 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 this sense 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 traffic engineering concepts, the architecture has to be tested and verified to get important feedback. Feedback can be provided by QoS measurements. For carrying out traffic control mechanisms, feedback in the sense of measurements could also be very helpful, so that the traffic control mechanisms are not de-coupled 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 QoS implementation, 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 [3]. Indirect measurement methods rely on network models and assumptions, e.g. measurements are done only at network ingress points and further evaluation is

[1] AQUILA (Adaptive Resource Control for QoS Using an IP base Layered Architecture) is an IST-project sponsored by the European Commission within the 5th framework program [1].

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.

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[2]. Examples of measurement systems which facilitate passive measurements are SNMP-based network management tools, tcpdump, Cisco's NetFlow or DAG-card based systems. Active measurement methods inject measurement traffic into the network and therefore interfere with operational traffic. Active measurement systems include e.g. NIMI [14], Surveyor [13] and AMP [15].

A third kind of classification of measurement methods is the distinction between aggregation-based measurement and sampling-based measurement [3]. 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.

2.2. AQUILA 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 over-provisioned. In such cases, the resource control can reject/admit further flows from/to the network during network operation. An automatic feedback loop can be established. This part is termed as *online result analysis*.

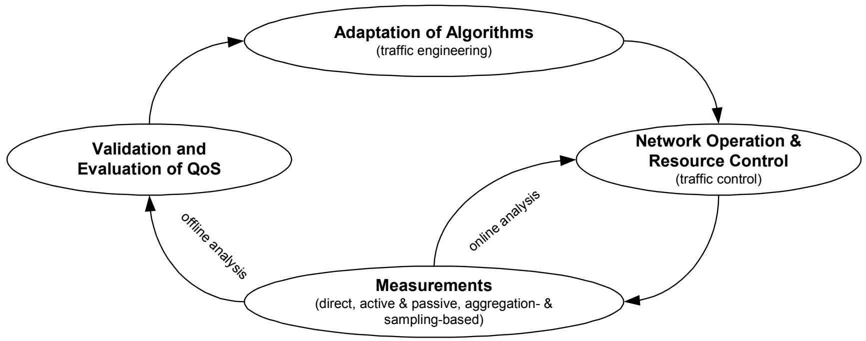


Figure 1: QoS Measurement Approach

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. The evaluation is done by loading the network with a background traffic and then start several flows of foreground traffic in the different traffic classes. The performance parameters of the foreground traffic are used to evaluate the delivered quality of service.

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

---

[2] 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.

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 Measurement Architecture

To be able to fulfill the measurement requirements for the evaluation and validation of the QoS based Internet infrastructures (especially based on DiffServ and MPLS technologies), AQUILA Distributed Measurement Architecture (DMA) has been designed and implemented.

3.1. Overview

AQUILA distributed measurement architecture consists of the following functional 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. As they are designed to support the network operation for ISPs, they are located at the providers edge systems.

- **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. While edge router monitoring enables per flow (i.e. reservation) analysis, core router monitoring enables per service class (i.e. queue) analysis.

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

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

- **Graphical User Interface (GUI):** Provides the interface for the user to control the single parts of the measurement system. The GUI is mainly divided into two parts, one for the scenario configuration and one for the measurement result analysis.

