Title:	Document Version:
Deliverable D2.2 **Global Management Architecture of Measurement**	2.1

Project Number:	Project Acronym:	Project Title:
IST-2001-37611	6QM	IPv6 QoS Measurement

Contractual Delivery Date:	Actual Delivery Date:	Deliverable Type* - Security**:
28/02/2003	18/05/2003	R – PU

* Type:	P - Prototype, R - Report, D - Demonstrator, O - Other
** Security Class:	PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Responsible and Editor/Author:	Organization:	Contributing WP:
Frédéric Le Garrec	FT	WP2

Authors (organizations):

Jordi Palet (Consulintel), Jean-Michel Combes (FT), Alexandre Dubus (FT), Renée Le Viol (FT), Emile Stephan (FT), Lidia Yamamoto (HEL).

Abstract:

This activity specifies the needs of a standardized management interface and scalable management architecture to permit inter-domain measurement.

Keywords:

Architecture, Collect, Inter-domain Measurement, Management.

Revision History

The following table describes the main changes done in the document since its creation.

Revision	Date	Description	Author (Organization)
v0.1	05/12/2002	Document creation	Yann Adam (FT)
v0.2	10/12/2002	Update	Alexandre Dubus (FT)
v0.3	10/12/2002	Update Test Packet section	Emile Stephan (FT)
v0.4	10/12/2002	Update Security section	Jean-Michel Combes (FT)
v1.0	13/12/2002	Delivery	Emile Stephan (FT)
v1.1	15/01/2003	Name correction	Emile Stephan (FT)
v1.2	15/02/2003	Update	Alexandre Dubus (FT)
v1.3	19/02/2003	Style correction	Alexandre Dubus (FT)
v1.4	27/03/2003	Revision of Executive Summary, Introduction, Section 3 “Requirements”, Conclusions. Use “QoS” instead of “QOS”. Eliminated unknown references.	Lidia Yamamoto (HEL)
v2.0	15/04/2003	Final version	Alexandre Dubus (FT)
v2.1	18/05/2003	Final Review	Jordi Palet (Consulintel)

Executive Summary

The goal of this deliverable is to define a management architecture for QoS measurement such that users and service providers can have a common understanding of the performance and reliability provided by the Internet clouds that are traversed.

Defining the management architecture involves the following:

- Looking at existing passive and active measurement architectures and potentially extending or enhancing them.
- Examining the requirements for both intra and inter-domain measurements.
- Defining methodologies for intra-domain measurements.
- Standardizing the exchange of measurement results among heterogeneous measurement systems and across administrative domains, thus allowing for concatenation of global metrics.
- Defining the setup process for creating end-to-end measurements across administrative domains.
- Standardizing the format and semantics of test packets for interoperability between probes.
- Cross-fertilization with the security section should be achieved.

Table of Contents

Table of Figures

1. INTRODUCTION

The goal of this deliverable is to define a management architecture for QoS measurement such that users and service providers can have a common understanding of the performance and reliability provided by the internet clouds that are traversed.

Defining the management architecture involves the following:

- Looking at existing passive and active measurement architectures and potentially extending or enhancing them.
- Examining the requirements for both intra and inter-domain measurements.
- Defining methodologies for intra-domain measurements.
- Standardizing the exchange of measurement results among heterogeneous measurement systems and across administrative domains, thus allowing for concatenation of global metrics.
- Defining the setup process for creating end-to-end measurements across administrative domains.
- Standardizing the format and semantics of test packets for interoperability between probes.
- Cross-fertilization with the security section should be achieved.

2. EXISTING MEASUREMENT ARCHITECTURES

2.1 Introduction

2.1.1 QoS Definitions

The documents [Inter-2], [P806-2], [P1008-1] provide good states of the art regarding QoS definitions and QoS measurement parameters. ITU-T E.800 [E800] specification defines the quality of service as “the collective effort of the service performance which determines the degree of satisfaction of the end-user”. It also provides the various relationships between existing QoS components. These relationships can be separated in two main classes:

- User oriented components such as support performance (i.e. resolution of problems), operability performance (i.e. ease of use of the service), serviceability performance and finally security performance aspects.
- Network oriented components include performance aspects related to resources and facilities, dependability and integrity.

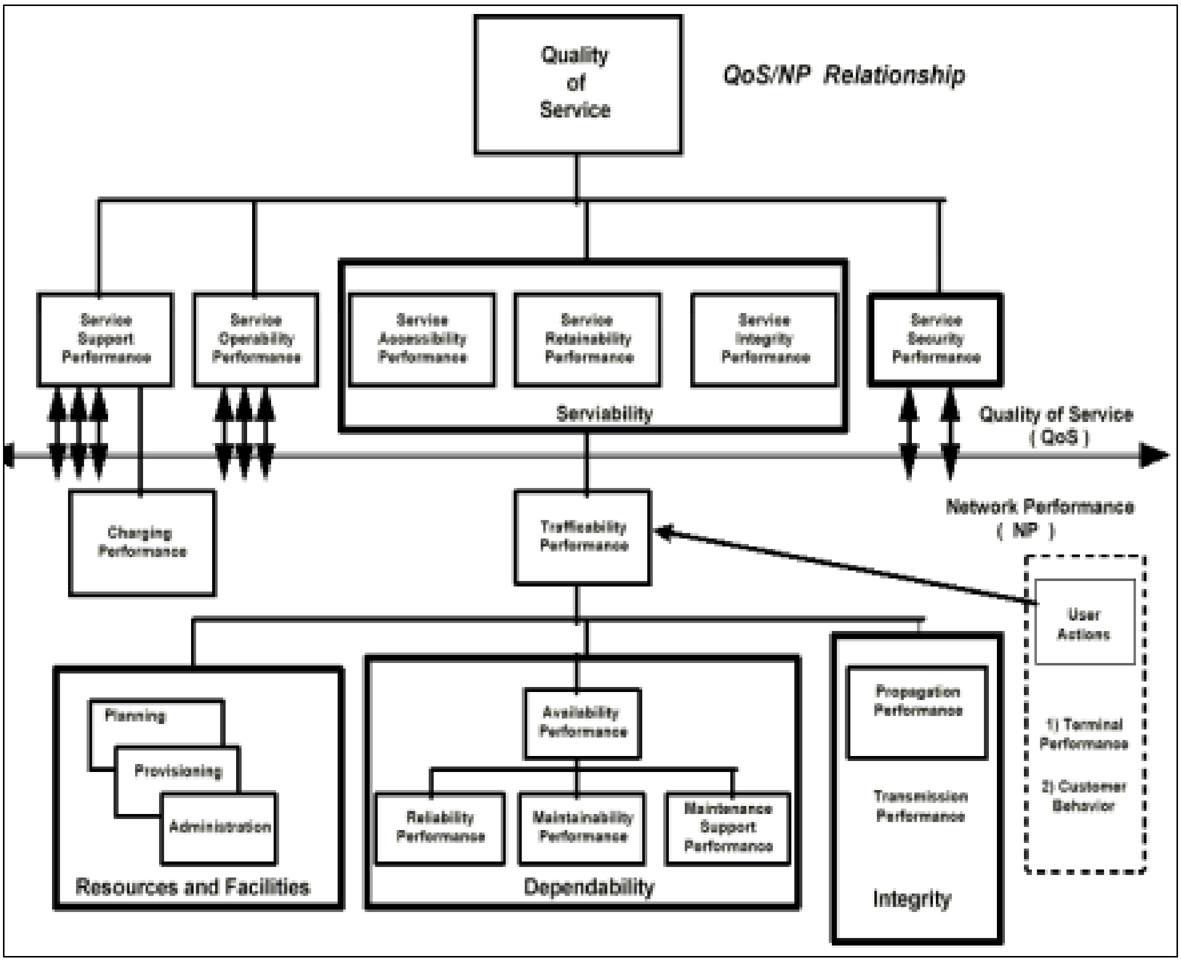

Quality of Service
QoS/NP Relationship
Service Support Performance
Service Operability Performance
Service Accessibility Performance
Service Retainability Performance
Service Integrity Performance
Serviability
Service Security Performance
Quality of Service (QoS)
Network Performance (NP)
Charging Performance
Trafficability Performance
User Actions
1) Terminal Performance
2) Customer Behavior
Planning
Provisioning
Administration
Resources and Facilities
Availability Performance
Reliability Performance
Maintainability Performance
Maintenance Support Performance
Dependability
Propagation Performance
Transmission Performance
Integrity

Figure 2-1: ITU-T E.800 Description of the Relations between QoS Components

2.1.2 Network Performance Measurement Parameters

2.1.2.1 ITU-T Activities

A lot of work has already been performed for the specification of network performance measurement parameters at ITU.

- ITU-T E.880 [E880] defines the guidelines for the collection of data related to dependability. Regarding measurement parameters, the main contribution of the specification is the definition of parameters regarding:
    - Reliability performance (e.g. failure rate, failure intensity, replacement intensity, mean operating time between failures and up time).
    - Maintainability performance (e.g. down time, technical delay, fault correction time, restart time, and various probabilities).
- ITU-T X.140 [X140] provides the definition of user-oriented QoS parameters such as access delay, incorrect access probability; access denial probability; user information transfer delay; user information transfer rate; user information error probability; extra user information delivery probability; user information misdelivery probability; user information loss probability; disengagement delay, disengagement denial probability; service availability; user information transfer denial probability and service outage duration in the case of data network.

2.1.2.2 IETF Activities

In the case of the Internet, most of the measurement parameters have been defined within the IPPM working group:

[RFC2330] specifies the general framework for QoS measurement within IP networks. The specification defines the general framework for the definition of IP performance metrics (IPPM), highlights the main measurement issues including the composition of metrics, time stamping issues, sampling methodologies as well as other measurement methodologies. Finally it defines the terminology to be used in the rest of IPPM related specifications. An important part of the terminology refers to the various types of metrics. In particular:

- A *singleton* metric is defined as an atomic metric resulting from measures made on one or several non-sampled datagrams.
- A *sample* metric is defined as a metric derived from a set of singleton metrics by selecting a number of instances of these metrics.
- A *statistical* metric is defined as a metric derived from a sample metric by computing some statistics on the singleton metrics located in the sample.

The following figure illustrates the relations between these three types of metric.

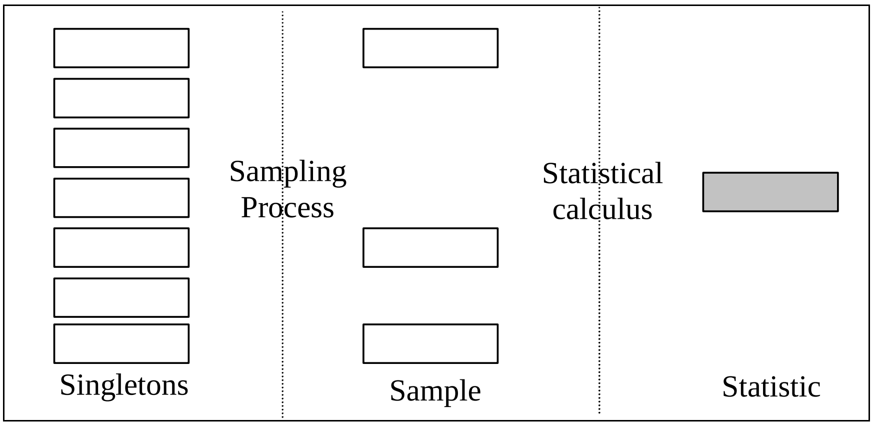


Figure 2-2: Metrics Computation Process

[RFC3148] extends the original framework for the measurement of Bulk Transfer Capacity (BTC). The main difficulty for the definition of a single BTC metric lies in the diversity of TCP implementations. In particular attention should be paid to the different types of congestion control methods implemented in existing implementations. As a result the framework suggest on this point to define BTC metrics per type of TCP implementation in order to perform unbiased measurements.

The following metrics have either been defined or are currently being defined by the IPPM working group:

- [RFC2678] defines a set of four metrics related to unidirectional and bidirectional connectivity measurement.
- [RFC2679] defines two one-way delay related metrics.
- [RFC2681] defines round six round trip delay related metrics.
- [Dem02] provides a set of metrics related to delay variation.
- [RFC2680] defines three metrics related to packet loss.
- [RFC3357] provides additional definitions derived from the singleton one-way packet loss metric provided in [RFC2680].
- [Mor02] provides a set of metrics related to packet ordering.

2.1.2.3 Comparison

Comparing ITU and IPPM metrics is an easy task. Even though metrics may carry similar names providing a precise mapping is difficult because metrics definitions while sharing similar goals are based on different assumptions. [Lel99] provides a comparison of ITU and IPPM approaches. [Jor00] provides an approximate mapping between IETF IPPM and ITU I.380 QoS measurement parameters. This comparison is provided in Table 1 below:

QoS Parameter	IPPM	I.380
Delay	One-way-delay,	IP Packet Transfer one-way-Delay
	One-way-delay-poisson-stream	Mean IPTD (one-way)
Delay variation	One-way-ipdv	IP Packet delay variation (one-way), end-to-end 2-point
	One-way-ipdv-stream	Average delay, Interval-based limits, Quantile-based limits
Loss	One-way-packet-loss	IP packet loss ratio (IPLR)
	One-way-packet-loss-stream	
	One-way-packet-loss-distance-stream	
	One-way-packet-loss-period-stream	
Transfer rate related	No	IP packet throughput (IPPT), Octet-based (IPOT)
		Destination limited source
		Throughput probe
Service Availability	No	IP Service Availability
Others	No	Spurious IP packet rate
	Non-Reversing-Order	No
Blocking	No	Defined in ITU-T E.493
Set-Up delay(s)	No	Defined in ITU-T E.493

Figure 2-3: Comparison between IPPM and I.380

2.2 QoS Measurement Architectures

2.2.1 QoS Measurement Definitions

In order to define QoS measurement methodologies and architectures, we first need to provide definitions regarding the components taking part in the measurement operations. ITU-T I.380 [Y1540] provides the following definitions:

- A *host* is a computer that communicates using the Internet protocol.
- A *router* is a host that enables communications between other hosts by forwarding IP packets based on the content of their IP header destination field.
- A *source host* is a host and a complete IP address where end-to-end IP packets originate.
- A *destination host* is a host and a complete IP address where end-to-end IP packets terminated.
- A *link* is a point-to-point connection used to transport packets between two hosts.
- A *network section* is a set of hosts and links that fall under a single jurisdictional responsibility.
- A *circuit section* is a link either connecting a source or destination host to its adjacent host or connecting two routers located in two different network sections.
- A *measurement point* is the boundary between a host and an adjacent link at which reference events can be observed.

[RFC2330] provides similar definitions for host, router and link and defines the following additional terms:

- A *path* is a sequence <h0, l0, h1,l1,…ln-1, hn> where hi is a host and li is a link.
- A *subpath* is a subsequence of a path. A subpath is itself a path.
- A *cloud* is an undirected graph whose vertices are routers and whose edges are links between routers.
- An *exchange* is a link connecting either a host to a cloud or two clouds together (i.e. equivalent to an ITU *circuit section*).
- A *cloud subpath* is a path where all host are routers within a given cloud (i.e. equivalent to an ITU *network section*).
- A *path digest* is a sequence <h0, e1, c1, …, cn, en, hn> where h0 and hn are hosts, ei are exchanges and ci are cloud subpaths.

In the rest of this section we will use IPPM terminology in the case where terms have a common meaning in IPPM and ITU specifications. In the other cases we will provide duplicate definitions when definitions are either lacking in one standard compared to the other or when definitions have different meanings.

2.2.2 QoS Measurement Architecture Classifiers

Consequently, existing QoS measurement architectures can be classified according to three main parameters.

- The intrusiveness of the measurement architecture. We define by measurement intrusiveness the level of alteration against the traffic flowing in the network generated by measurement operations.

- The organization of measurement component used to perform QoS measurement. We distinguish four main types of components and show how they can be combined.
- The scope of the QoS measurements that can be performed by the architecture. We define by scope the physical space in which measurements can be performed.

2.2.2.1 Measure Intrusiveness

We describe as a passive measurement architecture a measurement architecture where QoS measurements results are computed without introducing any alteration to the traffic monitored.

We define as active measurement architecture a measurement architecture where QoS measurements results computations require one or several of the following activities:

- The transmission of additional traffic in addition to the traffic to be measured.
- The modification of the original traffic to be measured.
- The generation of the traffic to be measured.

As a consequence the usage of active measurement techniques may introduce modifications in the measurement results.

According to the methodologies provided in [RFC2678], [RFC2679], [RFC2680], [RFC2681], [RFC3357], [Dem02], [Mor02] the relations between QoS measurement metrics and measurement architectures intrusiveness are provided in the following table. As can be seen IPPM activities mostly focus on active measurement methods. Note however that methodologies provided in these RFC are only examples. As a result, other relations between metrics and intrusiveness may exist. The table also provides the relationships between ITU-T I.380 metrics and related measurement methodologies. The Type-P semantic is explained in section 6.1.2.3.2.

Metric/Intrusiveness	Passive	Active
Type-P-*-Connectivity [RFC2678]		X
Type-P-*-One-Way-Delay [RFC2679]		X
Type-P-*-Packet-Loss [RFC2680]		X
Type-P-*-Round-Trip-Delay [RFC2681]		X
Type-P-One-Way-Loss-* [RFC3357]		X
Type-P-One-Way-ipdv-* [Dem02]		X
Type-P-Packet-Reordering-* [Mor02]		X
IP Packet Transfer Delay [Y1540]	X	
Mean IP Packet Transfer Delay [Y1540]	X	
Variation in IP packet Delay [Y1540]	X	
IP Packet Error Ratio [Y1540]	X	
IP Packet Loss Ratio [Y1540]	X	
Spurious IP Packet Rate [Y1540]	X	
IP Packet Throughput [Y1540]	X	
Octet Based IP Packet Throughput [Y1540]	X	
IP Service Availability [Y1540]	X	

Figure 2-4: Relationship between Metrics and Architecture Intrusiveness

2.2.2.2 Organization of Measurement Components

Most QoS measurement architectures follow the same goal. They try to capture some aspects of the traffic flowing in a network. However in order to do so, measurement architectures behaviors are bound between two opposites. On one hand the best way to understand the network traffic would be to capture every packet crossing the network and to forward it to a central location where packets could be stored and then matched in order to compute QoS measurement results. However this approach is usually not practical since it suppose to multiply network capacities in order to perform traffic measures.

On the other hand it may seem wise to perform all measurement operations as close as possible to the traffic in order to limit the size of the measurement information to be transported over the network. This approach can also be unpractical in many cases for several reasons. Network devices may have a limited computing power or limited capabilities regarding the building blocks described in the next section. As a result these devices may not be able to perform complex measurement operations. Moreover from a network operator point of view, the aggregation of information can be a good idea if the measurement parameters needed to solve network problems are known in advance. When this is not the case the aggregated information may be of little interest and less aggregated information about network traffic may be desirable. In order to satisfy the needs located between these two bounds, several different approaches coexist today.

In order to classify existing architectures we distinguish four basic components. It should be noted that many existing traffic measurement architectures combine several basic components into a single device and that oppositely, each basic components may be split between several physical components. Please also note many QoS measurement tools only provide the functionality associated with a given component.

Finally the four components we describe are based on basic building blocs for their implementation. These building blocs are described in detail in the next section.

2.2.2.2.1 Measurement point

We define the measurement point MP as the physical point where:

- An IP Packet Transfer Reference Event (IPRE) is captured according to ITU terminology.
- The wire-time at which a Type-P is captured can be measured according to IPPM terminology.

From a functional point of view the measurement point takes as an input a packet belonging to the traffic to be measured and produces as an output the original packet without modification (regarding the content of the traffic or its temporal characteristics), a copy of the packet and a timestamp precisely indicating at which time the packet has been captured. In addition to this information the measurement point may also add some local information related to the packet (e.g. originating network interface, BGP destination AS…). We later call this set of information (packet, timestamp, additional information) an extended packet. The measurement point function is based on the implementation of two building blocks:

- Packet copy.
- Packet time stamping.

2.2.2.2.2 Aggregation point

We defined the aggregation point AP as the physical point where time-stamped IP packets are selected according to an aggregation policy.

From a functional point of view the aggregation point takes as an input a set of extended packets provided by the measurement point an aggregation policy and produces as an output a set of aggregates. The aggregation policy for a set of time-stamped packets [(P1,T1) …(Pn,Tn)] expressed conditions and actions that may carry on:

- The temporal information associated to each packet (T1 …Tn).
- The content of each extended packet (P1…Pn).
- The rank i of a packet (Pi,Ti) within the set [(P1,T1) …(Pn,Tn)].
- The number of packets in the set [(P1,T1) …(Pn,Tn)].
- Predefined packet contents carrying either on the packet header or its content or additional information carried in the extended packet.
- Predefined temporal information such as time ranges, durations or specific random process distributions.
- A combination of conditions carrying on these parameters.

The aggregation point function is based on the implementation of three building blocks:

- Packet sampling.
- Packet classification.
- Packet hashing.

Note that oppositely to measurement points that have to be unique, aggregation points can be duplicated in order to chain aggregation policies.

2.2.2.2.3 Metrics computation point

We define as metrics computation point MCP the physical point where QoS measurement metrics are computed according to their definition.

From a functional point of view the metrics computation point takes as an input one or several aggregates (these aggregates can include all original IP packets if the aggregation policy does not exist) as well as a definition of the metrics to be computed (either using an ITU or IPPM definition) and produces as an output the value of a specific metric according to its definition.

Note that similarly to aggregation points, metrics computation points can be duplicated in order to compute several different metrics from the same aggregate.

2.2.2.2.4 User application point

We define as user application point UAP the physical point where QoS measurement metrics are actually used to provide QoS measurement related services.

From a functional point of view the user application point takes as an input one or several metrics values and produce an output according the application goal.

2.2.2.2.5 Example

The following figure provides a summary of possible relations between MP, AP, MCP and UAP.

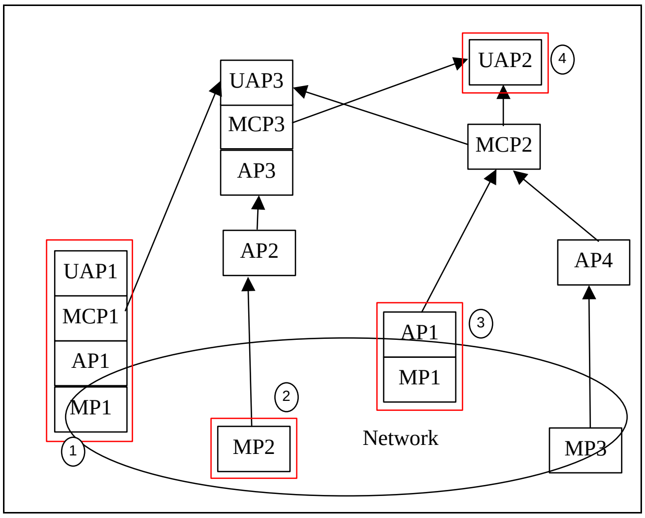


Figure 2-5: Examples of Possible MP, AP, MMP and UAP Combinations

In this example:

- Device (1) could in practice represent a host using the ping command to evaluate the IPPM ICMP-Request-ICMP-Response-Round-Trip-Delay between (1) and (3).
- Device (2) could in practice represent a DAG card exporting a time-stamped copy of packet received to an aggregation point.
- Device (3) could in practice be Cisco router exporting Netflow flows to a metric measurement point.
- Device (4) could in practice be a workstation running the HP openview software representing a summary of the flows flowing in the network. These flows are provided through a RTFM MIB located on Device (3).

2.2.2.3 Measurement Scope

Among various potential users of QoS measurement architecture, at least two main classes of users may have a particular interest in measurement architectures:

- For Network operators, QoS measurement is a mean to achieve several goals:
  - Measure the performance of the network.
  - Detect and identify problems.
  - QoS measurement is a mean to influence the billing of recorded service usage (accounting process) according to reported SLA violation and service problems
  - Measure the usage of the network and model future evolutions in traffic characteristics in order to plan future network evolutions.
- For Internet or network users, QoS measurement is:
  - Measure the quality of service provided by a network operator.
  - Measure the usage of the network and model future evolutions in traffic characteristics in order to plan future internal network evolutions.

In order to satisfy these different classes of users, measurement tools have been created that cope with the limitations carried by each class. For example network operators may not have access to network users infrastructure and similarly Internet users usually have limited access to network infrastructure since both usually have divergent interests. As a result it is clear that users of a given network have the ability to perform QoS measurement in different measurement scopes. We distinguish two main classes of measurement architectures related to these different scopes:

- *End-to-end measurement*. We define the end-to-end measurement architecture for a given measurement path P=<h0, l1, h1, …ln, hn> as the measurement architecture including two measurement points m1, m2 where m1 is located on host h0 and m2 is located on host hn.
- *Path measurement*. We define a k points path measurement architecture for a given measurement path P=<h0, l1, h1, …ln, hn> as the measurement architecture including k measurement points (p1…pk) where located on routers belonging to P such that:
  - For each i=1,k, there is one j, 0<j<n such that pi, is located on hj
  - There is no (i,j,l) (0<i,j<=k, 0<l<n) such that pi is located hl and pj is located on hl.

2.2.3 Standardized Measurement Architectures

In the case where measurement points, aggregation points, metrics computation and user application points are located on physically separated devices, transferring packets, aggregates and measurement results between these devices becomes essential. In this section we analyze existing standardized measurement architectures according to classifiers defined in the previous section.

2.2.3.1 RMON

[Wal02] defines the RMON framework. Remote Monitoring (RMON) is a standard monitoring specification that enables various network monitors and console systems to exchange network-monitoring data. The RMON specification defines a set of statistics and functions that can be exchanged between RMON-compliant console managers and network probes. The user community with the help of the Internet Engineering Task Force (IETF) defined RMON. It became a proposed standard in 1992 as RFC 1271 (for Ethernet). The current standard describing RMON is [RFC2819]. Several extensions have been defined that extend the capacity of RMON for different types of networks and environments.

RMON delivers information in ten RMON groups of monitoring elements, each providing specific sets of data to meet common network-monitoring requirements. Each group is optional so that vendors do not need to support all the groups within the Management Information Base (MIB). Some RMON groups require support of other RMON groups to function properly. Existing groups are described bellow:

The Ethernet statistics group contains statistics measured by the probe for each monitored Ethernet interface on this device.

The history control group controls the periodic statistical sampling of data from various types of networks.

The Ethernet history group records periodic statistical samples from an Ethernet network and stores them for later retrieval.

The alarm group periodically takes statistical samples from variables in the probe and compares them to previously configured thresholds. If the monitored variable crosses a threshold, an event is generated. A hysteresis mechanism is implemented to limit the generation of alarms.

The host group contains statistics associated with each host discovered on the network. This group discovers hosts on the network by keeping a list of source and destination MAC Addresses seen in good packets promiscuously received from the network.

The hostTopN group is used to prepare reports that describe the hosts that top a list ordered by one of their statistics. The available statistics are samples of one of their base statistics over an interval specified by the management station. Thus, these statistics are rate based. The management station also selects how many such hosts are reported.

The matrix group stores statistics for conversations between sets of two addresses. As the device detects a new conversation, it creates a new entry in its tables.

The filter group allows packets to be matched by a filter equation. These matched packets form a data stream that may be captured or may generate events.

The Packet Capture group allows packets to be captured after they flow through a channel.

The event group controls the generation and notification of events from this device.

2.2.3.2 IPPM

[Ste02] defines a MIB for managing the measures using the IP performance metrics specified by the IPPM Working Group. It specifies the objects to manage the results of the measure of metrics standardized by IPPM Working Group. They are built on notions introduced and discussed in the IPPM Framework document.

2.2.3.3 RTFM

The RTFM architecture is an attempt by IETF to standardize several aspects of flow definition, capture and metering operations [RFC2722]. The architecture has the following property:

- The traffic flow model can be consistently applied to any protocol, using address attributes in any combination at the 'adjacent', network and transport layers of the networking stack.
- Traffic flow attributes are defined in such a way that they are valid for multiple networking protocol stacks, and that traffic flow measurement implementations are useful in multi-protocol environments.
- Users may specify their traffic flow measurement requirements by writing 'rule sets', allowing them to collect the flow data they need while ignoring other traffic.
- The data reduction effort to produce requested traffic flow information is placed as near as possible to the network measurement point. This minimizes the volume of data to be obtained (and transmitted across the network for storage), and reduces the amount of processing required in traffic flow analysis applications.

From an architectural point of view the RTFM architecture is made of four components:

- Meters observe packets passing through measurement points classifies them into certain groups, accumulate usage data and store these results in flow tables. As such meters can be described as a combination of MP and AP according to our QoS measurement architecture classification.
- Manager: A traffic measurement manager is an application, which configures 'meter' entities and controls 'meter reader' entities. It sends configuration commands to the

meters, and supervises the proper operation of each meter and meter reader. It may well be convenient to combine the functions of meter reader and manager within a single network entity.

- Meter reader: A meter reader transports usage data from meters so that it is available to analysis applications.
- Analysis applications: An analysis application processes the usage data so as to provide information and reports, which are useful for network engineering and management purposes.

