Information Society Technologies

IPv6 QoS M

IPv6 Quality of Service Measurement

Title: **Deliverable D3.3** **Revised Specification**	**Document Version:** 1.3

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

Contractual Delivery Date: 31/12/2004	**Actual Delivery Date:** 05/12/2004	**Deliverable Type* - Security**:** 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: David Diep	**Organization:** HIT	**Contributing WP:** WP3

Authors (organizations):

Lutz Mark (FOKUS), Guido Pohl (FOKUS), Alessandro Bassi (HEL), Elisa Boschi (HEL), Jordi Palet (Consulintel).

Abstract:

This document presents the revised specifications of 6QM which defines additional features to the system defined in the deliverable D3.1

Keywords:

Revised specification, inter-domain, security, scalability

Revision History

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

Revision	Date	Description	Author (Organization)
v0.1	05/06/2004	Document creation	David Diep (HIT)
v0.2	01/11/2004	Added content	Guido Pohl (FOKUS)
v0.3	09/11/2004	Edited doc. to include partners contributions	Guido Pohl (FOKUS)
v0.4	10/11/2004	Content update	David Diep (HIT)
v0.5	15/11/2004	Added content	Guido Pohl (FOKUS)
v0.6	17/11/2004	Reviewed and updated chapter 2, updated references	Elisa Boschi (HEL)
v0.7	17/11/2004	Updated Requirement and Specification Analysis	Guido Pohl (FOKUS)
v1.0	24/11/2004	Validate changes	David Diep (HIT)
v1.1	24/11/2004	Content update	Guido Pohl (FOKUS), David Diep (HIT)
v1.2	26/11/2004	Update of section 7	David Diep (HIT)
v1.3	05/12/2004	Final review	Jordi Palet (Consulintel)

Executive Summary

During the 6QM project many aspects of the IPv6 QoS measurement have been addressed. This document proposes a revision of the initial specifications presented in the deliverable D3.1 “IPv6 QoS Measurement Specification”. The purpose of this document is to present some areas that could be improved or extended and specifies the means or techniques necessary to perform such an improvement. The investigated topics are presented hereafter.

During the revised specification activity, important efforts have been put to enhance the inter-domain design that was introduced in D3.1. The design details the interface for setup and the data export format between inter-domain agents.

This document also makes proposals on how to reach security for the 6QM measurement system. The addressed topics are the secured communication between system components, and the restriction for accessing system components and user management.

An important effort concerned the improvement of the system scalability. This examination includes techniques to increase performance for the network probes with the usage of sampling; but it also contains consideration on how to increase overall system performance by establishing extended levels of component hierarchies and a decentralized architecture. Finally a paragraph inspecting methods for measurement result storage concludes the argumentation about scalability.

The document at hand follows the announcement given in the previous deliverable D3.1 to look for additional metrics of interest. A section presents metrics that we selected for further analysis including the metric for evaluating the available bandwidth.

In a separate section, this document also provides some information about some prototype enhancements that have been performed after the original development time.

Additional considerations that establish similarities and dissimilarities between existing tools that are listed by the MOME coordination action of IST and the measurement platform developed within the project frame of 6QM are given in a chapter of this document.

The document concludes that there is still a lot of space for further improvement in design and implementation as introduced in the deliverable D3.4 addressing the guidelines for further research.

Table of Contents

1. Introduction........ 8

2. Inter-domain 9

2.1 Motivation 9

2.2 Monitoring configuration 10

2.2.1 Specification of Monitoring Service 10
2.2.2 Configuration process 13
2.2.3 Holes on the path 14
2.2.4 Nested SMSs........ 15
2.2.5 SMS Implementation 16

2.3 Data export........ 18

2.3.1 IPFIX........ 19
2.3.2 IDFIX 20
2.3.3 Inter-domain export with IPFIX 21

2.4 Scenario........ 21

2.5 Monitoring configuration process step-by-step 23

2.5.1 From the GUI to the SMS 23
2.5.2 SMS splitting 24
2.5.3 Intra-domain configuration........ 26

2.6 Security considerations........ 28

2.6.1 Public key, private key approach........ 28
2.6.2 AAA 28

3. Security........ 30

3.1 IPsec Secured channel 30

3.1.1 IPsec Overview........ 30
3.1.2 Selected Configuration........ 31
3.1.3 Relation to Implementation........ 31
3.1.4 IPsec Conclusion........ 32

3.2 Web server access security 33

3.3 User management, privileges, access control........ 34

4. Scalability........ 38

4.1 Packet Sampling 38

4.1.1 One-way delay and Sampling........ 38
4.1.2 Content-based selection of packets for sampling 41
4.1.3 Two-level Sampling........ 41

4.2 Decentralized Architecture........ 51

4.3 Data Storage Policy........ 53

5. Additional Metrics........ 55

5.1 Available Bandwidth Estimation........ 55

5.2 Top N One-Way Delays 56

6. Prototype Extension 57
6.1 Integration of Inter-domain Measurement Capabilities 57
6.2 Integration of PathChirp 60
6.3 Security 61
7. Requirement and Specification Analysis WP2/WP3 62
7.1 Requirements for Measurement Points 63
7.2 General Requirements for Measurements 63
7.3 Requirements for Passive Measurements 64
7.4 Requirements for 6QM Measurement Manager 67
7.5 Requirements for 6QM Evaluator and Collector 68
7.6 Requirements for Inter-domain measurements 70
7.7 Requirements for Fault Management 71
7.8 Requirements for Security Management 72
7.9 Security requirement (D2.5) 73
7.10 Conclusion of Requirement Analysis 73
8. 6QM System and the state-of-art 77
9. Summary and Conclusions 80
10. References 81

Table of Figures

Figure 2-1: SMS format ... 11
Figure 2-2: Monitoring configuration process ... 13
Figure 2-3: Monitoring configuration with "holes" on the path ... 15
Figure 2-4: Nested SMSs ... 16
Figure 2-5: IPFIX and IDFIX ... 19
Figure 2-6: IPFIX message format ... 20
Figure 2-7: IDFIX header ... 20
Figure 2-8: IPFIX format for inter-domain export ... 21
Figure 2-9: One-way delay measurement scenario ... 22
Figure 2-10: SMS (left) vs. 6QM meter configuration ... 23
Figure 2-11: SMS splitting ... 24
Figure 2-12: Flow path from T1 to T2 ... 25
Figure 2-13: Problem of probe selection ... 27
Figure 2-14: Inter-domain communication with AAA ... 29
Figure 3-1: IPsec Configuration ... 31
Figure 3-2: IPsec Configuration file ... 32
Figure 3-3: IPsec versus SSH ... 33
Figure 3-4: User roles and privileges ... 35
Figure 3-5: ORM diagram with sample population ... 36
Figure 4-1: PSAMP in 6QM Architecture ... 39
Figure 4-2: Correlation Problem with Sampling ... 40
Figure 4-3: Two level Sampling ... 42
Figure 4-4: Asymmetric Sampling ... 43
Figure 4-5: Sampling in Interdomain ... 44
Figure 4-6: Two-level Sampling Scenarios ... 46
Figure 4-7: Two level Sampling simulation 1 ... 47
Figure 4-8: Two level sampling simulation result 1 ... 48
Figure 4-9: Two Level Sampling Simulation 2 ... 49
Figure 4-10: Two-level Sampling Extension ... 50
Figure 4-11: Automatic selection of sampling ... 50
Figure 4-12: Distributed System for Large Network ... 52
Figure 4-13: Design for Scalable Data Collection ... 53
Figure 4-14: Data Value ... 54
Figure 5-1: Top one-way delays ... 56
Figure 6-1: 6QM Functional architecture and inter-domain components (controller) ... 57
Figure 6-2: 6QM Functional architecture and inter-domain components (collector) ... 58
Figure 6-3: Basic structure of the intra-domain controller ... 58
Figure 6-4: Basic structure of the collector ... 59
Figure 6-5: Result of Available Bandwidth Estimation ... 60
Figure 6-6: Secured system components ... 61
Figure 7-1: Requirements for Measurement Points ... 63
Figure 7-2: General Requirements for Measurements ... 63
Figure 7-3: Requirement Analysis for Passive Measurements ... 67
Figure 7-4: Requirement Analysis for 6QM Measurement Manager ... 68
Figure 7-5: Requirements for 6QM Evaluator and Collector ... 69
Figure 7-6: Requirements for Inter-domain measurements ... 71
Figure 7-7: Requirements for Fault Management ... 71
Figure 7-8: Requirements for Security Management ... 72
Figure 7-9: Security Analysis Requirement (D2.5) ... 73
Figure 7-10: Summary Requirements for Measurement Points ... 73
Figure 7-11: Summary Requirements for Measurements ... 73
Figure 7-12: Summary for Passive Measurements ... 74
Figure 7-13: Summary Requirements for Measurement Manager ... 74
Figure 7-14: Summary Requirements Evaluator and Collector ... 74
Figure 7-15: Summary Requirement for Inter-domain Measurements ... 74
Figure 7-16: Summary Requirements Fault Management ... 74
Figure 7-17: Summary Requirements for Security Management ... 75
Figure 7-18: Summary Requirements for Security Analysis (D2.5) ... 75

Figure 7-19: Prototype Compliance Summary 76
Figure 8-1: MOME tools including one-way delay measurement 78

1. INTRODUCTION

The WP3 generated four deliverables:

- The deliverable D3.1 specifies the technologies to be used.
- The deliverable D3.2 documents the prototype measurement system.
- The deliverable D3.3 provides a revised specification of the measurement system.
- The deliverable D3.4 provides guidelines for applications and further research.

During the 6QM project many aspects of the IPv6 QoS measurement have been addressed. This document proposes a revision of the initial specifications presented in the deliverable D3.1 “IPv6 QoS Measurement Specification”. The purpose of this document is to present some areas that could be improved or extended and specifies the means or techniques necessary to perform such an improvement. Concerning the revised specification, the investigated topics are:

- The inter-domain refinement.
- The security improvement.
- The scalability improvement.
- The additional metrics proposal.

In addition to the revised specification, the present document provides the overview of the prototyping modifications that have been performed during the extension time.

Moreover a section will also provide a comparison between 6QM system and other existing systems that were identified by the MOME coordination action of IST.

2. INTER-DOMAIN

The purpose of this section is to give an overview of the proposal on how to support for inter-domain measurement within the frame of 6QM.We designed a format for communication between different domains to set up measurement and retrieve results. These formats, the protocol, and the framework architecture are described in the following pages. Notes on the implementation and integration with the existing 6QM architecture are also included under 6.1 "Integration of Inter-domain Measurement Capabilities".

2.1 Motivation

When talking about flow monitoring from the users point of view, it would be interesting to have detailed information on end-to-end (E2E) performance. The problem is that the information a single ISP has would not be enough. While single ISPs usually monitor their networks, e.g. to obtain information on resource usage, traffic accounting, fault diagnosis and troubleshooting, and have all the necessary information about it, they need to request data to other ISPs as soon as the flow crosses other domains. To obtain accurate E2E information, it becomes necessary for ISPs to cooperate and make their metrics accessible to others.

A typical application would be measuring metrics such as one-way delay. While measuring one-way delay is possible with just meters at the two endpoints, it does not give details on how and where the biggest delay was. A cooperating inter-domain system can tell where the networks delays are occurring.

Currently this is hard to do because:

- Routers often make bad measurement targets because they protect themselves against DOS attacks by processing ICMP at a low priority.
- There are often large variations in delay between successive packets following the same route.
- The delays on the reverse path are likely to be different to those on the forward path.
- The path may change during a measurement.

ISPs are often compared unfairly because large parts of the delay are outside their control. Internet surveys take too little account of the service actually provided by the ISPs they measure. Partly this is because it is difficult. An ISP that wants to be able to discover and demonstrate the degree of delay introduced by its network can join the inter-domain framework/community.

Furthermore the system:

- Allows more accurate measurement.
- Supports measurement.
- Allows comparisons.
- Reduces the measurement overhead on the network by composing E2E results out of intra-domain results (often computed anyway).

2.2 Monitoring configuration

The process of configuring and End-to-End (E2E) measurement across several domains is carried out by a component called simply inter-domain controller. In this text we use interchangeably the words inter-domain controller and controller.

There is at least one controller in each domain. Due to scalability and to avoid the "single point of failure" problem, it is advisable to add some redundancy to the number of controllers in each domain. This can be done by using a DNS-like solution: one primary controller and a secondary one mirroring the primary and ready to becoming primary controller itself in case of failure. An even better solution, because more scalable, would be specifying controller names instead of addresses and having the DNS assigning different addresses associated to the same name (a bit like the browser redirecting www.google.com to www.google.de if the request comes form Germany, and to www.google.fr if it comes from France…).

The controller architecture is described in more detail in section 6.1.

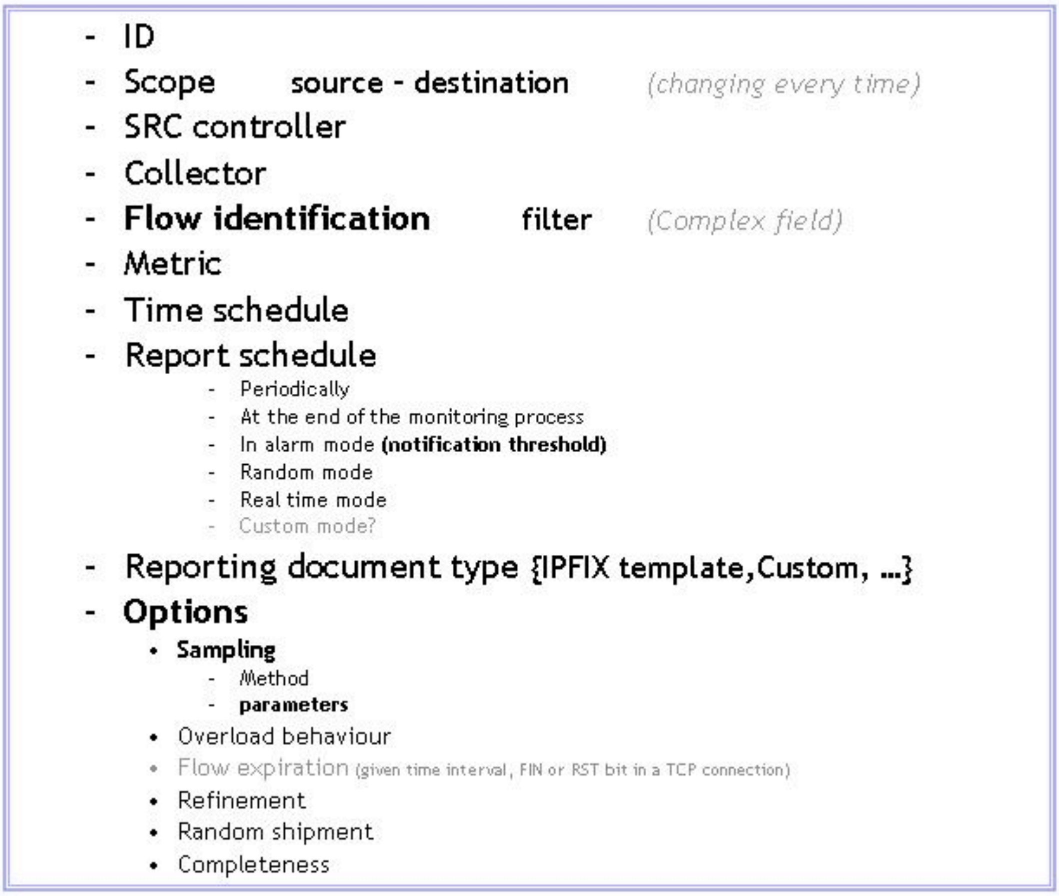
It must be said that in order for the system to work, different ISPs need to agree to cooperate with each other, authorize other ISP to configure measurements within their domain and specify clearly who is allowed to do what. We assume that this agreement has been made (being it a fundamental requirement for the system to work). How the agreements are stipulated is outside the scope of this document.

The configuration process uses a document called Specification of Monitoring Service (SMS) that will be described in detail in the following section. The configuration process itself is described in section 2.2.2. Security considerations are left for section 2.6.

2.2.1 Specification of Monitoring Service

The Specification of Monitoring Service (SMS) is a standard format used to specify monitoring configuration across domains. Given the heterogeneity of a multi-domain environment, it is necessary to provide a format that is general enough to be easily understandable in every domain.

The basic structure of an SMS is depicted in Figure 2-1.



- ID
- Scope source - destination (changing every time)
- SRC controller
- Collector
- Flow identification filter (Complex field)
- Metric
- Time schedule
- Report schedule
  - Periodically
  - At the end of the monitoring process
  - In alarm mode **(notification threshold)**
  - Random mode
  - Real time mode
  - Custom mode?
- Reporting document type {IPFIX template,Custom, ...}
- **Options**
  - **Sampling**
    - Method
    - **parameters**
  - Overload behaviour
  - Flow expiration (given time interval, FIN or RST bit in a TCP connection)
  - Refinement
  - Random shipment
  - Completeness

Figure 2-1: SMS format

In more detail, the different fields have the following meaning:

- SMS ID: a value that uniquely identifies each SMS. Being the value unique for all the domains, a part of this value must be the AS number of the domain generating the identifier.
- Scope: this complex field usually contains the following values:
  - Source: contains the IP address of the ingress point of the traffic flow.
  - Destination: contains the IP address of the egress point of the traffic flow.
  - The *scope* field could also contain one or more pointers to other SMSs. This particular option is described in detail in section 2.2.4 “Nested SMSs”.
- Source controller: contains the IP address of the controller of the domain from which the SMS has been forwarded. When a controller sends the document to the next domain on the path will put his own address here.
- Collector: contains the address of the collector to which the results must be sent. There might be many collectors in a single domain, when the controller forwards an SMS decides where the data must be sent back (so the controller is responsible for resource allocation).
- Flow identification: indicates which traffic flow has to be monitored; it provides rules to select the packets belonging to the traffic flow. The parameters correspond to those selected in D3.1 and are:
  - IPv4 or IPv6 source address.
  - IPv4 or IPv6 destination address.

    - IPv4 ToS field content/ IPv6 traffic class.
    - Flow label field content.
    - IPv4 protocol field content/ IPv6 Next header field content.
    - Port number.
- Metric: specifies which parameter needs to be measured (e.g. throughput, packet loss, one-way delay, jitter…).
- Time schedule: specifies start time and duration of the monitoring task.
- Report schedule: specifies when the results must be sent back. The value contained in this field may be one of the pre-defined options or a custom mode, defined by the customer (the definition, or a pointer to it, must then be carried with the SMS). The pre-defined modes are:
    - Periodically. The time interval must be then specified here.
    - At the end of the monitoring process. The requested data are collected during the monitoring process and sent back just once the whole process is completed.
    - In alarm mode. In this case, a Notification threshold is specified. Data are sent just when the monitored parameters reach the value specified in the field. In this case the result contains just those values exceeding the specified threshold.
    - Random mode. The results will be sent at random intervals.
    - Real time mode. This mode specifies whether the results must be received in real time. If the option is set, data must be sent as soon as collected.
- Reporting document type: defines the way the monitored data should be sent back. This field should contain a reference to the IPFIX/IDFIX template type to be used for data export. It is anyway left the possibility to define custom reporting documents. The type of document, or a reference to its description, should be specified here.
- Options: this field allows specifying one or more optional parameters related to the monitoring configuration. Again, custom options may be specified, and their description (or a reference to it) must be indicated here. The pre-defined options are:
    - Granularity: with this field it can be specified which domains should be configured. The options are:
        - All (completeness): a measurement is started just if the requested data can be collected in every ISP crossed to the final destination. The SMS is sent to all the ISPs on the path and all configure the monitoring locally. In case of incomplete configuration, an error is returned.
        - Best effort: the SMS is sent from one domain to the next domain on the path that is part of the agreement. In this case, there might be "holes" along the path, domains from which we don't get any measurements but it might still be better than no values at all.
        - Endpoints: the SMS is sent just to another domain (typically the flow destination domain).
    - Refinement: it must be defined which level of refinement is allowed in the results. Several refinement techniques exist and it is up to the client to specify one of them through this option. Refinement techniques are:
        - Rank-based: the top Nth results with respect to some QoS parameters are identified and reported.
        - Percentile-based: the Nth percentile of a metric is identified and reported. For example, the user may want to know the 90th percentile of delay.
        - Custom.

- Sampling: this composed field is necessary if sampling is allowed and indicates.
  - The sampling algorithm to be used.
  - Parameters necessary for that algorithm.
- Random shipment: one ISP may wish to give no hints at all to any eavesdropper that may intercept the traffic on the line. The size of data shipments may be therefore varied, by choosing the random shipment mode. The size of the packet can be varied, the same amount of data can be sent in packets with different size (or in bursts with different sizes);
- Overload behavior: it is possible to specify here what meters should do in case of overload to have a uniform reaction. Between the possibilities are stop monitoring, reduce sampling rate, etc.
- Flow expiration: it is possible to specify here to all meters when a flow should be considered expired by giving the timeout interval, indicating to wait for TCP FIN or RST bit…)

2.2.2 Configuration process

The configuration process uses a cascade model where a multi-domain SMS can be recursively split into two parts: a single domain SMS plus a remaining part related to elsewhere, i.e. one or more downstream domains. In other terms, after receiving the SMS, the generic controller selects the subset of information related to its own AS and translates it in configuration parameters to be sent to the appropriate meters within the same domain. The rest of the SMS document is sent to the peer controller en-route towards the destination.

Figure 2-2 describes the process. The controllers are indicated with the letter C.