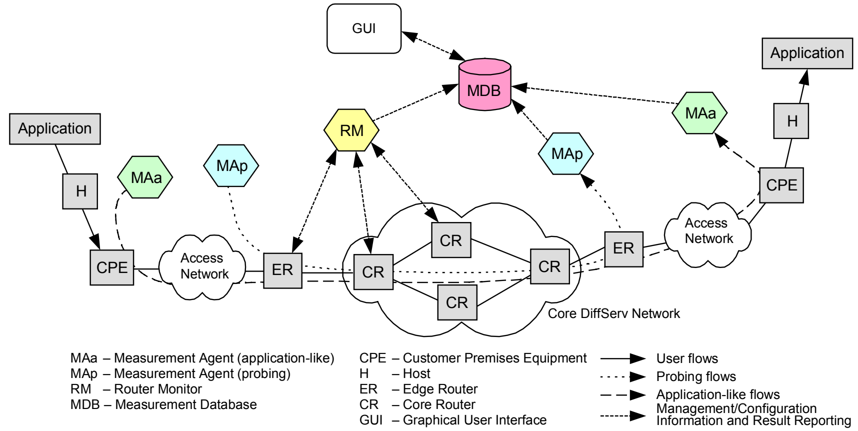


Figure 2: AQUILA Distributed Measurement Architecture

A more detailed description of the components of the AQUILA DMA can be found in [2]. The part of the AQUILA DMA which deals with active application-like measurements is covered by the Communication Measurement Toolset (CMToolset) and is described below in more detail.

3.2. "CMToolset" – a Measurement Tool in the Framework of AQUILA DMA

3.2.1. CMToolset Measurement Approach

The intention of application-like measurements 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 CMToolset produces synthetic traffic flows between two measurement stations either directly from trace-files which can be retrieved by network capturing tools like tcpdump or from different load models. The CMToolset load model is a generic, state-based model which supports different distributions for packet sizes and packet inter-departure times. The transition between the states is parameterized by a state duration and the probability to reach one of the following states. The parameters for the load models are stored in the measurement database and can therefore easily be reused. Also some pre-defined load models can be selected.

Once a trace is available or a traffic model has been parameterized, 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 to allow an offline result analysis of the achieved and a comparison with the expected performance. Measurement results for the synthetic flows are available on a per-packet basis (if selected), but can also be automatically aggregated in constant aggregation intervals.

Within AQUILA, resource reservations are necessary to request QoS from the network. For the synthetic measurement flows, also resource reservations can be invoked automatically for the measurement flows. If a resource reservation is specified, the resource request is sent to the AQUILA resource control layer. If the reservation was successful, the measurement flow will be started.

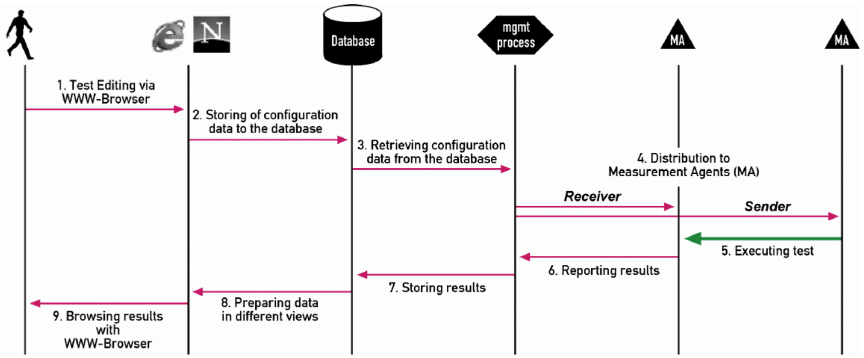


Figure 3: Process of a CMToolset Measurement

The CMToolset is implemented in a distributed manner, consisting of one central measurement server machine and several client machines. The server contains the measurement database, the graphical user interface and a management process for scenario distribution. The clients are equipped with distributed measurement agents, waiting for new measurement scenarios to be executed. The measurement results are communicated back to the management process on the measurement server, which is then in charge of storing the measurement results to the belonging measurement scenarios. The measurement process is depicted in Figure 3.

3.2.2. Measurement Scenario Configuration via GUI

To enable the management of the test scenarios, a platform independent graphical user interface (GUI) 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, monitoring of measurement flows and the online monitoring and offline browsing of the measurement results. In this section we focus only on the functions of the GUI, which are used for the configuration of application-like measurement flows. A detailed description of the GUI functionality can be found in [16].

Before single measurement flows can be specified, some preconditions must be met. E.g. there is a need, that at least two measurement clients are running and accessible from the measurement server. The measurement clients will register themselves at the measurement server.

For the generation of one or a group of application-like measurement flows, the parameters have to be specified as described in Table 1.