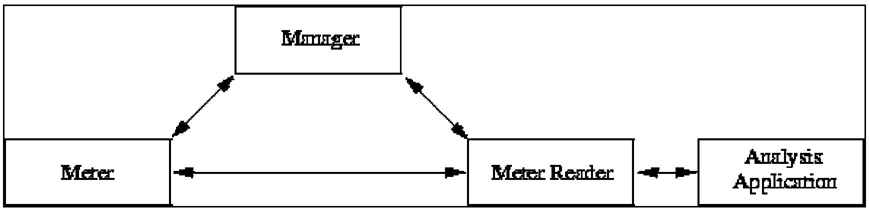


Figure 2-6: The RTFM Architecture

These components as well as the relation between components are presented in the RTFM architecture.

The RTFM working group has also defined additional components that may participate in the RTFM architecture:

- An RTFM MIB. [RFC2720] defines a Management Information Base (MIB) for use in controlling an RTFM Traffic Meter, in particular for specifying the flows to be measured. It also provides an efficient mechanism for retrieving flow data from the meter using SNMP.
- A rule set language. [RFC2723] defines a language for specifying rulesets, i.e. configuration files which may be loaded into a traffic flow meter so as to specify which traffic flows are measured by the meter, and the information it will store for each flow.
- Measurement Attributes Extensions for traffic flow measurement ([RFC2724]).

2.2.3.4 Sflow

[RFC3176] defines the sFlow technology. sFlow is a technology for monitoring traffic in data networks containing switches and routers. In particular, it defines the sampling mechanisms implemented in an sFlow Agent for monitoring traffic, the sFlow MIB for controlling the sFlow Agent, and the format of sample data used by the sFlow Agent when forwarding data to a central data collector.

The sFlow monitoring system consists of an sFlow Agent (embedded in a switch or router or in a stand alone probe) and a central data collector, or sFlow Analyzer. The sFlow Agent uses sampling technology to capture traffic statistics from the device it is monitoring. sFlow Datagrams are used to immediately forward the sampled traffic statistics to an sFlow Analyzer for analysis.

[RFC316] describes the sampling mechanisms used by the sFlow Agent, the SFLOW MIB used by the sFlow Analyzer to control the sFlow Agent, and the sFlow Datagram Format used by the sFlow Agent to send traffic data to the sFlow Analyzer.

2.2.3.5 IPFIX

[Nor02] defines the architecture for IPFIX. The main objectives of this document are to describe the key architectural components of IPFIX, define the architectural requirements, e.g., Recovery, Security, etc for the IPFIX framework, define the criteria to select the IPFIX Protocol and specify the control/data message formats and handshaking details to pass the IP flow information.

From an architectural point of view the IPFIX framework defines the following components:

- Collector: The collector receives flow records from one or more exporters. The collector might process or store received flow record.
- Observation Point: The observation point is a location in the network where IP packets can be observed. Examples are, a line to which a probe is attached, a shared medium, such as an Ethernet-based LAN, a single port of a router, or a set of interfaces (physical or logical) of a router.
- Metering Process: The metering process generates flow records. Input to the process are IP packets observed in an observation point. The metering process consists of a set of functions that includes packet header capturing, time stamping, sampling, classifying, and maintaining flow records.

Five protocols are currently proposed to implement the protocol specified in [Nor02]:

- Cisco Netflow is a feature available on almost all Cisco routers, which makes it the de facto standard. [Cla02] presents the version 9 of the architecture. Architecturally Netflow is based on two components:
  - The Exporter: A device with Netflow services enabled. The exporter monitors packets entering an observation point and creates flows out of these packets. The information from these flows are exported in the form of Flow Records to the collector.
  - Netflow Collector. The Netflow Collector receives Flow Records from one or more Exporters. It processes the received export packet, i.e. parses, stores the Flow Record information. The flow records may be optionally aggregated before storing into the hard disk.
- Diameter [Cal02] is a protocol under standardization by IETF for Authentication, Authorization, and Accounting purposes. Because of it’s flexibility Diameter can be easily extended to support flow information transport. However this flexible and general architecture render him more complex than other protocols.
- The LFAP protocol [RFC2124] LFAP was developed specifically for IP flow accounting. As such it is well suited to support the communication between the Exporting Process and an IPFIX Collecting process. From an architectural point of view LFAP is made of three main components: IPFIX devices that produce flow information, Collecting processes and finally applications. One Collecting process services multiple IPFIX Devices. Each IPFIX Device may have one or more backup Collectors. An application then retrieves the flow data from the Collecting devices. The LFAP protocol is used between the IPFIX Devices and Collecting process to exchange flow accounting data.
- The CRANE protocol [Zha02] can be viewed as an application that uses the data transport service provided by lower layer protocols. It relies on a transport layer protocol to deliver reliable, in-sequence data packets.
- The IPDR protocol evaluation document [Mey02-1] defines a document format, which offers a compact and efficient representation of usage accounting data. The encoding

format is based on XDR. The encoding supports a basic set of primitive data types and a number of additional types, which are derived from the primitive types. The mechanisms for encoding and transport are completely separate in IPDR. The Compact IPDR format can be used to serialize usage information to a file or it can be used to serialize usage information onto a reliable transport, such as TCP. For real time push oriented communication the streaming over a reliable transport is preferred, as described in Streaming IPDR [Mey02-0]. A file can also be used as the unit of exchange. IPDR's XML-Schema based format has the additional benefit of providing a well-defined equivalent XML encoding. Both the compact and XML formats are based on a common service definition specification. The service specification is expressed as one or more XML Schema documents. Service specifications are the primary means of extension in IPDR.

[Zs02] provides an analysis of the ability of IPFIX flows be used by additional components to provide IPPM metrics compliant measurements. These findings are summarized in table 3 below:

Metric	IPFIX as standardized	IPFIX with extension
Type-P-*-Connectivity [RFC2678]	Not considered	
Type-P-*-One-Way-Delay [RFC2679]	X	
Type-P-*-Packet-Loss [RFC2680]		X
Type-P-*-Round-Trip-Delay [RFC2681]		X
Type-P-One-Way-Loss-* [RFC3357]	X	
Type-P-One-Way-ipdv-* [Dem02]		X
Type-P-Packet-Reordering-* [Mor02]	Not considered	

Figure 2-7: IPFIX Ability to Provide IPPM Compliant Measurements

2.2.3.6 PSAMP

[Du02] describes the framework for Passive Packet Measurement (PSAMP). It provides a framework for a standard set of capabilities for network elements to sample packets and report on them. One motivation to standardize these capabilities comes from the requirement for measurement-based support for network management and control across multi-vendor domains. This requires domain wide consistency in the types of sampling schemes available, the manner in which the resulting measurements are presented, and consequently, consistency of the interpretation that can be put on them.

The framework for passive measurement includes three main parts: the selection of packets for measurement, the creation and export of measurement reports, and the content and format of the measurement records.

Compared to other work the PSAMP measurement capabilities are positioned as suppliers of packet samples to higher-level consumers, including both remote collectors and applications, and on board measurement-based applications. Indeed, development of the standards within the framework described in the PSAMP framework should take into account the measurement requirements of standards in other IETF working groups, including IPPM and TEWG. Conversely, it is expected that aspects of the PSAMP framework not specifically concerned with the central issue of packet sampling may be able to leverage work in other working groups. The prime example is the format and export of measurement reports, which may leverage the work of IPFIX.

2.2.3.7 Conclusion

The table below provides a comparison between existing standardized proposals.

Architecture	Passive/Active	SCOPE	Components Included	Packet	Flow	Metric	METRICS/OUTPUT
RMON	Passive	Path	MP, AP, MCP	X		x	Throughput, Flows, Packets
IPPM	Active	End to End/ Path	MP, AP, MCP			X	OWC,RTC,RTD, OWD,OWPL,OWR, OWPDV
RTFM	Passive	Path	MP, AP, MCP		X	x	Flows Throughput
Sflow	Passive	Path	MP, AP		X		Flows
IPFIX	Passive	Path	MP, AP		X		Flows
PSAMP	Passive	Path	MP, AP	X			Packets

Figure 2-8: Comparison of Standardized Proposal

2.2.4 Impact of IPv6

In this section we analyze what the impact of IPv6 would be on existing architectures. We may distinguish two main aspects that may be impacted by the use of IPv6:

- The first one is the nature of the information retrieved. Since the format of IPv6 packet is different from the one of IPv4 packets the information reported between measurement components may be different. For example:
    - In the case where packets are transmitted between measurement components, the information model should be flexible enough to allow IPv4 and IPv6 packets to be reported.
    - In the case of flows, the information model should be flexible enough to provide either IPv4 or IPv6 flow related information. According to [Qui02] three fields may be considered: Flow Label, Hop Limit (i.e. IPv4 TTL) and Traffic Class (i.e. IPv4 ToS).
    - In the case of metrics, metrics computation should be possible with IPv4 and IPv6 packets in order to accommodate the Type-P packet generation requirement. As a result active architecture should be able to generate IPv6 measurement packets and passive architectures should be able to capture IPv6 packets.

  Regarding this requirement we can state that existing architectures support an information model that allows information present in both IPv6 and IPv4 to be reported. For example RMON uses generic definitions of IP addresses allowing IPv6 and IPv4 addresses to be reported.

  On the other hand most architectures do not take into account specific IPv6 information. To our knowledge the only information model taking directly IPv6 specific fields into account is IPFIX where the IPv6 flow label field is one of the fields required in flow reports. Note that IPv6 flow label support is mentioned in [RFC2724] however the document does not specify how this should be done.

- The second one is the ability of QoS measurement to take IPv6 packets into account. Because of specificity within IPv6 packets, the mechanisms used to perform measurement operation may be affected. This aspect will be addressed in the next section.

2.3 QoS Measurement Operations

In order to perform QoS measurement, a set of operations usually has to be applied to the traffic flowing in the network. Although the location and the techniques used for QoS measurement operations may vary in different QoS measurement architectures, the nature of these operations can usually be classified into a few classes. Each implementation usually uses a combination of these basic operations in order to provide QoS measurement services. In this section we provide an overview of existing services, techniques and algorithms used to perform QoS measurement operations. For additional information, the interested reader may refer to [Zse02] which provides a good overview of existing sampling and filtering techniques that can be used in the case of IP networks. [Zse01] provides a good overview of some of the building blocks described in this section can be used for passive measurements in the case of a Linux based implementation.

2.3.1 Packet Copy

Since most trunk measurement operations aim at being transparent to the traffic being monitored, one of the first operations to be performed is to copy a relevant part of the packet stream when other measurement operations cannot be performed on the network device at line rate and perform these operations on a separate component. Section comes back on the reason justifying the separation of these functions and as a consequence the implementation and use of a packet copy functionalities. Let’s also mention that the packet copy functionality is also useful for QoS measurement unrelated tasks such as fault diagnostics or traffic tapping for security purposes.

Depending on the measurement needed, full traffic copy may not be required. For example [Duf00] notes that using the 40 first bytes of each IP packets is usually sufficient to identify uniquely each packet in the network and can therefore be used to perform adequate traffic measurement. Similarly DAG capture card series [Dag02] limit packet capture to the first 64 bytes of the physical frame format. Such cards are used by NLANR projects to compute round trip times at several American Internet exchange points. On the other hand some types of QoS measurement may require or take advantage of full packets.

From an implementation point of view four different techniques are currently available to perform full or partial packet capture.

2.3.1.1 Packet Sniffing

The most widely available is the one based on the broadcast nature of local area networks. Most Ethernet cards can be configured to capture all packets flowing on the Ethernet network segment they are connected to. This mode called promiscuous mode allows all the data-link frames to be forwarded to the operating system device driver [Ste94]. By using an adequate device driver it becomes possible build applications that interact with the device driver to capture all the traffic. The main limitation to this approach is that the volume of traffic that can be captured is limited by the bandwidth available between the network interface card and the general-purpose processor. As current PCI buses are limited to 622Mb/s, the current technology allows only relatively low bandwidth networks to be tapped. Please also note that this approach is only valid in the case of broadcast networks. As a result most backbone network links that use non-broadcast technologies (ATM, POS, …) are unable to take advantage of such architectures.

2.3.1.2 Port Mirroring

Another popular approach [Jun02], [Cis02] is to take advantage of the multicast capabilities of many network devices such as switches or routers. These devices usually have the ability to multicast a packet received on an input port to several output ports. This ability can be used to implement a service called port mirroring where every packet received on one port are simultaneously sent to an output port toward their destination as well as to a second output port where a system similar to the one described in the previous section can be connected. This approach solves the non-broadcast network technology problem but keeps the PCI bus bandwidth limitation. As a result this approach is usually combined with packet filtering prior to the traffic copy operation to limit the amount of traffic sent to the analyzer.

2.3.1.3 Optical/Electrical Splitting

Another possibility is to perform a copy of the traffic by duplicating the electrical or optical signal [ADC02]. Depending on the network technology such operations may be performed using an electrical or optical splitter. This component allows the whole traffic to be redirected while generating a negligible alteration to the original signal. An additional device such as those described in section can then treat the traffic.

2.3.1.4 Specialized Network Interface Cards

Finally most network management companies [Agi02], [Dag02], [IFT01] have developed network interface cards or complete systems that can be used to capture all the traffic flowing on a physical link at very high speed. Two approaches currently exist to feed these cards with network traffic. They can either be connected to network link through a splitting component such as those described in the previous section are can include a reception and emission port allowing the card to be plugged between two existing network nodes by using additional physical connections. These cards usually include a specialized processor combined with a synchronous memory component allowing the traffic to be captured in memory and reemitted while introducing a very small delay. Depending on the amount of memory, measurement operations and the measurement retrieval speed, these cards may either be able to store traffic within short period of time or be able to capture traffic in a permanent way.

2.3.2 Time Stamping

Few QoS measurement parameters can be computed without using a notion of time. Additionally even though a specific QoS measurement parameter may not require a precise time for its computation, it is still often necessary to be able to identify the time at which a measure has been made. As a result time measurement and time stamping are essential functionalities for QoS measurement activities.

2.3.2.1 Time Definitions

In order to perform one-way measurements [Ciu02] additional constraints usually apply to the way time is defined. These clock and time issues are widely discussed in [RFC2330] which also provide time related definitions used for the definition of QoS measurement parameters:

- Clock offset: the difference between the value of a clock and the true UTC time.
- Clock skew: the value of the clock offset derivative with respect to true time.
- Clock resolution: the smallest unit by which the clock's time is updated.
- Relative clock offset: the difference between the values of two clocks.
- Relative clock resolution. The worst resolution that can be achieved comparing time measurements performed by two clocks (i.e. the sum of the two clocks resolutions).
- Synchronized clocks: Two clocks with a null relative offset.

[RFC2330] also highlights the difference between "wire time" and "host time":

- Wire arrival time for a packet P at host H on link L: the first time at which the first bit of P reaches H on L.
- Wire exit time for a packet P at host H on link L: the first time at which all bits of P have reached H on L.
- Host time for a packet P at host H on link L: this first time at which all bits of P are received from L by the measurement application on H.

Although wire time can always be defined precisely for a given packet, host time may not be defined so precisely because of the delays that can be introduced by the hardware, operating system and application operations during the transfer of P from the wire to the measurement application. These delays are often generated by buffering and memory copy operations. In some cases host time may even be undefined because of packet fragmentation in the network. Such packets may have to be reassembled into a non-fragmented IP datagram for measurement operations. As a result host time in this case does not have a meaning. In order to avoid these problems precise time stamping techniques are usually performed as close as possible to the wire.

2.3.2.2 Implementation Means

In practice, several approaches currently exist to provide clock synchronization:

- The most popular one relies on NTP (Network Time Protocol – [RFC2030]). NTP computes time value by retrieving timing information from NTP server located around the globe. These servers are themselves synchronized by additional means (GPS, Radio). In order to take packet transit delays between servers and clients into account NTP clients use the values of the one-way delay in each direction in order to compute the clock offset. NTP usually provides a clock offset within the range of a few milliseconds.
- Another popular approach is to use time synchronization through GPS receivers. Clock offsets within the range of a few hundreds of nanoseconds [GPS99] can be reached. The main drawback of GPS based synchronization is the need for a line-of-sight connectivity between the GPS antenna and GPS satellites.
- Finally radio signals are also a mean to perform clock synchronization. Clock offsets within the range of a hundred microseconds can be reached [Mill97]. However radio signals sometimes suffers from alteration caused by landscape buildings or atmospheric

conditions. Similar architectures making use of cell phone communication signals are presented in [Prae].

2.3.3 Packet Selection

Packet selection can be performed using any combination of the following operations.

2.3.3.1 Filtering

Packet filtering is an operation allowing packets to be selected according to their content and a filtering policy. Depending on the information used to filter packets, we may distinguish two main types of filtering operations:

- Filtering operations relying solely on the content of the packet. This type is called stateless filtering. We may note such a filtering operation as a binary function f(p,o) where p is the packet to filter and o is the filtering policy. f(p,o) return 1 if p has to be selected and 0 in the other case.
- Filtering operations relying on the content of the packet as well as some additional information. This information may be related to the content of previous packets in which case the filtering operation is called statefull filtering. We may note such a filtering operation as a binary function $f(p_x,o,_{px-1},p_{x-2},\ldots,p_0)$ where p_x is the packet to filter, o is the filtering policy and $p_{i,\ 0<i<x}$ are packets received before p_x. f() return 1 if p has to be selected and 0 in the other case. In order to limit the size of information to be remembered to perform the filtering operation, statefull techniques usually only keep a summary of previously received packets.

From an algorithmic point of view the problem of packet filtering relates to the problem of packet classification. The packet classification based on d header fields can be viewed as a d-dimensional range match problem, which is equivalent to a classical problem in computational geometry called point location problem. The point location problem is to find the object that a point belongs to among n d-dimensional objects. The general form of this problem doesn’t have a nice algorithmic solution when d > 3. The best algorithm in terms of time complexity has a O(log(n)) complexity but requires a $O(n^d)$ working space. The best algorithm in terms of space complexity only needs a O(n) space but has a $O(\log^{d-1} n)$ time complexity.

Existing implementations can be classified in two main classes:

- Software implementations (PF, IPF, netfilter…) usually take advantage of the algorithmic research performed on this topic during the last five years ([Lak98], [Sri99], [Gu00], …) to introduce a nice balance between time complexity, space complexity and update complexity. These implementations usually support statefull classification as an extension to the basic stateless classification.
- Hardware implementations usually take advantage of the evolution of specific types of memory components such as CAM or TCAM [Sib02]. These implementations usually provide a very low time complexity and a reasonable space complexity but are usually less efficient in term of update complexity. As a result statefull classification is usually not supported by these components.

Software implementation can usually be found on most network operating system (unixes, MS windows, …) as well as on the general purpose CPU of most network devices (Routers, Switches, …). Hardware implementations can usually be found on most router line cards.

2.3.3.2 Sampling

Sampling allows the selection of a subset of packets by applying deterministic or random functions on the temporal of spatial packet position [Zse02]. Among sampling algorithms we may additionally distinguish two main families:

- Static sampling [Kla93] in which sampling behavior is defined in advance, independently from sampling results.
- Adaptive sampling [Cho02] in which sampling behavior is modified with sampling results in order to maximize some measurement aspects (measurement precision for example).

Within static traffic-sampling algorithms we may distinguish:

- Systematic sampling is based on the selection of a sampling starting point and a sampling duration interval. A popular example of systematic sampling is the every k-th element selection where every k-th element of a data set is selected.
- Random Sampling is based on the selection of a sampling starting point and a random sampling interval defined by a random process. Within random sampling techniques we may further refine definitions:
  - Stratified random sampling separates the data set into k buckets and selects an element within each bucked randomly.
  - Simple random sampling only selects randomly k elements within the data set.

From a practical point of view sampling capabilities are implemented in many existing network devices. For example routers [Jun02] generally support some form of systematic traffic sampling. Finally network operating systems can usually be extended to include some type of traffic sampling on top of packet capture capabilities [Ntop00]. However it should be noted that systematic sampling is usually considered as biased and unable to provide sound samples [RFC2330]. [RFC2330] suggest using random sampling using Poisson or geometric distributions to prevent such problem.

2.3.3.3 Hashing

Hashing [Duf00] is a specific type of packet sampling allowing the same packet to be selected at several point of the network. The scheme works as follows: The immutable part i(p) is extracted from each IP packet p. A hash value h(i(p)) is then computed and the packet is selected if h(i(p)) lies within a certain range. The use of the same hash function on each measurement point in the network allows a packet with the same invariant part i(p) to be sampled either everywhere or nowhere since h(i(p)) would be the same at each measurement point. As a result this technique allows the packet trajectory in the network to be recorded. However a precise trajectory recovery would be based on a widespread implementation of hashing techniques in network devices and the absence of conflicting packets.

Although hashing is today not implemented on existing network devices it would appear that the implementation of masking and hashing functionalities in routers line-cards would not pose too many problems [Chi02] if reasonable hash functions are chosen. Actually existing routers already use hardware implementation of simple hashing functions such as CRC32 in order to compute checksums over IP packets. On the other hand, it is still debatable if stronger hash functions such as MD5 or SHA-1 could be implemented at line speed [Deep01]. Such stronger hash functions would be necessary to allow measurement architectures to resist attacks from malicious users. Finally implementing hashing based packet selection within existing operating system shouldn’t pose too many difficulties but would provide far lower performance than a line-card based implementation.

2.3.4 Impact of IPv6

In this section we analyze what the impact IPv6 would have on the basic building blocs described in this section.

- For copy functions:
    - Packet sniffing, Port mirroring and specialized NICs: Since IPv6 headers are usually larger than IPv4 headers it is expected that IPv6 packets will be on average larger. Another reason to get larger packets is the possibility to use Jumbo-grams in the case of large MTU paths. In the case of packet sniffing and specialized NICs this would mean a need to increase the bandwidth between the network interface card doing the capture and the memory. In the case of port mirroring this would mean a need to increase the bandwidth between input and output line-cards as well as the capacity of these line-cards.
    - Specialized NICs. Since some NICs store received packets in an internal memory, the increase in the packet sizes and packet header sizes would necessitate larger memory components.
- For time-stamping operations. We currently don’t plan any noticeable impact.
- For packet selection operations:
    - Filtering operations:
        - Since the regular IPv6 header is larger than the regular IPv4 header, the classification rules used to filter IPv6 packets are likely to be larger than the one that would be used for IPv4. As a result the classification process is likely to require more memory to store a similar number of rules. For the same reason, since classification process performance are sometimes related to the size of the fields to be used for the classification, we may expect classification operations to be slower in the case of IPv6 packets.
        - The IPv6 may include one or several optional headers between the IP header and the Transport header. As a result the classification process may have to skip headers to find the relevant Transport information. This is likely to reduce the performance of the classification process compared to the one in an IPv4 environment.
        - Oppositely since IPv6 packets are likely to be larger than IPv4 packets, we can expect that, for a given bandwidth, the number of packets to be treated per second would decrease.
        - When IPv6 jumbo-grams are used, the size of the datagram is indicated in an optional header. As a result analyzing every header becomes necessary to know the size of a datagram. This operation is likely to decrease the performance of the filtering mechanism.
    - Sampling operations. We currently don’t plan any noticeable impact.
    - Hashing Operations:
        - Hashing functions may have different properties depending the structure of IP packets. Hashing functions should be proved to be independent from the possible content of IPv6 packet headers. A way to check that is to follow the procedure used in [Duf00].
        - The information used to compute an hashing value is based on some immutable fields of the IP packet. These fields include the IP source and

destination address. As a result it is expected that the size of information to be considered in the case of IPv6 would be larger than the one considered with IPv4. Since the performance of hashing functions usually depends on the size of the information to be hashed, the performance of the hashing process may be lower in the case of IPv6.

- As mentioned earlier the transport header may not lie directly after the IP header. As a result hashing operations may necessitate analyzing all optional headers in order to take the transport information into account. Since hashing operations should carry on immutable fields within the IP and Transport Header, it is expected that hashing operations would be less efficient with IPv6 packets.

2.4 Existing Products

We provide in this section a list of various product aimed at measuring quality of service in the Internet. For each product we classify the architecture using the parameters previously defined as well as by specifying the type of measures available, the type of operations performed, the location of the components composing the system and finally the protocols (if any) used to transport measurement results. The list of products given here mainly originates from CAIDA product directory [Cai02]. The classification of existing products is provided by [Inter-1].

2.4.1 Path Measurement Tools

The following table provides a list of path measurement tools.

Product	Passive / Active	Measurement	Aggregation	Metrics	IPPM/ITU Compliant	Building Blocks
Flow based						
Sflow (HP, Foundry Network, inMon) [Sflow]	Passive	X	X	/	/	Cpy, Class, Sampl,Time
Netflow (Cisco) [Netflow]	Passive	X	X	/	/	Cpy, Class, Sampl,Time
Cflowd (Juniper) [Jun02]	Passive	X	X	/	/	Cpy, Class, Sampl,Time
Caida Netramet [Netram] (RTFM implementation)	Passive		X	/	/	Cpy, Class, Sampl,Time
Packet based						
Agilent Advisor [Agi02]	Both	X	X	Round-Trip-Connectivity Throughput, Round-Trip-Delay Packet-Loss	Yes No Yes No	Cpy, Class, Time
Acterna Linkview [Link]	Both	X	X	Round-Trip-Connectivity Throughput,	Yes	Cpy, Class, Time

				Round-Trip-Delay Packet-Loss	No Yes No	
Spirent Smartflow [Smart]	Both	X	X	One-Way-Connectivity One-Way-Throughput, One-Way-Delay One-Way-Packet-Loss	Yes No Yes Yes	Cpy, Class, Time
Brix 1000 verifier [Brix]	Both	X	X	One-Way-Connectivity One-Way-Throughput, One-Way-Delay One-Way-Packet-Loss	Yes No Yes Yes	Cpy, Class, Time
Sniffer-Pro [Sniff]	Passive	X	X	/	/	Cpy, Class,Time
Shomiti Explorer {Shom]	Passive	X	X	/	/	Cpy, Class, Time
Ethereal (+libpcap) [Ether]	Passive	X	X	/	/	Cpy, Class,Time
LanExplorer [Lexpl]	Passive	X	X	/	/	Cpy, Class
France Telecom R&D Internet Fast Translator [IFT01]	Passive	X	X	/	/	Cpy, Class,
University of Auckland DAG [DAG02]	Passive	X		/	/	Cpy, Time
Caida Coral Reef [Creef]	Passive		X	One-Way-Throughput,	No	Class,
Metric based						
Cisco RTT-MIB [RTTMIB]	Active	X	X	Round-Trip-Connectivity Round-Trip-Delay	No Yes	Cpy, Class, Time
RMON [RFC2819]	Passive	X	X	Throughput,	No	Cpy, Class, Time

Figure 2-9: Path Measurement Tools

2.4.2 End-to-end Measurement Tools

The following table provides a list of end-to-end measurement tools:

Product	Passive / Active	Measurement	Aggregation	Metrics	IPPM/ITU Compliant	Building Blocks
Netperf [Netperf]	Active	X	X	One-Way-Connectivity One-Way-Delay One-Way-Throughput	Yes Yes No	Class, Time
Pathchar [Pathc]	Active	X	X	Round-Trip-Connectivity Round-Trip-Delay One-Way-Throughput Round-Trip-Packet-Loss	Yes Yes No No	Class, Time
Pchar [Pchar]	Active	X	X	Round-Trip-Connectivity Round-Trip-Delay One-Way-Throughput Round-Trip-Packet-Loss	Yes Yes No No	Class, Time
QosMetrix [QosMetrix]	Active	X	X	One-Way-Connectivity One-Way-Delay One-Way-Throughput Jitter (Ipdv)	Yes Yes No Yes	Class, Time
Ttcp [TTCP]	Active	X	X	One-Way-Connectivity One-Way Throughput	Yes No	Class, Time
Iperf [Iper]	Active	X	X	One-Way-Connectivity One-Way Throughput One-Way-Delay One-Way-Delay-Variation One-Way-Packet-Loss	Yes No Yes Yes Yes	Class, Time
Ping [Ping]	Active	X	X	Round-Trip-Connectivity	Yes	Class, Time

				Round-Trip-Delay Round-Trip-Packet-Loss	Yes No	
Sting [Stin]	Active	X	X	One-Way-Packet-Loss One-Way-Connectivity	Yes Yes	Class, Time
Chariot [Char]	Active	X	X	One-Way-Connectivity One-Way Throughput One-Way-Delay One-Way-Delay-Variation One-Way-Packet-Loss	Yes No Yes Yes Yes	Class, Time

Figure 2-10: End-to-end Measurement Tools

3. Requirements for Intra and Inter-Domain Measurements

3.1 Introduction

The specification of the 6QM measurement architecture requirements provided in this document follows the terminology specification provided in [RFC2119] except noted otherwise.

6QM Deliverable D2.1 identified two main types of QoS measurement architectures:

- Passive architectures are mainly designed to perform workload (throughput, and traffic flow volumes) measurements and are usually based on flow measurement.
- Active architectures are mainly designed to perform QoS metrics measurement. These metrics include but are not limited to delay, loss or delay variation. The computation of such metrics is usually based on the generation and extraction from the traffic of a small set of packets having specific characteristics.