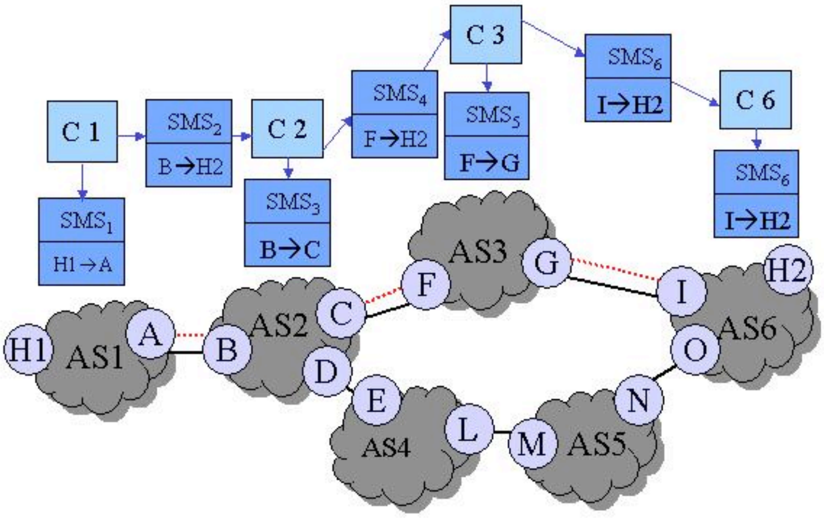


Figure 2-2: Monitoring configuration process

In the example presented in Figure 2-2, a flow going from H1 in the Autonomous System AS1 and ending in H2 in AS6 needs to be monitored. C1, the controller located in AS1, receives the monitoring request. It splits the SMS in two other documents: SMS1 that refers to the AS managed by C1 itself (the path between H1 and A), and SMS2 that is sent to the next controller on the way to H2 (C2) and refers to the remaining path to the destination (from B to H2). Each controller generates two documents out of the one obtained as input: the first one containing configuration information related to its own domain and the other to be forwarded to the next controller on the path. Only exception, the final AS whose controller (C6) configures the measurement for the last part of the path, i.e. from I to H2. Each controller will then translate the SMS entirely related to its domain in the appropriate meter configuration language (which might be domain specific).

2.2.3 Holes on the path

The systematic process described in 2.2.2 works fine as far as each domain on the path has joined the data exchange agreement. In a more realistic case, some of the domains might be out of this “consortium”, and be like “holes” in the configuration. The action to be taken in case of holes is specified in the SMS in the field *granularity.*

If the value specified is “all”, and one domain crossed by the flow is not part of the agreement, the configuration process terminates with an error. In the example shown in Figure 2-3, the domain 1 controller observes that AS2 (the next step crossed by the flow on its way to H2) is not part of the agreement. It then stops the configuration process and returns an error. The user can eventually request another measurement configuration with a lower level of granularity.

If the value specified is “best effort”, all crossed domains that are part of the agreement are configured and holes are bypassed. In the example in Figure 2-3 the controller 1 forwards the SMS to AS3; AS1, AS3 and AS6 are configured and AS2 is crossed by the flow without exporting information about it. Sometimes incomplete information is still better than no information at all.

When the value contained in the field granularity is set to “endpoint” the SMS is sent directly to the final domain (indicated in the field scope) without going through the splitting process. Only the directly addressed domains will then configure the measurement (typically the flow endpoints, e.g. AS1 and AS6). This option is particularly interesting if used in combination with nested SMSs (cf. 2.2.4).

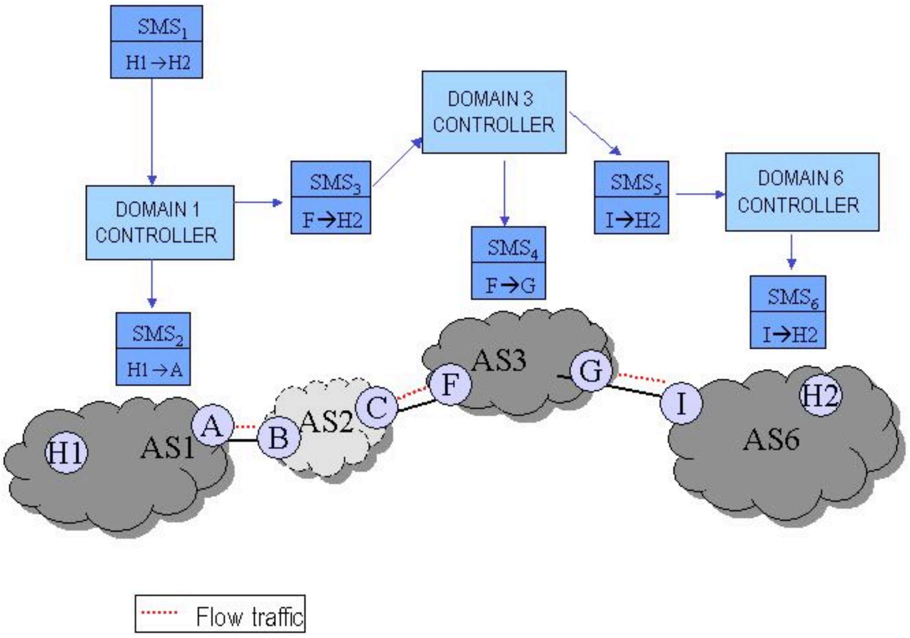


Figure 2-3: Monitoring configuration with "holes" on the path

2.2.4 Nested SMSs

One SMS can refer to other SMSs to obtain more flexible (and complex) measurement configurations. Different SMSs can form a tree by pointing at each other using their unique SMS IDs. As already mentioned, the pointers to the son SMSs, one or more, are contained in the field *scope*.

Nested SMSs are the solution for a number of applications between which are:

- End-to-end One-Way Delay (OWD) measurements: the OWD between two domains can be computed by using a two-level SMS (cf. Figure 2-4). The father SMS is sent to the controller of the "home" domain (in other words the one directly attached to the GUI from which the request arrives) and contains OWD as metric and two SMS IDs in its scope field. The two child SMSs are sent to the source and destination domains and have to report packet headers to the specified controller in the home domain where the one-way delay is finally calculated.
- Multicast: In this case, different destinations can be specified in the scope field. For each flow destination a child SMS is created.

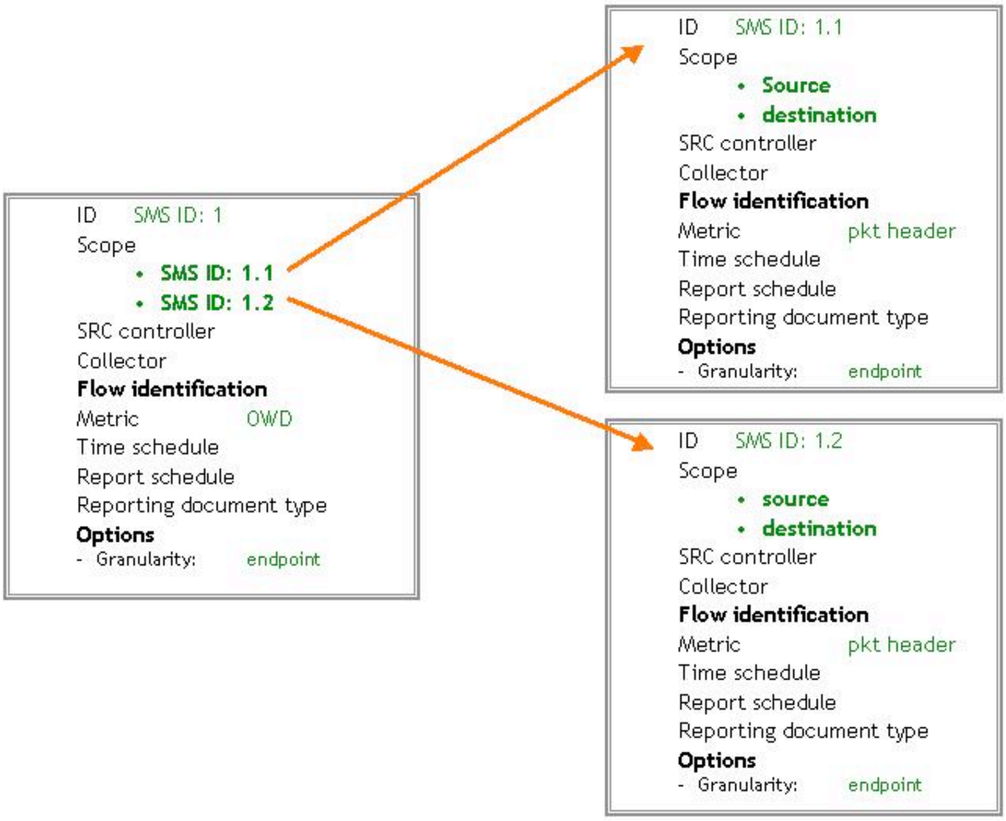


Figure 2-4: Nested SMSs

2.2.5 SMS Implementation

This section contains the SMS XML schema. We briefly explain why we've chosen XML before presenting the document itself.

2.2.5.1 Motivation for XML

The wide availability of XML aware tools and libraries for client devices is a primary consideration for this choice. In particular libraries for parsing XML documents are readily available. Also mechanisms such as the Extensible Stylesheet Language (XSL) allow for transforming a source XML document into other documents.

2.2.5.2 The XML document

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.6qm.org"
xmlns="http://www.6qm.org"
elementFormDefault="qualified">

<xs:simpleType name="IPAddress">
        <xsd:restriction base="xsd:string">
                <xsd:maxLength value="75"/>
                <xsd:pattern value="(([0-9]|[1-9][0-9]|[2][0-4][0-9]|[2][5][0-
5])\.){3}([0-9]|[1-9][0-9]|[2][0-4][0-9]|[2][5][0-5])">
                </xsd:pattern>
                <xsd:pattern value="([A-Fa-f0-9]{1,4}:){7}[A-Fa-f0-9]{1,4}">
                </xsd:pattern>
        </xsd:restriction>
</xsd:simpleType>
```

```
<xs:element name=”ID”>
   <xs:simpleType>
      <xs:restriction base="xsd:string">
         <xs:patternValue=”([A-Za-z0-9]*)” />
       </xs:restriction>
   </xs:simpleType>
</xs:element>


<xs:element name="SMS">
    <xs:complexType>
      <xs:sequence>
         <xs:element name=”ID Number” type="xs:ID"/>
         <xs:element name=”Scope”>
            <xs:simpleType>
              <xs:sequence>
                  <xs:element name=”Ingress” type=”xs:IPaddress” />
                  <xs:element name=”Egress” type=”xs:IPaddress” />
              </xs:sequence>
            </xs:simpleType>
         </xs:element>
         <xs:element name=”SRCController” type=”xs:IPaddress” />
         <xs:element name=”Collector” type=”xs:IPaddress” />
         <xs:element name=”Flow Identification”>
           <xs:simpleType>
             <xs:sequence>
               <xs:element name=”SourceAddress” type=”xs:IPaddress” />
               <xs:element name=”DestinationAddress” type=”xs:IPaddress” />
               <xs:element name=”SourcePort” type=”xs:integer” />
               <xs:element name=”DestinationPort” type=”xs:integer” />
               <xs:element name=”ToSTrafficClass” type=”xs:string” />
               <xs:element name=”FlowLabel” type=”xs:string” />
               <xs:element name=”ProtocolNextHeader” type=”xs:byte” />
             </xs:sequence>
           </xs:simpleType>
         </xs:element>
         <xs:element name=”Metric”>
           <xs:simpleType>
             <xs:restriction base=”xs:string”>
                  <xs:enumeration value="THROUGHPUT" />
                  <xs:enumeration value="PACKETLOSS" />
                  <xs:enumeration value="ONEWAY" />
                  <xs:enumeration value="JITTER" />
             </xs:restriction>
           </xs:simpleType>
         </xs:element>
         <xs:element name=”TimeSchedule”>
           <xs:simpleType>
             <xs:sequence>
               <xs:element name=”StartTime” type=”xs:time” />
               <xs:element name=”Duration” type=”xs:unsignedLong” />
             </xs:sequence>
           </xs:simpleType>
      </xs:element>
      <xs:element name=”ReportSchedule”>
          <xs:simpleType>
             <xs:sequence>
                  <xs:restriction base=”xs:string”>
                    <xs:enumeration value="PERIOD" />
                    <xs:enumeration value="ALARM" />
                    <xs:enumeration value="END" />
                    <xs:enumeration value="RANDOM" />
                    <xs:enumeration value="RTIME" />
                 </xs:restriction>
                  <xs:element name=”TimeInterval” type=”xs:unsignedLong” />
                  <xs:element name=”Threshold” type=”xs:integer” />
              </xs:sequence>
           </xs:simpleType>
      </xs:element>
      <xs:element name=”ReportingDocumentType” type=”xs:string” />
       <xs:element name=”Granularity”>
           <xs:simpleType>
             <xs:restriction base=”xs:string”>
                 <xs:enumeration value="ALL" />
                 <xs:enumeration value="BestEffort" />
                 <xs:enumeration value="EndPoints" />
             </xs:restriction>
           </xs:simpleType>
         </xs:element>
         <xs:element name=”Options”>
           <xs:simpleType>
             <xs:sequence>
                 <xs:element name=”Refinement” type=”xs:unsignedLong” minOccurs=0 >
                   <xs:simpleType>
                     <xs:restriction base=”xs:string”>
                         <xs:enumeration value="Rank-Based" />
                         <xs:enumeration value="Percentile-Based" />
                         <xs:enumeration value="Custom" />
                     </xs:restriction>
                   </xs:simpleType>

                 </xs:element>
                 <xs:element name=”Sampling” type=”xs:string” minOccurs=0 />
                 <xs:element name=”RandomShipment” type=”xs:string” minOccurs=0 />
                 <xs:element name=”Overload” type=”xs:string” minOccurs=0 />
                 <xs:element name=”Flow” type=”xs:string” minOccurs=0 />
              </xs:sequence>
           </xs:simpleType>
         </xs:element>

     </xs:complexType>
</xs:element>


</xs:schema>
```

2.3 Data export

Measured data are collected, composed, and exported by a component referred as collector in this text. There are more collectors in each domain and it is during the configuration phase that the controllers decide where the data have to be reported to. Having the controller an overview of the “load” on each collector, this guarantees load balancing within the domain. The architecture of the collector is described in detail in section 6.1.

Given the heterogeneous nature of a multi-domain environment, there is a strong need for standard formats to carry messages and information across domains. A standard format for data export already exists (even if it is still work in progress); it is the work done in the IP Flow Information eXport (IPFIX) Working Group of the IETF. The IPFIX protocol has been designed to provide a standard format to export measured IP flow data from a Meter to a collector, i.e. it refers to an intra-domain environment. We extended IPFIX to inter-domain, calling the new protocol IDFIX (Inter Domain Flow Information eXport), and adding fields required for inter-domain data export while removing fields that have a local scope. One of the advantages of this approach is the easy interoperability between intra-domain data export (IPFIX) and inter-domain data export (IDFIX). The relation between the two formats is shown in Figure 2-5.

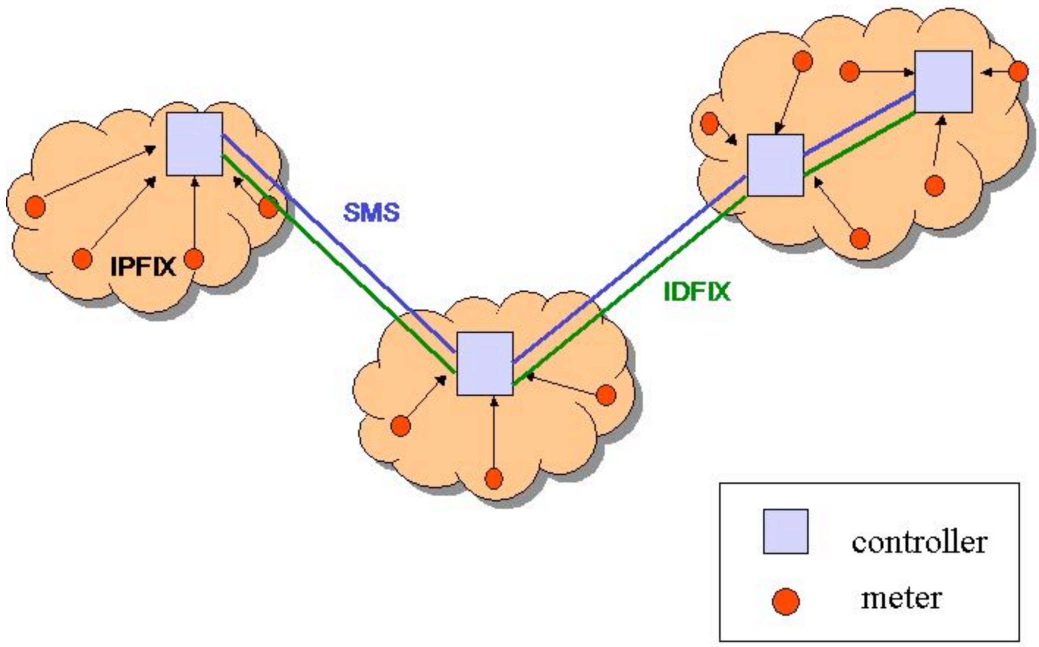


Figure 2-5: IPFIX and IDFIX

The following sections will provide an overview of both protocols and data export formats.

2.3.1 IPFIX

The IPFIX protocol defines how IP Flow information can be exported from routers, measurement probes or other devices. It is intended to provide this information as input for various applications, such as SLA validation, accounting, intrusion detection, network planning. A data network with IP traffic, primarily consists of IP Flows passing through its network elements. It is often interesting, useful or even a requirement to have access to information about these flows (e.g. for administrative purposes). In order to export flow information to a collecting process, a common method of representing the flow data and a standard means of communicating them from an exporter to a collector are required. The architecture for the export of measured IP flow information out of an IPFIX exporting process to a collecting processing is defined in [IPFIX-ARCH], per the requirements defined in [IPFIX-REQ]. [IPFIX-PROTO] specifies how IPFIX flow record data, options record data and control information is carried via a congestion-aware transport protocol from IPFIX exporting process to IPFIX collecting process. IPFIX has a formal description of IPFIX information elements (fields), their name, type and additional semantic information, as specified in [IPFIX-INFO]. Finally [IPFIX-AS] describes what type of applications can use the IPFIX protocol and how they can use the information provided.

Figure 2-6 shows the general IPFIX message format. For more details refer to [IPFIX-PROTO].

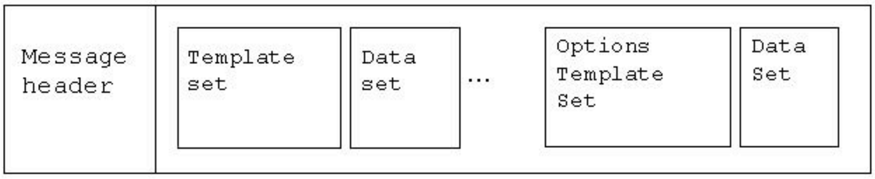
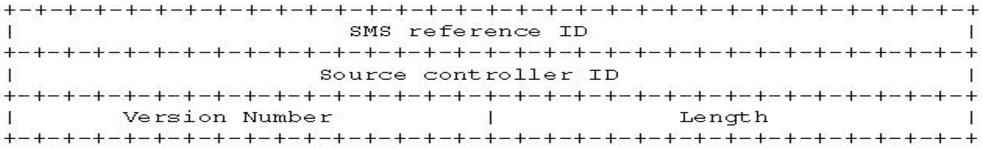


Figure 2-6: IPFIX message format

Although IPFIX has been designed to export IP flows, an extension has been proposed to export per-packet information [PoMB04]. When talking about IPFIX, we refer to both.

2.3.2 IDFIX

IPFIX has been thought of as a protocol to export flow data from a meter to a collector. The scope is therefore local, and the protocol as-is is not suitable for an inter-domain environment. IDFIX (Inter Domain Flow Export) extends IPFIX for this purpose. The extension is anyway limited to a different header and to the parameters that are allowed in the data templates; being the data exchanged across domains, there are of course limitations on the type of data that can be exported.



```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      SMS reference ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Source controller ID                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Version Number          |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2-7: IDFIX header

Figure 2-7 represent the IDFIX header. The fields have the following meaning:

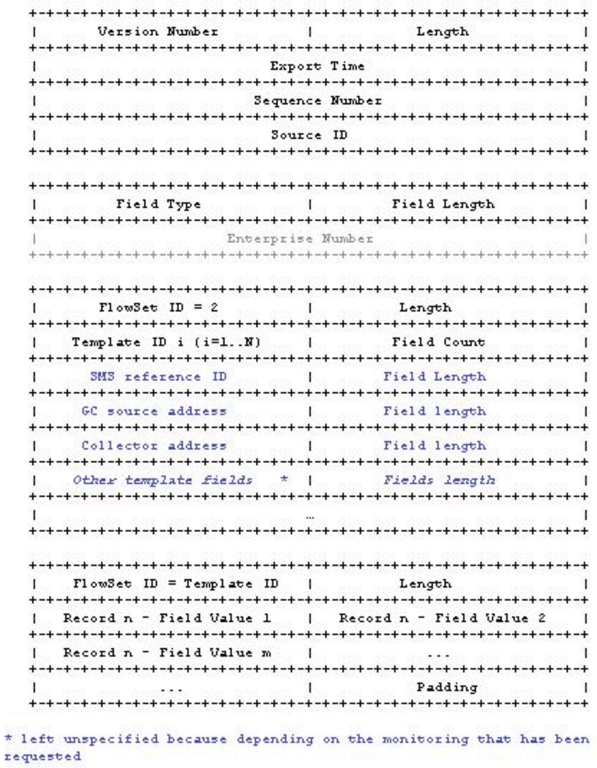
- SMS Reference Number: A value that identifies the SMS that started the measurement whose results are reported here. A reference to the SMS also means a reference to the metering parameters.
- Destination ID: A value that identifies the Controller of the neighboring domain the packet is addressed to.
- Source ID: A value that identifies the Controller Domain. Collecting Process SHOULD use the combination of the source IP address and the Source ID field to separate different export streams originating from the same Exporting Process. This could also simply be the IP address.
- Version: Version of Flow Record format exported in this message. The value of this field is 0x000a for the current version (cf. IPFIX).
- Length: Total Length is the length of the IDFIX message, measured in octets, including message Header and FlowSet(s).