Parameter	**Description**
Sender and Receiver Host	The IP addresses of the sender and the receiver of the measurement connection have to be selected out of a list of available measurement clients.
Reservation	A resource reservation parameter set can be selected, if a reservation should be invoked for this flow. Currently the AQUILA resource reservation is supported. It can chosen, whether the measurement flow should also start, when the reservation fails.
Traffic	A traffic model for the traffic generator must be selected, which can be either a state-based, distribution-based or trace-file-based generator. The traffic models with their parameters are configured in a separate task. The already defined traffic models can be easily reused.

Multiplex	A number of flows can be entered, which will be generated simultaneously with the same behavior. This functionality allows the generation of flow aggregates in a single step.
Aggregation Time	A constant time interval, in which the gathered results will be aggregated to a so called aggregated result set. If no aggregation time is specified, there will be only one aggregated result set available over the whole measurement flow.
Path Discovery	An option, which forces the sender to discover the path via "traceroute" to the receiver. The path will be stored to the measurement database. The path can change over time or when using different service classes.
Schedule	The flow can be scheduled to a specific start time. To specify the length of the flow either a number of packets or an end time can be selected.
Multiflow	The GUI also provides a function, where multiple flows can be generated with one configuration. If this function is chosen, a number of flows and the start time of the first flow has to be specified. The duration of the flows is specified in seconds. The inter-arrival time of the flows is also specified in seconds. Both can be either constant or exponential distributed.

Table 1: Flow configuration parameters

The graphical user interface is implemented on a web-server and can therefore be used with most of the current Internet browsers. Result graphs are automatically generates using the measurement results available in the database.

3.2.3. Example Usage Scenario for QoS Measurement

This section presents an example for a usage scenario of the CMToolset . The aim is to find out the optimal multiplexing of application traffic for given resource reservation. In this example we emulated the multiplexing of a streaming video application.

The optimal number of streaming video connections which can be admitted to a link with 2Mbit/s must be found to meet the QoS requirement for no packet loss. Every 30 seconds an additional streaming video connections (160kbit/s) is started.

The scenario was configured with the GUI by using the "multiflow" feature. The number of flows in this configuration was 14, the inter-arrival time between the flows is defined with 30 seconds and the flow duration of all flows was 100 minutes.

Figure 4 shows the packet loss of the start period of the eleventh and twelfth streaming video connections, 10 streaming video connections are already running. It can be seen that packet loss increases at the start time of the eleventh streaming video connection at about 13:43:12.

A further decrease of the QoS parameter packet loss comes up at the start time of the twelfth streaming video connection.

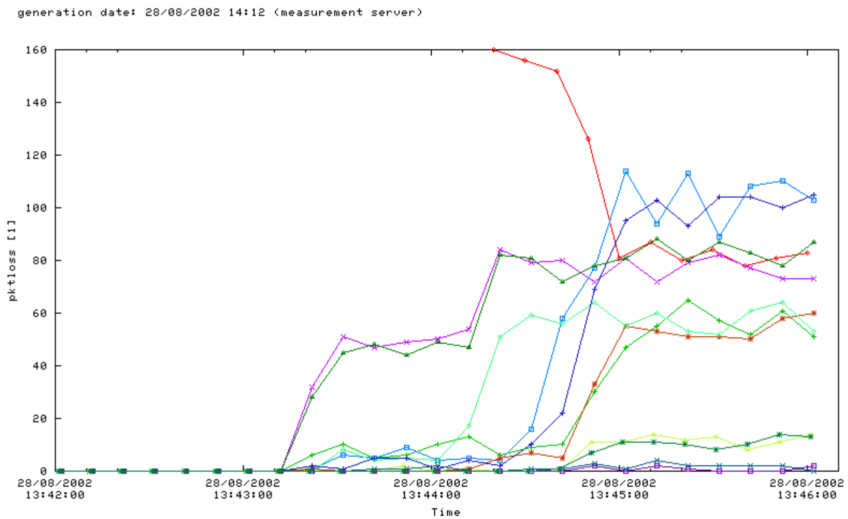


Figure 4: Multiplexed Streaming Video Connections

The experiment has shown, that in the case of starting the eleventh streaming video flow, there is significant increase of packet loss. Therefore the maximum number of multiplexed streaming video connections for this scenario would be ten.