However it should be noted that nothing prevents in theory QoS metrics to be computed from flow information as long as the information associated with the flow description is sufficiently rich and precise [Zs02]. It should also be noted that most active and passive architecture share several measurement component thus making it possible to provide passive and active measurement capabilities based on the same devices.

As a result, in order to provide the widest capacities in term of QoS measurement the 6QM architecture must be able to provide both passive and active measurement capabilities. Both measurement capabilities should be based upon a common infrastructure base. Additionally the measurement architecture must provide a single QoS measurement management interface that allows both measurement capabilities to be managed remotely.

In the rest of this section we provide Intra-Domain and Inter-Domain measurement requirements for passive and active measurements components.

3.2 Problem Space

All measurement systems may be divided into two areas: Inter-domain and Intra-domain. Intra-domain measures occur within the boundaries of an enterprise. Inter-domain measurements occur when measurements occur across enterprise boundaries.

Given these broad divisions, all measurement systems have common properties and components, which collaborate to provide a user with QoS measurements. These common properties and components are the essential architectural components that every measurement system must have in order to fulfill the function of measuring traffic flow through any given domain. Each of these architectural components has certain requirements that must be fulfilled in order for them to function as a measurement system. The requirements for these architectural components are provided in the sections below.

3.3 Functional Components

All measurement systems must have functional components. These functional components are the elements of a system engaged in the collaborative actions that give the users statistical information on the QoS of their networks. These functional components operate at various levels and have different roles and responsibilities. The functional elements for a QoS measurement system are given below.

3.3.1 Measurements Points/Points of Measure

3.3.1.1 Definition

Points of measure are the locations at which QoS metric measurements take place. It is to be noted that a point of measure may be performing more than one measure at any given time. This means that a Point of Measure has more than one measurement associated with it and the relationship between measures and point of measures is that a point of measure houses measurements and the activities pertaining to taking those measurements.

3.3.1.2 Requirements

Type of requirement	RID	Requirement	Level of requirement
Container	PM1.1	Ability to contain one or more measurement entities.	Must
Contactable	PM1.2	A Point of Measure must be contactable by a management entity in order to setup measurements within that Point of Measure.	Must
Configuration	PM1.3	A Point of Measure must be able to setup measurements when requested to do so.	Must
Setup	PM1.4	A point of measure must be able to remove or stop measurements when requested to do so.	Must

Figure 3-1: Types of Requirements

3.3.2 Measure

3.3.2.1 Definition

A measure is an abstraction with behaviors that measure the QoS of an given traffic flow. It is contained in a point of measure and its sole purpose is to filter the traffic flow for the packets that it is interested in and perform measurements on those packets.

3.3.2.2 General Requirements

A measure must have the following characteristics:

Type of requirement	RID	Requirement	Level of requirement
Begin-Point Configuration	M1.1	A Measure must have a source address. This means that every measurement must begin somewhere and the source from which it begins is source associated with the Measure.	Must
End-Point Configuration	M1.2	A Measure must have a destination address. This means that every measurement must end somewhere and the place at which the measurement ends is the end-point/destination of the Measure.	Must
Metric Configuration	M1.3	A Measure must be associated with a given QoS metric to look for in the traffic flow. These metrics are well defined and include One-Way-Packet-Loss, One-Way-Delay, round trip Delay and so on.	Must
Duration and Time Configuration	M1.4	The Measure must have a starting time and an ending time that determine when to begin and when to end measuring. A measurement may not be run indefinitely. It may begin and end at various times and this start time and end time must be specified at the Measure.	Must
Access Control Configuration	M1.5	A measure must be associated with an administrator. Some measurements are potentially dangerous to network operation and should be managed by high level administrators.	Must

Figure 3-2: Measure Requirements

3.3.2.3 Passive Measurement Requirements

Besides configuration requirements, the measure must have certain operational characteristics. These characteristics for both active and passive measures are given below:

Type of requirement	RID	Requirement	Level of requirement
Measurement Operations Traffic Copy	MI0.1	Ability to perform packet capturing in order to obtain a copy of the traffic without introducing modifications in the original traffic.	Must
Measurement Operations-Classification	MI1.1	Ability to classify packet according to IPv4 or IPv6 source address.	Must
	MI1.2	Ability to classify packet according to IPv4 or IPv6 destination address.	Must
	MI1.3	Ability to classify packets according to IPv4 ToS field content / IPv6 Traffic class	Must
	MI1.4	Ability to classify packets according to IPv6 flow label field content.	Must
	MI1.5	Ability to classify packets according to the IPv4 Protocol field content / IPv6 Next header field content	Must
	MI1.6	Ability to classify packets according to Transport	Must

		addresses.	
	MI1.7	Ability to classify packets according to previous packets information within a flow.	Must
	MI1.8	Ability to classify packets according to BGP information (Destination AS, Source AS).	May
	MI1.9	Ability to classify tunneled packets (v4 over v6, v6 over v4)	Should
	MI1.10	Ability to classify packets according to incoming interface.	Should
	MI1.11	Ability to perform classification operations at line-rate	Should
	MI1.12	Ability to perform classification operations within fixed duration bounds.	Should
	MI1.13	Ability to configure classification process with classification parameters	Must
	MI1.14	Ability to perform IPv6 and IPv4 configuration consistently.	Should
Measurement Operations- Time-Stamping	MI2.1	Ability to time-stamp the first packet of a flow	Must
	MI2.2	Ability to time-stamp the last packet of a flow	Must
	MI2.3	Ability to perform time-stamp operations before other operations.	Should
	MI2.4	Ability to perform time-stamp operations after classification or sampling.	May
	MI2.5	Ability to perform time-stamp operations on a remote device.	Should not
	MI2.6	Ability to indicate time-stamping source as well as time-stamping source characteristics (resolution)	Must
	MI2.7	Ability to choose time-stamping source if several available	Should
	MI2.8	Ability to perform time-stamping operations at line-rate	May
	MI2.9	Ability to perform time-stamping operations within fixed duration bounds.	Should
	MI2.11	Ability to synchronize clocks from a single source.	Must
	MI2.12	Support several clock synchronization sources	Should
	MI2.13	Support several clock synchronization methods	May
Measurement Operations-Sampling	MI3.1	Ability to perform systematic sampling	Must
	MI3.2	Ability to perform random sampling	Should
	MI3.3	Ability to perform hash based sampling	May
	MI3.4	Ability to perform stratified sampling	May
	MI3.5	Ability to perform classification before sampling	Must
	MI3.6	Ability to perform sampling before classification	May
	MI3.7	Ability to configure sampling process with sampling parameters	Should
	MI3.8	Ability to perform sampling operations at line-rate	Should
	MI3.9	Ability to perform sampling operations within fixed duration bounds.	Should

Measurement Operations-Coordination	MI5.1	Ability to perform pre-defined sequences of time stamping, classification and sampling operations.	May
	MI5.2	Ability to express any sequence of time stamping, classification and sampling operations.	May
	MI5.3	Ability to indicate if sequences are impossible to execute according to measurement architecture and timing model.	Should
	MI5.4	Ability to optimize operation placement depending on the sequence to execute.	May
	MI5.5	Ability to start and stop measurement operations given specific time conditions.	May
	MI5.6	Ability to start and stop measurement operations when a specific event is detected.	May
Accounting operations	MI6.1	Ability to account number of packets per flow	Must
	MI6.2	Ability to account number of bytes per flow	Must
	MI6.3	Ability to account duration of flow	Must
	MI6.4	Ability to classify flows according to their type.	Should
	MI6.5	Ability to account packets based on their actual size	Must
	MI6.6	Ability to account IPv6 packet based on the payload length	Must not
	MI6.7	Ability to deal with fragmented packets.	Must
	MI6.8	Ability to compute fragmentation rate of flow.	May
	MI6.9	Ability to measure measurement cost (CPU/memory consumption)	May
Measurement operations configuration	MI7.1	Ability to retrieve flow information. This flow information complies with IPFIX requirements. [Qui02]	Must
	MI7.2	Ability to provide full packets.	May
	MI7.3	Ability to perform measurement configuration and to retrieve measurement results remotely.	Must
	MI7.4	Ability to pull results from measurement devices to measurement manager.	Must
	MI7.5	Ability to push results from measurement devices to measurement manager.	Should
	MI7.6	Ability to perform exports operations depending on the type of flow (Long lived, Short lived).	Should
	MI7.7	Ability to perform measurement operations configuration and measurement through a single interface. (MP side)	May
	MI7.8	Ability to perform measurement operations sequences configuration through the same interface.	May
	MI7.9	Ability to signal or detect failure or dysfunction of any component of the system.	Must
	MI7.10	Configuration and result retrieval protocol is loss and error resilient	Should
	MI7.11	Support several measurement operations in parallel.	Should
	MI7.12	Support several measurement requesters.	May

	MI7.13	Ability to express measurement conditions (type of clock synchronization, clock resolution, value of results) for the acceptation of measures.	May
	MI7.14	Ability to report resources consumption regarding a measurement operation.	May
	MI7.15	Support several collectors for fail over operations	Should
Impact on network traffic	MI8.1	The impact of passive measurement operations on the traffic measured is negligible.	Must
	MI8.2	The impact of passive measurement operations on existing network devices is negligible.	Should
	MI8.3	The impact of traffic measurement configuration on the traffic measured is negligible.	Must
	MI8.4	The impact of traffic measurement configuration on existing network devices is negligible	Should
	MI8.5	Remote management operations have a negligible effect on existing traffic.	Should

Figure 3-3: Passive Measurement Requirements

3.3.2.4 Active Measurement Requirements

Type of requirement	RID	Requirement	Level of requirement
Measurement Operations Traffic Copy	MI0.1	Ability to perform a copy of the whole traffic without introducing modifications in the original traffic.	May
Measurement Operations-Classification	MI1.15	Ability to classify packet according to IPv4 source and destination addresses	Must
	MI1.16	Ability to classify packet according to IPv6 source and destination addresses	Must
	MI1.17	Ability to classify packets according to IPv4 ToS field content / IPv6 Traffic class	Must
	MI1.18	Ability to classify packets according to IPv6 flow label field content.	Must
	MI1.19	Ability to classify packets according to the IPv4 Protocol field content / IPv6 Next header field content	Must
	MI1.20	Ability to classify packets according to Transport addresses.	Must
	MI1.21	Ability to classify packets according to previous packets information within a flow.	May
	MI1.22	Ability to classify packets according to BGP information (Destination AS, Source AS).	May
	MI1.23	Ability to classify tunneled packets (v4 over v6, v6 over v4)	Should
	MI1.24	Ability to classify packets according to incoming interface.	Should
	MI1.25	Ability to perform classification operations at line-rate	Should
	MI1.26	Ability to perform classification operations within fixed duration bounds.	Should

	MI1.27	Ability to configure classification process with classification parameters	Must
	MI1.28	Ability to perform IPv6 and IPv4 configuration consistently.	Should
Measurement Operations- Time-Stamping	MI2.1	Ability to time-stamp the first packet of a flow	May
	MI2.2	Ability to time-stamp the last packet of a flow	May
	MI2.3	Ability to perform time-stamp operations before other operations.	Should
	MI2.4	Ability to perform time-stamp operations after classification or sampling.	Should not
	MI2.5	Ability to perform time-stamp operations on a remote device.	Must not
	MI2.6	Ability to indicate time-stamping source as well as time-stamping source characteristics (resolution)	Must
	MI2.7	Ability to choose time-stamping source if several available	Should
	MI2.8	Ability to perform time-stamping operations at line-rate	Should
	MI2.9	Ability to perform time-stamping operations within fixed duration bounds.	Should
	MI2.10	Ability to time-stamp every packet	Must
	MI2.11	Ability to synchronize clocks from a single source.	Must
	MI2.12	Support several clock synchronization sources	Should
	MI2.13	Support several clock synchronization methods	May
Measurement Operations- Sampling	MI3.1	Ability to perform systematic sampling	May
	MI3.2	Ability to perform random sampling	May
	MI3.3	Ability to perform hash based sampling	May
	MI3.4	Ability to perform stratified sampling	May
	MI3.5	Ability to perform classification before sampling	Must
	MI3.6	Ability to perform sampling before classification	May
	MI3.7	Ability to configure sampling process with sampling parameters	Should
	MI3.8	Ability to perform sampling operations at line-rate	Should
	MI3.9	Ability to perform sampling operations within fixed duration bounds.	Should
Traffic generation operations	MI4.1	Ability to generate a Type-P IPv4 packet	Must
	MI4.2	Ability to generate a Type-P IPv6 packet	Must
	MI4.3	Ability to generate Type-P packet fields according to the content of the fields of a Type-P' packet received.	Should
	MI4.4	Ability to generate encapsulated packets (v4 over v6, v6 over v4)	May
	MI4.5	Ability to time-stamp packet emission in a [RFC2330] way.	Must
	MI4.6	Ability to schedule packet emission to a specific time.	Should

	MI4.7	Ability to schedule packet emission according to a specific event.	Should
	MI4.8	Ability to configure packet emission (content, schedule) policy.	Must
Measurement Operations-Coordination	MI5.1	Ability to perform pre-defined sequences of generation, time stamping, classification and sampling operations.	May
	MI5.2	Ability to express any sequence of generation, time stamping, classification and sampling operations.	May
	MI5.3	Ability to indicate if sequences are impossible to execute according to measurement architecture and timing model.	Should
	MI5.4	Ability to optimize operation placement depending on the sequence to execute.	May
	MI5.5	Ability to start and stop measurement operations given specific time conditions.	Should
	MI5.6	Ability to start and stop measurement operations when a specific event is detected.	Should
Accounting operations	MI6.1	Ability to account number of bytes per flow	May
	MI6.2	Ability to account duration of flow	May
	MI6.3	Ability to classify flows according to their type.	May
	MI6.4	Ability to account packets based on their actual size	Should
	MI6.5	Ability to account IPv6 packet based on the payload length	Should not
	MI6.6	Ability to deal with fragmented packets.	Must
	MI6.7	Ability to compute fragmentation rate of flow.	May
	MI6.8	Ability to retrieve flow information.	May
	MI6.9	Ability to measure measurement cost (CPU/memory consumption)	May
	MI6.10	Ability to compute singleton One-Way Delay metrics.	Must
	MI6.11	Ability to compute sample One-Way Delay metrics	Should
	MI6.12	Ability to compute singleton Round-Trip Delay metrics.	Must
	MI6.13	Ability to compute sample Round-Trip Delay metrics	Should
	MI6.14	Ability to compute statistical Round-Trip Delay metrics	May
	MI6.15	Ability to compute singleton One-Way Loss metrics.	Must
	MI6.16	Ability to compute sample One-Way Loss metrics	Should
	MI6.17	Ability to compute statistical One-Way Loss metrics	May
	MI6.18	Ability to compute singleton Unidirectional Connectivity metrics.	Must
	MI6.19	Ability to compute sample Unidirectional Connectivity metrics	Should
	MI6.20	Ability to compute singleton One-Way Delay Variation metrics	Should
	MI6.21	Ability to compute sample One-Way Delay Variation metrics	Should
	MI6.22	Ability to compute statistical One-Way Delay Variation metrics	May
	MI6.23	Ability to compute singleton packet reordering metrics	Should

	MI6.24	Ability to compute sample packet reordering metrics	Should
	MI6.25	Ability to compute singleton Multi cast metrics	May
	MI6.26	Ability to compute sample Multicast metrics	May
	MI6.27	Ability to compute statistical Multicast metrics	May
Measurement operations configuration	MI7.1	Ability to retrieve flow information. This flow information complies with IPFIX requirements. [Qui02]	May
	MI7.2	Ability to provide full packets.	May
	MI7.3	Ability to perform measurement configuration and to retrieve measurement results remotely.	Must
	MI7.4	Ability to pull results from measurement devices to measurement manager.	Must
	MI7.5	Ability to push results from measurement devices to measurement manager.	Should
	MI7.7	Ability to perform measurement operations configuration and measurement through a single interface. (MP side)	May
	MI7.8	Ability to perform measurement operations sequences configuration through the same interface.	May
	MI7.9	Ability to signal or detect failure or dysfunction of any component of the system.	Must
	MI7.10	Configuration and result retrieval protocol is loss and error resilient.	Should
	MI7.11	Support several measurement operations in parallel.	Should
	MI7.12	Support several measurement requesters.	May
	MI7.13	Ability to express measurement conditions (type of clock synchronization, clock resolution, value of results) for the acceptation of measures.	May
	MI7.14	Ability to report resources consumption regarding a measurement operation.	May
	MI7.15	Support several collectors for fail over operations	Should
Impact on network traffic	MI8.1	The impact of passive measurement operations on the traffic measured is negligible.	Must
	MI8.2	The impact of passive measurement operations on existing network devices is negligible.	Should
	MI8.3	The impact of traffic measurement configuration on the traffic measured is negligible.	Must
	MI8.4	The impact of traffic measurement configuration on existing network devices is negligible	Should
	MI8.5	Remote management operations have a negligible effect on existing traffic.	Should

Figure 3-4: Active Measurement Requirements

3.3.3 Collectors

3.3.3.1 Definition

A collector, as defined by this document is a dual role component in a QoS system. The two roles that a Collector may play in a QoS measurement system are an information service role and an administrative service role.

- In the role of an information service, a collector is a functional component that collects and persistently stores the measurement results given to it by a Point of Measure and makes this information available to interested clients. The implications of this functionality are that the clients accessing the QoS measurement information in a Collector may originate in different domains. Some of them may be a part of the same domain in which the Collector resides. On the other hand, the client accessing the Collector may originate from a foreign domain i.e. a domain, which is external to the Collector. Given that the clients accessing the Collector may be from heterogeneous and foreign domains the collector, must have the functionalities of authentication, access control, service provisioning so that clients from these networks may not compromise the security of the Collector or the components within it.
- In the administrative role, the Collector has the additional responsibility for the configuration of inter-domain measurements. This means that two collectors in separate sovereign domains must interact with each other so that Points of Measure may be setup and activated and the results of all measurements may be viewed.

The requirements for a collector, as given below, are based of this dual role that a Collector plays in a QoS measurement system.

3.3.3.2 Requirements

Type of requirement	RID	Requirement	Level of requirement
Measurement Operations- Time-Stamping	C0.1	The Collector has the ability to check remote time-stamping resolution (Cross Check with other measurement source)	May
Measurement operations: Configuration	C0.2	The Collector has the ability to request the measurement operations in other domains (operation fully performed in foreign domain).	Must
	C0.3	The Collector has the ability to initiate measurements starting in mother domain and finishing in a foreign domain (cross domain measurement).	Must
	C0.4	The Collector has the ability to receive synchronous measurement results from other domains.	Must
	C0.4	The Collector has the ability to share measurement definitions between domains.	Must
Standardization	C0.5	Measures indicate if the measurement metrics complies with standards and which standards it complies to.	Must
	C0.6	The measures indicate if the measurement methodology complies with standards and which standards it complies to.	Should
	C0.7	The Collector has the ability to indicate that a specific metric is not supported or a specific measurement request is not possible.	Must

Publisher/Directory Service	C0.8	The Collector has the ability to advertise measurement capacities (measurement points, measurement point capacities)	Must
	C0.9	The Collector has the ability to advertise measurement capacities (measurement points, measurement point capacities) of foreign partner domains.	May
	C1.0	The Collector has the ability to identify and log measurement requests.	Must
Proxy Server	C1.1	The Collector has the ability to provide “proxy” measurements to other domains for n points measurements.	May
	C1.2	The Collector has the ability to request “proxy” measurements from other domains for n points measurements.	May
	C1.3	The Collector has the ability to export measurement to other domains asynchronously. Periodic flow export/flow beginning-end notification.	Should
	C1.4	The Collector has the ability to receive asynchronous measurement results from other domains.	Should
Broker	C1.5	The Collector has the ability to find an appropriate service that will satisfy a client’s request. This service may on different machines in the same domain or it may be in external domains. To the requesting client, the Collector’s Broker functionality is transparent. The client neither knows, nor should it care, how the service is provided.	Must
	C1.6	The Collector stores QoS information in a persistent repository.	Must
Authentication Service	C1.7	The Collector provides an authentication service to user who are accessing the system	Must
Access Control Service	C1.8	The Collector provides access control to any client that is attempting to access a service in the QoS measurement system.	Must
Service Activation	C1.9	The Collector provides Service activation functionality for all clients interacting with the QoS Measurement system. This means that the service for a particular request may be activated upon demand.	Must
Persistent Service	C2.0	The Collector stores QoS measurements in a persistent repository.	Must

Figure 3-5: Collector Requirements

3.4 Management Areas

All measurement systems must have certain management areas associated with them. These management areas are:

- Configuration management: This management area is responsible for the activation and de-activation of collectors, points of measures and measures.
- Fault Management: This management area is responsible for notifications when any threshold in the functional components is exceeded.

3.4.1 Configuration Management

The following are the requirements for configuration management:

Type of requirement	RID	Requirement	Level of requirement
Measurement operations configuration	I9.1	Flow information complies with IPFIX requirements. [Qui02]	Must
	I9.2	Ability to perform active and passive measurement configuration and to retrieve measurement results from measurement devices	Must
	I9.3	Ability to perform configuration and measurement retrieval through a single interface.	Should
	I9.4	Ability to perform measurement operations sequences configuration through the same interface.	Must
	I9.5	Support several measurement operations in parallel.	Must
	I9.6	Support several measurement requesters.	Must
	I9.7	Support several requests from several requesters simultaneously.	Must
	I9.8	Ability to advertise measurement capacities (measurement points, measurement point capacities)	Should
	I9.9	Configuration interface enables administrator to express measurement conditions (type of clock synchronization, clock resolution, value of results, maximum duration, measurement location, measurement method …) for the acceptation of measures.	Should
	I9.10	Ability to report resources consumption regarding a measurement operation along with measurement results.	Should
	I9.12	Ability to report measurement conditions and limitations along with. This include clock synchronization, sampling method, classification method, computation method, type of measure (active, passive) …	Should
	I9.13	Ability to provide measurement results through several methods. (Flow based/ Active measurement) —Several results would be provided.	May
Result Storage	I10.1	Ability to store measurement results in separate DB.	Should
	I10.2	Ability to query DB to retrieve past measurement.	Should
	I10.3	Ability to combine new and past measurement results (e.g. statistical values) through DB queries	May
MP configuration	I11.1	Ability to translate measurement configuration in MP configuration.	Must
	I11.2	Ability to translate MP measurement results to common format results.	Must
	I11.3	Ability to pull results from measurement devices to measurement manager.	Must
	I11.4	Ability to push results from measurement devices to measurement manager.	Should
	I11.5	Ability to report failure or dysfunction of any component of	Must

		the system.	
	I11.6	Configuration and result retrieval protocol is loss and error resilient.	Should

Figure 3-6: Configuration Management Requirements

3.4.2 Fault Management

Fault management is that area which monitors and notifies the management entity when system components exceed their functional thresholds. The requirements of this area are:

Type of requirement	RID	Requirement	Level of requirement
Configuration	F1.1	Ability to configure thresholds in critical system components.	Must
Notification	F1.2	Ability to issue notifications to users when normal operating parameters have been exceeded.	Must

Figure 3-7: Fault Management Requirements

3.5 Structural Model

The following diagram gives an overview of the structural model for a measurement system. The components given here can be further sub-divided should the need arise. However, for the purposes of generalization these components reflect the high-level structure of measurement systems whose requirements have been given above.

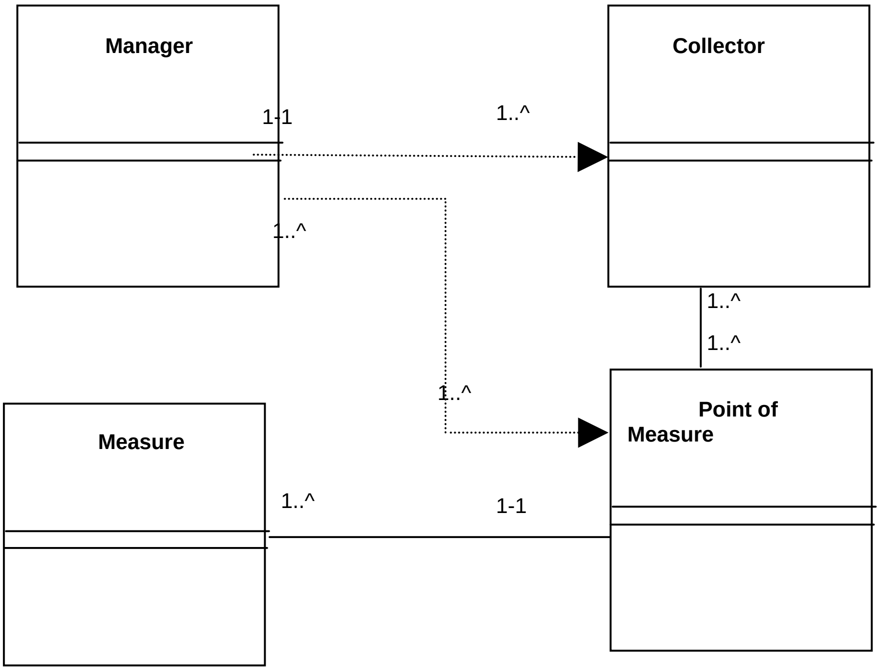


Figure 3-8: Structural Model of Requirements

4. STANDARDIZATION OF THE EXCHANGE OF MEASUREMENT RESULTS

4.1 Terminology

Measurement is a process that includes gathering, recording and post-processing of information with the direct goal of retrieving a certain characteristic (called metric) of the measured object.

Monitoring is a subset of measurement, denoting the continuous or repeated measurement of certain metrics with the purpose of checking the existence of a pre-defined condition (i.e. exceeding a threshold value).

QoS-specific Measurement Information means the set of key performance metrics and service events that need to be monitored to support inter-domain management of IP QoS services.

Transit Inter-domain IP QoS service is the forwarding of IP packets with a specific QoS treatment through an IP network domain.

End-to-End Inter-domain IP QoS service consists of one Transit Inter-domain IP QoS service or is formed as the concatenation of several of TIPQoS services.

Traffic Profile Metrics are metrics or events that characterize the IP packet stream constituting the IP Service at any point of the end-to-end path.

Service Performance Metrics are metrics or events that characterize the impact of an IPND on the IP packet stream of the inter-domain IP service.

Type-P packet [RFC 2330] A fundamental property of many Internet metrics is that the value of the metric depends on the type of IP packet(s) used to make the measurement. Important factors determining the type of IP packets are the protocol ID (UDP or TCP), port number, packet size, arrangement for special treatment (IP precedence or RSVP).

4.2 Introduction

The ability for a service provider to offer a reliable service over several interconnect IP network domains requires a mechanism to ensure that quality objectives and commitments can be met through a standardized exchange of measurement results.

There are various *actors* involved in the delivery of an end-to-end service, and the relationships between these *actors* are considered both from the business perspective as well as from the perspective of the underlying network infrastructure. The business perspective focuses more on the measurement methodologies and contractual commitments, whereas the network perspective provides the metrics that the respective parties can measure.

The goal of this document is to examine the various actors, their relationships, and to define and standardize a methodology for exchange of measurement results in the form of an “Implementation Agreement”.

4.3 Scope and Objectives

The scope of this document is illustrated in figure-1 below. There are three types of services that will be examined:

- End-to-End Service

  This scenario focuses on the end-to-end service offered by service provider A in which interconnected IP network domains (IPND) provide the underlying means to deliver the service. In this case, the service provider offers a service to the service customer from the ingress point in IPND A to the egress point in IPND C. As such, service provider A is responsible for the overall performance from A to C.

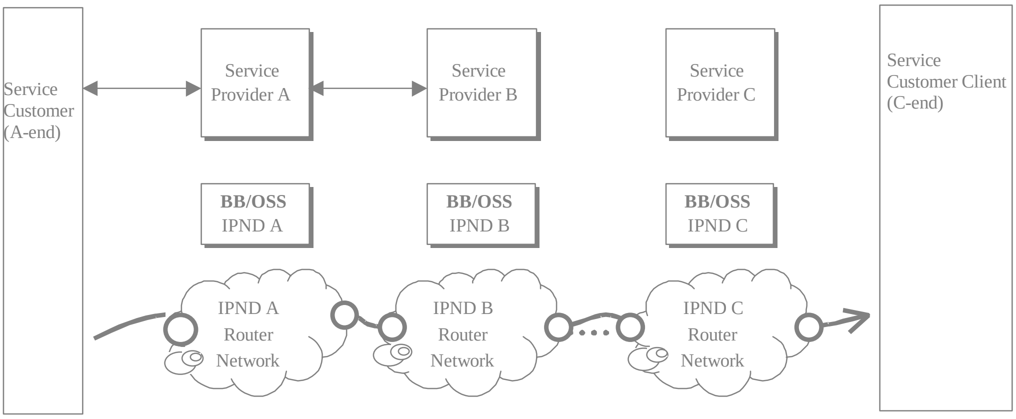
- Transit Services

  Generally, transit services are provided by long distance or other carriers, and enable end-to-end delivery of services as described above. In this scenario, service provider B provides transit services to service provider A, and as such is responsible for all traffic between the ingress point in IPND B and the egress point in IPND B.