- Export Time: Time in seconds since 0000 UTC 1970, at which the Packet leaves the Controller.
- Sequence Number: Incremental sequence counter of all IDFIX Messages sent from the current Domain by the Controller. This value MUST /SHOULD be used by the Collecting Process to identify whether any IDFIX Messages have been missed.

Note that the IPFIX template has been specified with the SMS.

2.3.3 Inter-domain export with IPFIX

An alternative solution to IDFIX, is defining appropriate templates for IPFIX to define the inter-domain information it is currently missing. While this solution would be less efficient than IDFIX (it introduces a lot of overhead) it might be faster to deploy. The IPFIX template in this case would be like the one shown in the figure below.



```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Version Number         |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Export Time                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Source ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Field Type           |         Field Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Enterprise Number                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        FlowSet ID = 2         |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Template ID i (i=1..N)     |          Field Count          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      SMS reference ID         |         Field Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     GC source address         |         Field length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Collector address         |         Field length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Other template fields    * |         Fields length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               …                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    FlowSet ID = Template ID   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Record n - Field Value 1    |   Record n - Field Value 2    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Record n - Field Value m    |              ...              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ...              |            Padding            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

* left unspecified because depending on the monitoring that has been
requested
```

Figure 2-8: IPFIX format for inter-domain export

2.4 Scenario

The scenario we propose is strongly based on the one-way delay measurement information presented in 6QM deliverable D3.1. The configuration of one-way delay in an inter-domain environment (and of course the visualization of the results) could be a good demo to be presented at the end of the project.

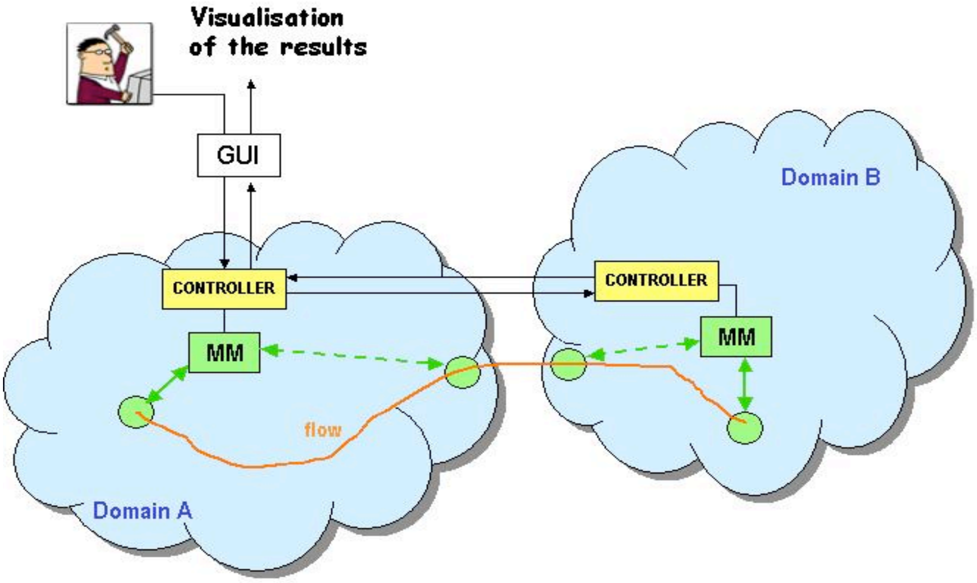


Figure 2-9: One-way delay measurement scenario

To show the basic functionalities, two different domains would be sufficient (cf. Figure 2-9). A generic user sends the request to its own domain using the GUI. One SMS is created, a part of which is related to the same domain and the rest to other domains. The controller splits the document into two new SMSs: the SMS with local scope is translated into the appropriate meter configuration language; the other one is directed to the controller in the upstream domain. Being domain B the last on the path, it doesn't split further the incoming SMS, but translates it into the appropriate meter language. The remaining part of the measurement process is then configured.

It should be noticed that the one-way delay (OWD) obtained would in this case be the sum of the OWD in domain A and the OWD in domain B and thus slightly incomplete because missing the delay between the two domains. This delay however is usually constant and therefore less interesting from our point of view than detailed information of the delay components in each domain on the path.

Figure 2-10 sketches the relation between SMS and the meter configuration language to configure one-way delay measurements.

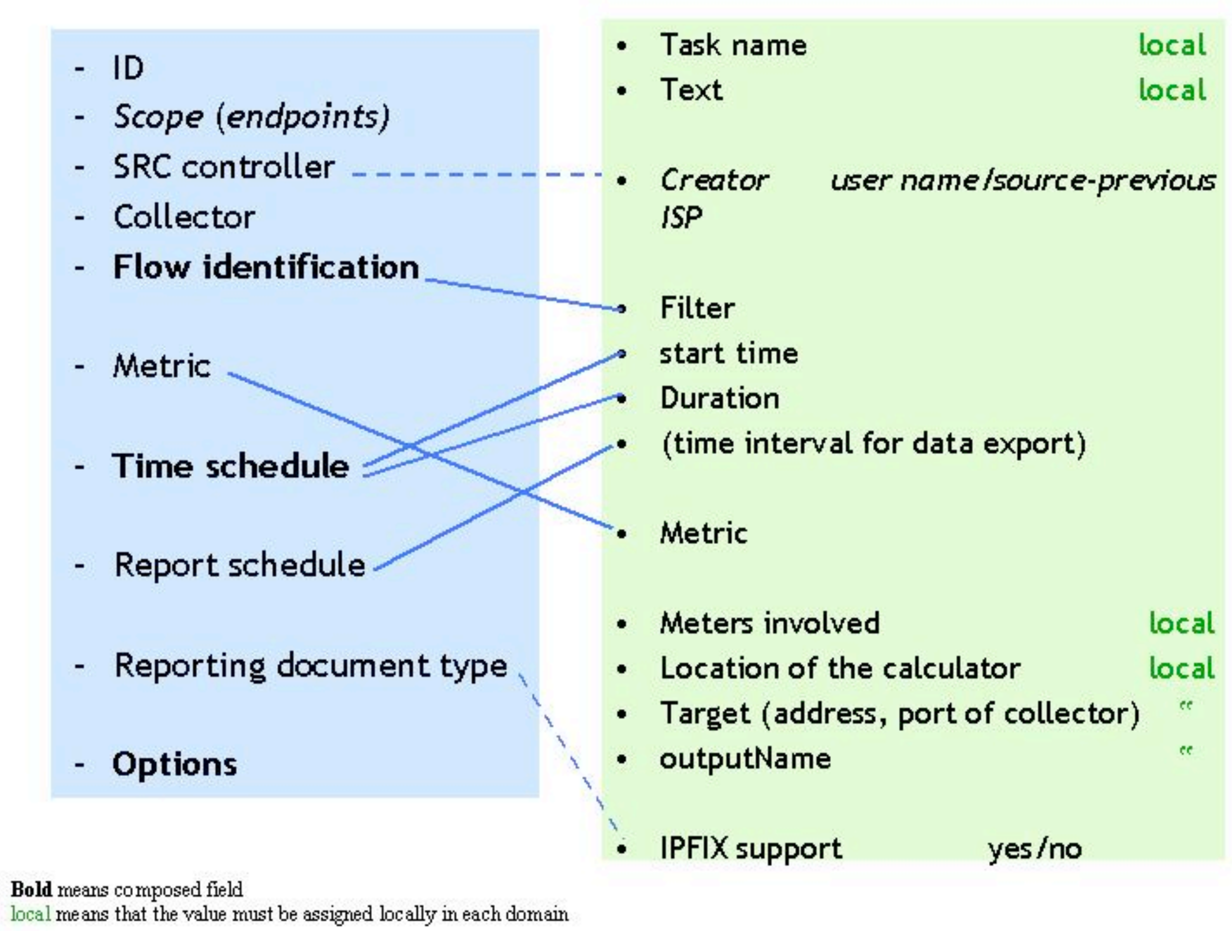

- ID
- Scope (endpoints)
- SRC controller
- Collector
- Flow identification
- Metric
- Time schedule
- Report schedule
- Reporting document type
- Options
• Task name local
• Text local
• Creator user name/source-previous ISP
• Filter
• start time
• Duration
• (time interval for data export)
• Metric
• Meters involved local
• Location of the calculator local
• Target (address, port of collector) "
• outputName "
• IPFIX support yes/no
Bold means composed field
local means that the value must be assigned locally in each domain

Figure 2-10: SMS (left) vs. 6QM meter configuration

The "translation" of SMSs into meter configuration language is relatively straightforward because:

- Many parameters can be taken directly from the SMS (e.g. time and report schedule, metric, etc).
- Other ones must be set locally as these data are known just inside the domain (e.g. meters involved) and are marked with the word "local" in the figure.
- Other parameters again can be derived from the information in the SMS and are represented with a dotted line (e.g. the indication of the IPFIX template in the field "Reporting document type" corresponds to "IPFIX support = yes" –and a bit more-).

The data export process takes advantage of the similarity between IPFIX and IDFIX: once IPFIX data are collected (and aggregated), the controller changes the header and sends IDFIX data to the controller that sent the SMS initially. The controller in the first domain will compose the data and visualize them for the final user.

2.5 Monitoring configuration process step-by-step

2.5.1 From the GUI to the SMS

Measurements can be started through the GUI. The administrator or authorized user, types on the GUI what kind of measurement should be done, when and how it should be performed. These data are inserted in an XML document, which is a first version of the SMS. This document arrives at the controller where the remaining SMS fields are inserted:

- The SMS ID is assigned.
- The Controller source address is inserted.
- The collector that will receive the measurement data is selected and its address inserted.

The controller provides the system with a consistent version of the SMS. From this point the configuration process can be started.

2.5.2 SMS splitting

Each controller receives an SMS as input and has two outputs: the intra-domain SMS that will be sent to the “core controller” for intra-domain monitoring configuration, and the extra-domain SMS that will be passed to the forwarder and from there sent to the next hop. Only exception is the controller in the last domain on the path; in that case just the intra-domain SMS is created.

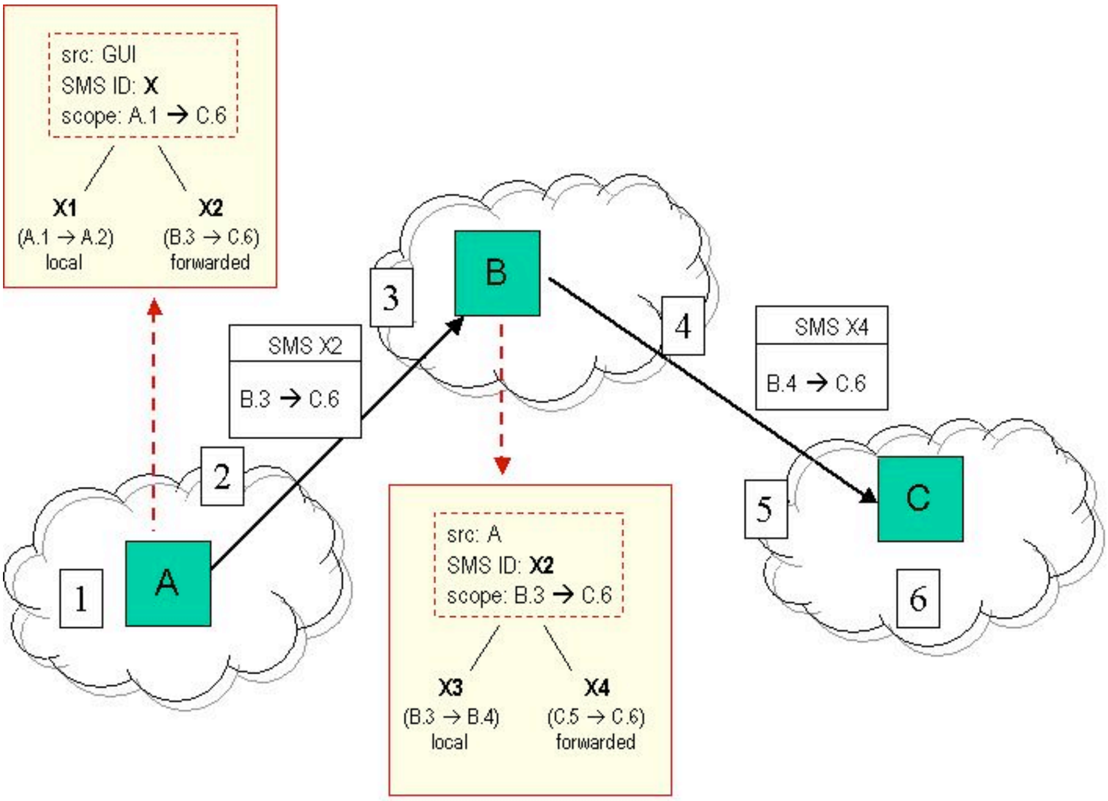


Figure 2-11: SMS splitting

The forwarder, in turn, will find out the address of the next controller on the path based on BGP data and send there the extra-domain SMS.

To describe in detail the configuration process, we refer to a simple scenario with only two domains, D1 and D2. We suppose that D1 receives a request to monitor a flow that goes from the endpoint T1 in Domain 1 to T2 in Domain 2 (cf. Figure 2-12).

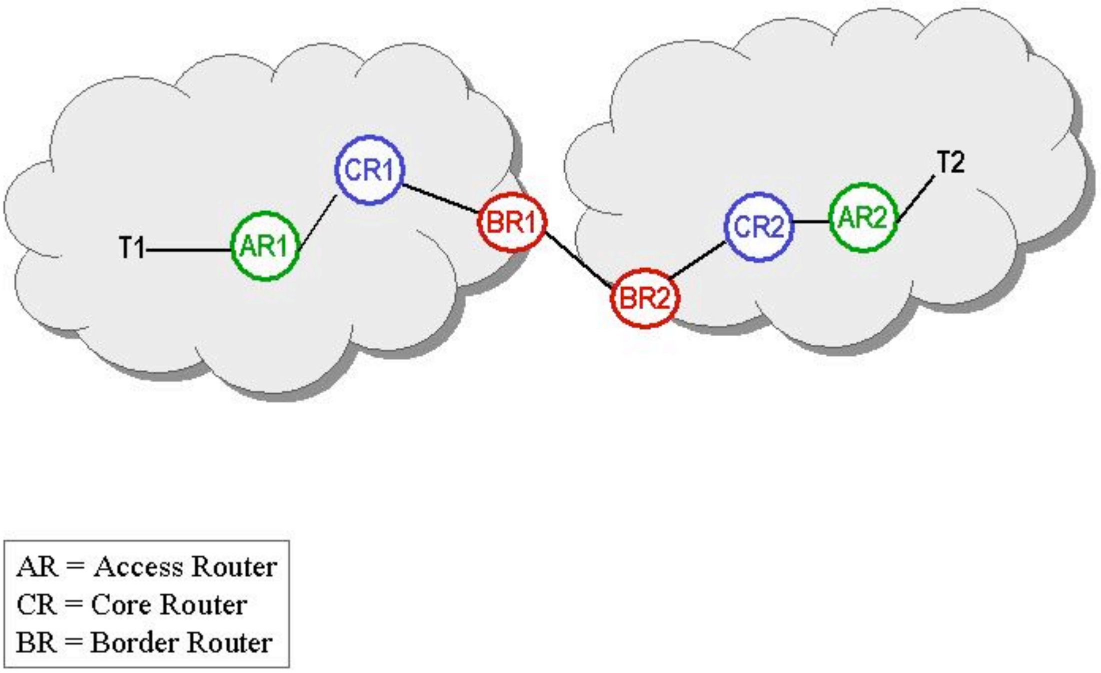


Figure 2-12: Flow path from T1 to T2

When the controller receives the SMS, the *SCOPE* field contains the values T1, T2. In order to split properly the SMS between the local domain and the other domains along the path, and to configure the measurement internally, the controller uses several tables.

- A table (Subnet, Access Router) is used to find out which Access Router is associated to the subnet T1 belongs to. In case of multiple access routers, one is picked randomly.
- The table that contains the associations (Meter, Router, Interface) is then used to select the appropriated probe to be configured, in our case P1. This process assures us to start the measurement from a point close to the endpoint T1.

At this point, the controller knows that T1 is in his domain, that it is close to the Access Router AS1 and knows the address of the meter to be configured.

What needs to be done now, is to find out which is the border router (which is also the domain egress point) close to which another measurement point has to be set and the next AS to forward the rest of the SMS to. This can be done either running a *traceroute* from AR1 to T2 or by looking at the BGP tables. The result is the path AR1, CR1, BR1, BR2, CR2, AR2, T2. The controller can now look again in the table (Meter, Router, Interface) to find the address of the probe close to BR1 to be configured, P2.

Now the splitting process can properly begin as the controller has all the necessary information:

- The probes P1 and P2, both located in Domain 1 are configured using the parameters specified in the SMS. More details on this “intra-domain” measurement configuration are provided in the following section.
- The controller now knows that the next step is BR2 in Domain 2. It then performs the following actions:

- Changes the SCOPE field of the SMS into BR2, T2.
- Looks in the table (AS, Controller name) for the address of the controller of Domain 2. This table has been created during the agreement phase.
- Puts its address in the filed source controller.
- Puts the collector address in the field *COLLECTOR*. Each domain has many collectors and a table in the controller counts how many flows are reported to each of them. The controller selects one of the less loaded and puts its address here. This guarantees resource sharing and scalability.
- Saves a copy of the SMS locally and sends the SMS (or a reference to it) to the selected collector.
- Forwards the SMS to domain 2.

The controller in domain 2 performs similar actions.

2.5.3 Intra-domain configuration

Upon arrival of the intra-domain SMS at the measurement controller of a domain, a measurement within the domain can be configured. This configuration process is actually two-parted. The process consists of meter selection followed by actual measurement configuration. The problem statements for both of these parts follow up.

2.5.3.1 Meter Selection

The Measurement Controller of a specific domain is in charge to realize the part of an inter-domain measurement that falls into its domain. Prior to this, the inter-domain controller of its domain has determined whether the source and/or destination of the flow specified in the SMS lie within its own domain. If the flow is destined for another domain, a modified SMS is passed on towards the flow's destination to the next inter-domain controller.

For the measurement within a domain, it is necessary to determine which network probes are at disposal for examining the flow specified in the SMS. Then, as a second step, appropriate probes have to be selected and measurement tasks according to the SMS must be initiated for these probes.

In fact, the measurement controller needs topology information enabling a selection of meters in relation to flow and location of a meter on the path of the flow. An example for this problem statement is presented with Figure 2-13. For this example, we assume two distinct cases for which we want to inspect first, the black flow (f1) entering the domain at ingress point i1 and leaving the domain at egress point e1 and secondly, the blue flow (f2) entering at i1 and leaving at e2. Specifically, we intend to measure the one-way delay within this single domain. Since this is a two point measurement the probe selection process has to come up with two appropriately located probes for each of the stated cases.

As depicted in the figure, there exists a set of probes P1 to P4. Some of these probes are able to examine both the black and blue flow, namely P1 and P2. On the other hand, the probe P3 can exclusively examine the black flow whereas P4 can examine the blue one respectively.

Optimal selection of probes for case one *"examining black flow f1"* would yield probes P1 and P3; for case two *"examination of blue flow f2"* would yield probes P1 and P4. The optimal selection of probes should be based on the location of probes described in terms of distance to ingress and egress point, respectively. As a detail, the selection process must determine that probe P2 it not a feasible selection for neither of the two cases although P2 is able to observe both f1 and f2.

Please note that without loss of generality, when the source and/or the destination of the flow under observation lie within the domain, the same problem of selecting probes arises; and for this scenario, the term *ingress* can be interchanged by the term *source* and the term *egress* by *destination*.

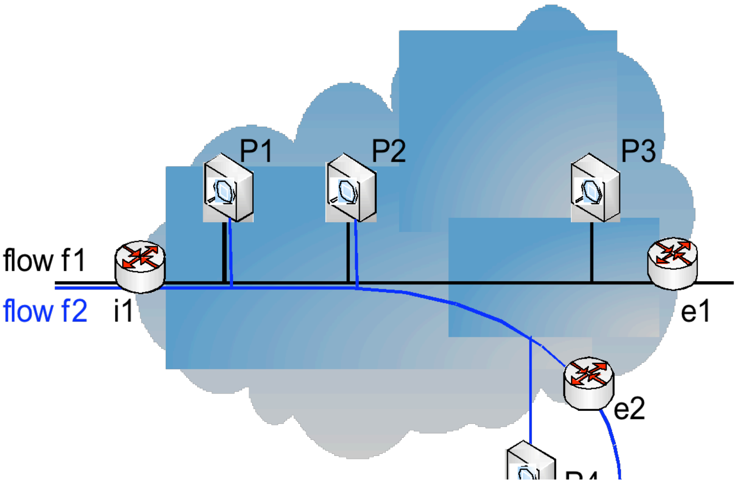


Figure 2-13: Problem of probe selection

A first stage of the selection algorithm establishes for a flow an ordered set of probes that are able to examine the flow. For our example, the set for f1 would be <P1, P2, P3>. The order of the set elements is based on the probes distance to ingress or source of flow f1. The second stage of the selection algorithm selects the first (P1) and the last element (P3) of the ordered set of probes. These probes are the optimal selection for examining one-way delay of black flow f1. Likewise, for case two: the algorithm would first establish <P1, P2, P4> as ordered set of probes and secondly, selects P1 and P4 for the examination of blue flow f2's one-way delay.