3.2.4. Loss Estimation

Within this section it is discussed, how loss rate measurement results can be exploited to find the optimum operation point of a network.

3.2.4.1. Accurate Estimation of the Loss Rate

The mean loss rate will be estimated for independent and dependent loss events:

- **independent losses:** The number of losses is binomial distributed and the accuracy of the mean loss rate estimation can be calculated.

- **dependent losses:** The aggregation of a sufficient large number of losses into independent batches results in (Lindeberg-Levy Lemma) normal distributed batch mean losses and the confidence interval can be calculated.

The mean value estimation of a QoS value can be done in both, a stationary and a non-stationary environment. In a non-stationary environment older values are less weighted by applying smoothing algorithms.

Assuming a stationary environment the mean value can be estimated by using the Chi-square (χ^2) test (in case of a known variance) or by using the t-distribution (in case the variance is unknown).

Table 2 gives an overview on this approaches. The loss measurements of the flows are given by single loss events: loss_meas = {0,0,0,1,0,1,0,...}; 0 = packet sent, 1 = packet lost. By aggregation of loss measurements we mean the sequence of batches of summarized single loss events: {0,0,1,0,1,0} => 2/6. Such values are the result of the router loss statistics monitoring.

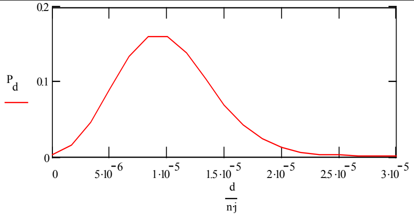
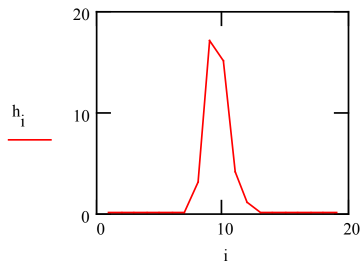
analysis type sample type	mean estimation M[parameter]=?
0,1 (e.g. loss events)	aggregation to batches {0,0,1,0,0,1} => M[loss event] = $^2/_6$ independent => binomial distributed dependent => normal distributed
Independet aggregated losses: binomial distributed (e.g. collected from router loss statistics)	
Dependent aggregated losses: normal distributed (e.g. collected from router loss statistics)	variance known: χ^2 test variance unknown: t-test  i ... number of loss events per interval h_i ... number of intervals with i loss events

Table 2: Mean Value Estimation

The objective of measurement control is to get a predefined level of accuracy with a minimum number of measurements.

In [16] there is a differentiation between “left” and “right” side confidence interval because it is less important to find

- the left side interval for the low target loss rate e.g. 10^{-6} (better is not critical)

- the right side interval for the higher tolerance loss rate e.g. $2*10^{-6}$.

In case this differentiation is not required both sides will be considered.

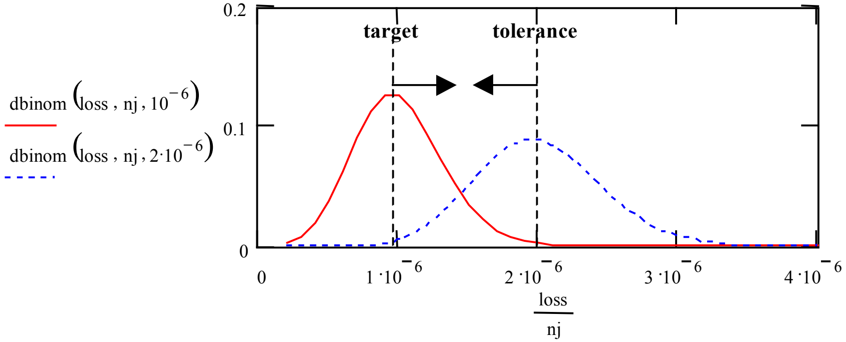


Figure 5: Left (tolerance) and right (target) side confidence intervals

Table 3 shows the right side α=0.05 intervals of the target loss rate for different loss probabilities p and sample sizes n*j.

n*j	2/p	3/p	4/p	5/p
right side α-interval	2p	1.9p	1.75p	1.7p

Table 3: Right side α=0.05 intervals for different loss probabilities and sample sizes

Example: For a loss probability of 10^{-5} we need $5*10^{5}$ samples to get the 1.7p right side confidence interval.