- 3rd Party Services

  Third party services are services offered to another service provider. In this scenario, the service provider that is actually contracted to provide a service to the end service customer enters into a business relationship with the 3rd party service provider. The third party service may be offered to the service customer via the service provider who has a contracted business relationship with the customer.

The objective is to examine the three types of services from both the business and technical perspectives in order to determine how best to exchange measurement result information for contract conformance and verification purposes. It is assumed that measurement systems are already in place and that intra domain measurements are well established.



IPND - IP Network Domain BB - Bandwidth Broker OSS - Operation Support System

Figure 4-1: Service Relationships

4.4 Actors and Responsibilities

4.4.1 Service Customer

The service customer is the entity that desires to “buy” a service from the service provider, otherwise known as the “seller”. The consumer enters into a contractual agreement with the service provider by means of signing a service level agreement. It is the customer’s responsibility to ensure that he does not knowingly or willingly abuse the service being offered by the provider, i.e. through continued spamming or intentional security attacks.

4.4.2 Direct Service Provider

The direct service provider “sells” a service to a service customer and enters into a direct contractual business relationship with the customer. This type of service provider assumes all contractual obligations as outlined in the service level agreement, and as such is responsible for ensuring the measurement and reporting of IP performance metrics across domains.

4.4.3 Indirect Service Provider - Intermediator

The indirect service provider “sells” a service to a service customer and enters into a direct contractual business relationship with the customer, however acts only as an intermediator of the service being provided i.e. does not directly offer the service. This type of service provider may “buy” the actual service from a 3rd party service provider. However, the indirect service provider assumes all contractual obligations as outlined in the service level agreement with the service customer, and as such is responsible for ensuring the measurement and reporting of IP performance metrics across domains.

4.4.4 Transit Service Provider

The transit service provider, or indirect provider, “sells” a service to another service provider and not directly to the end service consumer. This type of service provider enters into a contractual relationship with the “direct” service provider to whom transit services are offered. The transit service provider is responsible for the service performance between the ingress and egress points in his IP network domain, while the “buyer” of the service shall conform to certain traffic profiles at the ingress point.

4.4.5 3rd Party Service Provider

The third party service provider “sells” a service, other than transit services, to another service provider. An example might be an xDSL offering that is part of a bundled service offered by an ISP. In this case, the 3rd party service provider enters into a contractual agreement with the other service provider, and not the end service consumer. He is obligated to resolve all issues associated with the proper functioning of the service being provided, and to report any anomalies.

4.5 Relationship Between Actors

The business and technical relationship between “buyers” and “sellers” of service are regulated by means of different offers and agreements. In this document, we consider only the exchange of technical information incorporated in those agreements (SLA), i.e. the Service Description (SD), Traffic Conditioning Specification (TCS) and Traffic Forecast (TF). Both parties can perform measurements for internal purposes or as a validation of the agreed service parameters.

The following table depicts the relationships between the actors as defined in the previous section. One can interpret the table by "the actor in row X has the following set of responsibilities to the actor in column Y". When there is no direct relationship between actors, or between actors of the same type, the table entry indicates "NONE".

	Service Consumer	**Direct Service Provider**	**Indirect Service Provider**	**Transit Service Provider**	**3rdParty Service Provider**
Service Consumer	NONE	Buys service. Signs SLA. Requests verification of SLA.	Buys service. Signs SLA. Requests verification of SLA.	NONE	NONE
Direct Service Provider	Sells direct service SLA Traffic forecasting. May verify traffic conformance to TCS. End-to-end measurement results.	NONE	NONE	Buys service. Signs SLA. Requests verification of SLA. Ensures incoming traffic conforms to TCS.	Might buy 3rd party service, if required to deliver end-to-end service. Signs SLA if service needed. Requests verification of SLA.
Indirect Service Provider	Sells third party service. SLA Traffic forecasting. May verify traffic conformance to TCS. End-to-end measurement results.	NONE	NONE	Buys service. Signs SLA. Requests verification of SLA. Ensures incoming traffic conforms to TCS.	Buys third party service. Signs SLA. Requests verification of SLA.
Transit Service Provider	Transparently provides underlying transit network.	Traffic forecasting. Verifies traffic conformance. Domain measurements.	Traffic forecasting. Verifies traffic conformance. Domain measurements.	NONE	Transparently provides underlying transit network.
3rdParty Service Provider	NONE	SLA Provide service. Measure service.	SLA Provide service. Measure service.	NONE	NONE

Figure 4-2: Relationship Between Actors

4.6 Requirements for Standardized Exchange of Measurement Results

4.6.1 Contract Conformance

In this scenario, either a direct or an indirect service provider has “sold” a service to a service customer. The service provider might have contracted with both a 3rd party service provider for access services, and a transit service provider for long distance transportation of IP packets.

The direct, or indirect service provider enters into a business relationship with both the transit service provider and the 3rd party service provider and “buys” service from each of them. In this sense, the service provider is both a “seller” and a “buyer” of services. The following figure illustrates this relationship.

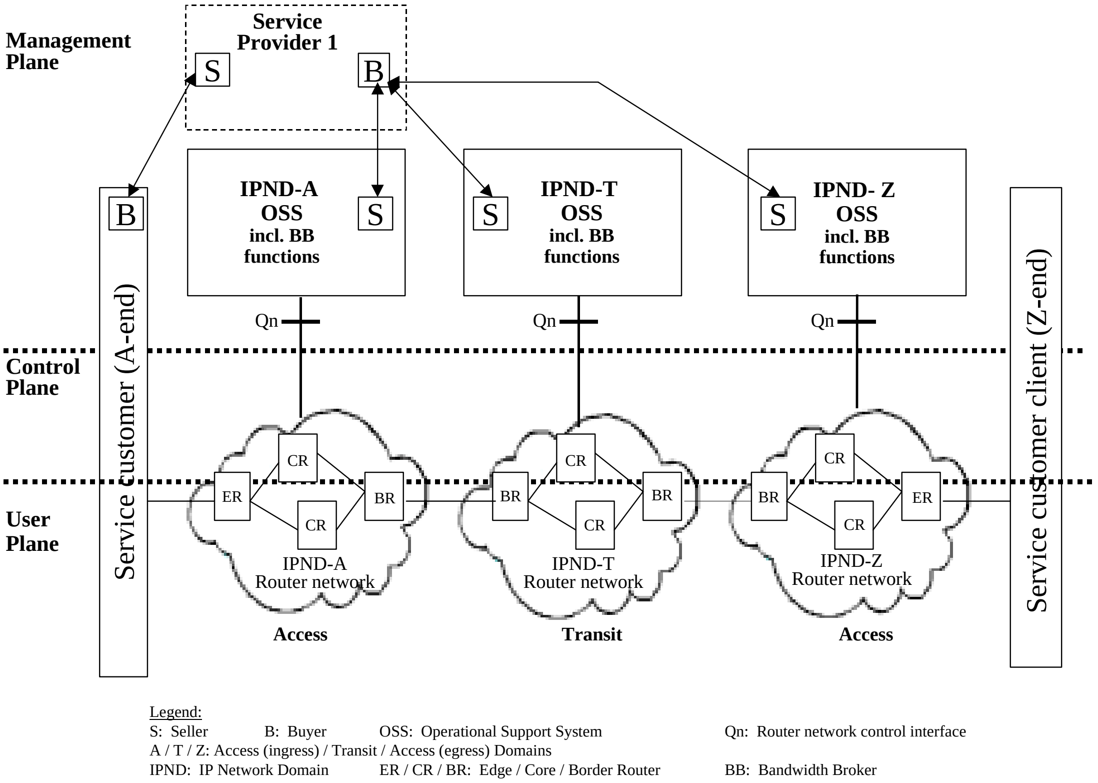

Management Plane
Service Provider 1
S
B
B
IPND-A OSS incl. BB functions
S
S
IPND-T OSS incl. BB functions
S
IPND- Z OSS incl. BB functions
Qn
Qn
Qn
Control Plane
User Plane
Service customer (A-end)
Service customer client (Z-end)
CR
ER
BR
CR
IPND-A Router network
BR
CR
BR
CR
IPND-T Router network
BR
CR
ER
CR
IPND-Z Router network
Access
Transit
Access
Legend:
S: Seller
B: Buyer
OSS: Operational Support System
Qn: Router network control interface
A / T / Z: Access (ingress) / Transit / Access (egress) Domains
IPND: IP Network Domain
ER / CR / BR: Edge / Core / Border Router
BB: Bandwidth Broker

Figure 4-3: End-to-End Service Offering over Interconnected IP Domains [EURESCOM]

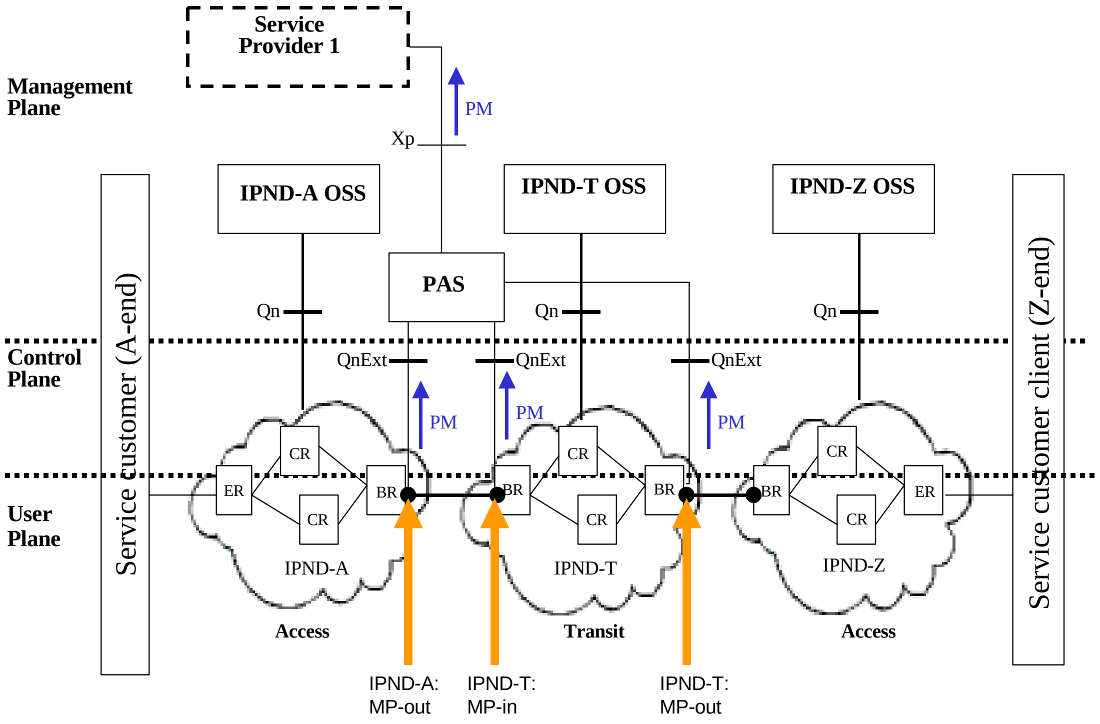
The traffic leaving a buyer IPND must comply with the SLA between the buyer and seller IPNDs or between the buyer IPND and the Service Provider.

The service offered by a seller IPND must comply with the terms of the SLA between the seller and buyer IPNDs or between the seller IPND and the Service Provider.

4.6.1.1 Requirements

- Each domain specific OSS must be capable of collecting IP performance measurement data for its’ own domain and exporting it over a standardized performance interface to the OSS of the Service Provider.
- The service provider OSS shall collect and consolidate performance measurement data from the various 3rd party and transit service providers with which it has entered into a contractual agreement.

- The data exported by the 3rd party and/or transit service provider may be used to validate packet loss, throughput, jitter, and other contractual obligations to the "buying" service provider.
- The service provider OSS shall be capable of exporting information to each IPND OSS to prove conformance of incoming traffic to characteristics as outlined in the SLA.
- In the event that the service provider can not "trust" the IPND's to make accurate, truthful measurements, an objective entity called a *Performance Assessment Service* (PAS)may be required [See figure-3 below]
- The PAS is expected to have the access rights for making the necessary performance measurements at the border routers of the relevant IPND's.
- The measurements made by the PAS shall be capable of being exported to the service provider OSS over a standardized performance interface

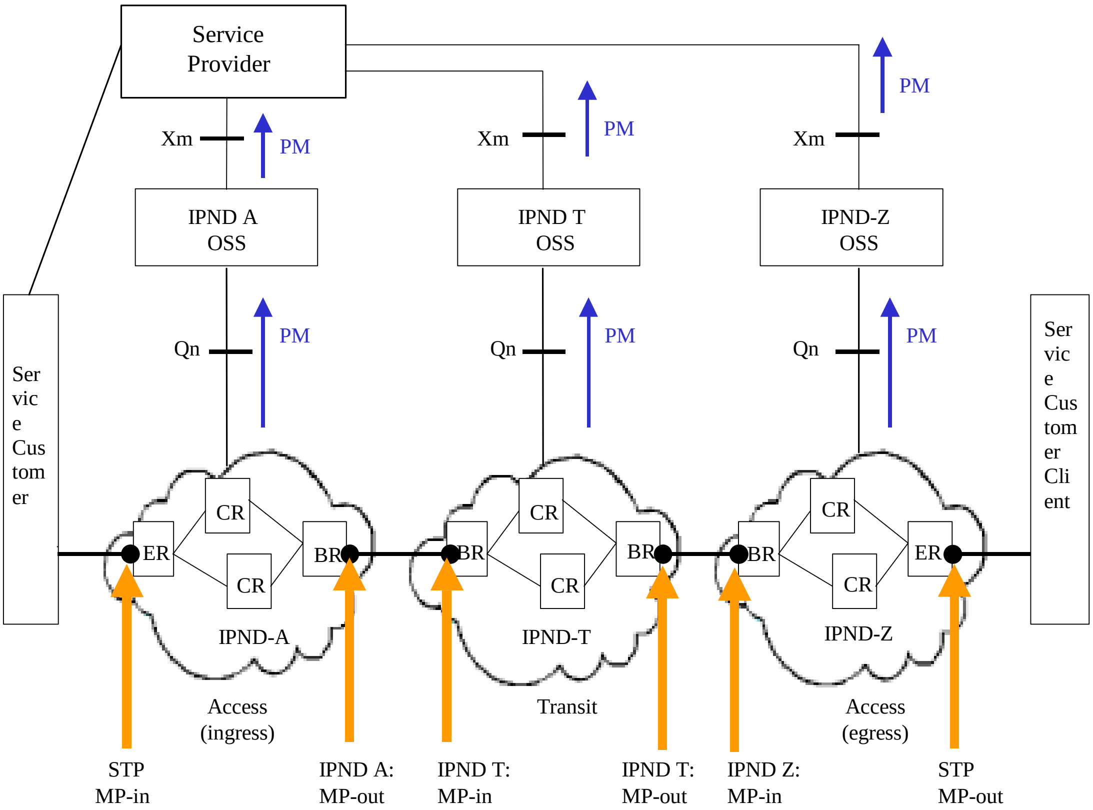


Legend:
OSS: Operational Support System
MP: Measuring Point
ER / CR / BR: Edge / Core / Border Router
Qn: Router network control interface
PAS: Performance Assessment Service
IPND: IP Network Domain
PM: Performance Measurement
A / T / Z: Access (ingress) / Transit / Access (egress) Domains
QnExt: External interface for router network control
Xp: Interface for exchanging performance data

Figure 4-4: Trustable Performance Assessment System [EURESCOM]

4.6.2 End-to-end Performance Measurement

In the above figures the service offered by the Service Provider 1 spans three IPNDs. To ensure that the end-to-end service performance meets the requirements of the SLAs in force between the Service Provider 1 and its service customers, the service performance needs to be measured between the two points where customer traffic enters and exits the Service Provider's network. These have been identified as *Service Termination Points* (STP) in the figure below.



Legend:

OSS: Operational Support System	IPND: IP Network Domain
MP: Measuring Point	PM: Performance Measurement data transfer
ER / CR / BR: Edge / Core /Border Router	A / T / Z: Access (ingress) / Transit / Access (egress) domains
Qn: Network control interface	Xm: management interface supporting Performance Data exchange

Figure 4-5: Performance measurement of an end-to-end service

4.6.2.1 Requirements

- End-to-end service performance metrics (e.g., one-way-delay, round-trip-delay, IPDV, etc.) can be measured between the two STPs and checked against the SLA requirements to detect any SLA violations.
- The end-to-end measurements can be compared to the aggregation of the individual measurements obtained from each IPND. Any discrepancy might indicate that one or more of the measurements obtained from the individual IPNDs are incorrect.

5. Setup Process for Creating End-To-End Measurements

5.1 Introduction

The ability to collect and exchange measurements across administrative domains implies that there are processes for setting up measurements.

Setting up such processes across administrative domains requires peering agreements that specify the factors to be taken into consideration and the mechanisms employed to implement the setup process.

The issues are to standardize the exchange of measurement results among heterogeneous measurement systems and across administrative domains, thus allowing for concatenation of global metrics, and to define the setup process for creating end-to-end measurements across administrative domains.

5.2 Problem Scope

The fact that Network Management Systems are usually confined to an area of control within their own domain boundaries make it impossible to a Network Management System to perform end-to-end measurement among several administrative areas. The reason is that the setup of a measure needs to setup a couple of probes and that one of them is not in the initiator domain.

The only scalable solution consists in providing a standard interface for the control of the initiator access, for the reception of measure results, and for the setup of measures.

A proposal is to allow the QoS measurement system entity that we will name proxy to receive and perform measure setup queries from initiators.

It consists in providing an interface for Network Management Systems located in other domains, so they can setup measures.

The setup of measure has a standardized interface. The QoS measurement system proxy controls the systems of measure of its own domain. On reception of a measure query sent by an initiator, the QoS measurement system proxy controls that the initiator is granted for the measure requested. It setups the points of measure involved which are in its domain. Probes save the results they compute in the entity of the QoS measurement system proxy named the collector. The collector controls the access to the results of the measure.

It also consists in providing an interface to access to the results of measure performed inside each domain. It allows Network Management Systems that are not granted to make measures in a domain to read the results of measures that the QoS measurement system proxy of the domain granted.

Measures are performed inside each domain. The results of the measures are saved per domain in the entity of the QoS measurement system proxy named the collector. The collector controls the access to the results of the measure from outside the domain. The domain that initiated the measure accesses the results of its measure saved in the collector of the domain that terminated the results.

Granted Network Management Systems access the results from outside each domain and collect the average of per domain measures in order to get an end-to-end measure.

5.3 Proposal

A quick glance to a typical scenario that an initiator who wants to set up inter-domain measurements may give a good overview of the various step of set up process.

The initiator must authenticate itself to a dedicated service that may reside within or outside its domain. As a positive result of the authentication phase the initiator will be identify or known by his profile.

Then with his profile the initiator will try to create end-to-end measurements across administrative domains, which in term of management can be translated into, will try to perform operations on managed objects in various distinct QoS measurement systems.

The QoS measurement system must have a dedicated service that:

- Check if this initiator can access the domain is requesting for
- Then if this initiator can access the service is requesting for
- And finally if this initiator is allowed to perform the operation is requesting.

The QoS measurement system must also have a dedicated service that is able to handle:

- The concurrent access to data
- The synchronization between requests
- The consistency of requested operations

The QoS measurement system must also have a repository within each domain of all the registered services as well as their associated access profile.

This means that the entity in charge of those described functionalities needs to communicate between domains one with each other using messages. Therefore the encoding/decoding of the exchanged messages as well as marshalling/un-marshalling of the exchanged client request objects become a necessity in order to achieve running platform neutrality.

On the other hand this implies that each services within a domain needs to register itself to the QoS measurement system when it becomes available and also that an administrator initiate the set up of those profiles.

The client issues a request for an operation upon a given QoS measurement object with the appropriate attribute values filled in.

The request goes to the client “proxy” which in turn contacts the QoS measurement system entity in charge of service registration in order to locate the server that can satisfy this request.

The QoS measurement system checks for user access then for server access. If the client has successfully logged in and has the requested server access clearance, it returns the server's contact information to the client “proxy”.

The client “proxy” processes the marshalling encoding of the request and sends it to the server “proxy” for processing.

The server “proxy” processes the un-marshalling decoding of the request, performs the requested operation by dispatching the request to the server obtains the reply from the server then processes the marshalling encoding of the response and sends it back to the client “proxy”.

So by following the preceding description mechanism the QoS measurement system needs to offer an object request broker dispatcher capability that can be proxy.

The QoS measurement system also needs a persistent mechanism in order to store or retrieve data as for example its own configuration parameters.

5.4 Requirements

The QoS measurement system entity that encapsulates the data and the services provided by functional components and therefore allows for inter-domain QoS measurement operations to take place has a lot of commonalities with a middle-tier infrastructure.

In the context of a middle-tier infrastructure, all clients and services components must interact with this infrastructure in order to perform any operation upon any managed object in a QoS measurement system.

Among all the functionalities associated with a middle-tier infrastructure, at least the following requirements should characterize this infrastructure.

Type of requirement	RID	Requirement	Level of requirement
Security Level	L1	The system infrastructure has an Access Control mechanism in order to solve the inter domain as well as intra domain security issues.	Must
Transaction Level	L2	The system infrastructure has a Transaction Control mechanism in order to solve the problem of concurrency, synchronization and consistency.	Must
Communication Level	L3	The system infrastructure has a Service Repository and Service Location mechanism.	Must
	L4	The system infrastructure has an object request broker, dispatcher and proxy agent handler mechanism.	Must
	L5	The system infrastructure has an encoding decoding mechanism as well as a marshalling un-marshalling mechanism for inter domain request.	Must
	L6	The system infrastructure contains plug-in protocol in order to interact with the requested service.	May
Persistency Level	L7	The system infrastructure has a persistent mechanism or repository in order to store or retrieve data as for example its own configuration parameters.	Must

Figure 5-1: Requirements for End-to-end Measurements

6. STANDARDIZATION OF THE FORMAT AND SEMANTICS OF TEST PACKETS

Despite the growing availability of good measurement platforms, it is still impossible to generalize IPPM metrics measurement among heterogeneous points of measure nor to couple active and passive techniques. To do so, the extra information inserted in the IP packets to perform the measurement has to be standardized.

There is a need for interoperability among heterogeneous manufacturer equipments to measure the performance of IP networks for different Type-P.

This effort is motivated too by the need to perform end-to-end measures across administrative areas and composite networks. Currently there is only one solution that consists in the concatenation of end-to-end metrics. As these measures are not performed simultaneously and the test packets metered are different and the accuracy of the concatenation is extremely variable.

The test packets exchanged by active probes are filtered efficiently by the passive points of measure. Spatial metrics **[Ste03]** are computed using the end-to-end information and the intermediary information. These metrics are mandatory for troubleshooting and for SLA management. The following is directly inspired of individual draft of standard test packets **[Ste04]**.

6.1.1 The IPPM Framework

The IPPM Framework consists in 4 major components:

- A general framework for defining performance metrics, described in the Framework for IP Performance Metrics, RFC 2330;
- A set of standardized metrics, which conform to this framework. The IPPM Metrics for Measuring Connectivity, RFC 2678. The One-way Delay Metric for IPPM, RFC 2679. The One-way Packet Loss Metric for IPPM, RFC 2680. The Round-trip Delay Metric for IPPM, RFC 2681;
- Emerging metrics which are being specified in respect of this framework;
- A Reporting MIB to exchange the results of the measures. It is an interface between a system of measure and the administrative entities interested in these results. This proxy controls the access to the results. These entities use the results to compute statistics and aggregated metrics.

6.1.2 Terminology

6.1.2.1 IP Measurement Packet Definition

A measurement packet is a regular Internet packet that contains additional fields needed for measurement inserted somewhere in the packet.

6.1.2.2 IPPM Measurement Signature Definition

An IPPM measurement signature is a regular Internet packet that contains a standard block of fields needed for performing IPPM measure.

This block of fields is named IPPM measurement signature (IMS).

The type of the Internet packet determines the Type-P of the measure.

6.1.2.3 State of the Art

As Internet is designed for 'best effort' the specifications do not include the control of the quality of service provided.

The IPPM WG has started an effort to standardize the protocols for measuring the performance of the network.

It encounters 2 main issues. The measurement packets must not be easy to detect by the network equipments and must be easy to detect by measurement tools. The standard solution must be easy to implement and must not be a security hole.

This section analyses the existing solutions and extracts the common parts to provide inputs for a first proposal.

6.1.2.3.1 ICMP

ICMP if the only test packet that is standardized. There is no doubt that it is dramatically used. On the other hand ICMP has serious limitations. Basically it cannot emulate type-P like UDP or TCP.

6.1.2.3.2 Type-P

The Type-P corresponds to the suite of protocols present in the IP and SUB IP headers of the packet.

Software based devices inserts the data needed for the measure after the header of the packets. That increases the speed of software processing in the source and in the sink. The additional fields needed for measurement differs according to the Type-P. Typically its contents changes when fields of the packet header are used as data for the measurement. Moreover its location in the packet changes according to the Type-P. That makes it difficult to receive and timestamp the test packets at wire time. This technique is not adapted to high-speed rate.

At high-speed rate (e.g. over OC12), hardware based devices insert the data needed for the measure in a fixed block of fields located at the end of the packet. That simplifies dramatically the design because the measurement data are located in the regular part of the packets. It generalizes the concept of Type-P for IPng, SUB IP and permit operational interoperability among heterogeneous SUB IP links. On the receiver's side it facilitates the detection of the test packet at wire speed, the time stamping of the packet on the fly and finally the extraction of the test signature.

6.1.3 Wire Time and Time Stamping

This IPPM framework defines the wire time of a packet inserted on the network as the time that the last bit of the packets is sent.

Most of the hardware implementations insert the timestamp on the fly and complement the binary value of the packet in a timestamp complement field. The delay between the timestamp insertion and the packet insertion on the wire is deterministic. The timestamp inserted is increased by the value of this delay while preserving the accuracy of the timestamp. This technique is generalized for link rate over gigabit/s. It is very accurate but requires larger field.

Software devices do not share a common technique. The timestamp is inserted at different places, at different time according to the implementation and the operating system. There is not a common technique to adjust the timestamp value. The delay between the timestamp insertion and the packet insertion on the wire is not deterministic and introduces an initial jitter.

6.2 Requirements

6.2.1 Existing Measurement Fields

IPPM measurement systems share the same semantic. The information inserted in the packet is very closed. The measurement packets differ only by the fields order, the field name, the field unit and the field size.

The common fields are the following:

- The device that has sent the packet.
- The interface that has sent the packet.
- The identifier of the stream the packet belongs to.
- The absolute timestamp corresponding to the time the packet is sent.
- The sequence number of the packet.
- The absolute timestamp corresponding to the time the packet is received.
- A checksum computed on the previous fields.
- One-complement of timestamp.

6.2.2 OWAMP Requirements

The IPPM WG is standardizing a measurement protocol. The requirements are listed in the "A One-way Active Measurement Protocol Requirements" (draft-ietf-ippm-owdp-reqs-01.txt). The test protocol needs to have the following characteristics:

- Be lightweight and easy to implement.
- Be suitable for implementation on a wide range of measurement nodes.
- Since the protocol needs to be able to measure individual packet delivery time and has to run on various machines, it needs to support UDP as transport protocol.
- It should be possible to use varying packet sizes and network services, as negotiated using OWDP-Control.
- To be a lowest common denominator, OWDP-Test packet format should only include universally meaningful fields, and minimum number of them.

- It should be possible to make packets generated by OWDP-Test as small as possible, to be able to accurately measure paths where packet-splitting technologies such as ATM are used.

6.2.3 Interoperability Requirements

The requirement is to have operational interoperability among heterogeneous manufacturers and to perform one-way delay measurement across administrative areas and among composite networks.

The constraints to gain inter domain interoperability and interoperability between heterogeneous manufacturers devices does not differ.

The meaning of the common fields has to be adapted to the interoperability context. In a test involving heterogeneous equipments the values set up by the source have no meaning for the sink. The meaning of the administrative fields must be set by an entity common to the source and the sink. A field value sets by the source has no meaning for the sink excepted if the value refer to a shared referential. A typical example is the need to create a referential for the time.

The issue does not consist only in having the IMS format to be recognized. The main need is to have the results of the measurement packets to be assigned to the same measure setup in the sink and in the source. The measure must be identified in the IMS. It must include a field that identifies the measure in the scope of the initiator of the measure.

6.2.4 SUB IP and IPNG Needs

The measurement packet must consider the measure of the performance of multicast services, mobile IP services and Ipv6. The protocol translation mechanisms and the coexistence between Ipv4 and Ipv6 are potential sources of interoperability of the measures.

6.2.5 Relationship with Other Organization

The aim is to increase operational interoperability. Basically it consists in promoting the need to share the same measurement packets identification mechanism to unambiguously detect the measurements packets and avoid overlapping regarding the fields' values chosen.

6.3 IPPM Measurement Signature Framework

The aim is to provide a standard signature of the packet to perform measurements of the IPPM metrics across administrative areas and among heterogeneous devices.