Conclusively, the information, which the probes can monitor, which flows (prefix and prefix length) must be kept by the measurement controller. Additionally, an order of probes must be established for the selection process of meters. As an alternative of manually feeding the information, the probes could learn the flows that they are able to monitor and report those results as part of their capability list to the measurement controller. However, this is not sufficient for finding the order of probes. Assuming that all probes within the domain are timely synchronized, the order can be found by inspection of absolute timestamps for packets of a flow. This process is the same as it is for determination of one-way delay: for a packet and a set of N probes that the packets passes N 3-tuples of the form (packet id, timestamp, probe) can be collected. Because timestamps within the 3-tuples monotonously increase along the transportation path of the packet the order of timestamps will ascertain the order of network probes. The applicability of this algorithm surely depends on the accuracy of the underlying time synchronization process.

Instead of using timestamps as describes prior, the probes can also report the value of time-to-live (TTL)/hop-limit fields within the packet header along with the prefix and prefix length values for the flows they can observe.

2.5.3.2 Measurement Configuration

The inter-domain controller generates a variant of an incoming SMS. This intra-domain variant contains the information that is needed for configuring a measurement within the domain of the controller:

- From the specification within flow identification part of the SMS a filter expression is constructed to restrict the measurement to the identified flow based on source, destination address, etc.
- Furthermore, the measurement start time and duration are taken from time schedule part of the SMS.
- The export of measurement data from the probes to the collector is configured according to the information in the report schedule part of the SMS.
- The collector to which intra-domain measurement data have to be sent, has been chosen previously by the inter-domain controller. The details are passed within the intra-domain SMS.

Finally, measurement tasks are generated with the information from the intra-domain SMS on probes that have been determined prior. (Refer to 2.5.3.1.)

2.6 Security considerations

The goal of this section is to give some guidelines on how the process described up to now can work securely in an inter-domain environment (where ISPs are even more, if possible, sensible to the problem…). We proposed two solutions: the first one is simpler and therefore easier and faster to implement while the second one involves AAA and its standard protocol, Diameter [diameter].

2.6.1 Public key, private key approach

An easier way to carry out the inter-domain communication would be using the asymmetric key exchange. Each domain has his own private key which is kept secret and either has or can retrieve from a trusted third party the public key of its peer. The two peer domains (or better the inter-domain controllers) can use asymmetric keys for authentication and authorization (and to send the SMS) and then agree on a session key, more efficient in terms of computation, for the data.

2.6.2 AAA

The communication between two different ISPs goes through their local AAA servers. Each AAA server will store in a table, which ISP (or, more in detail, which user or group of that ISP) has the right to do / get what, according to previously negotiated agreements.

The possible “actions” are data collection and configuration (e.g. ISP A wants ISP B to measure something and configures a task using AAA).

Figure 2-14 shows how a procedure between different domains can be started. If ISP A is interested in data owned by ISP B; the domain controller of ISP A asks the local AAA server to prepare a Data Request message, where it is specified what kinds of data are needed and ISP A identifies itself (Step 1). Afterwards, the request is sent by the local AAA component to the AAA component of ISP B (Step 2).

This component verifies together with the local inter-domain interaction component whether ISP A is authorized to obtain the requested data (Steps 3 and 4) and sends the answer (authorized, not authorized, error) back to ISP A (Steps 5 and 6). This communication is done using the Diameter protocol [diameter]. Only upon confirmation, the controller fetches data from the Database (Step 7) and sends them, using an encrypted channel, to ISP A (Step 8). The session key is provided by the AAA component after the transaction approval.

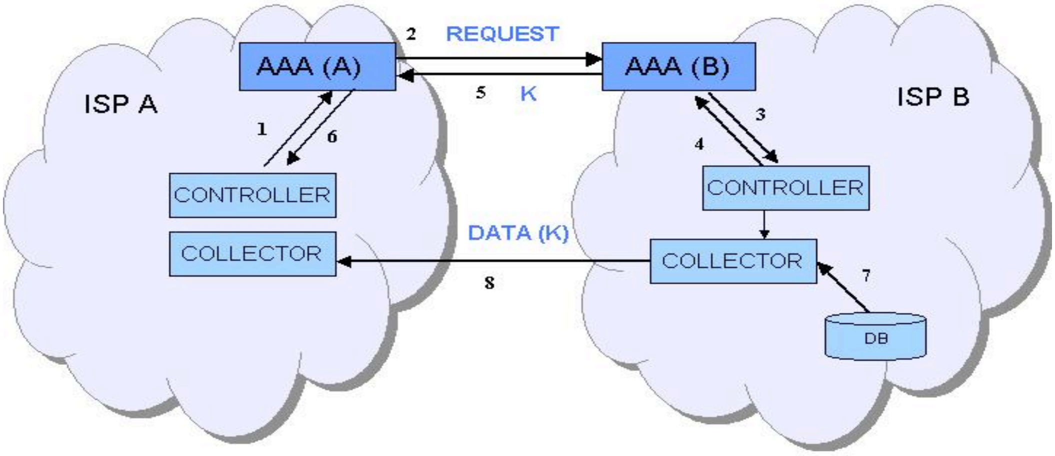


Figure 2-14: Inter-domain communication with AAA

3. SECURITY

The purpose of this section is to propose enhancement necessary to enforce the security of the measurement system. The topics addressed in the following of the documents are:

- The security of inter-components communications by using IPsec.
- The meter component access restriction by using a firewall.
- The management server access restriction.
- The organization of users and privileges.

3.1 IPsec Secured channel

In this part of the document, we focus on the security required between the meter hosts and the control/collection hosts. For a fully secured system, the security would be required for both signaling and data export. The signaling part, that is actually the communication between the controller and the meter, requires authentication and integrity. The data export part, that is actually the communication between the meter and the collector requires confidentiality, authentication and integrity.

Some security mechanism has been implemented in prototype based on asymmetric key authentication and SSH2. However, this concerns only the data export from the meter to the collector as a consequence there is a need to review the security of the prototype especially concerning the signaling part. Securing the data export was seen as a priority; so consequently this issue was covered at the first stage. A stronger security policy protecting both signaling part and data part is required. In the following, we propose some configuration using IPsec instead of SSH2.

3.1.1 IPsec Overview

The purpose of this sub-section is not to provide a deep description of IPsec which can be easily found in the literature, but just to give the component outline.

There are three core IPsec components:

- The authentication header (AH - RFC2402) verifies the identity of a packet's sender and the authenticity of the packet's contents.
- The encapsulating security payload (ESP - RFC2406) encrypts a packet before transmitting it; ESP may also encapsulate the original IP packet in tunneling mode.
- The Internet key exchange (IKE) manages key transfer between senders and receivers.

IPsec is very flexible, indeed AH and ESP can be used with various authentication and encryption schemes, besides IPsec can be used in several mode namely transport or tunnel mode. Moreover, IKE can be used with different flavors based on, for example:

- Share secret, which is known not be very scalable.
- Public key.
- Certificate and public key infrastructure.

Because of the high degree of flexibility offered and the multiplicity of components and options, IPsec is a complex issue. As a consequence the study required before going for IPsec deployment should not be under-evaluated as greatly pointed out in [BELL].

3.1.2 Selected Configuration

In order to ease real deployment, we simplified our approach by assuming that both controller-meter and meter-collector channel could have the same security configuration. Indeed using the most stringent policy should be able to fulfill our former requirement. From now we will assume the use of a unique security configuration enabling: confidentiality, authentication, integrity.

The simplest and most realistic configuration we propose is based on:

- Transport mode.
- ESP: authentication.
- ESP: encryption.
- IKE: public key scheme.

The transport mode is preferred to the tunnel mode because a meter is very likely to export a lot of data; thus in transport mode, the protocol overhead and the complexity should be lighter than in tunnel mode as a consequence it looks more appropriate for the exporting device.

For IKE, the public key scheme is preferred to the share secret to avoid a secret management that may not be very scalable when the number of meters increases in the system. The public key infrastructure with certificate scheme has been discarded in this approach because it is a strong assumption to believe that there will be a PKI available in the network of deployment. Actually this is not very likely to be the case at least under the present circumstances.

3.1.3 Relation to Implementation

We carried out some basic experiment to test the feasibility of IPsec deployment in IPv6. Several implementations are available on IPv6, for example from:

- USAGI for Linux based on Free/SWAN implementation.
- KAME for BSD.

In our case we used USAGI implementation. We setup the configuration pictured on Figure 3-1.

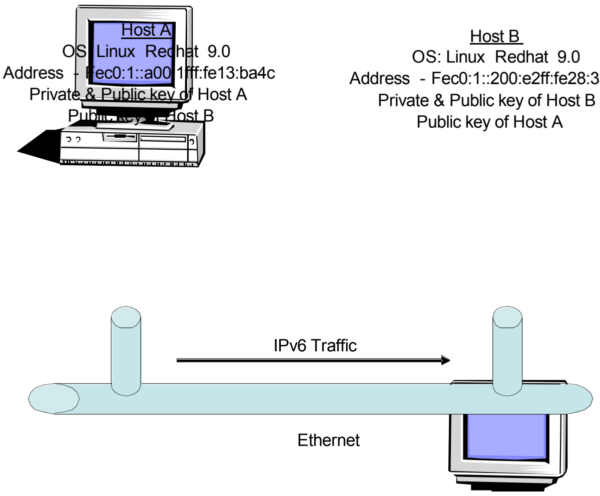

Host A
OS: Linux Redhat 9.0
Address - Fec0:1::a00:1fff:fe13:ba4c
Private & Public key of Host A
Public key of Host B
Host B
OS: Linux Redhat 9.0
Address - Fec0:1::200:e2ff:fe28:3a85
Private & Public key of Host B
Public key of Host A
IPv6 Traffic
Ethernet

Figure 3-1: IPsec Configuration

The following features were used:

- Transport mode.
- ESP: authentication (hmac-md5).
- ESP: encryption (3des-cbc).
- IKE: public key – RSA.

The configuration file used on both hosts can be found on Figure 3-2. This basic test was concluded successfully. As consequence the usage IPsec in this mode seems feasible; however one remark we can make concerning a deployment issue is that in our case the installation of USAGI IPsec required the recompilation of the host kernel. We mention this latter point as this has some impact on the deployment in real production networks.

```
############################################################
# USAGI sample configurations
# USAGI IPv6 Transport mode sample configuration
############################################################

conn testv6
        af=inet6
        type=transport
        auth= esp
        authby =rsasig
        left=fec0:1::a00:1fff:fe13:ba4c
        right=fec0:1::200:e2ff:fe28:3a85
        esp=3des -md5-96

        rightrsasigkey  =0sAQO1dpEhusM3GeNbNDqdMqCqCtzz6BU6uD19
pXhHscvMYSiN4YgwOqjCpGgRTWVqFSN7QXufQ5Mirxt13Nut6GvV1zR740fL
QC03Vu93dPBUvuzhFpZquk0+tVtQmWL/uvx8YvPPRlYAbT/fGSIqtO+f5eZGE
0hVSC8sc7iwHtyNTQ==

        leftrsasigkey  =0sAQO6OVRvhULf5EhEUqke4O1uWiNEjPoj9TYq+kP
h1WUne2bD+FCtqt37syHkM7WvJZdpbxmF7cE1tuy5qRdUh0x7mNcs+sga7Vo
BgWkcCxk5fCGi15//2qfRg/c/Xscad/Z/sJ4j5kUHJq+KiFzkLLI8yleGRSGnWck        m
dMKDIGRBXQ==

############################################################
# End of USAGI sample configurations
############################################################
```

Figure 3-2: IPsec Configuration file

3.1.4 IPsec Conclusion

To provide a little more details on our analysis the following figure summarizes and compares IPsec and our existing SSH usage in the context of our prototype.

As a conclusion of this subsection IPsec appears as a possible candidate to secure 6QM prototype in replacement of our existing scheme based on SSH. The usage of IPsec could enforce the security that currently exists in the prototype and also takes more advantage of IPv6 feature as IPsec is mandated in IPv6.

The hosts used in the infrastructure should take advantage of firewall protection to avoid malicious access. This is especially true concerning the meter host. Indeed in many deployments a separate management network will not be available meaning that the network interface used for control will be connected to the network under observation which could simply be the a network opened to public. As a consequence there is an opened door for malicious access that makes meter access restriction necessary.

	Advantages	Drawbacks
IPsec	- Support any transport protocol (appropriate for IPFIX extension with UDP/SCTP) - Usage managed independently from the prototype code - Could cover both signaling and data part without code modification - Should become highly available in the future as mandated in IPv6	- Currently not always available on production hosts (may require kernel recompilation) - Implementation issue: different implementation may have interoperability issue - May interfere with NAT and firewall (however NAT may not be often used in IPv6 environment)
SSH	- Highly available (usually enable on existing hosts) - Application granularity	- Specialized in TCP services (problematic for IPFIX extension with UDP/SCTP) - Integration may require prototype some code modification in order to integrate usage in the control part generating modification work

Figure 3-3: IPsec versus SSH

The solution would simply be to use a firewall protection on the meter host.

It has to be noted that the firewall rules set at the meter should allow the connection from the QoS server to the control port of the meter.

3.2 Web server access security

One particular aspect when considering *security* for the measurement system is the access to the Web server that realizes the Graphical User Interface (GUI). For the access to the Web Server we will consider here the three important processes, these are:

- Authentication.
- Authorization.
- Privacy of communication.

Authentication verifies that a communication partner is the one he claims to be. Usually this process includes the exchange of username and password, but it can include means that are more sophisticated.

Authorization clarifies whether a formerly successfully authenticated person has allowance to use a certain resource. The conditions in the process of authorization that must be fulfilled to get access to the resource depend on the particular application. They may include verification of membership to a group or others.

The Apache Web server that has been used for the implementation of the GUI to the measurement system supports authentication and authorization processes by means of Secure Socket Layer (SSL) protocol widely known as HTTPS. This protocol is in fact HTTP exchanged over SSL above the transport layer (e.g. TCP). SSL provides a secure way of communication between client and server. It allows mutual authentication of the client against the server and vice versa. Moreover, SSL optionally uses digital signatures for integrity of messages. Very important and commonly known is the circumstance that it supports encryption to guarantee *privacy of communication*.

SSL initiates a session by means of a handshake sequence. The result of that handshake is a proper selection of session parameters. A successful negotiation is mandatory for the communication to take place afterwards. The SSL connection is parameterized by:

- Negotiation of the cipher suite (including key exchange method, cipher for data transfer, message digest for creating the Message Authentication Code) to be used during data transfer.
- Establishment and exchange of a (symmetric) session key between server and client.
- Optionally authentication of the server to the client.
- Optionally authentication of the client to the server.

Authentication uses certificates in a chain of trust. Therefore, in order to use authentication users of a secured Web access must have valid certificates issued.

Concluding the discussion, we note that access to the GUI of the measurement system that has been realized with the Apache Web server can be secured by the aforementioned authentication and authorization process by means of HTTPS. As a prerequisite, once a user is identified further access policies and restriction can be implemented.

3.3 User management, privileges, access control

User management goes hand in hand with the authentication and authorization process described under section 3.2 as access to the measurement controller is concerned. In a multi-user system, it is necessary to register identities and assign them privileges. A *privilege* is a right to execute a particular type of command or function of the measurement system. Furthermore, *roles* are created by administrative persons and are used to group together privileges or roles; roles represent a means of facilitating the granting of multiple privileges or roles to users.

We exemplarily give here some thoughts about a user and role management. It naturally makes sense to distinguish the roles of administrative personal from other users since their tasks are quite different. As a first approach, we define basic roles that relate to the usage of the measurement system:

- System administrator.
- User manager.
- Backup Operator.
- System user.

For the previously defined roles, we exemplarily list the tasks assigned to members of those roles: The *system administrator* will superintend the measurement system. In order to fulfill this task he has to manage the system components. Eventually he has to add and remove system components. He has to enter information about system components. He has to register additional probes, tear down, or install system services. The *user manager* in contrast is solely in charge for user information. His business includes the enrolment for new system user and the deletion, prolong expired user accounts. He has to generate certificates for users and so on. The *backup operator* mainly deals with the data repository of the system. Therefore, this role needs grant to provided access the database backup functions; this includes both, access to repositories of measurement data and user information. Finally, the system user executes actual measurement related function as he creates measurement tasks. In order to fulfill his assignments he needs access to certain entities of the measurement system components. The user manager maintains the authorization, that is, the relation between users and accessible components.

Next figure shows a minimal set of user groups that should be distinguished within the measurement system. The table describes resources or functions of the measurement system that are access or used by a user group. As indicated in the table for these resources and functions either full access is granted or policies apply. Finally, functionality that is not relevant for a member of a role group might be hidden from access or view.

Role	**Resource/Function**				
	Management User Accounts	**Assignment User Privileges**	**Management System Components**	**Database Backup & Restore**	**Management Measurement Tasks**
System Administrator	Hidden	Hidden	Full access	Hidden	Hidden
User Manager	Full access	Full Access	Hidden	Hidden	Hidden
Backup Operator	Hidden	Hidden	Hidden	Full access	Hidden
System User	Hidden	Hidden	Hidden	Hidden	Explicit Permissions & Measurement Access Control

Figure 3-4: User roles and privileges

For the role of system users, we identified that mainly the creation and deletion of measurement tasks and examination of measurement results are of interest. Since certain measurements, such as full packet captures, are suited to reveal sensitive information, we vote for the explicit assignment of permissions to every single type of measurement for user groups. Additionally, it makes sense to restrict the access to certain entities of probes and meters or other system components.

As expressed briefly in the figure above, an access control should prevent contradicting settings of measurements. In a scenario in which for instance passive and active measurements are conducted concurrently but without intention results could be driven worthless and would in a worst case be wrongly interpreted. Furthermore, the access control must enforce that tasks of different system user groups are isolated: initiation, modification and cancellation of measurement tasks must not be allowed for individuals that do not belong to the task owner's group.

The considerations we made above testify that realization of full-fledged user administration including fine-grained privilege management and access control is challenging; nevertheless, it is important in a multi-user environment in order to enable a coordinated work with the system.

In the following paragraph, we propose how to implement the privilege management to allow or forbid access to functions of measurement servers (probes and meters) on a user basis. In order to describe the implementation we have chosen to use Object-Relation modeling method.

We start off by stating elementary facts that we want to express and handle with the privilege management:

- f1: user U1 is member of user group G1
- f2: user U2 is member of user group G1
- f3: user U2 is member of user group G2
- f4: user U1 is authenticated with credential C1
- f5: user U2 is authenticated with credential C2
- f6: function F1 is implemented on measurement server S1
- f7: function F2 is implemented on measurement server S1
- f8: group G1 is allowed to use function F1 on measurement server S1
- f9: group G2 is allowed to use function F2 on measurement server S1
- f10: group G2 is forbidden to use function F1 on measurement server S2

The stated elementary facts are instances of the fact types:

- f1, f2, f3 ➔FACT1: user is member of a user group
- f4, f5 ➔ FACT2: user is authenticated with credential
- f6, f7 ➔ FACT3: function is implemented on measurement server
- f8, f9 ➔ FACT4: group is allowed to use function on measurement server
- f10 ➔ FACT5: group is forbidden to use function on measurement server

The resulting fact types (which are entities and relations) are sketched in above along with the fictive sample population according to the elementary facts we initially stated.

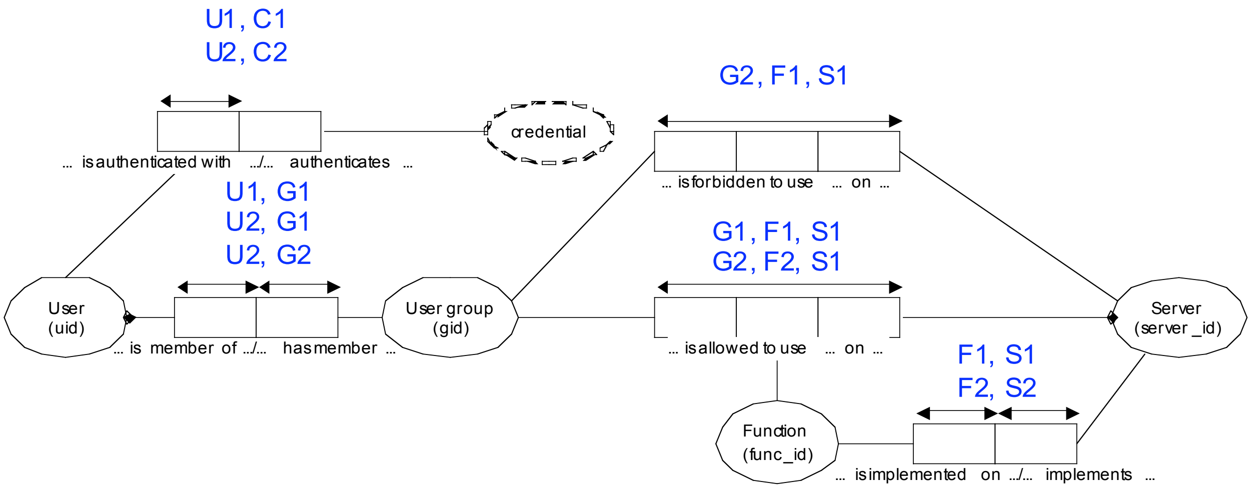

U1, C1
U2, C2
credential
... is authenticated with ../... authenticates ...
G2, F1, S1
... is forbidden to use ... on ...
U1, G1
U2, G1
U2, G2
User
(uid)
... is member of ../... has member ...
User group
(gid)
G1, F1, S1
G2, F2, S1
... is allowed to use ... on ...
Server
(server_id)
F1, S1
F2, S2
Function
(func_id)
... is implemented on ../... implements ...