This knowledge allows us to quantify the run length of tests and to evaluate accuracy of the short term loss estimation in an operational monitoring.

3.1.4.2. Hypothesis Testing between Two Loss Rates

Above we have seen, that accurate loss rate is difficult to calculate because loss events are "rare events". For operational purposes the accurate loss rate is often not needed, a lower level of knowledge about the "truth" of one of the two hypothesis

- Hypothesis 0 = The actual status is QoS level 0: the target QoS e.g. loss rate $p0 = 10^{-5}$ and
- Hypothesis 1 = The actual status is QoS level 1: the tolerance QoS level, e.g. loss rate $p1 = 10^{-3}$

is sufficient. Such a test needs less samples compared with the accurate loss rate estimation. The hypothesis testing between two loss rates can be realized for two approaches:

- Maximum likelihood estimation of the crossing point from QoS Level i to QoS Level j, i,j=0 or 1 [17]).
- Wald's sequential testing [18].

Figure 6 shows the mean number of samples for α=0.05, p0=p_target=0.00001 in dependence of the tolerance loss rate p1. It can be seen, that a differentiation between p0 and 10^{-4} needs more samples then a differentiation between p0 and 10^{-3}.

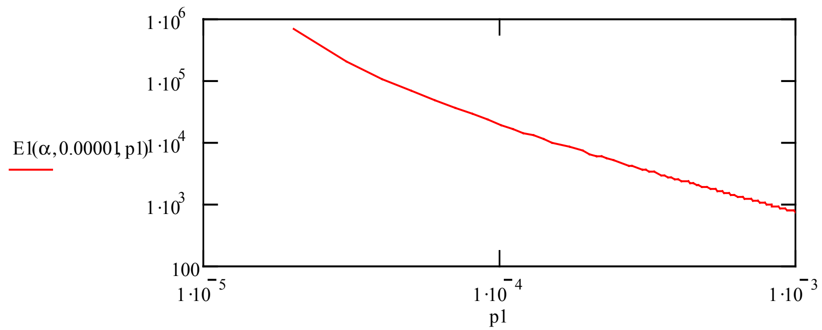


Figure 6: Mean sampling number for H1 hypothesis testing
H0: p = p0 = 0.00001; H1: p = p1 = 0.001

Obviously the necessary number of samples increases for p1 → p0, this is the reason why the differentiation between "near" values is more "difficult".

This leads to a heuristic approach: The differentiation between "near" (e.g. 10^{-5} and 10^{-4}) values is not so important for QoS management than the differentiation between values like 10^{-5} and 10^{-3}.

So a α-type1-error-function can be assumed. Figure 7 shows a linear α–function for $p0 = 10^{-5}$.

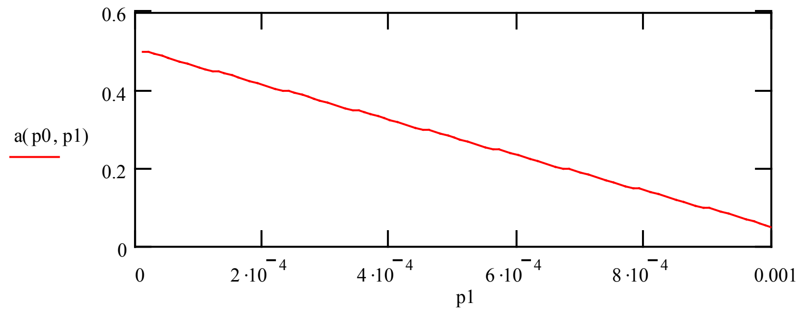


Figure 7: α-function: $\alpha\,(10^{-5},10^{-5})= 0.5$; $\alpha(10^{-5},10^{-3})=0.05$

Figure 8 shows the reduced mean number of necessary samples.