The framework has the following requirements:

- Respects the IPPM Framework requirements (RFC2330) regarding the Type-P and the accuracy.
- Respects the requirements for the test protocol of OWAMP.
- Specifies a format that allows the integration of the future needs.
- Specifies a test packet format for including the IMS in regular packets.
- Integrates the existing test packets format and concepts whenever it is possible.

The requirements and the needs may be gathered in a strong constraint: To have operational interoperability among heterogeneous manufacturers and to perform one-way delay measurement across administrative areas for the different Type-P, including those that have short packet-length constraints.

Basics Type-P is obviously UDP and TCP.

6.3.1 Measurement Packet Identification

To distinguish measurement packets among regular packets, the last field of the IMS is a protocol identifier.

6.3.2 RFC2330 Type-P

The header of the packet defines the type-P of the test packet from the network point of view.

6.3.3 RFC2330 Wire Time and Time Stamping

There is a strong requirement in the IPPM framework to have a timestamp consistent with the time the packet is inserted on the network. The specification must not increase the complexity of the time stamping at multi gigabit rates.

6.3.4 IPPM Measurement Signature Format

It permits the IPPM WG to define several versions of the measurement packets.

6.3.5 State Machine Limitation

To guaranty operational interoperability the model is stateless excepted for semantic needs such as packet sequence order and security.

That does not preclude the definition of more complex test packets in the future.

6.3.6 Packet Sequence Number

More and more services cross gateways. They may change the sequence numbering of the packets in the header (e.g. the initial value). A lot of metrics computation relies on the analysis of the order of the packets. To provide a trustable sequence of results there is a need for the sequence number to be integrated in the IMS.

6.3.7 Measure Initiator Scope

The management framework of the IPPM-REPORTING-MIB defines a namespace for each initiator (owner) of a measure. Each measure is identified by its owner and by the number chosen by the initiator for this measure within its scope. These values provides to the source and sink with a shared identifier. This field is mandatory to discriminate concomitant measures set up between 2 points measures. It permits different initiators to set up measures between 2 point of measures while concerning different metrics and Type-P.

6.3.8 Discussion on IMS and OWAP Base Formats

Measures performed with a timestamp resolution under the second are out of the scope of this memo.

The OWAMP timestamp format reduces the maximal resolution of the NTP timestamp to improve the accuracy of operational measure of performance and to permit an efficient interoperability of the measure. It reuses the MSB of the 'fractional part of the second' field to carry both the source clock precision and the source synchronization state.

Its resolution, ~50ns excludes ipdv measures on gigabit networks. As an example, consider the measure of IPDV of small packets (or cells) on the next generation of gigabit link, the 40G. The timeslot of such a packet is closed to 10 nanoseconds (400 bits* 1/40 ns). With such a timer resolution the first variation metered will correspond to 5 times the size of the packet itself. It means that the jitter will be computable only for packet of which the size is over 400 bytes. That excludes most of the real time applications, such as VoIP.

The proposed timestamp is respectful of the OWAMP timestamp design while preserving the maximal resolution of the NTP timestamp format. It permits a timestamp resolution suitable for the measure over multi gigabit path. It preserves the NTP timestamp format. It differs because it counts the second since 1 Jan 2000 0H00 instead of 1 Jan 1900 0H00. It will wrap in year 2068 (The NTP timestamp will wrap in year 2036).

As it does not count the second of the last century, the most significant bit of the part that represents the second is not needed for counting the second. It is set to indicate if the fractional part of the second contains a precision field. When this bit is not set the resolution is maximal. The maximal resolution is closed to 250 picoseconds (see NTP RFC).

The field Prec has the same semantic than in OWAMP. Its definition differs because it counts only the trusted bits of the fractional part of seconds.

The field S is 3 bits long to describe the current level of clock synchronization (Status 0 to 7+).

The proposal is to define a common packet signature format common to the OWAP test packet and to IMS. It is directly inspired from the 'unauthenticated test packet' defined for the OWDP. This base is enhanced to define different types of IMS and to define the different type of OWAP test packets.

It permits top define up to 16 different type and up to 4 versions. Currently that is enough both for IMS and OWAP needs and for the future.

It is 12 bytes length.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Tx Timestamp Integer part of seconds             |P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Tx Timestamp Fractional part of seconds    | Prec    |  S  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Sequence Number     |Ext| Type  |Ver|  Protocol Id  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

6.3.9 Interdomain

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Owner Id                            |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |      Measure Id               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|            Tx Timestamp Integer part of seconds             |P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|    Tx Timestamp Fractional part of seconds                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|           Sequence Number     |Ext| Type  |Ver|  Protocol Id  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
```

6.3.10 RoundtripDelay

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|                      Rx Timestamp                            |
|                                                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|            Tx Timestamp Integer part of seconds             |P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|    Tx Timestamp Fractional part of seconds                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|           Sequence Number     |Ext| Type  |Ver|  Protocol Id  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
```

6.3.11 Interdomain RoundtripDelay

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|                      Rx Timestamp                            |
|                                                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|                           Owner Id                           |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|                               |      Measure Id              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|            Tx Timestamp Integer part of seconds             |P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|    Tx Timestamp Fractional part of seconds                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|           Sequence Number     |Ext| Type  |Ver|  Protocol Id  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
```

6.3.12 CHK

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|                            CHK                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|            Tx Timestamp Integer part of seconds             |P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|    Tx Timestamp Fractional part of seconds                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
|           Sequence Number     |Ext| Type  |Ver|  Protocol Id  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
```

6.3.13 Proprietary Extension

Manufacturers may insert proprietary extension at the beginning of the IMS while preserving measurement interoperability. The field 'Ext' indicates the number of blocks of 8 bytes, which carried proprietary data.

Example:

An measurement point that need an 5 bytes of extra information to been inserted in an interdomain IMS use the following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Extra information block                 |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Owner Id                            |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |      Measure Id               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Tx Timestamp Integer part of seconds             |P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Tx Timestamp Fractional part of seconds                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Sequence Number     | 1 |  1    |Ver|  Protocol Id  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

6.4 Software and Hardware Techniques Interoperability

Both hardware and software solutions used the header of the packet to define the type-P, insert an IMS in the payload of the packets. They differ on the location of the IMS in the packet.

It is obvious that interoperability between software and hardware technique will be reached when they will use the standardized IMS with no extra data in the type-P SDU.

The maximum interoperability is gained when the type-P packet PDU consists only of the IMS.

6.5 Security

To avoid the measurements systems to be used to make attacks there is a strong requirement to propose a security mechanism to control the access to the setup of the network measures.

From the network security point of view, the main security hole in a network measure is the control test packet. The standardization of a packet signature does not facilitate the control of a probe to perform a DOS attack.

7. MANAGEMENT LEVEL SYSTEM

7.1 Terminology

7.1.1 Model Related Definitions

7.1.1.1 Infomodel

An information model describes a set of objects independent of their implementation and storage. From that information model, one can derive one or more data models. Formally, these are described as a set of mappings. That is, while the information stays constant, classes may be added, or repository-specific concepts may be used to define the mapping.

7.1.1.2 Datamodel

Each data model is the result of mapping from the information model to a specific type of repository.

7.1.1.3 Schema

A schema is a collection of data models for a particular repository. It is not just a collection of data models.

7.1.2 Service Related Definitions

7.1.2.1 Contract

Is a written document that is negotiated between a subscriber and a service provider during the purchase of a service-bundle. The contract contains specifications and expected performance of the service(s) that compose the service bundle [NMF-504].

7.1.2.2 Organization

In the service provider environment, an organization is an entity that can either provide services or subscribe to them or both.

7.1.2.3 Recipient

Is the role of an end-user that receives a service but does not pay for it.

7.1.2.4 Service

A meaningful set of capabilities (a network function, network connection or operational function, or a combination of these), that is offered as a component of a Service Bundle by a

Service Provider. Customers, End-users, Network Providers and Service Providers see a different perspective of the service.

7.1.2.5 Service Level Agreement (SLA)

Contains the technical information relating to a service. It groups together the technical information relating to the service(s) that comprise a service bundle [TMF-701]. It is the documented result of a negotiation between a customer and a provider of a service, that specifies the levels of availability, service ability, performance, operation or other attributes of the service. (See also "Service Level Objective.") [RFC2475]

7.1.2.6 Service Level Objective (SLO)

Partitions an SLA into individual metrics and operational information to enforce and/or monitor the SLA. It is a set of parameters and their values. It may be defined as part of an SLA, an SLS (Service Level Specification, see later), or in a separate document.

7.1.3 Role Related Definitions

7.1.3.1 Role

Concept used to define the allocation of responsibility. A role is a name associated with a set of well-defined rights and privileges. The IETF Policy WG has a different definition for a role, as the administratively specified characteristic of a managed element (e.g. an interface).

[ITU-M3208.1] and [ITU-M3320] have a set of formal definitions for these participants, and therefore for the rest of this document we will refer to the actors and systems under the following role identifications:

7.1.3.2 Partner (a.k.a. Tenant in [SP-DNA])

Object that represents either side of a service contract. The two parties that are contractually bound are referred to as the customer and the provider.

7.1.3.3 Customer

One of the two roles that exist in services dealing. It is an organization that contracts with a service provider for service(s). The customer purchases communications and or data services from a service provider and/or network operator.

7.1.3.4 Service Customer

Can initiate one or many service requests. The service customer is acting in the role of a customer when requesting services provided by the service provider according to a contract with him.

7.1.3.5 Service Provider

Object that provides the services used by the service users. The services can be provided either by the Provider itself or by groups of Providers. A Service Provider may (and then be a Network Operator) or may not operate a network. A Service Provider may or may not be a Customer of another Service Provider, where one provider may "sub-contract" with other providers to fulfill the customer’s needs.

7.1.3.6 Network Operator

Object that provides the transport network upon which service provider's and service user's services run. A Network Operator may be a Service Provider and vice versa. The network operators produce usage data for the service provider's billing.

7.1.3.7 End-user

Entity that makes use of network services. The end-user entity may be an individual person, a device, a group, on organization or something else, depending on the service. The end-user may be the direct client of a service provider or may receive services as a result of his association with an organization.

7.1.3.8 Subscriber

Role of an organization or individual to make use of services provided by a service provider. The subscriber is the ultimate buyer of a network service. An end-user who is an individual person and pays for the service it receives also has the role of a subscriber.

7.1.4 Network/QoS Related Definitions

7.1.4.1 Domain

A collection of elements and services, administered in a coordinated fashion. Known as AdminDomain within CIM.

7.1.4.2 SLS

Specifies handling of customer's traffic by a network provider. It is negotiated between a customer and the provider, and (for example) in a DiffServ environment, defines parameters such as specific Code Points and the Per-Hop-Behavior, profile characteristics and treatment of the traffic for those Code Points. An SLS is a specific SLA (a negotiated agreement) and its SLOs (the individual metrics and operational data to enforce) to guarantee quality of service for network traffic. (See also SLA and SLO) [RFC3060]

7.1.4.3 QoS (Quality of Service)

Refers to the ability to deliver network services according to the parameters specified in a SLA. "Quality" is characterized by service availability, delay, jitter, throughput and packet loss ratio.

At a network resource level, "Quality of Service" refers to a set of capabilities that allow a service provider to prioritize traffic, control bandwidth, and network latency. The way QoS is offered to the user by the network can either be by opening the whole spectrum of possible values for one or more traffic parameters, or by packaging them in a set of discrete parameter values. There are two different approaches to "Quality of Service" on IP networks: Integrated Services [RFC1633], and Differentiated Service [RFC2475].

7.1.4.4 CoS (Class of Service)

Refers to the provisioning of relative levels of service amongst different packet flows. The resulting service perceived by a CoS flow is dependent on the number of other flows that share network resources and are members of the same (or different) CoS. [TEQUILA]

7.1.5 Flow Related Definitions [IPFIX]

7.1.5.1 Flow

There are several definitions of the term 'flow' being used by the Internet community. Within this document we use the following one:

A flow is defined as a set of IP packets passing an observation point in the network during a certain time interval. All packets belonging to a particular flow have a set of common properties. Each property is defined as the result of applying a function to the values of:

- One or more packet header field (e.g. destination IP address), transport header field (e.g. destination port number), or application header field (e.g. RTP header fields).
- One or more characteristics of the packet itself (e.g. number of MPLS labels, etc...).
- One or more of fields derived from packet treatment (e.g. next hop IP address, the output interface, etc...).

A packet is defined to belong to a flow if it completely satisfies all the defined properties of the flow.

This definition covers the range from a flow containing all packets observed at a network interface to a flow consisting of just a single packet between two applications with a specific sequence number.

7.1.5.2 Observation Point (OP)

The observation point is a location in the network where IP packets can be observed. Examples are a line to which a probe is attached, a shared medium, such as an Ethernet-based LAN, a single port of a router, or a set of interfaces (physical or logical) of a router. [IPFIX] Note that one observation point may be a superset of several other observation points. For example one observation point can be an entire line card. This would be the superset of the individual observation points at the line card's interfaces.

7.1.5.3 Metering Process (MP)

The metering process generates flow records. Inputs to the process are packet headers observed at an observation point and packet treatment at the observation point, for example the selected output interface. The metering process consists of a set of functions that includes packet header capturing, time stamping, sampling, classifying, and maintaining flow records. The maintenance of flow records may include creating new records, updating existing ones, computing flow statistics, deriving further flow properties, detecting flow expiration, passing flow records to the exporting process, and deleting flow records. The sampling function and the classifying function may be applied more than once with different parameters.

7.1.5.4 Flow Record

A flow record contains information about a specific flow that was metered at an observation point. A flow record contains measured properties of the flow (e.g. the total number of bytes of all packets of the flow) and usually also characteristic properties of the flow (e.g. source IP address).

7.1.5.5 Exporting Process

The exporting process sends flow records to one or more collecting processes. One or more metering processes generate the flow records.

7.1.5.6 Collecting Process

The collecting process receives flow records from one or more exporting processes. The collecting process might store received flow records or further process them, but these actions are out of the scope of this document.

7.2 Introduction

The goal of this section is to provide an architecture based on the requirements specified in the previous sections of this document.

This work is organized as follows:

- The requirements for each structural component in a QoS system is referenced and then the services provided by that structural component is given.
- The services of each structural component solve the problem and address the issues of the listed requirements.
- The collaboration and interaction of these components form the architecture for a QoS measurement system.

Additionally, the approach taken with each structural component is from “the ground up”. This means that the fundamental components involved in a QoS measurement system are provided first. After this, the document goes to the next component level that is organized around these fundamental/core components. This is the management layer/component. Finally a middleware component, which provides data and service encapsulation for the purposes of intra and inter-domain access as well as security, is described. This model might be described as follows:

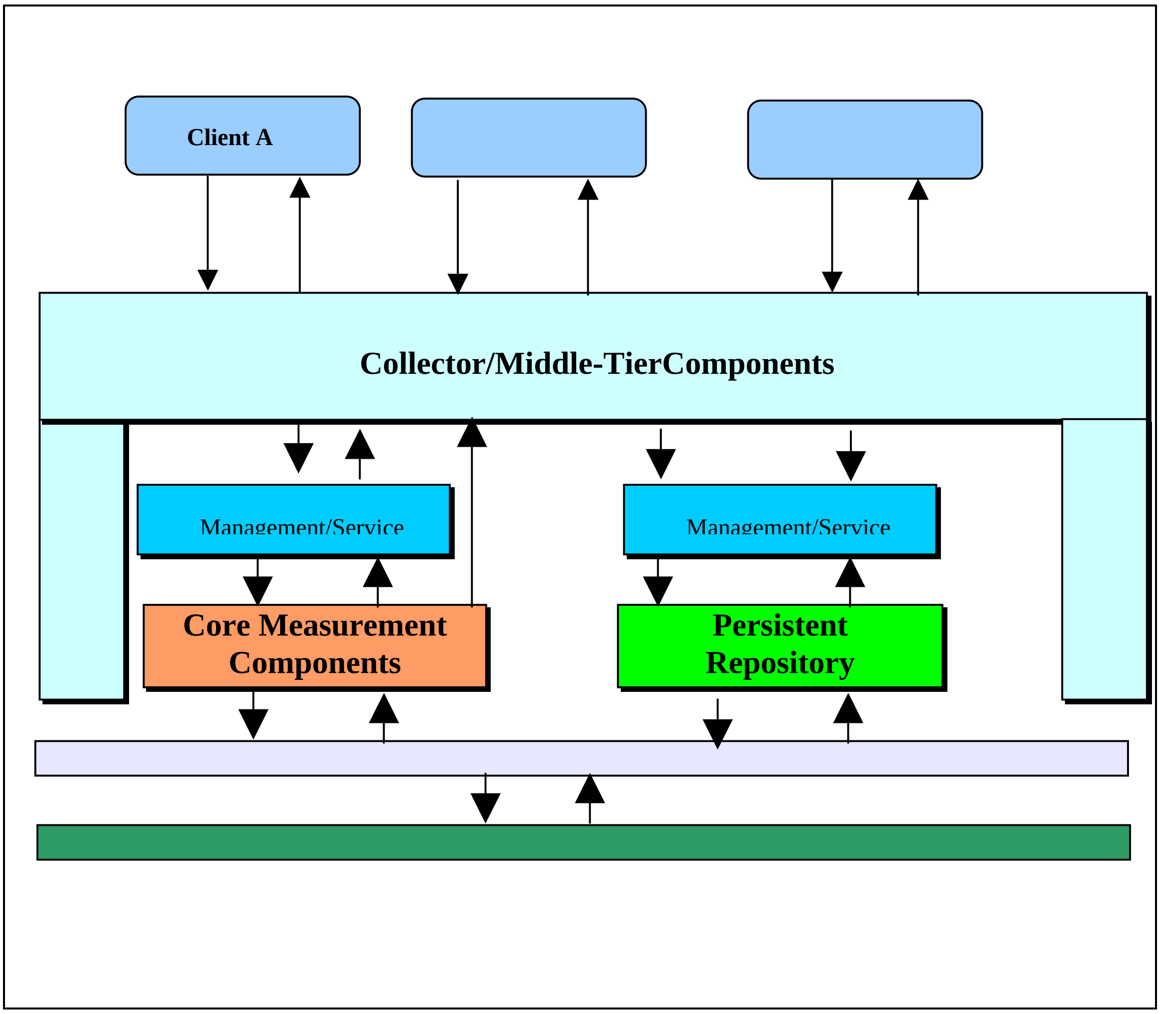


Figure 7-1: Management System Level Model

Please Note: In this document, the attributes described in the structural components are by no means complete. Their sole purpose is to suggest some of the properties present in each structural component and to highlight the associations between these structural components. They are not a detailed specification of all the possible attributes that these structural components may finally possess.

7.3 Scope and Objectives

The objective of this document is to propose a architecture for the management level system (hereby designated as MLS) based on the requirements identified in the other tasks of the 6QM project.

This document deals about measurement level, management level, storage, dialog between the metering points (hereby designated as MP) and the MLS, some aspects of the security, and customer access level.

As an architecture document, it does not preclude any implementation specifics.

7.4 Summary of Requirements

The requirements that were detailed in the section 3 indicate that there are certain fundamental components that collaborate to perform QoS measurements. These fundamental components are contained within two broad divisions. The first division is the functional division and the second is the management division. The functional division contains those core measurement components that perform QoS measurements, QoS reporting, QoS logging...etc. The management division has the responsibilities to configure, monitor and perform fault management upon these core functional components. Above the management component is a middle-tier component, which encapsulates the data and the services provided by the functional, and management components and allows for inter-domain QoS operations to take place. The following definitions, taken from the QoS measurement architecture requirements, review these components.

7.4.1 Functional/Core Components

Points of Measure: the locations at which QoS metric measurements take place. It is to be noted that a point of measure may be performing more than one measure at any given time. This means that a Point of Measure has more than one measurement associated with it and the relationship between measures and point of measures is that a point of measure houses measurements and the activities pertaining to taking those measurements. This definition corresponds to the Observation Point in IPFIX; i.e. It is the point where “. Packets can be observed”.

Measure: A measure is an abstraction with behaviors that measure the QoS of any given traffic flow. It is contained in a point of measure and its sole purpose is to filter the traffic flow for the packets that it is interested in and perform measurements on those packets. This definition implies that a measure is a base abstraction and may be used to extend or derive other abstractions. Therefore there are network measures, which are used to perform measurements in any given network; there are aggregated measures, which are used to perform statistical analysis upon network measures; there are notifications, which extend measures such that threshold violations may be reported to interested clients. Note that a network measure has the activities of sampling, time stamping, and generating a flow of measured results to a receiving component. This activity corresponds to the IPFIX definition of a Metering Process.

7.4.2 Management Component

The management component contains the traditional network management functions of configuration management, provisioning, fault management, monitoring/status reporting...etc. Note that from a network management point of view all the core functional components like Points of Measure, Network Measures, Logging/History components are considered to be managed objects. These managed objects are therefore subject to some or all of the traditional network management operations that may be performed upon them. Some of these management areas are detailed below:

- Configuration management: This management area is responsible for the creation/activation and removal/de-activation of managed objects.
- Event Management: This management area is responsible for notifications when any threshold in the functional components/managed object is exceeded.
- Status Reporting: This management area is responsible for viewing any one or all of the attributes of any given managed object and it's associated attributes.

7.4.3 Collector Component

Collector: A collector is that component of a measurement system that is the central information point and provides all the functionality associated with a middle-tier component. Note that this definition promotes a Collector from the IPFIX definition of an area where measurements are stored and acted upon, to a much larger context. In the context of a middle-tier component, all clients must interact with the Collector in order to perform any operation upon any managed object in a QoS system. The collector's role in a QoS system therefore, contains the functionalities of persistence, object request brokering, access control, dispatching and proxy agent handling among others. It is therefore the middle-tier of a QoS system and is the over-arching component, which is used for all interactions that any client may have with a QoS measurement System.

7.5 Architectural Overview

This high level view of the structural components provide for an overview of a measurement system. In order to see how measurements are taken, collected and delivered to end-users, each of these components and their internal interactions must be viewed in greater detail.

The following is the architectural overview of a QoS measurement system.

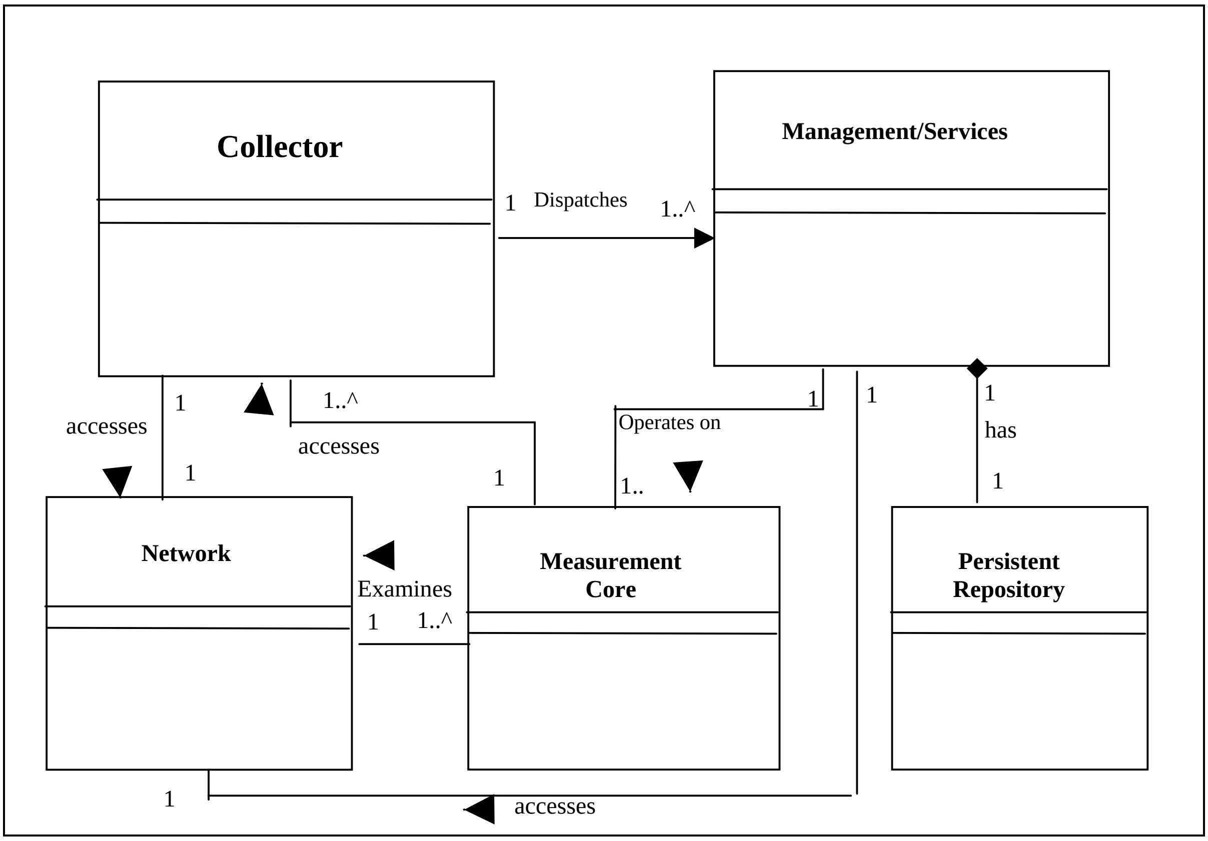


Figure 7-2: Architectural Overview of a QoS Measurement System

7.5.1 Measure Core Components

7.5.1.1 Measure

A measure is a fundamental abstraction, which describes what QoS metric is going to be measured, aggregated or reported on. Note that a measure is not the QoS metric; instead it describes all the properties that go into a measurement activity. These properties include ut are not limited to:

- The metric(s) that are to be measured.
- The time at which the measurement activity is to take place.
- The duration of the measurement activity.
- The interval at which measurement is to take place.
- The type of clock used to calculate the interval at which measurement is to take place.

This fundamental description of a measure is essential to all measures and may be extended and sub-typed into measurements that occur in a network, aggregated measurements that provided statistical surveys and reports that can provide details on the activities of any given network measure.

The structural representation of a Measure follows:

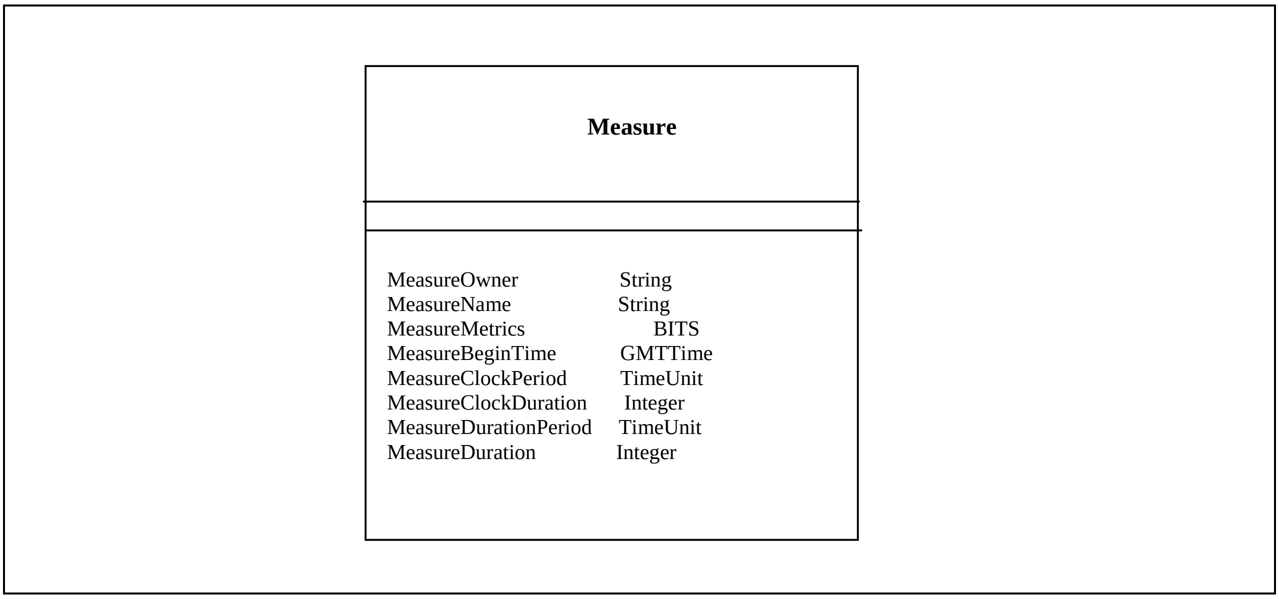


Figure 7-3: Structural Representation of a Measure

7.5.1.2 Metrics

Contained in the attributes of a measure are the metrics or types of measurements to be taken. In fact without these metrics, there could be no measurement activity. This metric component contains, but is not limited to, the identity of the metric, it's capabilities and it's description.

The following are the associations between the measure table and the metric table.