Figure 3-5: ORM diagram with sample population

The model depicted serves as a basis for implementing the information model into a database management system. The overall concept is as follows: A user is authenticated against the measurement controller by presenting his credential. These credentials are checked with the data in the database. On successful authentication, the user is henceforth referenced by the system by means of his user identification (uid). When the user is about to configure a specific type of measurement his privileges are checked prior by the controller:

- What is the set of groups the user a member of?
- What functions are required to setup the measurement? e.g. *packetid* for one-way delay measurement.
- What is the set of servers for which the access regarding the specific function is “allowed” and never “forbidden”?. Note: “Forbidden” takes precedence over “allowed”.

The Graphical User Interface presents the determined list of servers, to the user. For the introduced examples this means: user U1 is able to use function F1 on measurement server S1; user U2 is able to use function F2 on S2; but he is not able to use function F1 on S. Please note, that the use of access “forbidden” for a group is a convenient way for excluding a subgroup of users from access.

4. Scalability

The original design let many issues concerning scalability unaddressed. The following section makes some proposal in order to improve the system scalability. The following issues are addressed:

- Usage of sampling technique in the context of passive one-way delay.
- Proposal for a more decentralized component architecture.
- Proposal for data management.

4.1 Packet Sampling

The amount of data collected can increase very quickly especially in the case of passive measurement for quality of service measurement. As defined in the original specification our measurement system uses packet identifiers based on hashed function instead of raw packets in order to reduce the amount of exported data. The amount of data created can really vary depending on the traffic under observation. For example, a video streaming traffic is quite different from a web browsing traffic. The result is that the amount of data created for the measurement varies significantly and the amount of data to be manipulated can be huge in some cases.

High amount of data are known to be more difficult to store, manipulate, analyze and transmit. As a consequence data reduction technique could be recommended for an advanced system depending on the existing constraints and system usage. It could be especially recommended for flows creating a high number of packets such as streaming services.

The IETF PSAMP working group has defined the framework for the IP packet sampling. The following part of the document examines some usage of this sampling framework in order to reduce the amount of information to be processed for one-way delay measurement.

4.1.1 One-way delay and Sampling

As described in D3.1 in order to compute the delay, a packet is captured, time stamped, filtered and some packet identifier (packet ID) is created based on a hash function (CRC32). Those operations are performed at least at two observation points. Finally, the set of packet ID and time stamp is exported to a single point for a correlation that is based on packet ID matching and time stamp comparison.

In this measurement technique the purpose of sampling could be:

- To reduce the processing power which is required for the packet ID generation process.
- To reduce the bandwidth necessary for the packet data export process.
- To reduce the number of packets to be correlated at the central correlation point.

Reducing the packet ID generation required resource could be realized by having sampling between filtering process and packet ID generation process. As a benefit:

- The sampling would reduce the number of packets, which would be processed and then save computation resource.
- Another induced consequence would also be to reduce the amount of data to be exported from the measurement device itself to a central correlation point.

In this case the framework defined by PSAMP in [P-Arch] would be applicable and result in the framework previously shown, for example.

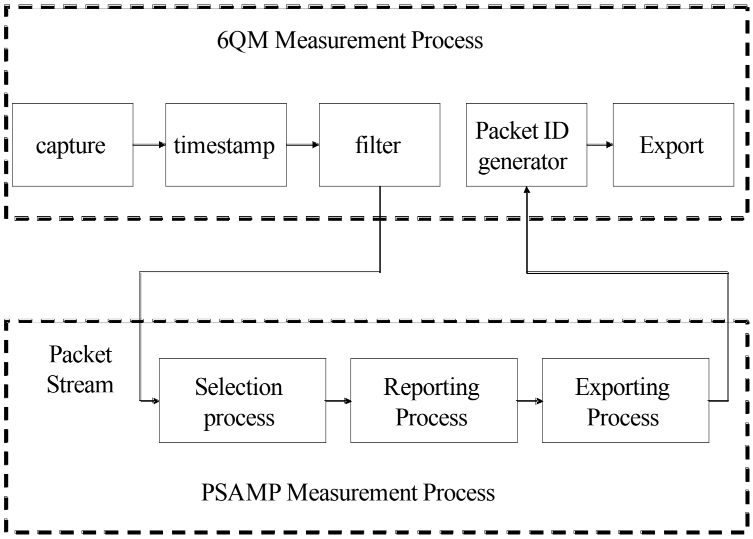


Figure 4-1: PSAMP in 6QM Architecture

Just reducing the export bandwidth and the number of packets to be correlated could be done by having a sampling process between the packet ID generator and the export process. However it is not possible to apply strictly PSAMP framework as the output of the Packet ID generator is not a “packet stream”, but a stream of packet ID and time stamps. By extension one can still generalize and apply PSAMP sampling concepts to the packet ID themselves instead of the real IP packets as an extension.

Packet sampling allows reducing the data production however in the case of passive 2-point measurements one should be careful when selecting the sampling algorithm. Indeed [PSP] refers to several sampling techniques however not all are applicable.

The 2-point measurements is based on packet matching that supposed that any applied packet sampling technique must be able to correlate packet selection at each measurement point even when packet loss occurs or the packet order changes. In other words the sampling techniques used at different points have to be synchronized in some extent in order to capture the same packet at different measurement points. This is a key point in the sampling selection, it is clear that if the same packet is not selected at each measurement point no correlation is possible and as a consequence it is impossible to compute any two point metric.

The problem that can be encountered is presented on the Figure 4-2. There is a flow composed of a set of packets P={P1, P2, P3, P4, P5}. The packets are captured at the meter 1 then the sampling process S1 selects the set of packets P(1) = {P1, P4} out of P. At the meter 2 due to network loss and order changes in the packet flow the meter 2 captures {P2, P1, P5, P3}, P4 is lost. Then the packet sampling process S2 selects P(2) = {P3, P2}. Obviously in this case P(1) and P(2) are disjoint meaning that not correlation is possible. This is exactly this problem that should be solved in order to compute one-way delay with non-intrusive method.

Figure 4-2: Correlation Problem with Sampling

As mentioned previously the key point is to be able to select the same packet at different points in the network in other words to have a synchronized sampling. [SSLA] provides an interesting study on sampling and one-way delay measurement and defines some architecture for sampling usage. As stated in [SSLA]:

- “Count-based trigger” sampling is almost impossible to be synchronized because of network loss and packet order change.
- “Time-based trigger” sampling could be applicable with some special window size to be defined. However, we think that there would still be a problem about selecting the correct window size as it is dependant on network delay and therefore is a variable value.
- “Packet-content-based trigger” sampling is possible to use to have a synchronized sampling.

In addition to this paper one can add that regular probabilistic sampling method would be difficult (certainly impossible) to use because of their lack of synchronization at different measurement points, which is due to their random selection nature.

As a conclusion one remaining candidate would be the “Packet-content-based trigger” for the sampling technique. A classical way to identify the packet in different points of measurement is to use a part of its content or an identifier based on its content. Typically this identifier can be a hash of a part of the packet content or a message digest, such as MD5. However this process is known to be resource consuming if we need to compute a hash for all the packets, this is the major drawback of this technique.

As a conclusion only using a packet sampling presents several problems concerning the algorithm sampling selection, but one solution exists with the “Packet-content-based trigger” to identify the same packet along the path. A simple practical solution would be to use a hashed based packet identifier and the same selection algorithm based of this identifier value at each measurement point.

4.1.2 Content-based selection of packets for sampling

In fact, the stated problem of synchronously selecting packets for sampling is the same as described in [DuGr00]. The authors propose sampling of packets as they traverse network elements such as routers. Identifying a packet at these elements makes is possible to track down which route a packet took coming from an ingress node and exiting the network at an egress node. This has been named as *trajectory sampling*. Sampling decision for a specific packet is taken grounded on a calculated hash value derived from parts of the packet content that are invariant throughout its transportation. Assuming every observation point has the same notion of the sampling decision, this way a packet either is sampled at any observation point or is not sampled at all. [DuGeGr02] presents realization of a trajectory sampling system with an analysis application.

Content-based sampling must assure that the sampled set of packets has the same distribution concerning a specific characteristic as the original distribution. As an example, for the case of one-way delay measurement the sampling process must not prefer packets with a certain payload size. The reason is that transport delay is also a function of packets size. If the selection algorithm selects packets of similar size, a biased estimation of one-way delay would result.

For the choice of a hash function used in content-based sampling process one has to be aware of the aforementioned problematic. To which extend a poor pick of the hash function affects the metrics needs to be investigated.

4.1.3 Two-level Sampling

In some cases we may want to have even more data reduction in the system while saving computational resource. The purpose of this section is to extend the architecture mentioned previously and presented in [SSLA] to take advantage of the 6QM design properties.

4.1.3.1 Two-level Sampling meter

In this part of the document we propose to extend the sampling approach by using a two level sampling component:

- The first level of sampling refers as “Packet Sampling” (S) applied just after the meter packet filter aims at reducing the processing power for the packet ID generation process.
- The second level of sampling refers as “Report Sampling” (RS) applied just before the export process, which aims at reducing the amount of per packet information to be exported. The “Report” contains information such packet ID and timestamp in our case.

This architecture is illustrated in Figure 4-3 where the colored components are extension from the [SSLA] architecture.

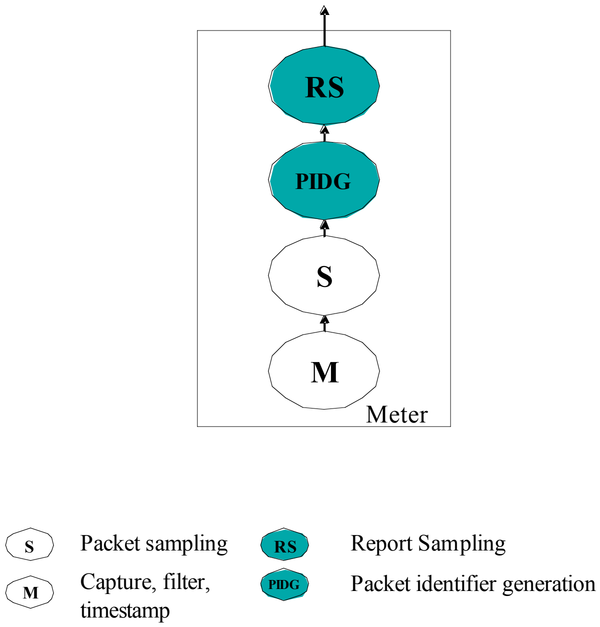


Figure 4-3: Two level Sampling

This proposal can raise several questions:

- Why don't we just use the regular packet sampling instead of RS?
    - The reason is that the output of the packet ID generator is not a packet flow but a stream of packet identifiers and time stamps.
- What is the base for selection at RS?
    - RS will be based on a "Packet-content-based trigger" more precisely the selection process will be based on the value of the packet identifier. As this value is already calculated for the regular one-way delay operation, it is easy and cheap from a computational point of view to reuse it for a selection decision too.

The philosophy of the two level sampling is to limit the number of packets to enter the packet ID generation process as it is an expensive process and take advantage of the packet ID generation process to identify packet. As a consequence the packet ID will serve both:

- As "Packet-content-based trigger" for sampling.
- And as a packet identifier for the packet matching at the final correlation point.

The report sampling could be synchronized easily by taking the same deterministic selection algorithm based on packet ID values at each measurement points however for the packet sampling part the synchronization is not solved yet if we do not use another "Packet-content-based trigger". A solution to this problem is provided in the following proposal.

4.1.3.2 Asymmetric Two-level Sampling Architecture

The synchronization of the packet sampling is the key point to ensure the success of the two-level sampling as in the case of the regular sampling. A simple solution is proposed on the Figure 4-4.

The packet sampling at the meter 1 has been suppressed. By doing so, no synchronization is required at the packet sampling level. Indeed Meter 1 will generate a packet identifier for all the captured packets while at the meter 2, a packet sampling process will be selecting the packets to be sent to the packet identifier generator (PIDG2). Concerning RS1 and RS2 it is possible to synchronize them as mentioned previously by using some hash value based technique ("Packet-content-based trigger").

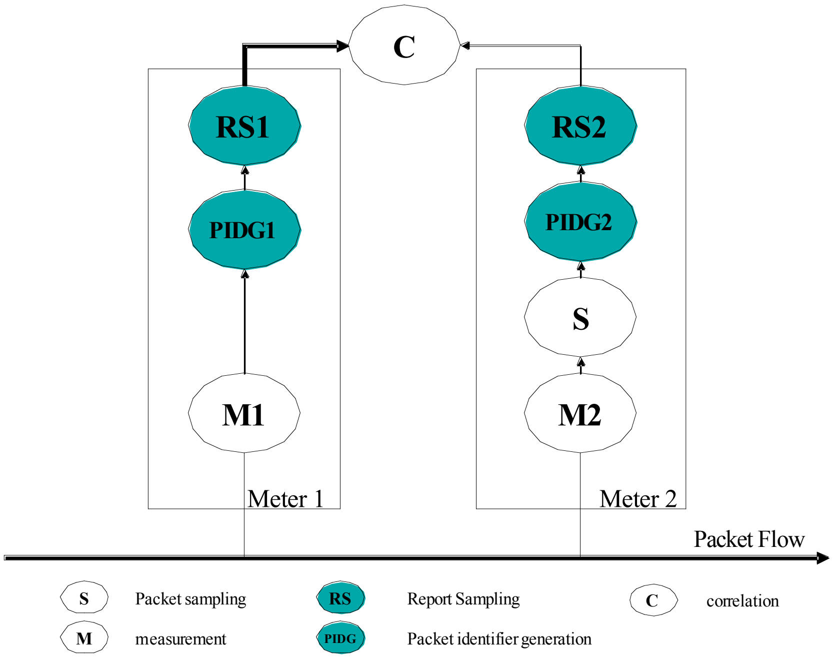


Figure 4-4: Asymmetric Sampling

Impact of this architecture:

- RS1 limits the amount of exported data from meter 1.
- RS2 limits the amount of exported data from meter 2.
- S limits the number of packets to be processed by PIDG2 as a consequence there is some saving of computational resource at meter 2.
- At packet sampling level no sampling synchronization is required as no packet sampling process exists at the meter 1. Packets selected out of S should be a subset of the packets capture by M1.
- S may be based on a simple algorithm (e.g. count based).
- This architecture is fundamentally asymmetric. One consequence is that the bandwidth required for the flow data export will be asymmetric too.

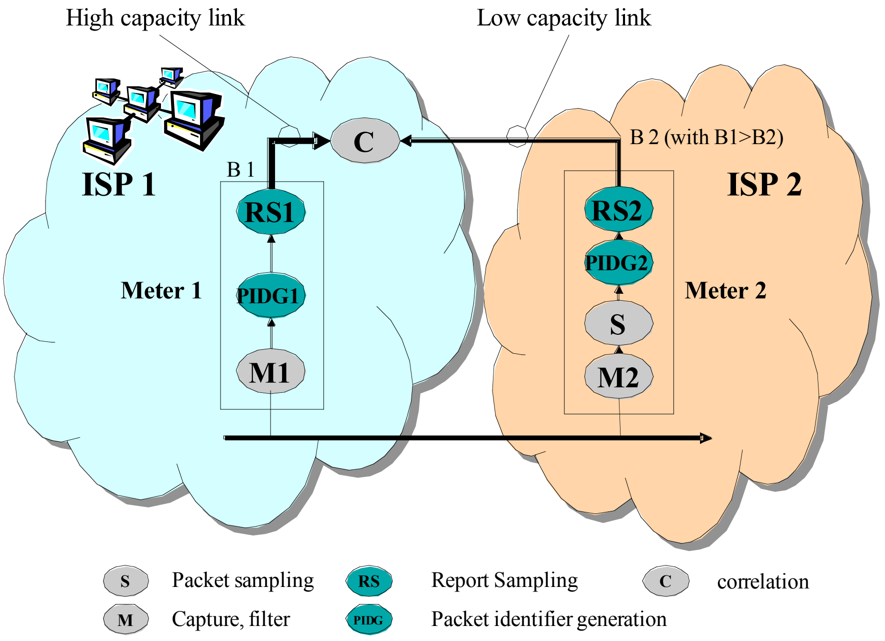


Figure 4-5: Sampling in Interdomain

Having such an asymmetric architecture in which the meter 1 is likely to export more data than the meter 2 could be interesting when:

- The meter 2 has lower resource than meter 1 for example meter 1 is a high performance meter while meter 2 is a low performance one. This may happen for various reasons such as difference in hardware, software or other factors such as the number of rules under execution as shown in WP4.
- The meter 2 is connected to the correlation server by a low bandwidth link while meter 1 has a large bandwidth link to the correlation server.
- Having an inter-domain measurement: in inter-domain it is certainly realistic to think that a carrier within its network will dedicate more bandwidth resource to its own measurement than the bandwidth offered to the other peered carriers. This concept is illustrated in the Figure 4-5 where the B1 the bandwidth used to export packet information at meter 1 is expected to be higher than B2, the bandwidth used to export packet information at the meter 2. In this case the asymmetry is interesting as the ISP 2 dedicates limited resource to the ISP1 (in term of bandwidth and computational resource) to carry out the measurement requested by the ISP 1. The latter scenario may change if ISP2 charges some fee to ISP1 for such a measurement; however, such a discussion is out of the scope of our discussion.
- In case of unidirectional multicast traffic, in which N instances of meter 2 at the receivers have an N-time reduced resource usage. A single meter instance of type 1 is located at the multicast sender.

4.1.3.3 Sampling Algorithms for Asymmetric Two-level Sampling

The asymmetric architecture proposed has the benefit to allow many different sampling techniques for S process, as the synchronization of IP packet sampling is unnecessary. The possible algorithm for S would include the following (defined in [P-Arch]):

- Systematic count-based.

- Systematic time-based.
- Random n-out-of-N.
- Random uniform probabilistic.

Concerning RS, the algorithm of selection should be based on the value of the packet identifier. In our case it will be based on the CRC32 value computed from the packet content. The simplest way to have a synchronization of the RS would be here to have the same deterministic RS algorithm at all the measurement points (e.g. RS1 = RS2) based on CRC32 value. This could be done for example by assuming that CRC32 output is a pseudo random variable distributed uniformly over [0,2^32-1], the report selection would be then based on the comparison between CRC32 value obtained for the packet ID and a limit value included in the interval depending on the selection probability required.

It has to be mentioned that the two-level sampling could clearly reduce some computational resource and bandwidth usage however some unwanted results might also appear. The figure below shows different scenarios of two-level sampling the columns represent the different output of each meter process in term of packets (P1, P2, P3, P4) or in term of packet identifiers (ID(P1), ID(P2), ID(P3), ID(P4)). The results are the following:

- Case 1: the output of RS is exactly the same for meter 1 and meter 2 meaning that the packet sampling S at the meter 2 eliminated only packet information that would have been eliminated at RS level. This allows proper packet correlation but this is likely to be a rare case as there is no special reason for RS and S to be correlated in such a way.
- Case 2: the samples output from RS at meter 2 are a subset of the samples collected at meter 1 then the delay is going to be computed on those samples. This is likely to be a common case with the samples of meter 2 being a subset of meter 1.
- Case 3: the output of meter 2 is empty. Here the samples are depleted so no correlation is possible. This is obviously a case to avoid. RS and S should be selected and tuned in a way to have enough samples available for correlation.

The presented scenarios show that one should be careful while applying such a sampling architecture. Precision issue put aside, the applicability really depends on the traffic under observation: while traffic with high rate of packets such as video could benefit from the sampling the situation could be very different for low packet rate flow where sampled packets could be too few for a measurement.

Case	_	Case 1	Case 2	Case 3
Observation point	Meter 1	Meter 2	Meter 2	Meter 2
Output RS (same selection at meter 1 and meter 2)	ID(P1), D(P4)	ID(P1), D(P4)	ID(P1)	Empty
Output of PIDG	ID(P1), ID(P2), ID(P3), ID(P4)	ID(P1), ID(P3), ID(P4)	ID(P1), ID(P3)	ID(P2), ID(P3)
Output of S	No sampling	P1, P3, P4	P1, P3	P2, P3
Output of M	P1, P2, P3, P4	P1, P2, P3, P4	P1, P2, P3, P4	P1, P2, P3, P4
comments		Ideal case: output of meter 1 and meter 2 are fully correlated	Standard: Meter 2 output is a subset of meter 1 output	Depletion case: no correlation is possible

Figure 4-6: Two-level Sampling Scenarios

4.1.3.4 Simulation for Asymmetric Two-level Sampling

In order to have some estimation about this two-level sampling and delay measurement we performed some simulation based on two trace files concerning a video streaming. Those two files are composed of 4206 and 4242 packet information including timestamp (those files contains information about 4203 packets in common). On those traces we performed some sampling and packet matching isolating a subset of packets and calculating a one-way delay for the streaming packets. The original stream under observation presented a mean delay of 275.2 ms and a standard deviation of 50 ms. Actually it has to be noted that 50 ms in standard deviation is a very large value reflecting a high dispersion in the delay value. This would impact any sampling negatively.

In the simulation for the packet sampling we simply use a systematic sampling (count based). For the report sampling we use the same algorithm on both trace (algorithm detailed in the following text). The results are presented in the next figures.

These figures should be read as follows:

- "S=x" means that the packet sampling selects one packet out of x.
- "RS=x" means that in average the number of reports selected is in average x times less than the number of input reports. This is done by assuming that CRC32 output is a pseudo random variable distributed uniformly over [0, 2^32-1], the packet selection is then based on the comparison between CRC32 value and 2^32/x. When the CRC32 value is less than 2^32/x the report is admitted. This algorithm is deterministic and allows synchronization at each measurement point.
- "Mean" is the mean value obtained finally over all the selected samples.
- "x/z" represents the numbers of samples that are finally obtained (and exported) for delay computation. More precisely:
  - x is the number of samples obtained after packet sampling and report sampling on the first file simulating the output of a meter

- y is the number of samples obtained after report sampling on the second file simulating the output of the other meter.