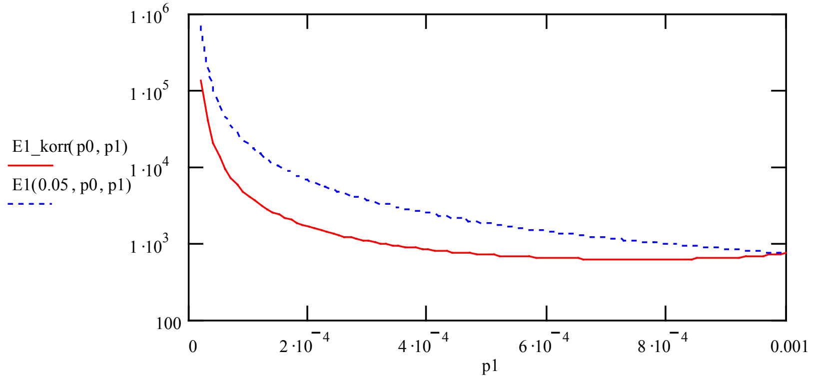


Figure 8: Reduced Number of Samples E1_korr for α(p0,p1), $p0=10^{-5}$

In the following simulation 200 experiments were made and the necessary number of samples until crossing the 10^{-3} threshold was determined. The solid line in Figure 9 shows the results of this simulation which is the number of samples c_n for the 200 experiments n=1,...200. The average number of necessary samples is 953.

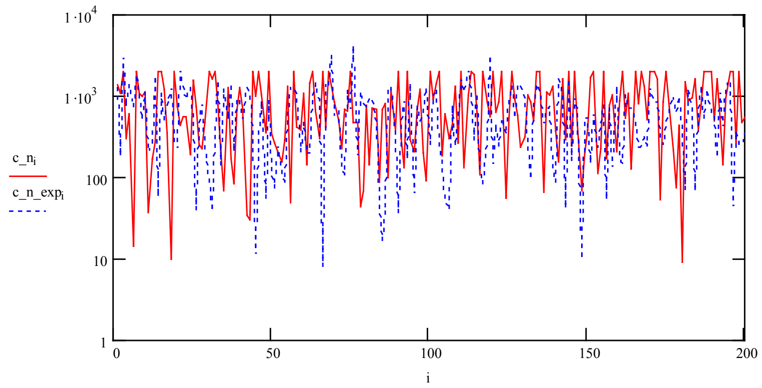


Figure 9: Number of necessary Samples for Hypothesis Admission

A second simulation was realized with the loss-epoch model c_n_exp [19]. The dashed line in Figure 9 shows the results when the series of lost packets shows exponential gaps. Using this model of losses the average number of necessary samples is 696.

For Poisson processes with independent arrivals the formulas are similar to the above formulas for the binomial loss process [20].

3.1.4.3. Conclusion

Sequential tests have benefits for operational applications because of their fast convergence. E.g. [18] shows that for α=β=0.05, p0=0.05, p1=0.1 the sequential test needs 144 samples and the Neyman-Pearson test needs 292 samples.

4. Inter-domain Distributed Measurement Architecture

The new concept of an inter-domain distributed measurement architecture (IDMA) enhances AQUILA DMA with novel techniques and components especially focused on QoS analysis in inter-domain environment.
Based on the experiences with the AQUILA DMA, especially the CMToolset, novel concepts are integrated in IDMA in order to

- enhance the QoS analysis of applications and services on hop-by-hop or inter-domain level, i.e. to provide measurement data for modeling of spatial composition of application QoS dependent on the routing path on IP or AS level

- to provide techniques for long and short term measurement based modeling of topology performance on IP or AS level

- to integrate QoS monitoring and modeling of application and service QoS based on SLA.

- to support inter-domain QoS/SLA verification of end-to-end services (on AS or IP level) .

IDMA is aimed at the integration of components for end-to-end QoS monitoring and modeling, inter-domain path discovery and performance of routing paths on AS and IP level as well as topology QoS and performance modeling on AS and IP level. IDMA components use common database.

The IDMA concept is illustrated in Figure 10.