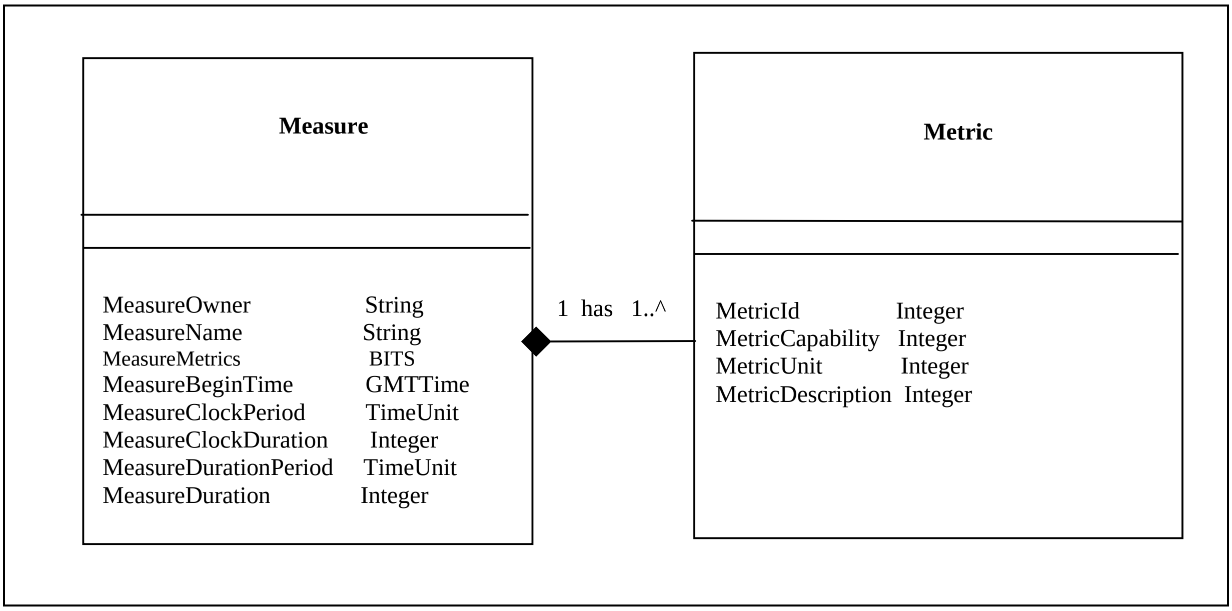


Figure 7-4: Associations between the Measure and the Metric Table

Note: In the above diagram the cardinality of the association between the measure and the metric indicates a one-to-many relationship between the measure and the metrics that it contains.

7.5.1.3 Measure Owner

Every measure must have an owner. This owner is actually the administrator of the measure. This means that an owner of the measure may create, retrieve, update and delete the measure that belongs to them. Additionally the measure owner may update attributes within the measure such as the MeasureBeginTime, the Measure Duration, etc. Care must be taken when doing this to make sure that all the components associated with the measure are also updated and that no value in the measure table is without it's corresponding value in an associated table.

Some of the attributes of the owner table are given below together with its association with the measure table and the metric table.

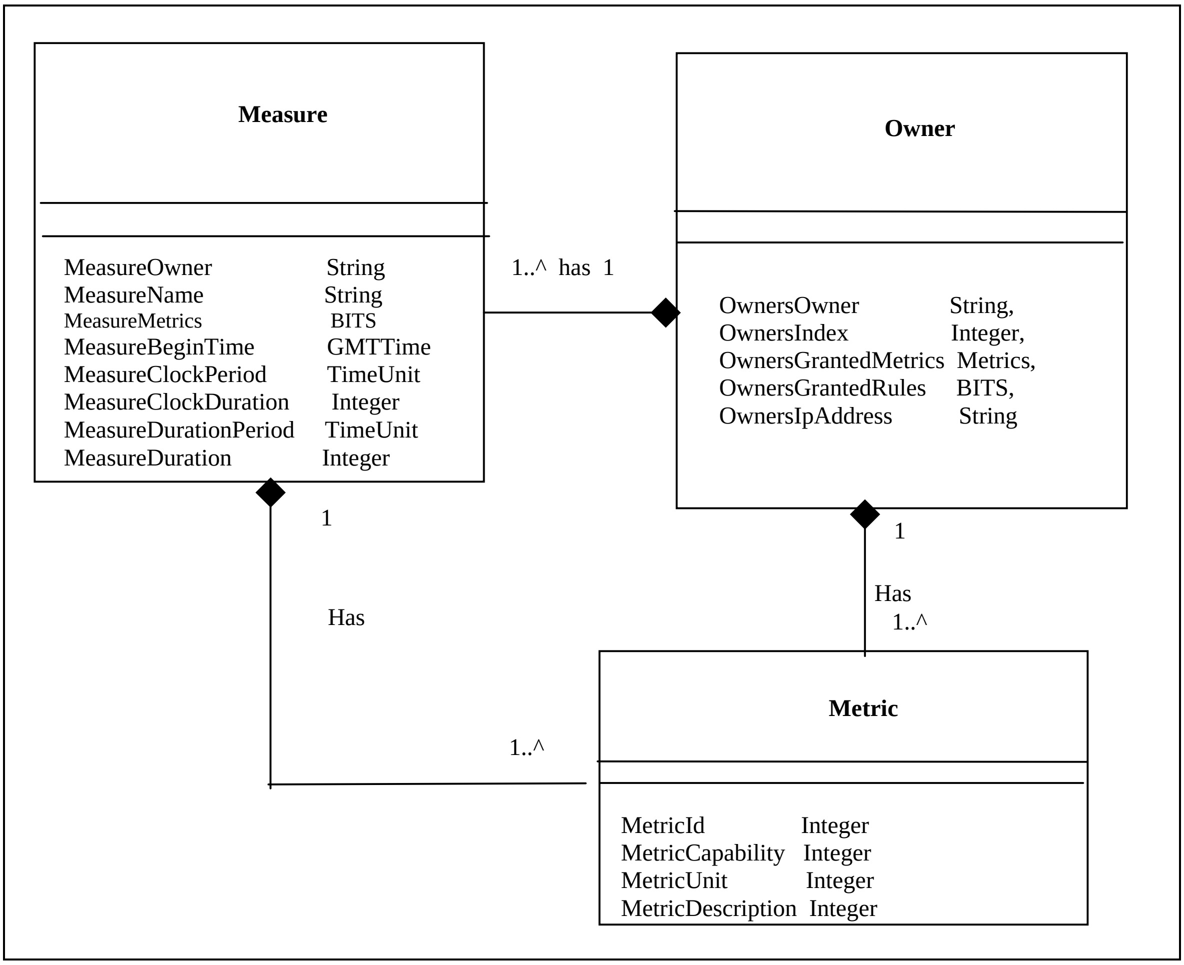


Figure 7-5: Attributes of the Owner Table

Note: In the above diagram the cardinality of the association between the measure and the owner indicates a one to many relationship between the owner and the measure. It also indicates a one to many relationship between the owner and the metric component. This association indicates that the Owner of the Measure must have ownership of the metrics that are in the measure. If this owner does not have ownership of the metrics in the measure, then the owner will not be able to perform the measurements described by the measure.

7.5.1.4 Network Measure

A network measure is an extension or sub-type of a Measure. It's relationship to a Measure is therefore of a child-parent association. The attributes of this component are to describe the end points at which measurements are to take place, as well as, how some of these measurements are to be taken. Together these two components, along with the metric component describe the measurement that is to be performed. Some of the attributes of a network measure are:

- Network Measure Source. The source or origination point of the measurement.
- Network Measure Destination: The destination or end point of the measurement.
- Network Measure Timeout delay: The time value at which the measuring entity considers the monitored packet(s) as being lost.
- Network Measure Clock Pattern: The clock pattern (a series of bits used to determine a valid instant of measure) used to perform measurements.

The following diagram illustrates some of the attributes of a network measure. It also indicates the association between the Measure and the Network Measure. This association illustrates that the network measure inherits attributes from it's parent, the Measure and extends these attributes with those characteristics that describe the network points (and the characteristics of these network points) at which the measurement are to be taken.

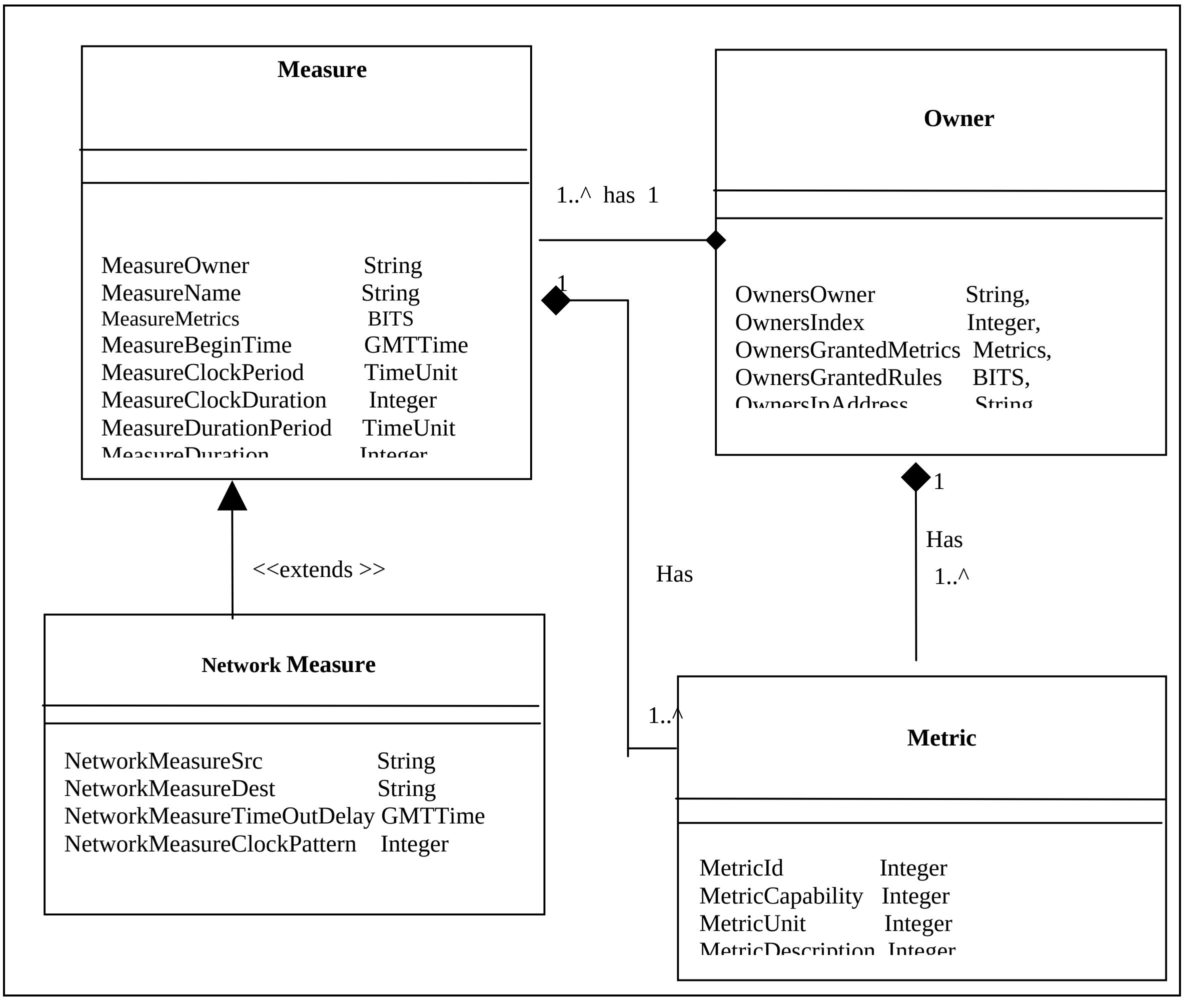


Figure 7-6: Attributes of a Network Measure

7.5.1.5 Aggregated Measures

An aggregated measure is also an extension or sub-type of a Measure. It's relationship to a Measure is therefore of a child-parent association. The attributes of this component are to describe the statistics that may be performed on network measures. In other words aggregated measures describe those measures that are to be aggregated into some form of statistical representation. The statistics may then be sent to users so that they may view trends in their networks. It is noteworthy that an aggregated measure gets the raw data for its operations from a persistent repository. This means that the aggregated measure examines a logging/history component in a persistent repository and uses the measures logged there in order to perform statistical computations. The results of these statistical computations may be re-entered into the logging/history component where all measures are stored; or they may be in a separate statistical/logging component defined for this purpose, in the persistent repository.

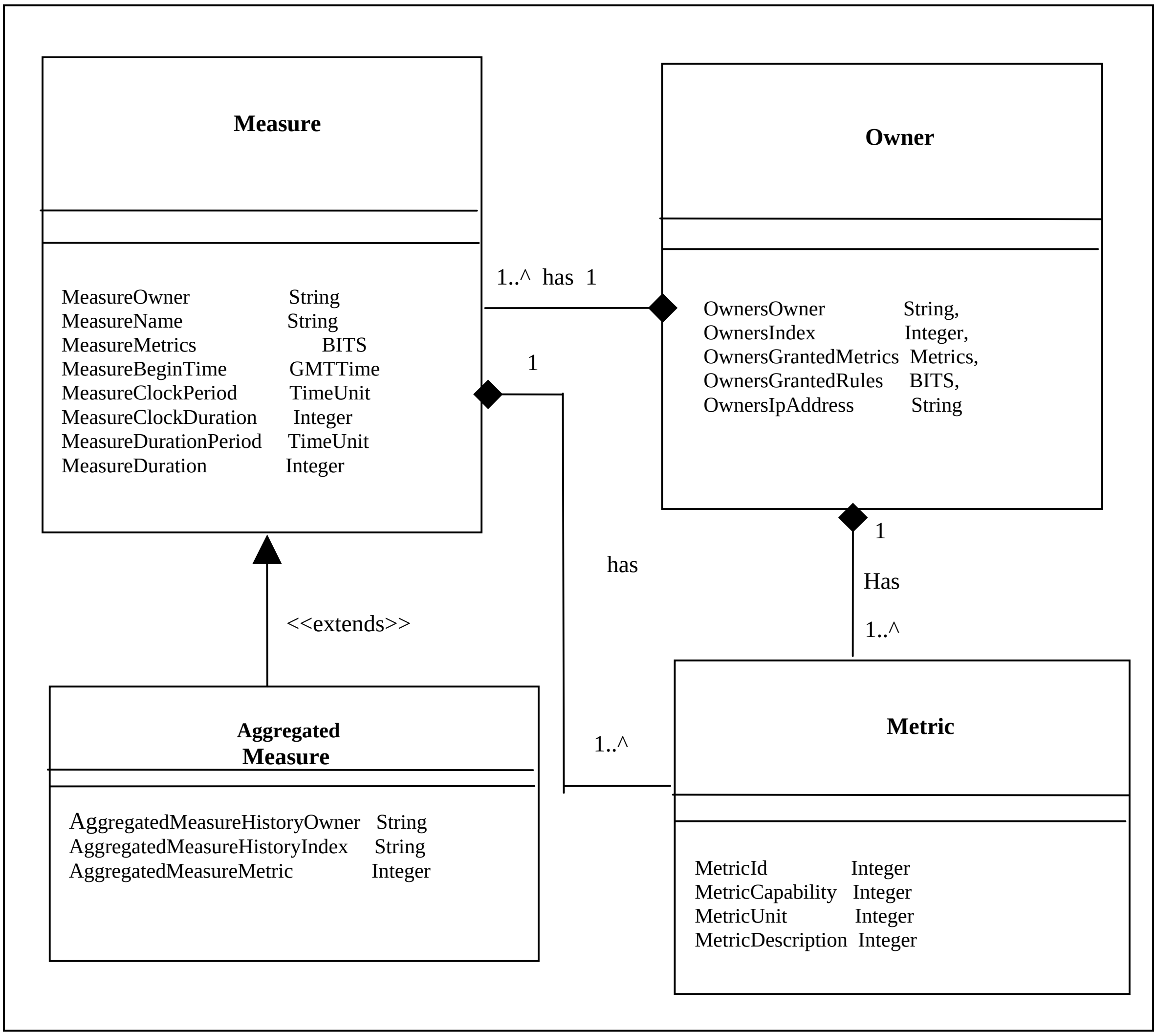


Figure 7-7: Aggregated Measures

7.5.1.6 Point of Measure

The measure, and all it's associated components describe the measurement to be performed. However, these measurements must have a location or a point at which they occur. This measurement area or measurement point houses, as it were, various measurement activities and is called a point of measure. Basically a Point of Measure represents the physical location/container which houses network measures.

Some of the attributes of a Point of Measure are:

- Identifier: There must be a unique identifier for the Point of Measure.
- Measure Address Type: This qualifies the type of the address that is defined in the management address. Not all addresses are IP addresses and in order to be flexible to different network protocols the type of address must be defined.
- Measure Address: This is the address of the measure. It may be an IP address or it may be some other form of Address. The format of this address depends upon its type as specified in the Measure Address Type.

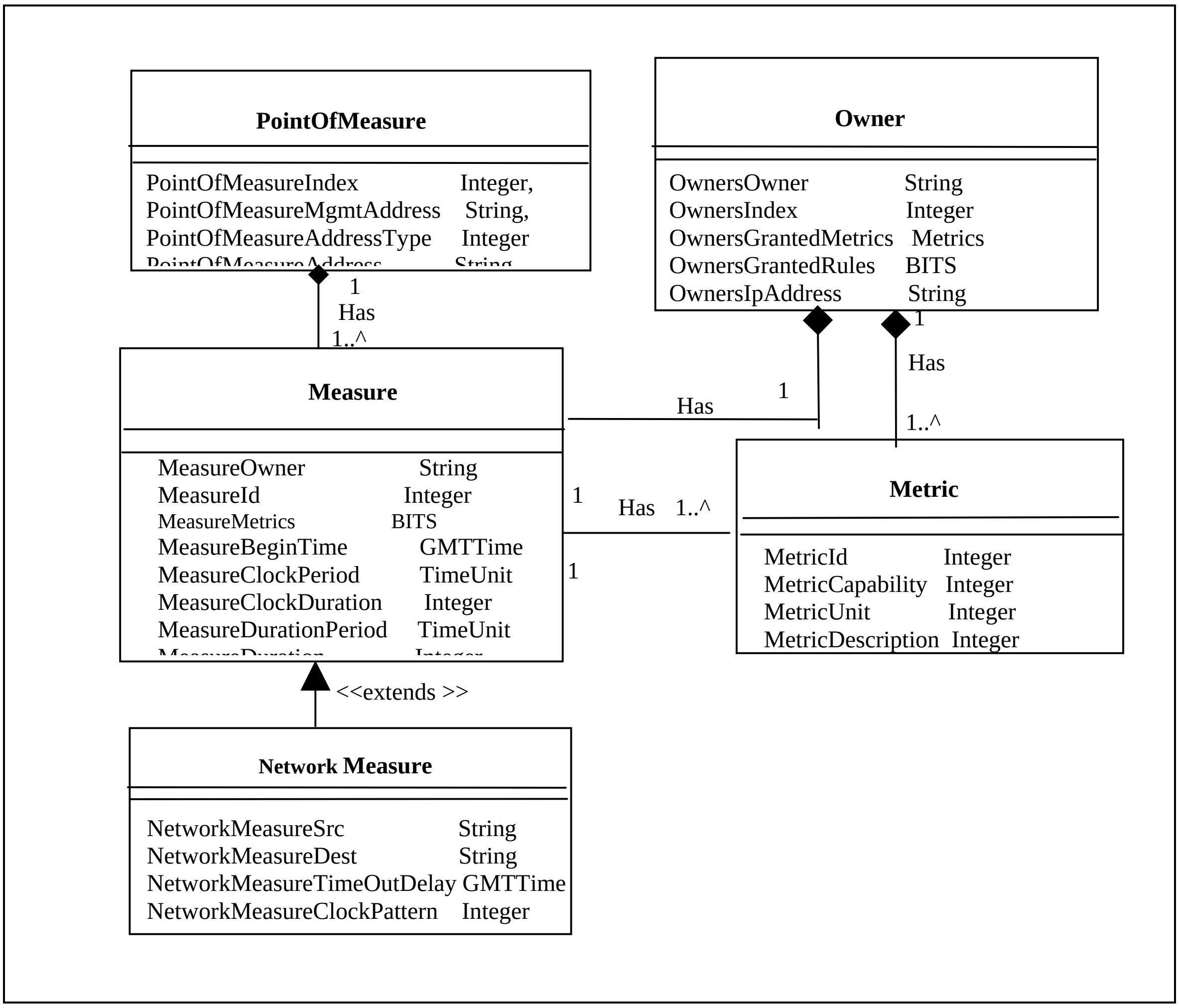


Figure 7-8: Point of Measure

7.5.1.7 History/Results/Logging

The Point of Measure and its associated measures is responsible for performing measurements between any two endpoints. The measurements performed at the Point of Measure are dynamic, in that, they occur at regular intervals and last for a given period of time. Given the dynamic nature of measurements there needs to be some component that will store the measurement results, so that end-users may access them. This component that stores results may be called the logging component, history component, or result-sharing component. For this purposes of this document, the term History component will be used. The attributes of the history table are given below.

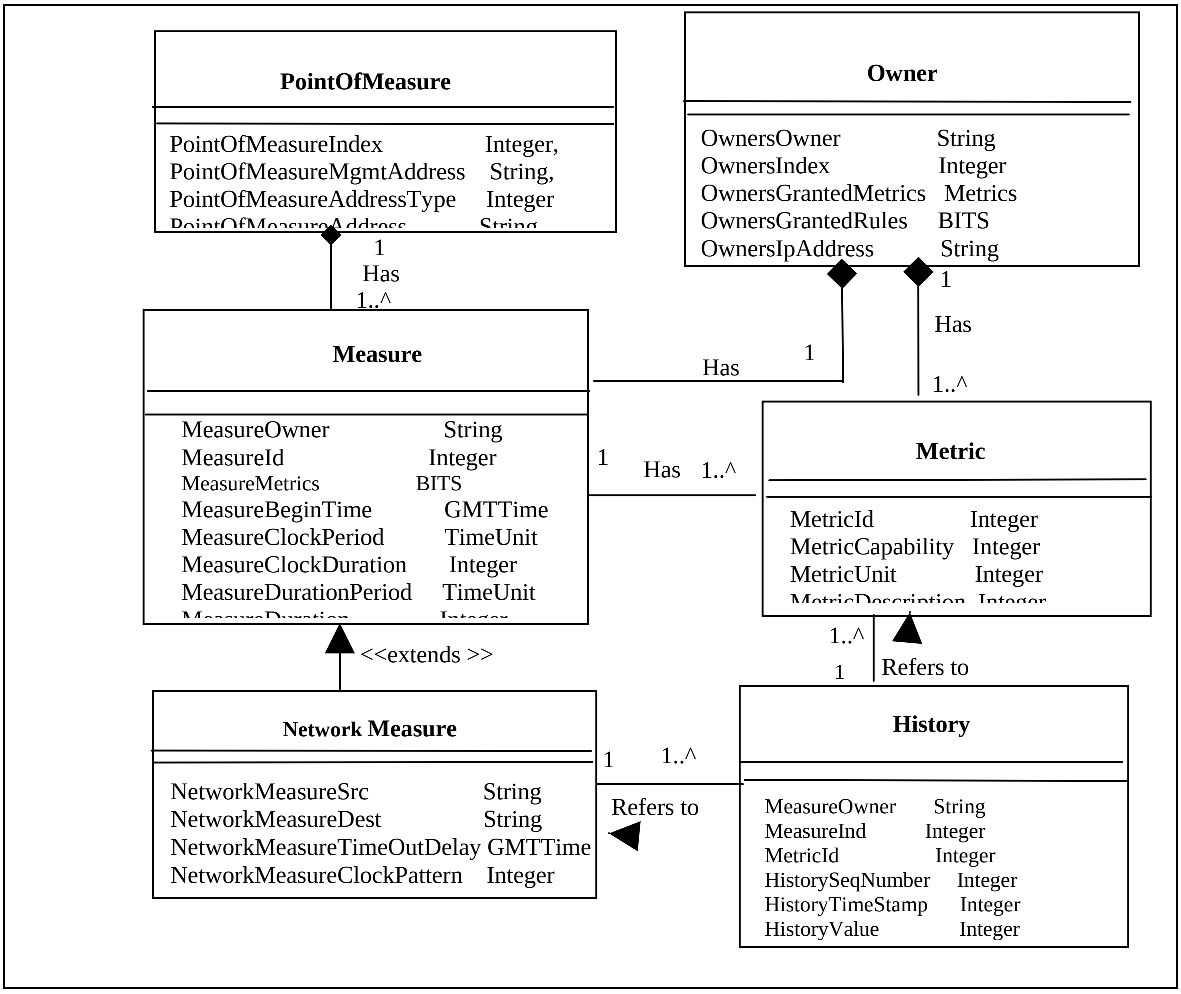


Figure 7-9: Attributes of the History Table

7.5.1.8 Event Notifications

All QoS measurement systems must posses a component that emits notifications. These notifications are to inform interested users of important events that are taking place in the QoS measurement system. These events may indicate when thresholds for certain measures have been exceeded. Some of these threshold exceptions or violations may have different degrees of severity that may range from minor events to major events that can impact network performance. The following are the characteristics of an event notification entity.

- The type of measure to be monitored must be defined. This allows for the notification entity to determine the measurement end points and the metric or metrics that are being monitored for threshold violations.
- The number of notification to emit. This determines if the metric is emitted one time, continuously, or a number of times over a give period.
- The type of notification to emit. This determines if the notification is logged or an email is sent out, or a PDU is emitted.... and so on.

The following is the Event Notifications entity, called for the purpose of this document, the Report Setup.

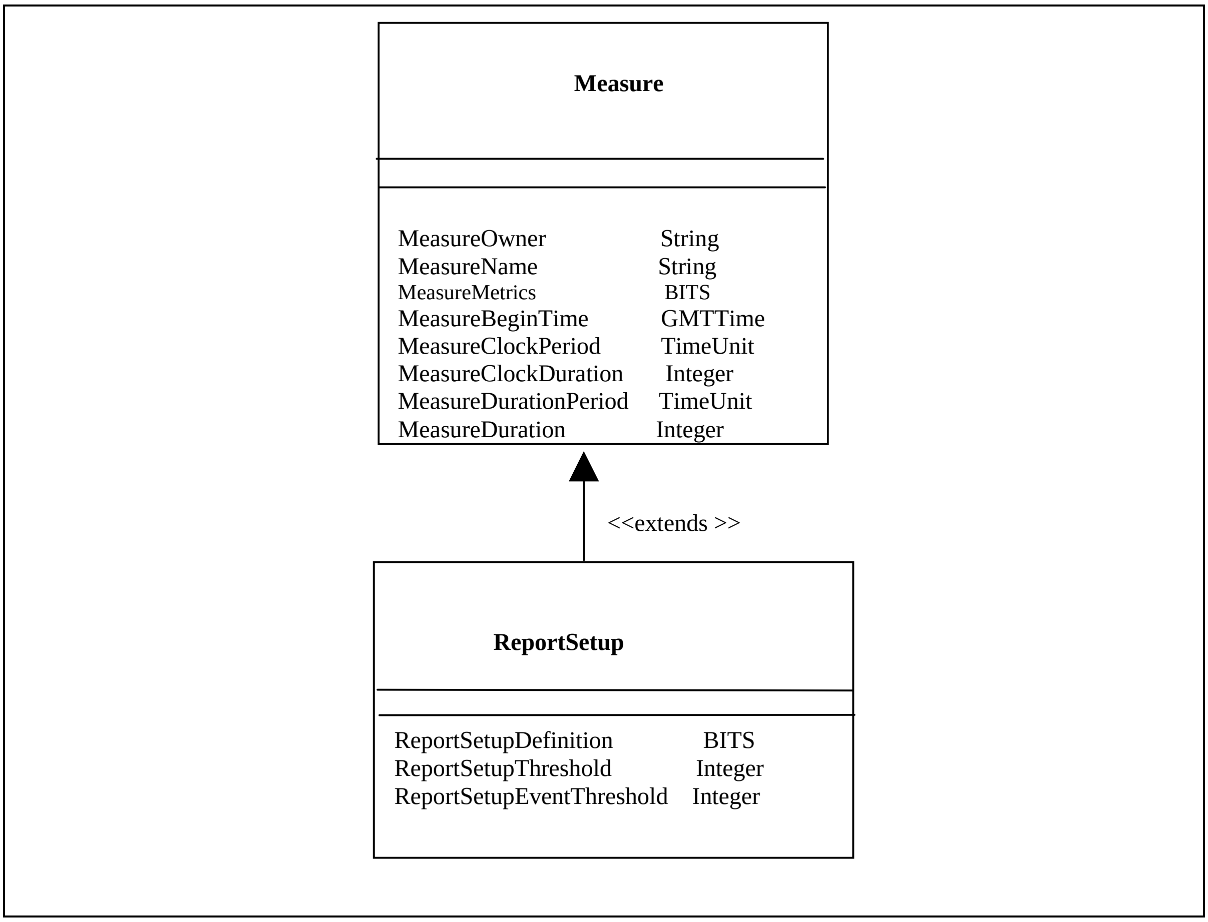

Measure
MeasureOwner String
MeasureName String
MeasureMetrics BITS
MeasureBeginTime GMTTime
MeasureClockPeriod TimeUnit
MeasureClockDuration Integer
MeasureDurationPeriod TimeUnit
MeasureDuration Integer
<<extends >>
ReportSetup
ReportSetupDefinition BITS
ReportSetupThreshold Integer
ReportSetupEventThreshold Integer

Figure 7-10: Event Notifications Entity

This concludes the core components that are required to perform measurements in any given domain. However, the problem of how to manage these components still exists. This management problem is addressed by a management entity, which has the responsibilities of configuring, maintaining and performing fault management on these components. The following section indicates some of the responsibilities of the management component in QoS measurements.