	RS=1	RS=2	RS=10
S=1	Mean=275.2 ms 4206/4242	Mean=276.4 ms 2077/2098	Mean=277.1 ms 416/433
S=2	Mean=274.9 ms 2103/4242	Mean=277.9 ms 1053/2098	Mean=282.4 ms 202/433
S=10	Mean=275.5 ms 421/4242	Mean=283.6 ms 207/2098	Mean=292.3 ms 34/433

Figure 4-7: Two level Sampling simulation 1

Comments:

- The case where S=1 and RS=1 represents the regular case where no sampling is used at all and a consequence the maximum precision and granularity that can be obtained.
- When S=1 no packet sampling is used but only report sampling, then the number of packet information that needs to be exported can be decreased at both measurement points at the price of a loss of precision of 1.9 ms in average for RS=10 where the sample number is a tenth of the original number packets. The problem of using RS only as said before would be the unchanged number of CRC32 computation. Export bandwidth is saved by a factor 10 at both meter for RS=10. However no computation resource is saved at the meter in addition to the precision loss.
- When RS=1 no report sampling is used but only packet sampling (in an asymmetric way). The precision of the mean value seems better than for the RS only schemes. But the problem of using only this asymmetric packet sampling is that it saves resource only at one measurement device for example for S=10 the number of samples exported and used is still 4242 for the second file (same as the original number of packets).
- When combining both packet sampling and report sampling (S>1, RS>1) there is a noticeable performance degradation that is greater than compared to the usage of RS only (which also reduce data at both meter). This can be seen by comparing (S=1,RS=10) to (S=2,RS=2) for example.

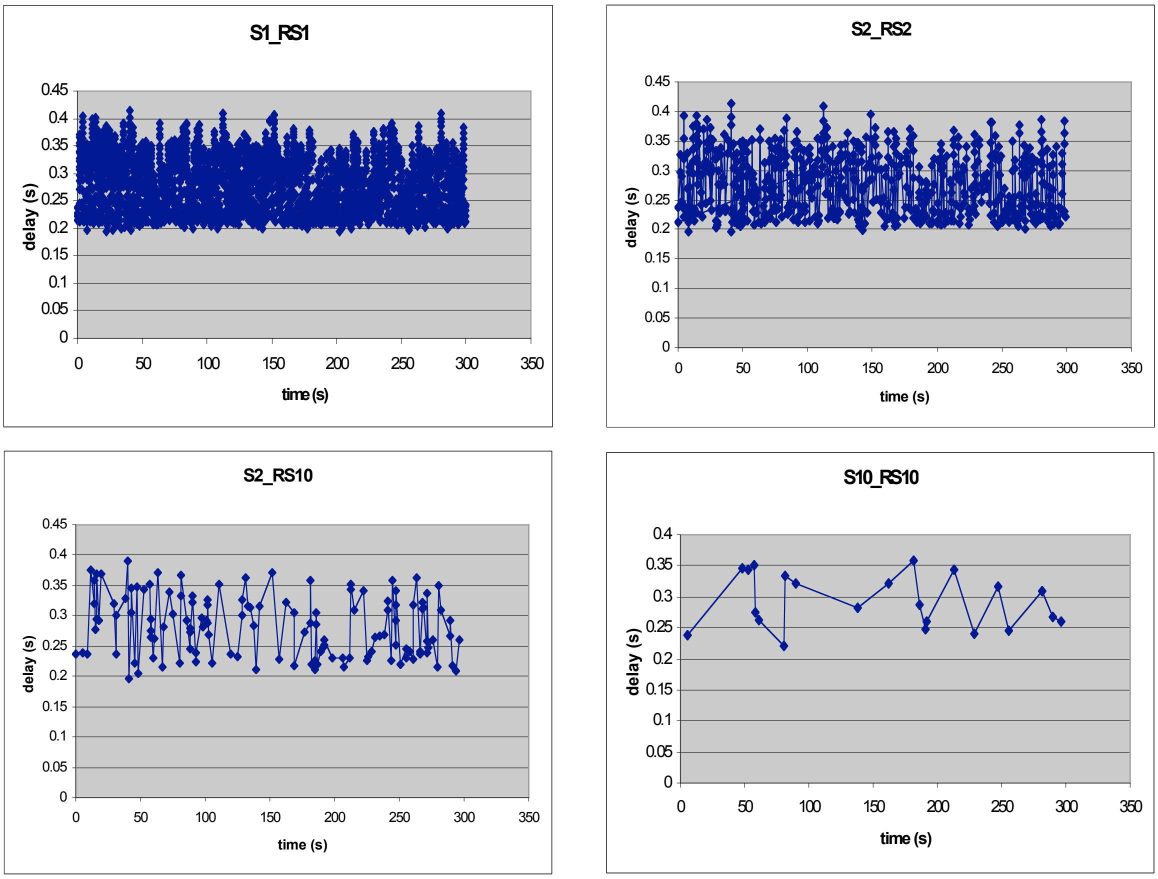


Figure 4-8: Two level sampling simulation result 1

The comparison in mean delays showed a serious precision degradation while combining report sampling and packet sampling. In order to improve the obtained results we performed another set of simulation for (S=2, RS=2) with a random sampling instead of a systematic sampling. We performed a set of ten tests with the following results:

- 1046 samples selected on the first file after S and RS on average.
- 2098 samples selected on the second file after RS.
- Average mean delay: 276.0 ms.

Given the fact that the mean delay is 275.2 ms for the original flow, and that with systematic sampling we obtained a mean of 277.9 ms previously there is a great improvement with the random sampling. This means that here the loss in precision is less than the millisecond for:

- A packet ID computation resource divided by 2 at the meter creating the first trace file.
- A bandwidth requirement divided by 4 at the meter creating the first trace file.
- A bandwidth requirement divided by 2 at the meter creating the second trace file.

As mentioned before the flow under observation has 50 ms in standard deviation which is a large value reflecting a high dispersion in the delay values. In order to have better understanding of those samplings we use a second set of files captured during former experiment containing a flow of packets with only 0.143 ms in delay standard deviation. We performed the same experiments as the ones leading to the results shown in the next figures (i.e. with systematic sampling). The results are presented in below.

As expected the precision obtained is much better while making a sampling of a low deviation flow of packets. For example for (S=2, RS=2) the error obtained was 2.7 ms in a systematic sampling while it is now 0.010ms.

In addition to the results presented in the table, we performed some experiment using random sampling for S and we could even reach a mean delay of 141.916 ms (mean over 10 experiments) which means almost no loss in precision. Another interesting result was obtained for (R=10, RS=10) and random sampling we obtained a mean of delay of 141.909 ms (i.e. 0.008 ms error) which can be considered very low given the high reduction of data.

	RS=1	**RS=2**	**RS=10**
S=1	Mean=141.917 ms 4000/4000	Mean=141.916 ms 1975/1986	Mean=141.906 ms 395/384
S=2	Mean=141.915 ms 2000/4000	Mean=141.907 ms 1007/1986	Mean=141.899 ms 203/384
S=10	Mean=141.916 ms 400/4000	Mean=141.905 ms 213/1986	Mean=141.904 ms 49/384

Figure 4-9: Two Level Sampling Simulation 2

The conclusion is that the sampling is really interesting in order to decrease the amount of data to be processed however one should be careful concerning the selection of the scheme and parameters to be set. This selection is really dependant on:

- The kind of constraint that exist in the system (bandwidth to export data, computation resource at each meter).
- The trade off on precision that the system user is willing to make. Indeed if the system user just wants to have an approximate estimation of delay trend heavy sampling may be enough.
- The deviation of the flow under observation as seen before low delay deviation flow can be sampled very highly with a relatively low loss in precision.
- The packet rate of the flow under observation (high packet rate flows are more suitable for sampling).

4.1.3.5 Revised Meter Structure

The structure of the passive meter has been introduced in the deliverable D3.2. The Figure 4-10 presents the extension of the meter structure to allow the proposed two-level sampling integration as follow:

- A “packet sampling” module is inserted between the “filtering/classifying” module and the “packet processor”.
- A “report sampling” module is inserted between the “packet processor” module and the “measurement store” module. This module is inserted before the storage module to save some storage resource at the device.
- Moreover some modifications are required on the control part to be able to activate, disable or configure each sampling module independently and per measurement task executed at the meter.

- An alternative location for packet sampling is before classification and filtering this way releasing computational power of the “filtering/classifying” module. This realization will override sampling per measurement task located after filtering.

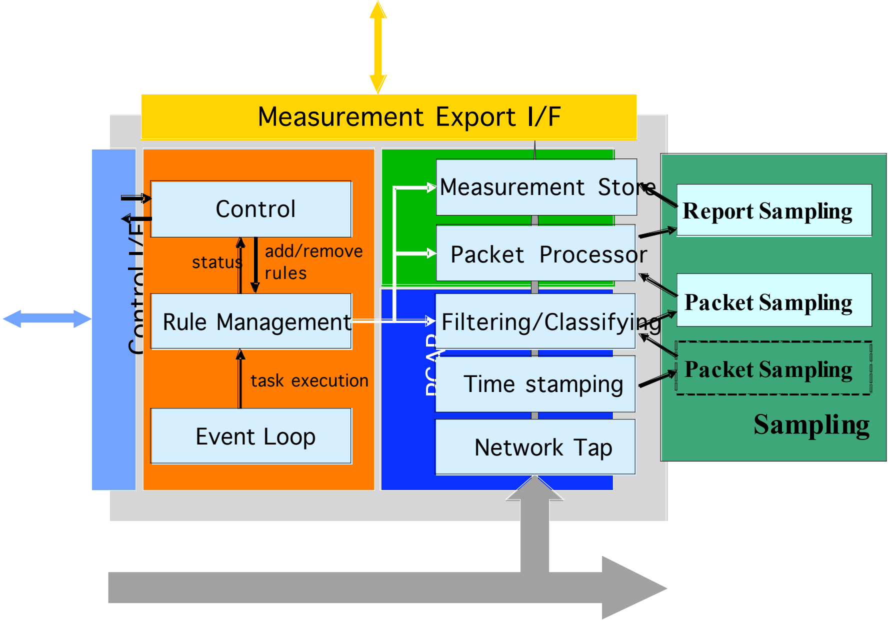


Figure 4-10: Two-level Sampling Extension

4.1.3.6 Automatic selection of sampling

In a fictive scenario the metering system could decide based on the packet rate whether measurement uses sampling or not. The packet rate for a measurement task is constantly examined. When packet rate exceeds a threshold, the system starts sampling in order to reduce resource usage. Figure 4-11 shows that sampling is activated when packet rate is raised above an upper boundary and is deactivated when packet rate decreases below a lower limit. The activation and deactivation events ought to be logged and marked in measurement results in order to make sure results will be interpreted justly.

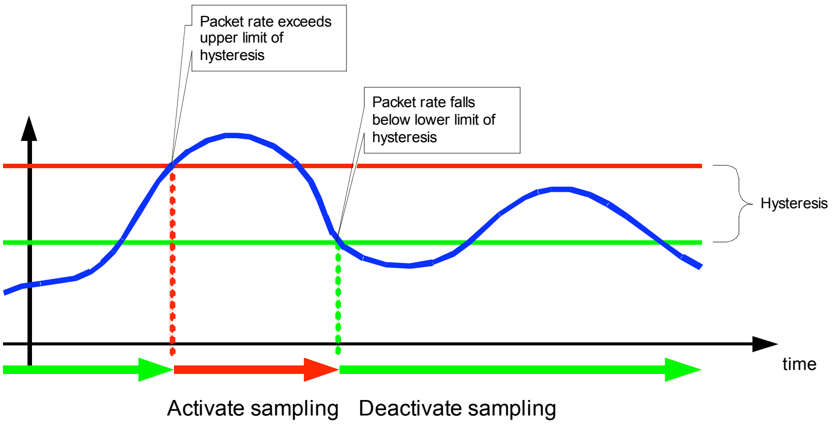


Figure 4-11: Automatic selection of sampling

As a conclusion several possibilities exist concerning the application of sampling within 6QM measurement architecture however the scheme to be selected for a measurement should be adapted to the deployment environment constraints and to the flow under measurement with a special concern for a low packet rate flows and high delay variation flows.

4.2 Decentralized Architecture

The scalability of the system can be addressed by having a very distributed architecture in which the components (meter, collector, calculator) are decoupled and multiple. Indeed in a large network with many meters having multiple collectors and calculators seems very beneficial to avoid a single failure point problem or distribute the workload per component.

Moreover a very specific impact of component distribution on our system is the optimization of the data flow that exists in the system. Indeed passive measurement creates a large amount of data when creating time stamps and packet identifiers, then this data needs to be sent over the network over a more or less long distance depending on the flow under observation and the location of measurement components.

As shown on Figure 4-12, the measurement data flow can be optimized to reduce bandwidth usage in the case of local user flow when using a highly distributed system. The network is divided in several logical areas with its own collector and calculator. When the local flow is measured in this area all the operations are processed within the area avoiding to send large amount of data over large distance.

For the one-way delay the meter sends per packet information, which can result in a large amount of data for this reason it looks interesting to reduce transfer distance. One key point of this architecture is to have a calculator per area. The calculator computes the metrics from the time stamp information then the resulting metric can be easily aggregated into values such as mean for example unlike the output of the meter which can not be aggregated in such a compact manner before time stamp correlation.

Finally in the case of local delay measurement aggregated results only could be exported from the calculator of each area to the final storage. This could avoid the transit of large amount of data over long distance in the case of large area networks.

In this highly distributed architecture there is a need:

- To allow a good flexibility in the data flow configuration
- To coordinate each component data flow in order to have a proper correlation and processing of the data.

In order to address the collection and correlation scalability the Figure 4-13 pictures a hierarchical representation of the measurement components including the meters, the Collectors, the Calculators and the final storage component in relation with the necessary information from the control plane.

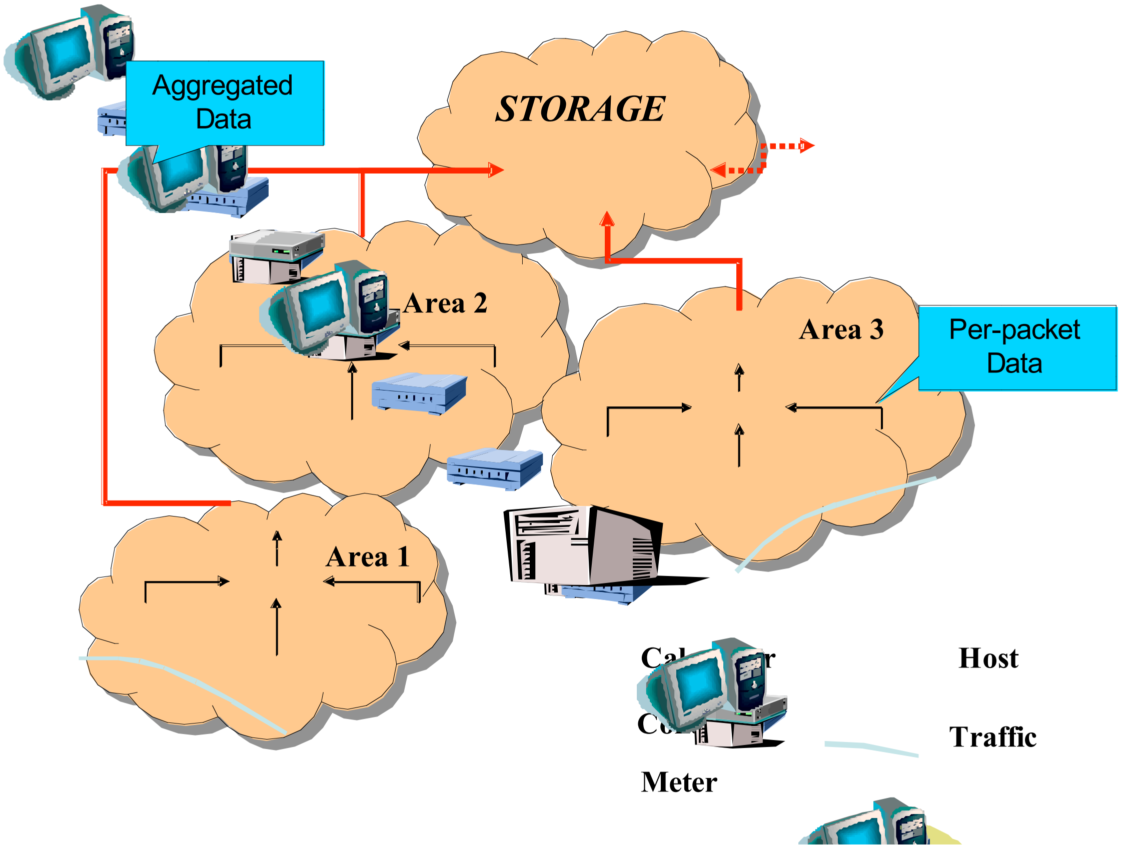


Figure 4-12: Distributed System for Large Network

Here we propose a simple way to coordinate the data flow:

- Each task is identified by a task identifier (Task ID). That identifier must define uniquely a measurement task within the whole system at a given time.
- The control plane sends to the each meters the location of the collector component for each measurement task (identified by a Task ID).
- Each collector involved in the measurement also receives the Calculator location per measurement Task ID. The constraint here is to have all the data concerning the same task gathered at the same Calculator for correlation.
- Moreover it has to be noted that a collector should also be able to send measurement data to the final storage directly too as some measurement such as 1-point volume measurement does not need to be correlated with another measurement.
- Finally the Calculator also gets the location of the final storage server.

Concerning the selection of the Collector or Calculator several selection policies could be used:

- Manual selection: this is clearly the simplest, the measurement system user (i.e. network administrator) chooses which component he desires to use for his measurement.
- Load based: the choice of the component could be made automatically based on the current or estimated load of the components. This could involve the CPU usage of the component or the number of task it has to perform for example. The purpose would be to avoid the overload of a single Collector or a single Calculator when some distribution of the workload would be possible.
- Location based: the choice of the component could be made automatically based on topology information. For example the collector could be chosen as close as possible to

the meter to reduce the impact on the network of exporting data from the meter to the Collector.

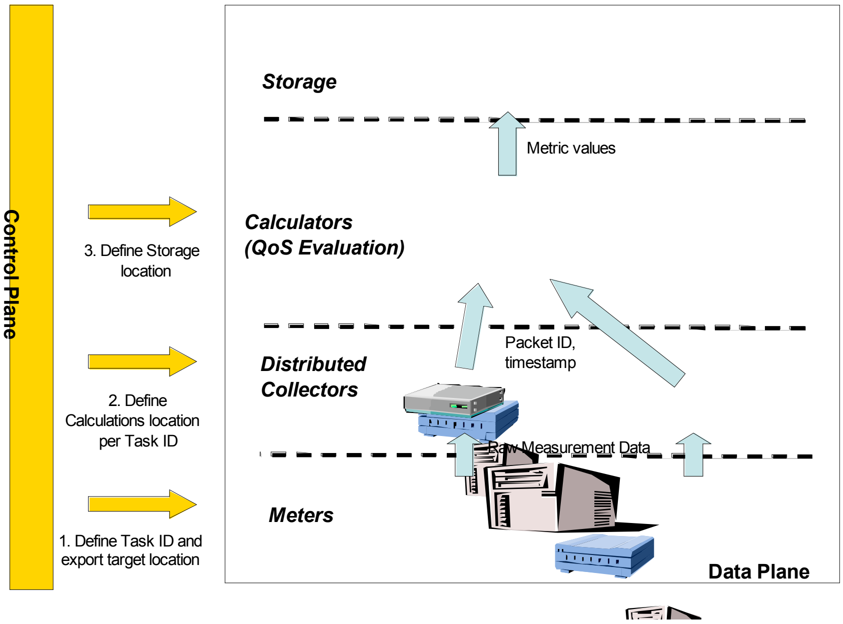

Control Plane
3. Define Storage location
2. Define Calculations location per Task ID
1. Define Task ID and export target location
Storage
Metric values
Calculators (QoS Evaluation)
Packet ID, timestamp
Distributed Collectors
Raw Measurement Data
Meters
Data Plane

Figure 4-13: Design for Scalable Data Collection

It has to be noted that this meter selection could be static or dynamic. In the latter case, the evolution of the system would be monitor in order to select the components or change their selection.

4.3 Data Storage Policy

The amount of data collected can increase very quickly especially in the case of passive measurement for quality of service measurement. Basically our system gathers all the packet information at the central server and correlates the packet information including time stamps in order to produce fine granularity results files such as the one-way delay for each matched packets. From those fine-grained results some aggregation are performed such as the mean or the variance of such a metric.

For long term measurement with many measurement points storing all this information is becoming difficult or very costly. In order to rationalize the data storage we need to assess the value of the data produced by our system with the time scope. For this we define the time scope as:

- Short-term for day or week duration.
- Mid-term for week duration.
- Long-term for weeks, months, year duration.

The figure below presents an exemplary of data value management over the time:

- The raw packet information is critical in a short-term period, as the correlation process needs to process it in order to compute the packet level metrics. After such a processing such a raw data is likely to become useless later on.
- The packet level metric is also critical in short-term as typically it is used to compute aggregated metric, analyze metric value distribution or having a fine grained information

in case of troubleshooting. This information may still have some value in mid-term for example to compare the evaluation of the metric value distribution but it really depends on the usage scenario. In some cases, such information may not be used in mid-term or long-term if only performance trend is desired.

- The aggregated metric is very likely to be valuable on the long term as long-term trends are usually information of interest. However, there is again the question for a higher degree of aggregation that might be applicable.

	Short-term	**Mid-term**	**Long-term**
Raw per packet information (i.e. packet timestamps)	High	Very low /none	None
Metrics at per-packet level	High	Medium /Low /none	Low /none
Aggregated metrics (i.e. mean delay)	High	High	High

Figure 4-14: Data Value

To reduce the final storage space requirements, data aggregation for the data to be stored on the final storage should be applied. The proposed approach to the data storage management is to divide the storage in several logical sections according to the lifetime of the data in the storage system. Several categories would exist.