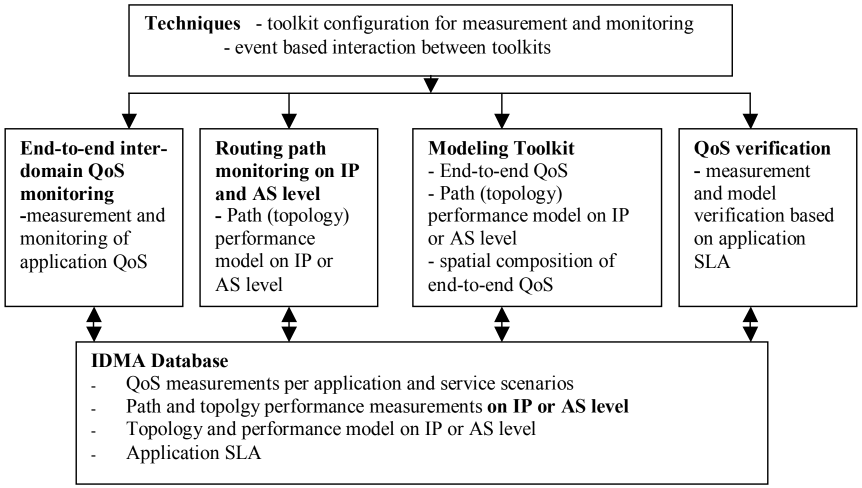


Figure 10: Concept of Inter-domain Distributed Measurement Architecture

5. Evaluation of QoS analysis architectures

AQUILA DMA [2] discussed in this paper is based on integration of different facilities:

- Measurement of QoS for emulated applications (VoIP, streaming audio and video)

- Scalable and distributed measurement for wide area networks,

- Data collection in a common measurement database,

- Integration of path QoS and application QoS monitoring with routing resource and MIB discovery.

Similar concepts of measurement architectures focussed on scalable network performance measurement and monitoring (especially path performance analysis) as well as integrated passive and active measurements are addressed by [4], [5], [7], [10], [12], [13], [14]. The paper "Measuring the Immeasurable: Global Internet Measurement Infrastructure" [8] compares different QoS measurement infrastructures with special focus on the different analysis types (passive and active techniques).

An example of active inter-domain QoS monitoring for interconnected DiffServ domains is the Qbone Internet2 network architecture [11]. An active measurement architecture focusing on collecting QoS parameter measurements from remote Internet hosts is discussed in [9]. RIPE NCC Test Traffic Measurement Service [12] for provision of active measurements as a regular service to ISP's is based on active measurements and traffic generation.

The CAIDA Network Modeling and Simulation project (http://www.caida.org/projects/nms/, see also [4]) for macroscopic Internet data measurement and analysis is aimed at integrated measurement and modeling on macroscopic Internet level.

The new focus proposed by IDMA discussed in this paper is to combine measurement and modeling of QoS in inter-domain environment. This approach is still not addressed in the current research on QoS analysis of applications in the inter-domain environment.

6. Conclusion

The QoS monitoring concept of AQUILA DMA and some scenarios based on the CMToolset tool are discussed in this paper. Based on the experiences with AQUILA DMA, the novel concept of the inter-domain distributed measurement architecture (IDMA) is developed and discussed in the paper.

The main trends in the QoS and performance analysis based on integrated measurement and modeling using common database are shown.

This work is continued in the INTERMON project [21]. Different measurement and modeling tools are integrated in INTERMON toolkit using common database for end-to-end and inter-domain QoS parameter as well as border router traffic flows obtained from IP Flow Import / Export (IPFIX) interfaces [22].

INTERMON further research is aimed at *visual data mining* for inter-domain traffic engineering and inter-domain QoS/SLA with focus on border router traffic flow modeling and spatial composition of inter-domain to end-to-end QoS.