7.5.2 Management Component

The management component in a QoS measurement system performs all the traditional functions associated with this role. It is therefore responsible for configuration management, provisioning, fault management, event notification and status monitoring. The association between a management component and the QoS functional components is that of a governing or directive component. This means that it is responsible for the configuration and maintenance of the QoS measurement system. The management component is divided into two parts, a manager and an agent. The manager forms the requests that are directed to an agent. The agent is that component which performs the actions associated with the request and returns results to the manager.

For the purposes of this document, a manager and its agent must reside in the same domain. This manager and its agent handle all the management operations for managed objects (Points of Measure and ALL their associated components) for the domain they reside in. Inter-domain configuration and monitoring is not the responsibility of the management component, as it has no jurisdiction over managed objects in an external domain. In order to perform cross-domain management operations, an administrative entity must act on it's behalf, as a proxy manager.

This administrative component is called the Collector and its functions are listed in a following sub-section.

Therefore a management component and its properties as given here, only perform intra-domain management operations.

- The management component must have an entity that has the role of a manager. This manager is the originator of all operation requests to all the agents that it knows about.
- The management component must have an agent, which performs the operations required of it, and return results to the manager.
- The management component must have a data description language in order to describe the managed object in the QoS measurement system.
- The management component must have a methodology for event notification.
- The management component must have a protocol for the exchange of messages between the manager and it's agents.

The following is a model of the manager; it's agent and its association with a measurement system. Note that the Point of Measure in the diagram below, indicates not just the Point of Measure but all it's associated components, including the measure, network measure, metric, owner.... etc.

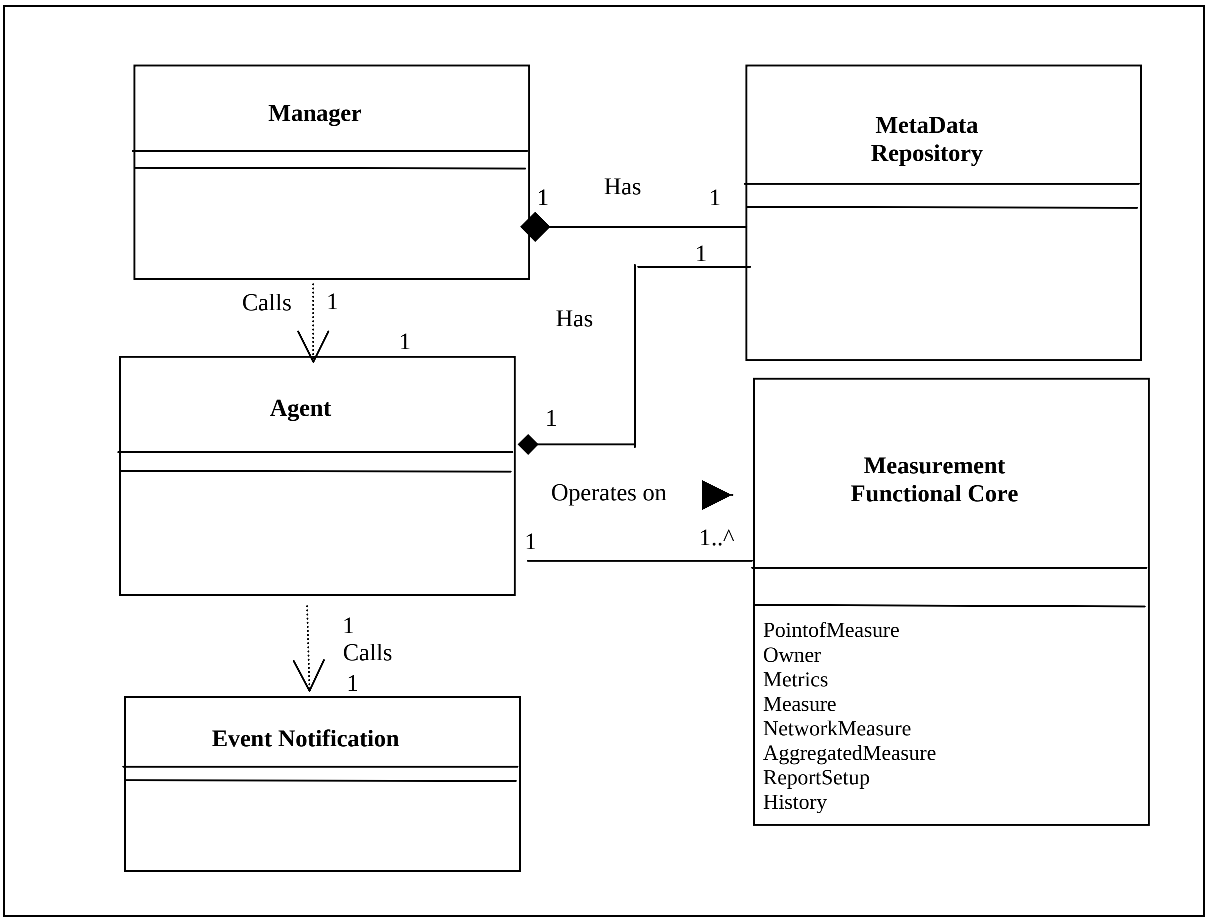


Figure 7-11: Management Components Model

7.5.3 Collector

The Point of Measure component and it's associated measures together with the management entity are the core components needed to setup a measurement between any two end points in an intra-domain network.

However, when the measurements need to be perform across domain boundaries, a different set of problems arise. This is because management systems are usually confined to an area of control within their domain boundaries. A single customer may not access another customers network unless authorized to do so. In the event that a customer is granted access to another domain, the issues of what operations that customer is allowed to perform arise. These issues belong to the domain of access control, security and cross-domain management. In order to allow an inter-domain measurement to take place and to allow the results of these measurements to be shared across domains, there must be a component that bridges the gap between two separate sovereign domains. This component, for the purposes of this document, is called the Collector.

The properties of the collector, in its role as a bridge between gives the Collector the responsibility for handling all security, access control, and even configuration management requests. It is noteworthy that the Collector also contains within it the persistent repository that is used to store all the data gathered from measurements taken at a Point of Measure.

Given the above component description, the Collector may be classified as the middle-tier component in any QoS measurement architecture and the functionality of this component and its sub-components may be summarized as follows:

Note: A Broker architectural pattern is used to describe the components of the Collector. This allows the Collector to be instantiated using Web Services, CORBA, EJB, or any other implementation that supports this middle-tier model.

- Collector-Broker: This is the component that is responsible for finding the appropriate service to fulfill the request and dispatching the request to that service. Note that the bottom end of the Collector-Broker contains plug able protocols that are used to interact with the requested service.
- Collector-Bridge: This component is responsible for contacting other Collectors that are in heterogeneous networks. Its role may become critical in inter-domain QoS measurements since it may be the entity that allows for interaction between two heterogeneous networks.
- Collector-Client Proxy Agent: The agent responsible for marshalling and unmarshalling requests.
- Container: This is the component that is the front end to all the services requested by clients. It provides the functionality of access control, transaction control and concurrency control. Additionally, this component has the Factory/Finder functionality and all other activation and passivation functionality associated with Service activation and de-activation.
- Collector-Server Proxy Agent: This is the component responsible, on the server side for unmarshalling and marshalling requests.
- Server/Service: The component that actually has the object that contains the behaviors to fulfill the request. It performs the request and sends back the response to the client.
- Persistent Repository: The component that is responsible for storing data.

A diagram illustrating the structural relationship of these components is given below. Note: the client side is also detailed so that the structural relationship between a client and the Collector may be realized.

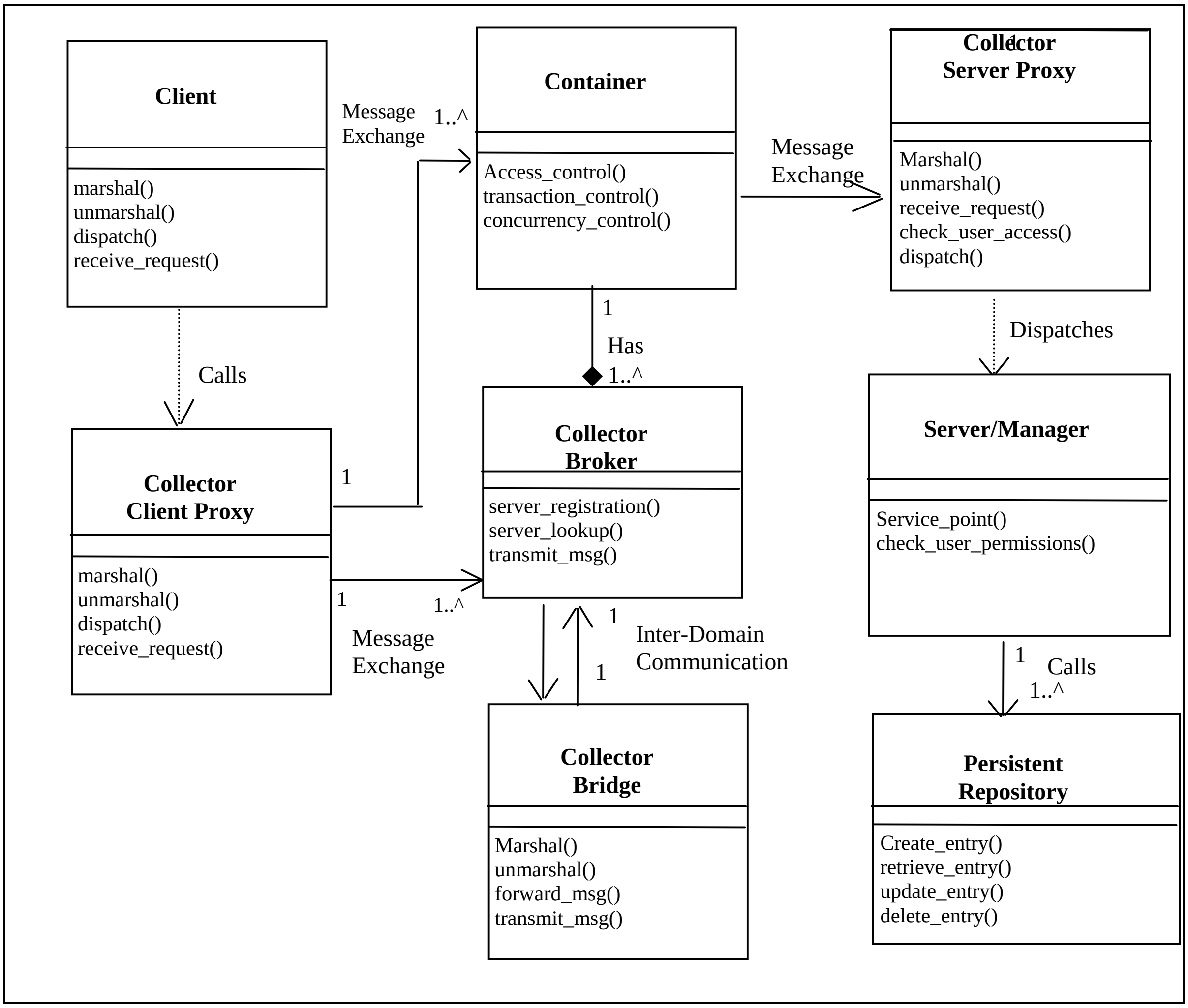


Figure 7-12: Collector

7.5.4 Use Cases/Interactions

The interactions that may occur between these various components may be divided into two broad categories: intra-domain interactions and the inter-domain interactions. In both cases the clients involved in these interactions may request the same operations that configure, access and otherwise manage their networks. However, in the case of intra-domain interactions all the operations are serviced by the management entity in the enterprise domain; in the case of inter-domain all the operations occur across all domains. It is noteworthy that in both cases all system interactions possess certain common properties. These properties can be generalized into a high-level system behavior. This generalized interaction between any given client and the QoS measurement system **applies both to intra-domain and inter-domain interactions** and may be detailed as follows.

7.5.4.1 Generalized QoS System Interaction

The following are the steps used to perform any operation a QoS measurement system in order that a measurement may be configured, activated, monitored and viewed.

- The administrative client issues a request for an operation upon a given QoS object with the appropriate attribute values filled in.
- The request goes to the client’s proxy agent that contacts the Collector Broker in order to locate the server that can fulfill this request.
- The Broker checks for the server. If the client accessing the server has that server as part of it's domain, it forwards the server's contact information to the clients Proxy Agent.
- The Proxy Agent marshals the request and sends it to the server/manager for processing.
- The server performs the operation and the response is sent back to the server's proxy agent that marshals the request and sends it back to the client.

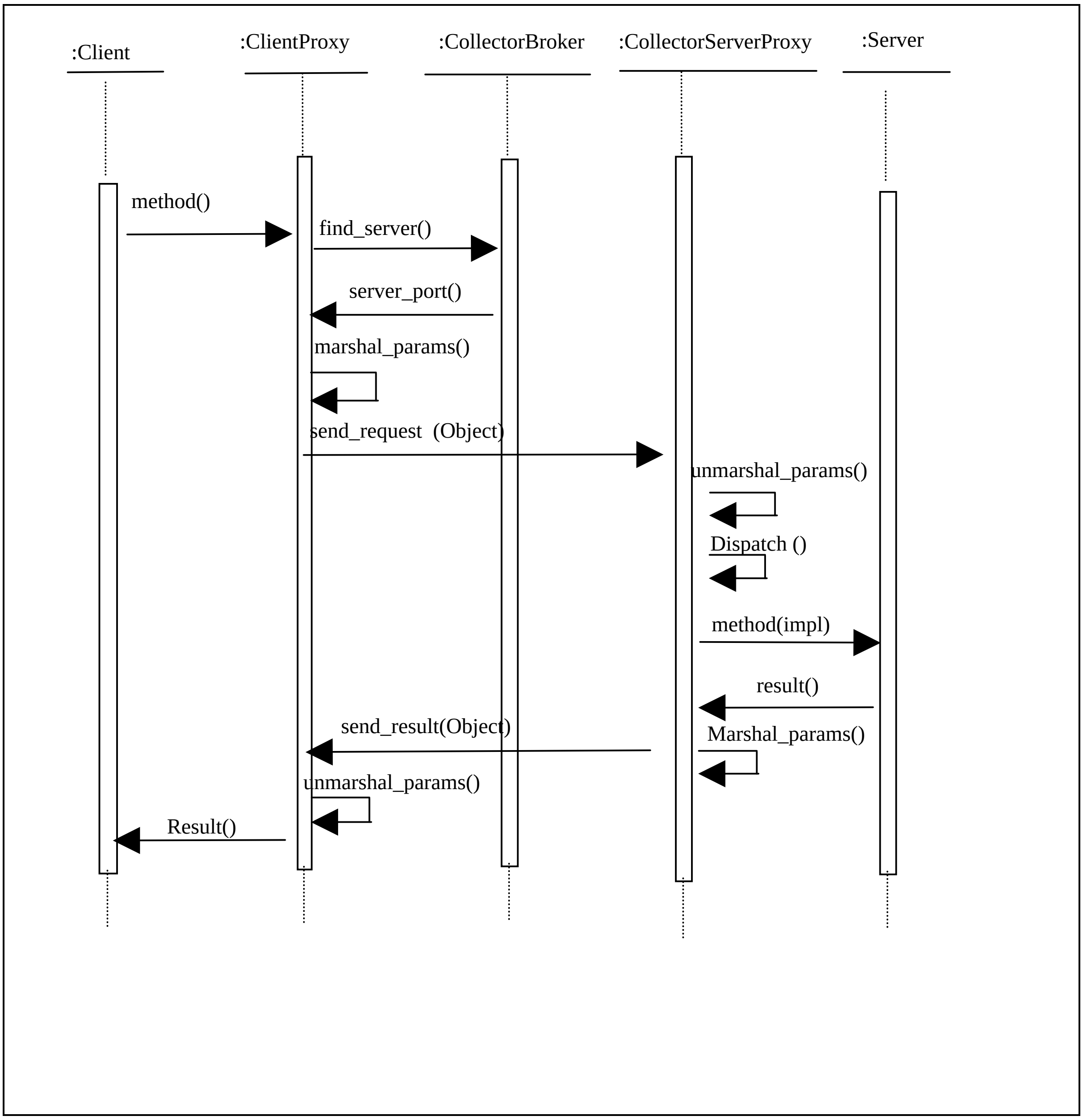


Figure 7-13: Generalized QoS System Interaction

All generalized system interactions, like the one above can only take place after the server side components have been registered and users are configured and logged into the system. After this, all interactions take place in two parts. The first part is authentication and all users must login or authenticate themselves prior to any system interaction taking place. The second part is actually the request for an operation that the system must perform.

These interactions are given below:

7.5.5 Collector Server Configuration

In order for any service to be accessed, the service must be registered in with the Collector-Broker. However, prior to its registration, its operational parameters must be configured. The configuration data for a service may be expressed in XML and may have the following entries.

- The service identifier. This is a unique identifier for the service.

- Access control information: The information on the users who may access the service and the roles they play when accessing the service.
- The location/address of the service.

In order for the configuration information to be created and added to the persistent repository, the following actions must take place.

Note: a system administrator performs all the following functions. This administrator is part of the factory level defaults issued with the system and the services that this administrator may activate are already part of the system. As already stated, the administrator must go through an authentication prior to performing the required administrative operations.

7.5.5.1 Authentication

The steps performed for authentication are as follows:

- A user/client logs in with a user name and password.
- The client sends the request to the Client proxy agent.
- The client proxy agent goes to the broker to request the login service.
- The broker finds the login service and returns the service point to the client proxy agent.
- The client proxy agent issues the login request to the login service. NOTE: the login service may run in the same domain as the client requesting a login or it may be running in an external domain. In the second instance, the login request is also considered a domain access request.
- The login service checks the persistent repository to validate the user name and password.
- If the user is allowed system access to the system/domain, a secret key is returned to them based off an MD5 algorithm using the user name, password and a shared secret. This methodology is well known and is part of existing authentication protocols. Kerberos is one such protocol and it may even be used for authentication in this case.

If the user is disallowed form system/domain access no further action takes place.

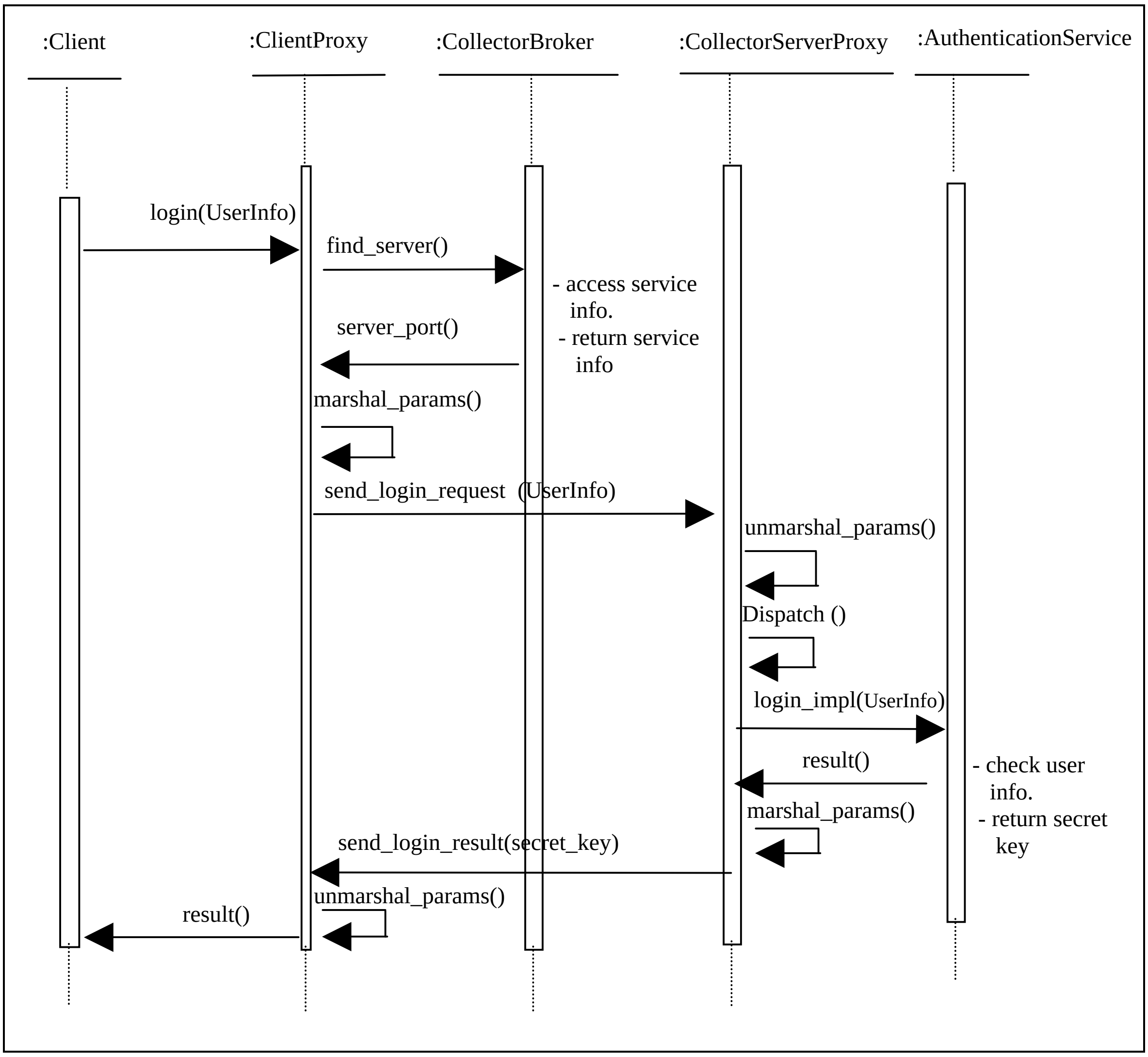


Figure 7-14: Authentication Diagram

7.5.5.2 Collector Server Configuration

- The administrator issues a configuration request for the client proxy.
- The client proxy initiates a request for a service to the Collector Broker.
- The Collector Broker returns the service point or location of the service, usually a port number, to the client proxy agent.
- The client proxy forwards the request to the Container that checks the access of the user.
- If the user has access to the service, the service request is forwarded to the server proxy.
- The server proxy agent unmarshals the request and dispatches the request to the appropriate service.
- The service performs the request and sends the response back to the server-proxy.
- The server proxy marshals the request and sends it back to the requesting client.

The diagram below details this interaction using SNMP. However, this is not the only protocol that may be used to perform this operation.

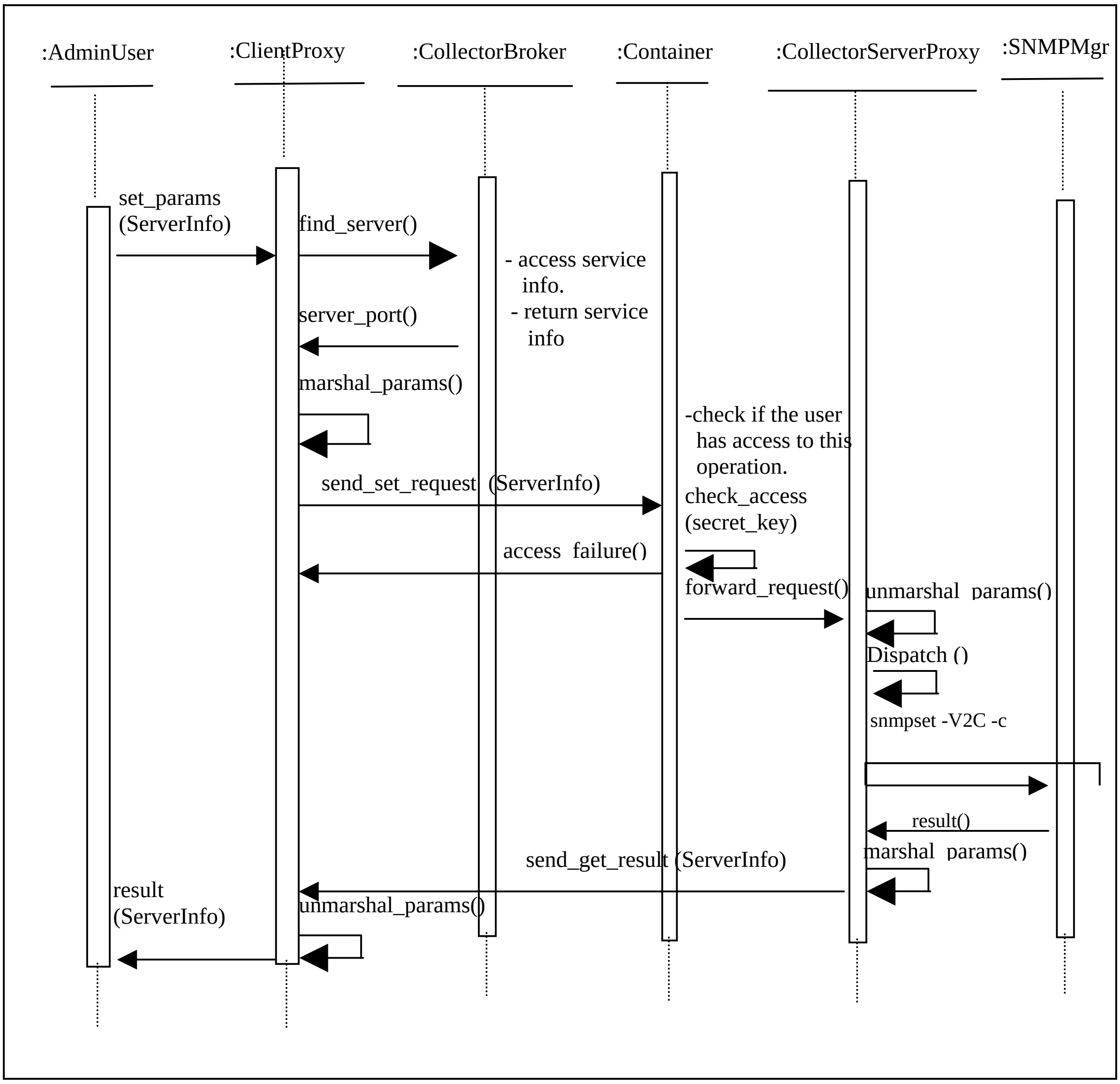


Figure 7-15: Collector Server Configuration

7.5.5.3 Collector Server Registration

- The server is initialized first, and as part of its inherent configuration, knows the port of the Collector's Broker that it must contact in order to register its services. It may determine this from a Directory service.
- The server registers its services and location with the Collector-Broker and listens for all contacts/connections for service requests.

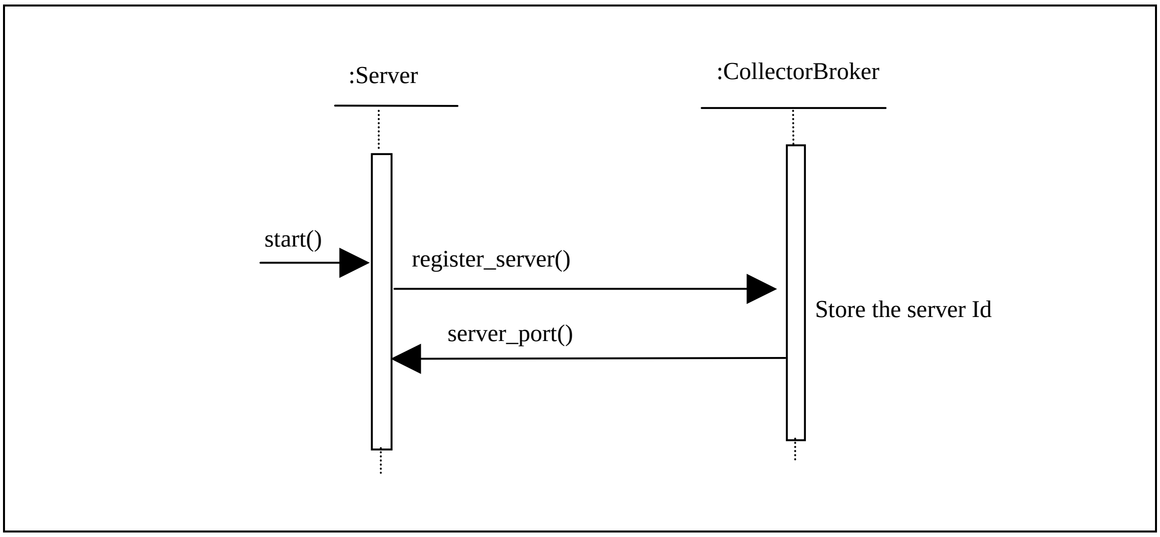


Figure 7-16: Collector Server Registration

7.5.6 Collector User Configuration

7.5.6.1 Authentication

The steps for authentication are already provided in the Authentication for the server as given above in section 10.1.