The measurement data would all be collected in a short-term storage then a Storage Agent would be in charge of feeding the mid-term storage with aggregated data from the short-term storage, feeding the long-term storage with aggregated data from the mid-term storage and finally erasing outdated data from short-term and mid-term storage. By proceeding this way it would be possible to reduce storage requirement while keeping information about the trend of the performance.

This concept is similar to some concept developed in RRDtool [RRDt]. Actually one easy solution to have long term trend analysis would be to avoid the final file based results and instead only use RRDtool for storage.

Moreover to reduce data storage, erasing unnecessary intermediate data should be applied. Typically after the correlation of timestamps and packet identifiers the system should delete the intermediate files and keep only result metrics. Indeed during real measurement the storage required for those files tends to increase quickly.

Such an analysis on the data life cycle is really depending on the system user requirement and several storage options should be enabled and configurable.

5. ADDITIONAL METRICS

5.1 Available Bandwidth Estimation

This sub-section deals with the measurement of the available bandwidth as another interesting QoS metric.

[pathload] provides a precise definition of the end-to-end capacity and available bandwidth.

- The capacity of a path is defined as: "the maximum rate that the path can provide to a flow, when there is no other traffic".
- The end-to-end available bandwidth is defined as: "the maximum rate that the path can provide to a flow, without reducing the rate of the rest of the traffic".

This project especially focused on the delay measurement however the available bandwidth would also be a useful piece of information for a more advanced system. It could be used for:

- Some capacity planning by a network operator.
- Estimating the bandwidth that would be available for an application on a given link (e.g. "I want to use as video streaming service at 155kbps do I have enough bandwidth available?").
- Evaluating the bandwidth of the connection provided by the service provider (e.g. "what is the real bandwidth of my ADSL connection?").

There exist several tools on the market. We investigated:

- Pathload [pathload].
- PathChirp [pathChirp].

Pathload is using the concept of the Self-Loading Periodic Streams (SLoPS) [pathload]. In SLoPS, the assumption is the following: some packets of constant size are sent at constant rate R; if the R is higher than the available bandwidth then the delay between packets at the receiving point will show an increasing trend because of the queuing delay. Based on this concept Pathload sends several streams at different rate (constant rate for each stream) to estimate the available bandwidth in an iterative way. However according to [pathChirp] Pathload presents some problem of convergence delay.

PathChirp approach is different in the sense that within one stream several rates are probed at the same time using chirp probing trains [pathChirp] that are exponentially spaced packets. According to [pathChirp] this approach could be more efficient than Pathload's approach as it requires only a light load to perform the probing while keeping an accurate estimation.

Both Pathload and PathChirp source code are available.

After confirmation with the authors both software do not support IPv6 and there is no plan for the porting in immediate future.

The integration of such a tool could be interesting for future development. It would require:

- Some porting to IPv6.
- Some integration of a new module in the meter component.
- Some light enhancement of the communication interface to configure the new module.

- Some integration for the collection and storage process and GUI.

It has to be noted that one concern of such a function is the precision. This topic would need further investigations.

5.2 Top N One-Way Delays

In accounting applications for network usage often a feature called *Top Talkers* is mentioned in their feature list. We introduce here something quite similar for obtaining a high score for one-way delay. The method is derived from trajectory sampling [DuGr00]. Referring to Figure 5-1 we explain this in detail: we have different networks that are connected an exchange data between computers therein. The communication links are color-coded based on source and destination of packets. For instance, host 2 communicates with host 3 as well as with host 4. Furthermore, there are three observation points A, B and C, each for one network. The probes at these observation points export for inspected packets the packet identifier, time stamp. In addition, at observation point C source and destination IP addresses are exported together with the packet identifiers. The one-way delay is calculated by matching the packet identifiers exported by A and C, respectively exported by B and C. This is nothing new so far as this is the standard procedure for determining one-way delay. However, at observation point C much more information can be derived from the additionally exported properties of the packets. Based on source and destination address that are correlated to packet identifiers separate analysis of the traffic mix at C can be applied on the delay data. By classifying the traffic mix by source and destination of packets, we can specify the top one-way delays between individual hosts. As an extension to this scenario flow labels, traffic class, et cetera might be exported to enable further information.

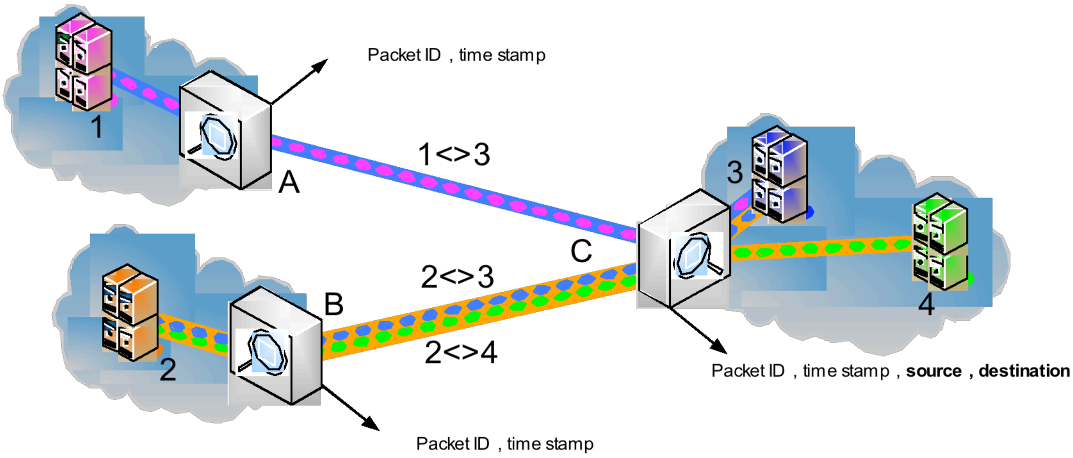


Figure 5-1: Top one-way delays

6. PROTOTYPE EXTENSION

6.1 Integration of Inter-domain Measurement Capabilities

As previously mentioned, the inter-domain communication process is carried out by a component called (inter-domain) controller and another one called collector. Figure 6-1 shows how the inter-domain controller can be integrated in the 6QM architecture and with which components it interacts. Figure 6-2 sketches the proposal on how to integrate the collector. The controller is responsible for the configuration of inter-domain measurements; firstly, it needs to be able to communicate securely with other domains, and secondly it needs to communicate with the proper Measurement Manager to redirect the configuration instruction related to the local domain. Moreover, it is the controller to decide where the results have to be sent to and has to inform the collector properly.

When talking about inter-domain communications, security aspect become very important, therefore, they have been treated separately in section 2.6 "Security considerations".

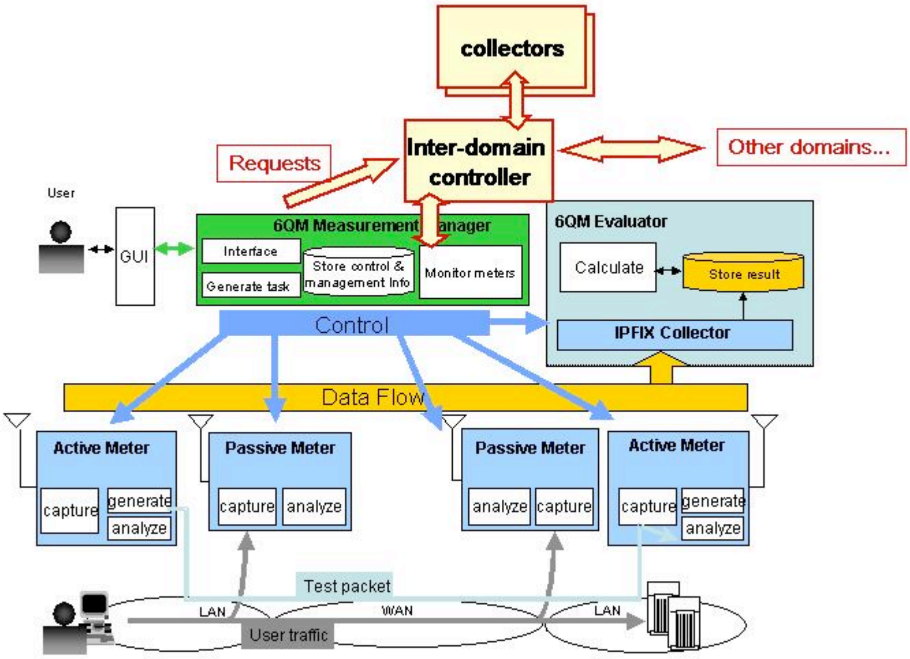


Figure 6-1: 6QM Functional architecture and inter-domain components (controller)

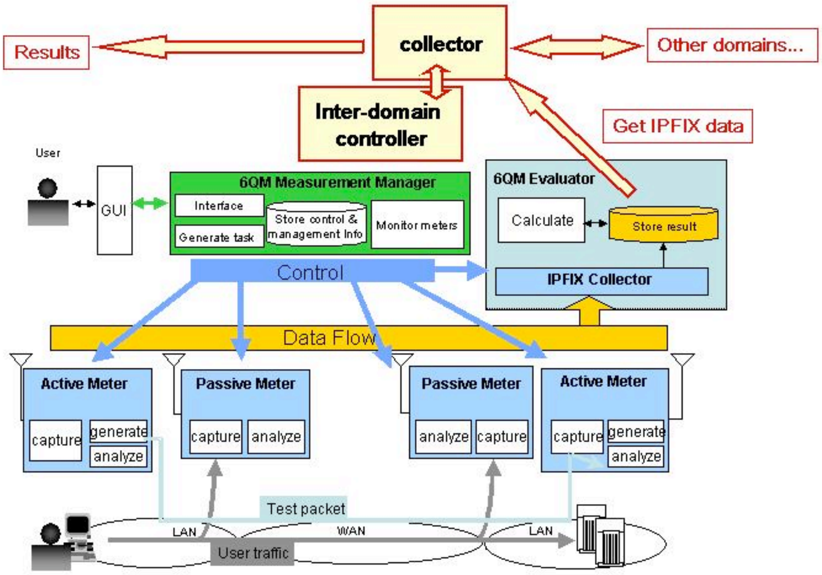


Figure 6-2: 6QM Functional architecture and inter-domain components (collector)

A basic structure for the controller is presented in Figure 6-3.

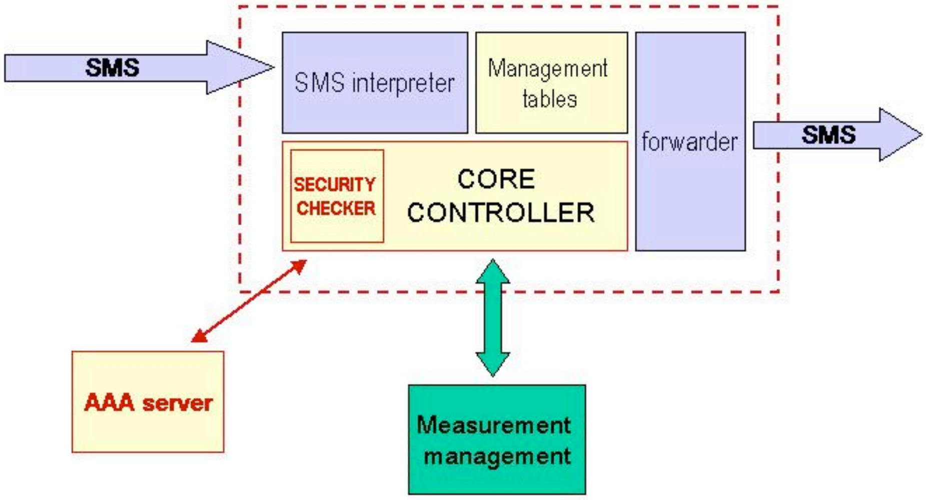


Figure 6-3: Basic structure of the intra-domain controller

The basic functions that the controller must be able to perform are the following:

- Receive SMS messages and generate the intra-domain and extra-domain SMSs out of them.
- Translate intra-domain SMSs into the specific meter configuration language.

- Use BGP information to find out the next step on the path to the destination and forward the extra-domain SMS to the next controller.
- Perform all the security (authorization, authentication…) checks. Eventually with the help of a AAA server
- Be aware of which meters are in its *competence area*[1] and choose the appropriate meter for the measurement. This could be done by requesting the information (list of meters) to the Measurement Manager. For each meter should be provided localization information.
- Send the SMS message further on the path to its final destination (to the next controller) using BGP information. BGP helps in finding the intermediate steps toward a particular destination. If the list of controllers per domain is known (and we can make this assumption and say it was completed during the agreement phase), the AS number will provide us with the address of the next controller[2].

The results of the measurement are stored locally in each domain in a Database inside a module called Evaluator. The collector must be able to retrieve those data, eventually further aggregate them, compose them with the data arriving from other domains and either transfer them further or display the results. As already mentioned, there are several collectors for each domain. The collector and controller need therefore to communicate; in particular, once the collector has been chosen to collect and export the data related to a particular SMS ID, it has to receive the SMS containing the details of the measurement. The collector basic structure is shown in Figure 6-4.

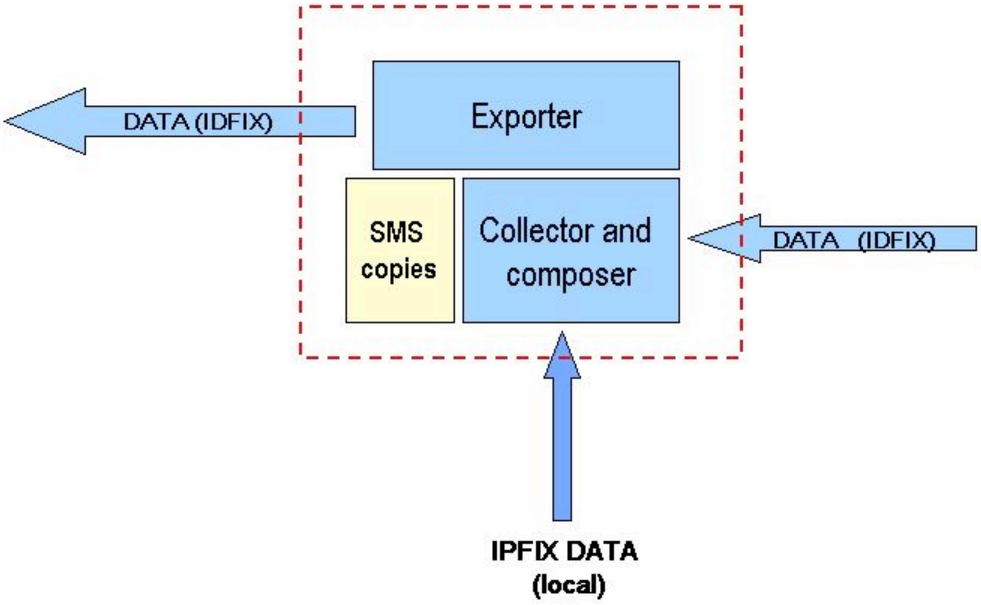


Figure 6-4: Basic structure of the collector

The basic functionalities of the collector are:

- Retrieve IPFIX data from the evaluator, i.e. from the local domain.

[1] *The term "Competence area" corresponds to "domain" if there is only one controller per domain. In case more controllers are allowed, the domain will be divided into competence areas. (still ongoing work)*

[2] *in case there's just one per domain*

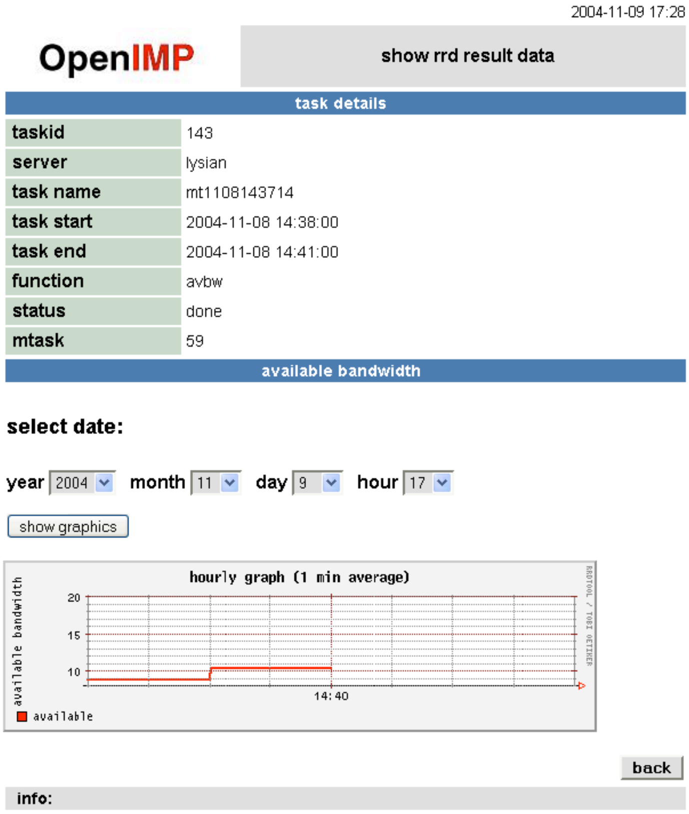
- Translate it into IDFIX and compose it together with other IDFIX data arriving from other domains.
- Eventually aggregate data.
- Export it further to the final destination.
- Provide the final results if the process is complete (i.e. the domain is the endpoint). This means composing the results and visualizing them with the GUI.

6.2 Integration of PathChirp

In order to measure (or estimate) available bandwidth we chose the tool *pathChirp* to be included into the measurement platform. A short introduction of this tool is given in 5.1 "Available Bandwidth Estimation". Initially this tool did not support IPv6. After modification of its source code within 6QM project, this tool now supports IPv6. Now the meters of the measurement platform do support the action for available bandwidth estimation. If pathChirp is available to a meter the capability to conduct available bandwidth measurements is reported to the measurement system.

Since pathChirp uses active probing for estimating the available bandwidth, sender and receiver addresses have to be configured when setting up a measurement along with additional parameters as average probing rate and packet size. A measurement period can be configured so that measurement results are periodically transferred to the collector. The results can be stored in a round-robin database. This allows for convenient examination of long term probing.

Support for configuring those measurements has been added to the Graphical User interface. The Figure 6-5 shows the probing result of available bandwidth estimation as presented.



2004-11-09 17:28

OpenIMP

show rrd result data

task details

taskid	143
server	lysian
task name	mt1108143714
task start	2004-11-08 14:38:00
task end	2004-11-08 14:41:00
function	avbw
status	done
mtask	59

available bandwidth

select date:

year 2004 month 11 day 9 hour 17

info:

Figure 6-5: Result of Available Bandwidth Estimation

6.3 Security

Based on the proposed updates concerning enhancements to prevent unsolicited access to components of the measurement system or eavesdropping of sensitive data, Figure 6-6 shows the protected communication between system components. As presented, it is suggested to use protocol IPsec in order to protect external communication with the components of the measurement controller instance. Likewise, the security is hardened with IPsec for the control and management message exchange and the exchange of measurement data. Authentication of a system user against the measurement controller and protection of the communication in between them is done via HTTPS that relies on SSL. Moreover the probe can also benefit from a firewall protection.

It has to be noted that the security schemes expressed on the figure are dependant on the type of deployment that is done as firewall, HTTPS, IPSec or SSH are not comprised in the core measurement system.

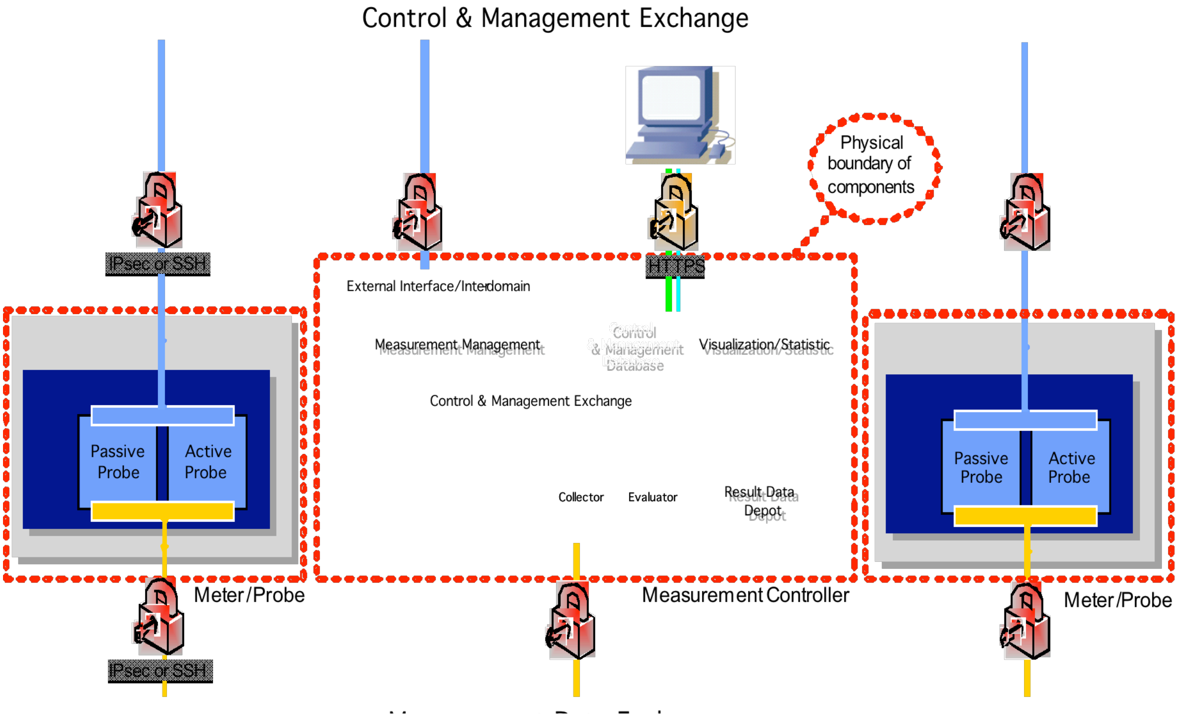


Figure 6-6: Secured system components

7. Requirement and Specification Analysis WP2/WP3

After we revised the specification for the 6QM measurement in this document, it is necessary to evaluate again, which of the requirements disposed in Work Package 2 Deliverable D2.2 have been met by the updated prototype. Please note that this is an update of the initial analysis given in a previous deliverable D3.1. Moreover the security analysis presented in D2.5 is also added in this section.