Acknowledgements

This work has been performed in the framework of the IST projects AQUILA (Adaptive Resource Control of QoS Using an IP-based Layered Architecture – IST-1999-10077) and INTERMON (Advanced architecture for INTER-domain quality of service MONitoring, modelling and visualisation – IST-2001-34123), both partially funded by the European Commission. Both of the projects are cooperating in a cluster activity of European projects aimed at monitoring and measurement together with the projects CADENUS, NGNI and ATRIUM.

References

[1] AQUILA consortium, “Adaptive Resource Control for QoS Using an IP-based Layered Architecture”, AQUILA homepage: http://www.ist-aquila.org

[2] F. Strohmeier, H. Dörken, B. Hechenleitner. AQUILA Distributed QoS Measurement. In Proc. of COMOCON8 Conference, Crete, Greece, pages 177-185, June 2001.

[3] N.G. Duffield, M. Grossglauser. Trajectory Sampling for Direct Traffic Observation. In Proc. of ACM SIGCOMM 2000. Stockholm, Sweden, August 2000.

[4] Cooperative Association for Internet Data Analysis (CAIDA), Homepage: http://www.caida.org

[5] CAIDA's Skitter project web page. http://www.caida.org/tools/measurement/skitter/

[6] INTERMON consortium, “Advanced architecture for INTER-domain quality of service MONitoring, modelling and visualisation”, INTERMON homepage: http://www.ist-intermon.org

[7] B. Cheswick. Internet mapping project. http://www.cs.belllabs.com/who/ches/map/.

[8] M. Murray, KC Claffy, Measuring the Immeasurable: Global Internet Measurement Infrastructure, In Proc. of PAM 2001, p. 159-167, Amsterdam, April 2001

[9] M.Schmidauer, U. Hofmann, L.Villasenor, L.Lamont, I. Miloucheva: “Global Measurement Infrastructure of Application Classes and Their Bundling over ATM”, 4th European Conference on Multimedia Applications, Services and Techniques ECMAST´99 Madrid (Spain) 26-28 MAY 1999

[10] T. Hansen, J. Otero, T. McGregor, H.-W. Braun. Active Measurement Data Analysis Techniques. Available at: http://moat.nlanr.net/Papers/AMP_case_studies/

[11] Internet 2 Qbone: http://www.internet2.edu/qos

[12] RIPE NCC Test Traffic Measurements: http://www.ripe.net/ripencc/mem-services/ttm

[13] S. Kalidindi and M.J. Zekausus, “Surveyor: An Infrastructure For Internet Performance Measurements” (1999). Available at: http://bsdi4.advanced.org/~kalidindi/papers/INET/inet99.html

[14] V. Paxson, A. Adams and M. Mathis, “Experiences with NIMI”, In Proceedings of PAM 2000 (2000).

[15] Active Measurement Project (AMP). NLANR. http://watt.nlanr.net/

[16] F. Strohmeier et al. “Report on the development of measurement utilities for the second trial”. AQUILA deliverable D2303, December

2001.

[17] P. Papantoni-Kazakos, "Algorithms for Monitoring Changes in Quality of Communication Links". IEEE Transactions on Communications, Vol. COM-27, No. 4, April 1979

[18] M. Fisz, "Wahrscheinlichkeitsrechnung und mathematische Statistik". (in German) Akademie Verlag Berlin, 1978

[19] Y. Zhang, N. Duffield, V. Paxson, S. Shenker, "On the Constancy of Internet Path Properties". ACM SIGCOMM Internet Measurement Workshop 2001, San Francisco, USA. November 2001.

[20] J. Kiefer, J. Wolfowitz: Sequential tests of hypotheses about the mean occurrence time of a continuous parameter Poisson process. Naval Research Logistics Quarterly 3, No. 3, 1956.

[21] D. Hetzer, I. Miloucheva, U. Hofmann, J. Quittek, F. Saluta, Integrated Information System for Inter-Domain QoS Monitoring, Modelling and Verification, EURESCOM Konferenz , Oct. Heidelberg, 2002

[22] J. Quittek, T. Zseby, B. Claise, G. Carle, S. Zander, K.C. Norseth, Requirements for IP Flow Export, August 2002 (work in progress), Internet-Draft http://www.ietf.org/internet-drafts/draft-quittek-ipfx-req-05.txt