7.5.6.2 **Configuration Interaction**

- In order for a user to be configured in the system, the following steps must be performed.
- The administrator issues a configuration request for the client proxy.
- The client proxy initiates a request for a service to the Collector Broker.
- The Collector Broker returns the service point or location of the service, usually a port number, to the client proxy agent.
- The client proxy forwards the request to the Container that checks the access of the user.
- The user configuration request is delivered to the server-proxy agent.
- The server proxy agent unmarshals the request and dispatches the request.
- The service performs the request and sends the response to the server-proxy.
- The server proxy marshals the request and sends it back to the client.

The diagram below details this interaction using SNMP. However, this is not the only protocol that may be used to perform this operation.

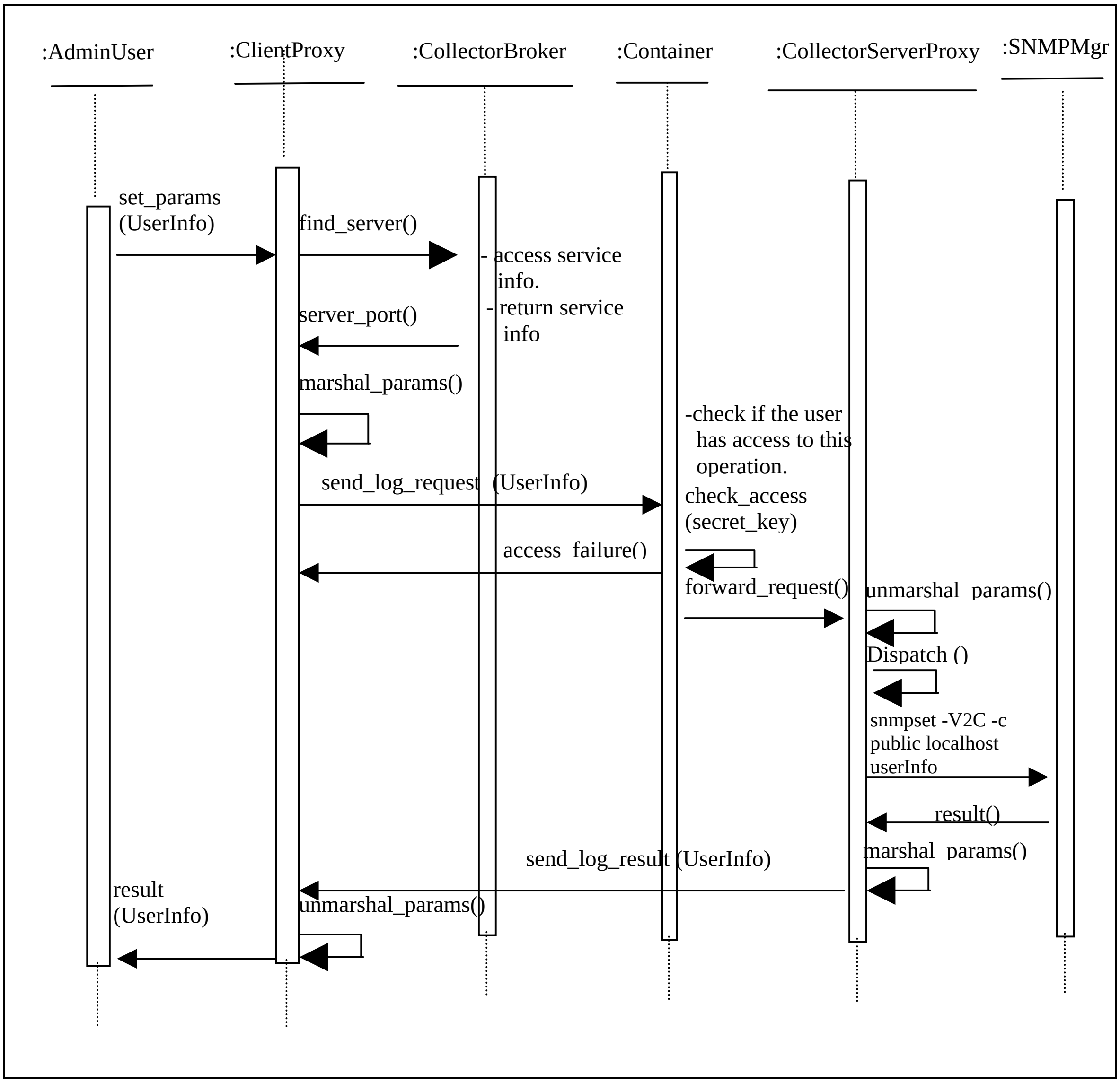


Figure 7-17: Configuration Interaction

7.5.7 Embedded System

All the above interactions are from the point of the user or the server. However, the embedded system containing the Point Of Measure may also be considered a client of the Collector. Users may be defined as Consumers/Subscribers of the QoS measurements in the Collector, while the Point of Measure (with all its associated components) may be considered the publisher of the QoS measurements in the Collector.

The interactions of a Point of Measure with a Collector may be detailed as follows:

- The measurement activities in a Point of Measure generate measurement flows.
- The Point of measure issues a logging request to the client proxy.
- The client proxy goes to the broker to find the location of the logging service.
- The Broker returns that logging service’s location (port, IP address).
- The client proxy marshals the request and sends it to the Container (which is listening on that port and address).
- The container checks the secret key of the Point of Measure.

- If the Point of measure has access to the service the logging request is delivered to the server-proxy agent.
- The server proxy agent unmarshals the request and dispatches the request.
- The logging service performs the request and sends the response to the server-proxy.
- The server proxy marshals the request and sends it back to the client, in this case, the Point of measure.

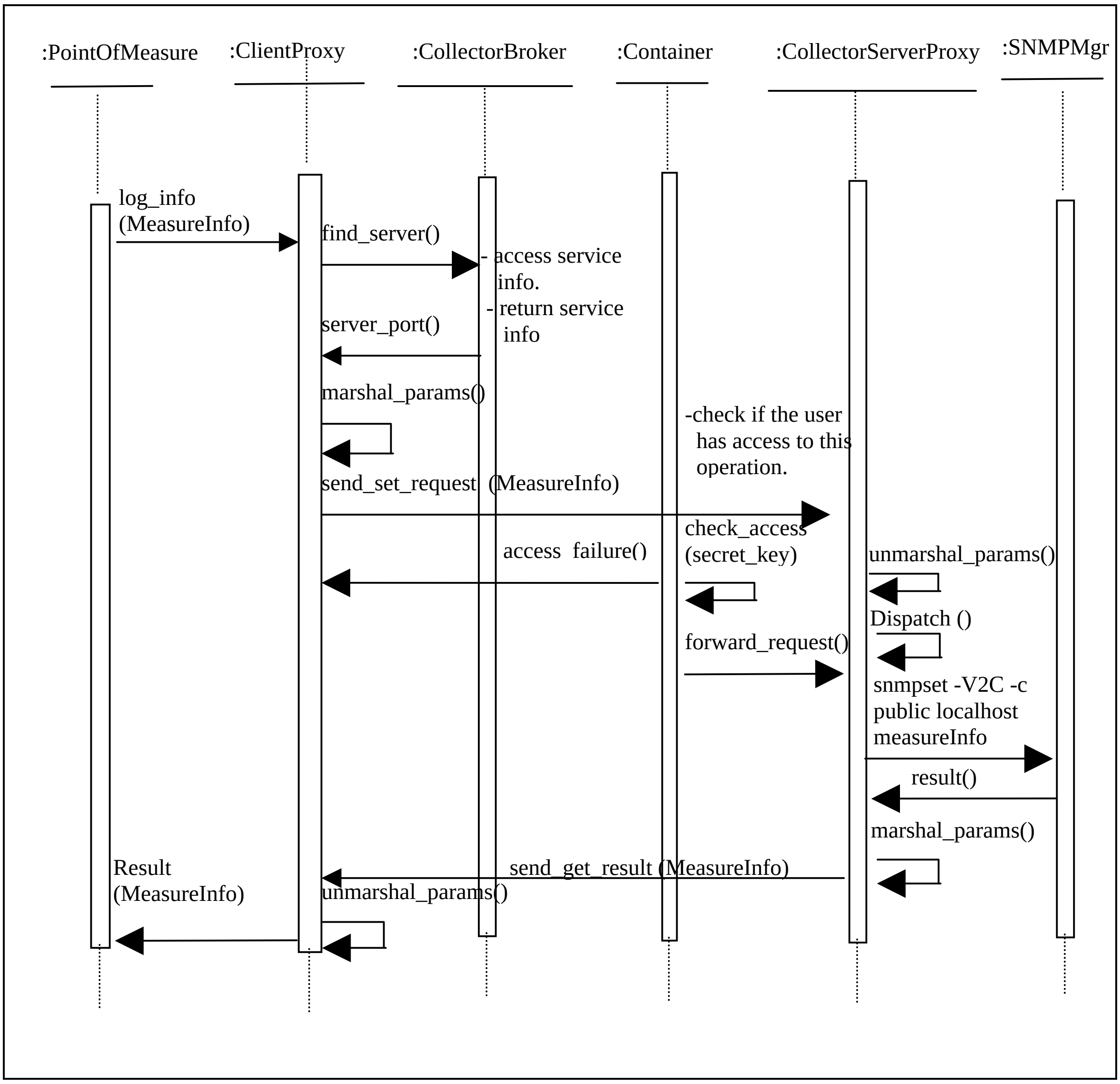


Figure 7-18: Embedded System

7.5.8 Conclusion

As can be seen from the above diagram, the interactions between the system components when configuring a server and a user differ only in the requested methods and services that are invoked. This indicates that all system interactions follow the above pattern and by instantiating different services and components, when needed, the system will be able to perform many varied operations using varied protocols and services, This makes the above system flexible, extensible and secure when performing intra-domain or inter-domain QoS measurements.

7.6 Limitation of the proposed architecture

Some points may be problematic in this architecture:

- It may be difficult to deploy an inter domain communication due to political reasons.
- There is no standard to exchange configuration data between domains.
- The standards are in development concerning the consultation of measures (IPPM)
- Security is an important issue in this architecture due the open interface.

8. SECURITY REVIEW

8.1 Security Requirements

Measurement points/Points of measure

Sec 1.a

These entities **MUST** be protected against DoS attacks and in particular flooding attacks.

Sec 1.b

The signalling to control a Measurement Point **MUST** be secured to ensure the authentication of the management entity and the integrity of its orders.

Measure

Sec 2.a

The results sent to the collector **SHOULD** be protected to ensure the authentication of the sender, ie. the Measurement Point, and the integrity of the measurement information.

Sec 2.b

The copied packets **SHOULD** be protected to ensure the authentication of the sender, ie. the Measurement Point, and the integrity of the measurement information.

Sec 2.c

The copied packets **MAY** be ciphered to ensure the confidentiality of the measurement information.

Collector

Sec 3.a

This entity **MUST** be protected against DoS attacks and in particular flooding attacks.

Sec 3.b

The signalling to control a Measurement Point **MUST** be secured to ensure the authentication of the management entity and the integrity of its orders.

Configuration Management

Sec 4.a

The signalling to control a Collector **MUST** be secured to ensure the authentication of the management entity and the integrity of its orders.

Fault Management

Sec 5.a

The signalling to inform a management entity **MUST** be secured to ensure the authentication of the Fault Management and the integrity of its information.

9. SUMMARY AND CONCLUSIONS

The aim of this document was to define both a management and measurement architecture for QoS measurement such that users and service providers can have a common understanding of the performance and reliability provided by the Internet clouds that are traversed.

This definition relies on several aspects:

- The overview of passive and active measurement architecture, introducing a common terminology. A global QoS measurement architecture is defined in this context.
- The overview of QoS Measurement Operations. In order to perform QoS measurement, a set of operations usually has to be applied to the traffic flowing in the network. Each implementation usually uses a combination of these basic operations in order to provide QoS measurement services.
- The overview of existing products, especially dealing with path measurement and end-to-end measurement.
- The requirement for both Intra and Inter-Domain measurements. In order to provide the widest capacities in term of QoS measurement the 6QM architecture must be able to provide both passive and active measurement capabilities. Both measurement capabilities should be based upon a common infrastructure base.
- The standardization of the exchange of measurement results. The aim is to define and standardize a methodology for exchange of measurement results in the form of an “Implementation Agreement”.
- The setup process for creating end-to-end measurements. Setting up such processes across administrative domains requires peering agreements that specify the mechanisms employed to implement the setup process.
- The standardization of the format and the semantics of test packets. The aim is to generalize IPPM metrics measurement among heterogeneous points of measure and to couple active and passive techniques.
- The definition of a global management layer relying on technical layers. The approach taken with each structural component is from “the ground up”.

All these aspects were developed with IPv6 concerns in mind. This should be a good basis to build a 6QM management and measurement architecture following the Work Package 2 priorities:

- Troubleshooting.
- Network and transport SLA.
- Standard configuration and reporting interfaces.
- Security and reliability of the control and reporting planes.
- Peering management of the measurement systems.

10. References

Name	Title	Version	Date
[Adc02]	Fused Fiber Technology Couplers, Splitters, Wavelength Division Multiplexers, ADC technologies, Available at http://www.adc.com/Library/Literature/1400.pdf, 2002.		2002
[Agi02]	Internet Advisor, Agilent Technologies, Available at http://onenetworks.comms.agilent.com/internetadvisor/default.asp, 2002.		2002
[Brix]	Brix 2500 Verifier, Brix Networks Inc, available at http://www.brixnet.com/products/brix2500.html, 2002.		2002
[CADENUS-D21]	"QoS control in SLA networks —Rev 1.0", Cadenus IST Project, March 2001		March 2001
[Cai02]	Workload Measurement Tools Taxonomy, CAIDA, http://www.caida.org/tools/taxonomy/workload.xml , 2002.		2002
[Cal02]	Diameter Base Protocol, draft-ietf-aaa-diameter-15.txt, Pat R. Calhoun, John Loughney, Erik Guttman, Glen Zorn, Jari Arkko, October 2002.		October 2002
[CIMV26]	http://www.dmtf.org/spec/cim_schema_v26.html		2002
[Cis02]	Configuring the Catalyst Switched Port Analyzer (SPAN) Feature, Cisco systems, available at http://www.cisco.com/warp/public/473/41.html, 2002.		2002
[Ciu02]	Measuring one-way metrics without a GPS, Augusto Ciuffoletti, Passive & Active Measurement Workshop (PAM 2002), March 2002.		March 2002
[Char]	Chariot, NetIQ Corporation, Available at: http://www.netiq.com/products/chr/default.asp, 2002.		2002
[Chi02]	Some hardware implications of packet sampling, Derek Chiou, available at http://www.iesg.org/proceedings/02jul/slides/psamp-2/, July 2002.		July 2002
[Cho02]	Adaptive Random Sampling for load Change Detection, B. Choi, J. Park, Z. Zhang, ACM Sigmetrics 2002, June 2002.		June 2002
[Clai02]	Cisco Systems NetFlow Services Export Version 9, draft-bclaise-netflow-9-00.txt, B. Claise, June 2002.		June 2002
[Creef]	The architecture of the CoralReef Internet Traffic monitoring software suite, Ken Keys, David Moore, Ryan Koga, Edouard Lagache, Michael Tesch, and K. Claffy, CAIDA, available at : http://www.caida.org/outreach/papers/2001/Coral		2001

	Arch/, 2001.		
[Dag02]	DAG 4 architecture, University of Waikato, available at http://dag.cs.waikato.ac.nz/dag/dag4-arch.html, 2002.		2002
[Dee01]	FPGA Implementation of MD5 Hash Algorithm, J. Deepakumara, H. M. Heys and R. Venkatesan, Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2001), Toronto, Ontario, May 2001		May 2001
[Dem02]	IP Packet Delay Variation Metric for IPPM, Internet Draft, draft-ietf-ippm-ipdv-10.txt, C. Demichelis, P. Chimento, August 2002.		August 2002
[Duf00]	Trajectory Sampling for Direct Traffic Observation, N. G. Duffield, M. Grossglauser, ACM SIGCOMM'00, October 2000.		October 2000
[Duf02]	Trajectory Engine, A Backend for Trajectory Smapling, N. G. Duffield, M. Grossglauser, IEEE NOMS'02, April 2002.		April 2002
[Du02]	A Framework for Passive Packet Measurement, draft-ietf-psamp-framework-00, Nick Duffield, September 2002.		September 2002
[E430]	ITU E.430, Quality of service framework, ITU-T, June 1992.		June 1992
[E800]	ITU E.800, Terms and definitions related to quality of service and network performance, ITU-T, August 1994.		August 1994
[E880]	ITU E.880, Field data collection and evaluation on the performance of equipment, networks and services, ITU-T, November 1988.		November 1988
[Ether]	The ethereal network analyzer, available at http://www.ethereal.com/, 2002.		2002
[GB908]	Network Management Detailed Operations Map (GB908), Telemanagement Forum, march 1999		March 1999
[GB910]	Telecoms Operations Map, evaluation version 2.0 (GB910), Telemanagement Forum, November 1999		November 1999
[GPS99]	GPS Time Transfer Performance, United States Naval Observatory (USNO), Automated Data Service, available at ftp://tycho.usno.navy.mil/pub/gps/gpstt.txt, June 1999.		June 1999
[Gu00]	Dynamic Algorithms with Worst Case Performance for Packet Classification, P. Gupta, N McKeown, in proc. of the IFIP NETWORKING 2000 Conference, May 2000.		May 2000
[HOWTO_COS]	"VTHD —Utilisation de la différentiation de services dans les applications", ENST Bretagne, 2001		June 2001
[IFT01]	IP Fast Translator (IFT) —TEQUILA white paper, Tequila consortium, Available at http://www.ist-tequila.org/white-papers/ift.pdf, 2001.		2001

[Inter-1]	INTERMON-IST-2001-34123, INTERMON Deliverable 1, “Analysis and Requirements Report”, IST Intermon project, June 2001.		June 2001
[Iper]	Iperf Version 1.6.3, The TCP/UDP Bandwidth Measurement tool , NLANR, available at: http://dast.nlanr.net/Projects/Iperf/, Oct. 2002.		Oct. 2002
[ITU-M3208.1]	ITU-T Recommendation M.3208.1, http://193.248.95.128:63000/dri/doc/uitten/m3208_1e.pdf		
[ITU-M3320]	ITU-T Recommendation M.3208.1, http://193.248.95.128:63000/dri/doc/uitten/m3320e.pdf		
[Jor00]	QoS/GOS parameter definitions and measurement in IP/ATM networks, Jorma Jormakka Kari Heikkinen, First COST 263 International Workshop, QofIS 2000, Berlin, Germany, September 25-26, 2000.		September 25-26, 2000
[Jun02]	JunOS 5.3 Software documentation, Juniper networks, http://www.juniper.net/techpubs/software/junos53/, 2002.		2002
[Kla93]	Application of Sampling Methodologies to Network Traffic Characterization, K. Klaffy, G. Polyzos, H. Braun, ACM SIGCOMM’93, September 1993.		September 1993
[Lak98]	High-Speed Policy-based Packet Forwarding Using Efficient Multi-Dimensional Range Matching, T.V. Laksham, D. Stiliadis, in proc. of ACM SIGCOMM'98, September 1998.		September 1998
[Lexpl]	LanExplorer Protocol and Internet Traffic Analyser, Analyzer Sales Ltd., available at http://www.lan-explorer.co.uk/, 2002.		2002
[Link]	Linkview, Acterna LLC. available at http://www.acterna.com/global/products/descriptions/LinkView/index.html, 2002.		2002
[Mey02-0[	Reliable Streaming Internet Protocol Detail Records, draft-meyer-ipdr-streaming-00, J Meyer, August 2002.		August 2002
[Mey02-1[	Evaluation Of Streaming IPDR Against IPFIX Requirements, draft-meyer-ipfix-ipdr-eval-00, J. Meyer, September 2002.		September 2002
[Mil97]	A Precision Radio Clock for WWV Transmissions, David Mills, Technical Report 97-8-1, Electrical Engineering Department University of Delaware, August 1997.		August 1997
[Mor02]	Reordering Metric for IPPM, Internet Draft, draft-ietf-ippm-reordering-00.txt, A.Morton L.Ciavattone, G.Ramachandran, S.Shalunov, J.Perser, 2002.		2002
[MRTG]	The Multi Router Traffic Grapher, MRTG, available at: http://people.ee.ethz.ch/~oetiker/webtools/mrtg/, 2002.		2002
[NDM-U]	Network Data Management —Usage (NDM-U) for		October 200

	IP-based Services version 2.0, IPDR organization, October 2000, http://www.ipdr.org		
[Netflow]	Netflow, Cisco Systems Inc. available at http://www.cisco.com/warp/public/732/Tech/nmp/netflow/index.shtml, 2002.		2002
[Netperf]	Netperf, available at : http://www.netperf.org/, 2002.		2002
[Netram]	NeTraMet - a Network Traffic Flow Measurement Tool, CAIDA, available at http://www.caida.org/tools/measurement/netramet/, 2002.		2002
[NEW_DS_TERM]	"New terminology for DiffServ", draft-ietf-diffserv-new-terms-05.txt, August 2001		August 2001
[Nor02]	Architecture Model for IP Flow Information Export, draft-ietf-ipfix-architecture-02.txt, K. Norseth, Ganesh Sadasivan, June 2002.		June 2002
[Ntop00]	Ntop a Network Top, an overview, João Paulo Almeida and Yohannes Albertino Ramlie, Technical report, University of Twente, available at http://www.ntop.org/ntop-overview.pdf, 2000.		2000
[P806-2]	EURESCOM P806 —EqoS A Common Framework for QoS/Network Performance in a multi-Provider Environment , Eurescom P806 project, September 1999.		September 1999
[P1008-1]	Project P1008 Inter-operator interfaces for ensuring end-to-end IP QoS, Deliverable 1, State of the art of IP Inter-domain management and supporting measurements, Eurescom P1008, 2001.		2001
[Pathc]	PathChar, Van Jacobson, available at: http://www.caida.org/tools/utilities/others/pathchar/, 1997.		1997
[Pchar]	pchar: A Tool for Measuring Internet Path Characteristics, Bruce A. Mah, available at: http://www.employees.org/~bmah/Software/pchar/, 2001.		2001
[Ping]	The story of the ping program, Mike Muuss, available at: http://ftp.arl.mil/~mike/ping.html, 1999.		1999
[POLTERM]	Policy terminology, A. Westerinen, J. Strassner, Shai Herzog, …, http://search.ietf.org/internet-drafts/draft-ietf-policy-terminology-04.txt		2002
[Prae]	Præcis Ct - Computer Time Source, EndRun Technologies, available at: http://www.endruntechnologies.com/network-time-source.htm, 2002.		2002
[QosMetrix]	QosMetrix (former Imedia-technologies) , available at : http://www.qosmetrix.com/.		2002
[RFC2030]	Simple Network Time Protocol (SNTP) Version 4		October 1996

	for IPv4, IPv6 and OSI, RFC 2030, David L. Mills, October 1996.		
[RFC2124]	Cabletron's Light-weight Flow Admission Protocol Specification, Version 1.0, RFC 2124, P. Amsden, J. Amweg, P. Calato, S. Bensley, G. Lyons, March 1997.		March 1997
[RFC2330]	Framework for IP performance Metrics, RFC 2330, V. Paxson, G. Almes, J. Mahdavi, M. Mathis, May 1998.		May 1998
[RFC2475]	"An Architecture for Differentiated Services", RFC2475, December 1998		
[RFC2678]	IPPM Metrics for Measuring Connectivity, RFC 2678, J. Mahdavi, V. Paxson, September 1999.		September 1999
[RFC2679]	A One-way Delay Metric for IPPM, RFC 2679, G. Almes, S. Kalidindi, M. Zekauskas September 1999.		September 1999
[RFC2680]	A One-way Packet Loss Metric for IPPM, RFC 2680, G. Almes, S. Kalidindi, M. Zekauskas September 1999.		September 1999
[RFC2681]	A Round-trip Delay Metric for IPPM, RFC 2681, G. Almes, S. Kalidindi, M. Zekauskas September 1999.		September 1999
[RFC2720]	Traffic Flow Measurement: Meter MIB, RFC 2720, N. Brownlee, October 1999.		October 1999
[RFC2721]	RTFM: Applicability Statement, RFC 2721, N. Brownlee, October 1999.		October 1999
[RFC2722]	Traffic Flow Measurement: Architecture, RFC 2722, N. Brownlee, C. Mills, C. Ruth, October 1999.		October 1999
[RFC2723]	SRL: A Language for Describing Traffic Flows and Specifying Actions for Flow Groups, RFC 2723. N. Brownlee, October 1999.		October 1999
[RFC2724]	RTFM: New Attributes for Traffic Flow Measurement. RFC 2724, S. Handelman, S. Stibler, N. Brownlee, C. Ruth, October 1999.		October 1999
[RFC2768]	"Network Policy and Services - A Report of a Workshop on Middleware", RFC2768, many contributing authors including J. Strassner and B. Carpenter, February 2000, http://www.ietf.org/rfc/rfc2768.txt		February 2000
[RFC2819]	Remote Network Monitoring Management Information Base, RFC 2819, S. Waldbusser, May 2000.		May 2000
[RFC3060]	"Policy Core Information Model", RFC3060, February 2001, http://www.ietf.org/rfc/rfc3060.txt		February 2001
[RFC3148]	A Framework for Defining Empirical Bulk Transfer Capacity Metrics, RFC 3148, M. Mathis, M. Allman, July 2001.		July 2001
[RFC3176]	InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed		September 2001

	Networks, RFC 3176, P. Phaal, S. Panchen, N. McKee, September 2001.		
[RFC3357]	One-way Loss Pattern Sample Metrics, RFC 3357, R. Koodli, R. Ravikanth, August 2002.		August 2002
[RTTMIB]	RTTMON-MIB, Cisco Systems Inc., available at ftp://ftpeng.cisco.com/pub/mibs/v2/CISCO-RTTMON-MIB-120_5_T.my, 1999.		1999
[SAA]	http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t5/saaoper.htm		2002
[Sib02]	Sibercore products, available at http://www.sibercore.com/products.htm, 2002.		2002
[Sflow]	Sflow, available at www.sflow.org, 2002.		2002
[Shi02]	SeungJae Shin and Martin B.H. Weiss, A Simulation Approach for Internet QoS Market Analysis, Technical Report, MIT, 2002.		2002
[Smart]	Smartflow, Spirent Communications Inc., available at http://broadband.spirentcom.com/solutions/products/applications/pdf/SmartFlow/, 2002.		2002
[Sniff]	Sniffer Pro, Analyzer Sales Ltd, available at : http://www.snifferpro.co.uk/, 2002		2002
[SP-DNA]	Service Provider Directory-enabled Network Applications, Open Group		2002
[Sri99]	Packet Classification Using Tuple Space Search, V. Srinivasan, S. Suri, G. Varghese, in proc. Of the ACM SIGCOMM'99 Conference, September 1999.		September 1999
[Ste94]	TCP-IP illustrated, volume 1, The Protocols, W. Richard Stevens, Addison Wesley, 1994.		1994
[Ste02]	IPPM Reporting MIB, draft-ietf-ippm-reporting-mib-00.txt, E. Stephan, J. Jewitt, June 2002.		June 2002
[Ste02]	IPPM Reporting MIB, draft-ietf-ippm-reporting-mib-00.txt, E. Stephan, J. Jewitt.		June 2002
[Ste03]	IPPM spatial metrics measurement, draft-stephan-ippm-spatial-metrics-00.txt, E. Stephan		Sept 2002
[Stin99]	Sting: a TCP-based Network Measurement Tool, Stefan Savage, Proceedings of the 1999 USENIX Symposium on Internet Technologies and Systems, pp. 71-79, Boulder, CO, October 1999.		October 1999
[TEQUILA]	"D1.1: Functional Architecture Definition and Top Level Design", IST Project number "IST-1999-11253-TEQUILA", September 2000		September 2000
[TTCP]	The Story of the TTCP Program, Mike Muuss, available at: http://ftp.arl.mil/~mike/ttcp.html, 1999.		1999
[X140]	ITU X.140, General quality of service parameters for communication via public data networks. ITU-T, September 1992.		September 1992
[Wal02]	The RMON Framework, draft-ietf-rmonmib-framework-01.txt, Steve Waldbusser, R.G. Cole,		August 2002

	C. Kalbfleisch, D. Romascanu, August 2002.		
[Zha02]	XACCT's Common Reliable Accounting for Network Element (CRANE) Protocol Specification Version 1.0, draft-kzhang-crane-protocol-05.txt, Kevin Zhang, Eitan Elkin, August 2002.		August 2002
[Zse01]	Evaluation of Building Blocks for Passive One-way-delay Measurements, Proceedings of Passive and Active Measurement Workshop (PAM 2001), Amsterdam, The Netherlands, April 23-24, 2001		April 23-24, 2001
[Zse02]	Sampling and Filtering Techniques for IP Packet Selection, T. Zseby, M Molina, F. Raspall, Internet Draft, draft-ietf-psamp-sample-tech-00.txt, April 2002.		April 2002

Figure 10-1: References