As in the foregoing analysis, we chose a tabular form. Numbering of requirements (RID) in the tables below is in accordance to Version v2.3 of deliverable D2.2. For convenience, we listed the information given already in D3.1 and we use the same symbols as in D3.1, which are the following:

- An "X" means that the requirement is addressed in the prototype or is at least conceptually investigated and planned to be supported.
- A "~" means that the requirement will partially be addressed in the prototype.
- An empty case means that the prototype does not consider this requirement.

The information given in the tables below is to be interpreted by considering the following scheme:

Information provided in D3.1		
	X	requirement is addressed by prototype
	~	requirement is partly addressed by prototype
		(empty case) requirement is not addressed by prototype
Updated information (not contained in D3.1)		
	X	requirement is addressed by prototype (and revised specification)
	☒	requirement is addressed by revised specification
	~	requirement is partly addressed by prototype
	☞	requirement is partly addressed by revised specification
		(empty cell) requirement is neither addressed by revised specification nor in prototype

7.1 Requirements for Measurement Points

Type of requirement	RID	Requirement	Level of requirement	Status
Container	PM1.1	Ability to contain one or more measurement entities.	Must	X
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	X
Configuration	PM1.3	A Point of Measure must be able to setup measurements when requested to do so.	Must	X
Setup	PM1.4	A point of measure must be able to remove or stop measurements when requested to do so.	Must	X
Reporting	PM1.5	A point of measure must be able to export measurement data	Must	X

Figure 7-1: Requirements for Measurement Points

7.2 General Requirements for Measurements

Type of requirement	RID	Requirement	Level of requirement	Status
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	X
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	X
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	X
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	X
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	☒
IP Configuration	M1.6	A measure may have an option that specifies whether IPv4 or IPv6 traffic is being measured	May	X
TOS Configuration	M1.7	A measure may have an option to override the TOS bit in an IPv4 packet with another value	May	
Flow Label Configuration	M1.8	A measure may have an option to specify the Flow Label Field contents for an IPv6 packet.	May	X

Figure 7-2: General Requirements for Measurements

7.3 Requirements for Passive Measurements

Type of requirement	RID	Requirement	Level of requirement	Status
Measurement Operations Traffic Copy	MI1.0	Ability to perform packet capturing in order to obtain a copy of the traffic without introducing modifications in the original traffic.	Must	X
Measurement Operations-Classification	MI1.1	Ability to classify packet according to IPv4 or IPv6 source address.	Must	X
	MI1.1	Ability to classify packet according to IPv4 or IPv6 destination address.	Must	X
	MI1.2	Ability to classify packets according to IPv4 ToS field content / IPv6 Traffic class	Must	X
	MI1.3	Ability to classify packets according to IPv6 flow label field content.	Must	X
	MI1.4	Ability to classify packets according to the IPv4 Protocol field content / IPv6 Next header field content	Must	X
	MI1.5	Ability to classify packets according to Transport addresses.	Must	X
	MI1.6	Ability to classify packets according to previous packets information within a flow.	Must	
	MI1.7	Ability to classify packets according to BGP information (Destination AS, Source AS).	May	
	MI1.8	Ability to classify tunneled packets (v4 over v6, v6 over v4)	Should	
	MI1.9	Ability to classify packets according to incoming interface.	Should	
	MI1.10	Ability to perform classification operations at line-rate	Should	
	MI1.11	Ability to perform classification operations within fixed duration bounds.	Should	X
	MI1.12	Ability to configure classification process with classification parameters	Must	X
	MI1.13	Ability to perform IPv6 and IPv4 configuration consistently.	Should	X
Measurement Operations-Time-Stamping	MI2.1	Ability to time-stamp the first packet of a flow	Must	X
	MI2.2	Ability to time-stamp the last packet of a flow	Must	X
	MI2.3	Ability to perform time-stamp operations before other operations.	Should	X
	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	X (choice between NTP or GPS depending on meter configuration)
	MI2.6	Ability to indicate time-stamping source as well as time-stamping source characteristics (resolution)	Must	~ (information from NTP)
	MI2.7	Ability to choose time-stamping source if several available	Should	~ (statically by configuration of NTP)
	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	X (performed by a task scheduler)
	MI2.11	Ability to synchronize clocks from a single source.	Must	X (by means of NTP)
	MI2.12	Support several clock synchronization sources	Should	~ (statically by configuration of NTP)
	MI2.13	Support several clock synchronization methods	May	X (GPS/NTP and NTP)
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 count number of packets per flow	Must	X
	MI6.2	Ability to count number of bytes per flow	Must	X
	MI6.3	Ability to determine duration of flow	Must	X
	MI6.4	Ability to classify flows according to their type.	Should	X
	MI6.5	Ability to count packets based on their actual size	Must	X (IP)
	MI6.6	Ability to count IPv6 packets based on the payload length	Must not	
	MI6.7	Ability to handle fragmented packets.	Must	X
	MI6.8	Ability to compute fragmentation rate of flow.	May	
	MI6.9	Ability to measure measurement cost (CPU/memory consumption)	May	X
Measurement operations configuration	MI7.1	Ability to retrieve flow information. This flow information complies with IPFIX requirements. [Qui02]	Must	X
	MI7.2	Ability to provide full packets.	May	X (for accordingly configured captures)
	MI7.3	Ability to perform measurement configuration and to retrieve measurement results remotely.	Must	X
	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	X (addressed by collector)
	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	~(meter status, measurement status)
	MI7.10	Configuration and result retrieval protocol is loss and error resilient	Should	X
	MI7.11	Support several measurement operations in parallel.	Should	X
	MI7.12	Support several measurement requesters.	May	X (addressed at measurement manager)
	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	~ resource consumption is globally reported (not task/measurement based)
	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	X
	MI8.2	The impact of passive measurement operations on existing network devices is negligible.	Should	X
	MI8.3	The impact of traffic measurement configuration on the traffic measured is negligible.	Must	X
	MI8.4	The impact of traffic measurement configuration on existing network devices is negligible	Should	X
	MI8.5	Remote management operations have a negligible effect on existing traffic.	Should	X

Figure 7-3: Requirement Analysis for Passive Measurements

7.4 Requirements for 6QM Measurement Manager

Type of requirement	RID	Requirement	Level of requirement	Status
Measurement operations configuration	I9.1	Flow information complies with IPFIX requirements. [Qui02]	Must	X
	I9.2	Ability to perform active and passive measurement configuration and to retrieve measurement results from measurement devices	Must	X (addressed by combination of Measurement Manager and Collector)
	I9.3	Ability to perform configuration and measurement retrieval through a single interface.	Should	X (addressed by GUI)
	I9.4	Ability to perform measurement operations sequences configuration through the same interface.	Must	
	I9.5	Support several measurement operations in parallel.	Must	X
	I9.6	Support several measurement requesters.	Must	X
	I9.7	Support several requests from several requesters simultaneously.	Must	X
	I9.8	Ability to advertise measurement capacities (measurement points, measurement point capacities)	Should	~ (measurement points are presented in GUI)

	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	~(partly this information is defined upon measurement task creation)
	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	X (collector stores IPFIX export in DB, otherwise collector stores pointer to result files)
	I10.2	Ability to query DB to retrieve past measurement.	Should	X (collector issue)
	I10.3	Ability to combine new and past measurement results (e.g. statistical values) through DB queries	May	X (collector issue)
MP configuration	I11.1	Ability to translate measurement configuration in MP configuration.	Must	X
	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	X (addressed at collector)
	I11.5	Ability to report failure or dysfunction of any component of the system.	Must	X (meter status, measurement status)
	I11.6	Configuration and result retrieval protocol is loss and error resilient.	Should	X

Figure 7-4: Requirement Analysis for 6QM Measurement Manager

7.5 Requirements for 6QM Evaluator and Collector

Type of requirement	RID	Requirement	Level of requirement	Status

Measurement Operations-Time-Stamping	C1.1	The Collector has the ability to check remote time-stamping resolution (Cross Check with other measurement source)	May	
Standardization	C1.2	The Measures indicate if the measurement metrics complies with standards and which standards it complies to.	Must	
	C1.3	The measures indicate if the measurement methodology complies with standards and which standards it complies to.	Should	
	C1.4	The Collector has the ability to indicate that a specific metric is not supported or a specific measurement request is not possible.	Must	X (managed at the Measurement Manager)
Publisher Directory Service	C1.5	The Collector has the ability to advertise measurement capacities (measurement points, measurement point capacities)	Must	X (Measurement Manager provides information)
	C1.6	The Collector has the ability to identify and log measurement requests.	Must	~ (intra-domain requests are logged at Measurement Manager, inter-domain requests at domain controller)
Broker	C1.7	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	
Authentication Service	C1.8	The Collector provides an authentication service to users who are accessing the system	Must	☞ (at the measurement controller)
Access Control Service	C1.9	The Collector provides access control to any client that is attempting to access a service in the QoS measurement system.	Must	☞ (at the measurement controller)
Service Activation	C2.0	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	C2.1	The Collector stores QoS measurements in a persistent repository.	Must	X

Figure 7-5: Requirements for 6QM Evaluator and Collector

7.6 Requirements for Inter-domain measurements

The functionality for inter-domain measurements is provided by dedicated components for inter-domain scenario; these are the domain-controllers and domain-collectors. They act atop of the intra-domain measurement system. The requirements disposed for these inter-domain specific components are listed below. The topic of inter-domain measurements has been worked on during the extension period of the project. Although conceptually the requirements have been regarded the implementation is work-in-progress and therefore subject to change.

Type of requirement	RID	Requirement	Level of requirement	Status
Authentication	C2.2	The collector must be able to authenticate users from foreign domains	Must	
	C2.3	The collector must be able to determine whether a foreign user may setup a measure	Must	
	C2.4	The collector must be able to determine for which metrics a foreign user may setup a measure	Must	
Configuration	C2.5	The Collector has the ability to initiate measurements starting in his home domain and finishing in a foreign domain (cross-domain measurement setup)	Must	X (addressed by domain-controller)
	C2.6	The collector may also setup measures that originate in a foreign domain and terminate in the home domain	Must	
	C2.7	The Collector has the ability to indicate that a specific metric is not supported or a specific measurement request is not possible to a foreign domain	Must	
Publisher Directory Service	C2.8	The Collector has the ability to advertise measurement capacities (measurement points, measurement point capacities) to and from foreign partner domains.	Must	
	C2.9	The Collector has the ability to identify and log measurement requests from foreign domains	Must	☒ (addressed by domain-controller)
Broker	C3.0	The Collector has the ability to find an appropriate service that will satisfy a foreign client's request.	Must	☒ (domain-controller configures probes within its own domain)
Service Activation	C3.1	The Collector provides Service activation functionality for all clients interacting with the QoS Measurement system	Must	
Authentication	C3.2	The collector must be able to determine whether a foreign user may access measurement results	Must	
Time-stamping	C3.3	The collector has the ability to check remote time stamping resolution, in particular between domains	Must	

Configuration	C3.4	The Collector has the ability to receive synchronous measurement results from other domains.	Must	X
	C3.5	The collector has the ability to share measurement results with other domains.	Must	X
Proxy Server	C3.6	The collector has the ability to provide “proxy” measurements to other domains for “n” points of measure	May	
	C3.7	The collector has the ability to request “proxy” measurements from other domains for “n” points of measure	May	
	C3.8	The collector has the ability to export measurements to other domains asynchronously. Periodic flow export/flow beginning-end notification.	Should	
	C3.9	The collector has the ability to receive asynchronous measurement results from other domains	Should	X
Persistent Service	C4.0	The Collector has the ability to store QoS measurement results from foreign domains.	Must	X
Broker	C4.1	The Collector has the ability to find an appropriate service that will satisfy a foreign client’s request for the exchange of measurement results.	Must	☒ (inter-domain controller)
Service Activation	C4.2	The Collector provides Service activation functionality for all foreign clients interacting with the QoS Measurement system	Must	☒ (inter-domain controller)

Figure 7-6: Requirements for Inter-domain measurements

7.7 Requirements for Fault Management

Type of requirement	RID	Requirement	Level of requirement	Status
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 7-7: Requirements for Fault Management

7.8 Requirements for Security Management

Type of requirement	RID	Requirement	Level of requirement	Status
Configuration	S1.1	Ability to configure users of the QoS measurement system	Must	☞
	S1.2	Ability to configure access to individual components of the system, such as points of measure, collectors, etc.	Must	☞ (access to measurement server's functions)
	S1.3	Ability to define access rights for individual users based upon the type of metrics that individual would like to access	Must	☞ (access to measurement server's functions)
	S1.4	Ability to allow authorized users the possibility to setup measures.	Must	☞ (access to measurement server's functions)
	S1.5	Ability of user to authorize other users to view his measures	Should	
	S1.5	Ability to allow authorized users the possibility to read results of measurements that he has setup	Must	
	S1.6	Ability to allow authorized users the possibility to read results of measurements created by another user	Should	
	S1.7	Ability to configure who should receive measurement results sent as notifications, and to where they should be sent.	Should	
Notification	S1.8	The ability to send a notification event when a security violation takes place	Should	

Figure 7-8: Requirements for Security Management

7.9 Security requirement (D2.5)

Type of requirement	Req. ID	Requirement	Level of requirement	Status
Measurement points/ Points of measure	A1.1	Protection against DoS attacks and in particular flooding attacks.	Must	
	A1.2	Authentication of the signaling to control a Measurement Point.	Must	X (if IPsec)
Measure	A2.1	Authentication of the results sent to the collector.	Should	X
	A2.2	Authentication of the copied packets.	Should	X
	A2.3	Ciphering of the copied packets.	May	X
Collector	A3.1	Protection against DoS attacks and in particular flooding attacks.	Must	
	A3.2	Authentication of the signaling to control a collector.	Must	X (if IPsec)
Management	A4.1	Authentication of the signaling to control a collector.	Must	X (if IPsec)
	A4.2	Authentication of the signaling to inform a management entity.	Must	

Figure 7-9: Security Analysis Requirement (D2.5)

7.10 Conclusion of Requirement Analysis

The lists of requirements from above sections allow us to make up a kind of scoreboard to which extend the requirements have been considered in the 6QM prototype. Those tables display the number of requirements that have been addressed fully or at least partially in the prototype.

Level of requirement	Number of requirements disposed	Number of addressed requirements	Percentage of addressed requirements
Must	5	5	100%
Should	0		
May	0		

Figure 7-10: Summary Requirements for Measurement Points

Level of requirement	Number of requirements disposed	Number of addressed requirements	Percentage of addressed requirements
Must	5	4	80%
Should	0		
May	3	2	67%

Figure 7-11: Summary Requirements for Measurements

Level of requirement	Number of requirements disposed	Number of addressed requirements	Percentage of addressed requirements
Must	26	21	81%
Should	24	11	46%
May	20	5	25%

Figure 7-12: Summary for Passive Measurements

Level of requirement	Number of requirements disposed	Number of addressed requirements	Percentage of addressed requirements
Must	10	7	70%
Should	9	6	67%
May	2	1	50%

Figure 7-13: Summary Requirements for Measurement Manager

Level of requirement	Number of requirements disposed	Number of addressed requirements	Percentage of addressed requirements
Must	9	4	44%
Should	1	0	0%
May	1	0	0%

Figure 7-14: Summary Requirements Evaluator and Collector

Level of requirement	Number of requirements disposed	Number of addressed requirements	Percentage of addressed requirements
Must	17	4	24%
Should	1	1	100%
May	2	0	0%

Figure 7-15: Summary Requirement for Inter-domain Measurements

Level of requirement	Number of requirements disposed	Number of addressed requirements	Percentage of addressed requirements
Must	2	0	0%
Should	0		
May	0		

Figure 7-16: Summary Requirements Fault Management

Level of requirement	Number of requirements disposed	Number of addressed requirements	Percentage of addressed requirements
Must	5	0	0%
Should	4	0	0%

May	0		

Figure 7-17: Summary Requirements for Security Management

Level of requirement	Number of requirements disposed	Number of addressed requirements	Percentage of addressed requirements
Must	6	3	50%
Should	2	2	100%
May	1	1	100%

Figure 7-18: Summary Requirements for Security Analysis (D2.5)

The evaluation above show up to which degree requirements defined in Work Package 2 have been considered for the prototype of 6QM. The figure below provides a compact table summarizing the different requirement aspects, the degree of compliance of the prototype and also the main points that are missing in the prototype. Moreover it shows the WP2 coverage with the revised specifications (independently from the prototyping).

The WP2 described an ideal system and it was not planned to implement all the features described in WP2 because of the resource available in WP3. Indeed the WP2 requirements were not build with in mind the WP3 resource restrictions and some topics were not originally considered as part of the WP3 technical annex (i.e. Inter-domain).

However, it was considered as an interesting topic as a consequence some extra resource has been allocated to this activity. The challenge was high on this activity and we tried to address the key points that is to say the global architecture and the measurement setup.

Moreover during the project lifetime, new ideas came up with the adaptation of IPFIX for per-packet information, the combination of active and passive measurements, or the multi-point measurement that created an additional shift in WP3 resource allocation as those topics were considered as interesting for the project and the state-of-art in general.

In order to allocate resource to those new activities some restrictions have been made in the original WP2 requirements allowing this shift of resource. During the revised specification period we tried to provide design solutions to answer some of the WP2 main issues which were omitted or weakly addressed by WP3 (i.e. sampling or security).

Category	Percentage of MUST covered in prototype	Percentage of MUST finally covered in the specifications	Prototype compliance comments
Measurement Points	100%	100%	_
Measurements	80%	100%	No measurement ownership - not addressed until revised specifications
Passive Measurements	81%	85%	No sampling - not addressed until revised specifications No pull export implemented

Measurement Manager	70%	70%	No sequence configuration implemented No common results format implemented No pull export implemented
Evaluation and Collector	44%	67%	Collector was implemented as an internal component just in charge of collecting data and not exposed to the user unlike the service broker concept introduced in WP2 but the specifications have been enhanced for inter-domain
Inter-domain	24%	47%	Only core functions are addressed in 6QM
Fault management	0%	0%	Simpler management functions have been implemented to detect task or component failure
Security Management	0%	80%	Originally no measurement ownership and access privileges - not addressed until revised specifications
Security Analysis (D2.5)	50%	50%	No DoS attack prevention

Figure 7-19: Prototype Compliance Summary

8. 6QM SYSTEM AND THE STATE-OF-ART

The document D3.1, first WP3 deliverable, provided a survey of existing systems for QoS. This survey was a major input to define the scope and direction of R&D for WP3. After the development of the specifications and the development of the technology realized in the work package, this section proposes to compare the final system used by 6QM based on OpenIMP and MGEN to competitors.

In order to provide the most up-to-date state this section relies on the work performed by the MOME working group, which released in June 2004 the most complete and up-to-date survey [MOME] known by the consortium. [MOME] identified more than 350 tools in the area and selected 58 of them for a closer review.

First of all, 6QM is dealing with one-way delay measurement as a key metric for QoS as a consequence we first selected the tools allowing IP one-way delay measurement out of the 58 tools introduced by [MOME]. The list of selected tools is presented in the figure below, the information presented is partially selected from [MOME] with the following comments and restrictions:

- When "?" symbol is used it means that the information was not available (neither available from [MOME] or from the tool website).
- Category with "*" mark are available from [MOME].
- MGEN and OpenIMP tools information have not been included as they are used in the 6QM tool.
- The 6QM tool has been added to the table.
- The tool called "Sting" which was presented as enabling delay measurement in [MOME] is not presented, as after verification it does not provide one-way delay measurement.
- The tool called "Iperf" which was presented as enabling delay measurement in [MOME] is not presented as after verification it deals with bandwidth estimation and cannot provide delay for every IPv6 packets.
- QoSmetrix identified in D3.1 as an advanced product has been added to the table and updated with its current status (according to writer's knowledge).

Name	IPv6*	Passive/ Active*	Passive Delay	GUI*	IPFIX*	Inter-domain	Security	Comments
CMToolset	Yes	Active	No	Yes	No	No	Yes	_
D-ITG	No	Active	No	No	No	No	No	_
E2ETT	?	Active	No	Yes	No	?	?	No reference found
eHealth	No	Both	No	Yes	No	?	?	No answer received from the vendor
NetMate	Yes	Passive	Yes (but no analysis)	Yes	No (planed)	No	Yes	_
QoSmetrix	Yes	Both	No	Yes	Yes	?	Yes	_
6QM tool	Yes	Both	Yes	Yes	Yes	Partially	Yes	_

Figure 8-1: MOME tools including one-way delay measurement

Comments:

- From the selection of [MOME] most of the QoS solutions allowing one-way delay are active solutions. Indeed the active measurement is traditionally covered by many existing systems unlike the passive measurement. As a consequence there already is a lot of competitions in active delay measurement.
- Concerning the support of IPv6, several solutions already support it and it is certainly realistic to believe that most of the tools will support IPv6 in a more or less near future.
- Few solutions use IPFIX, actually one can mention that QoSmetrix has evolved from the time of writing D3.1 with additional IPFIX support to report flow information.
- The table shows only a minority of systems with passive one-way delay, except 6QM only NetMate (Network Measurement and Accounting System), seems to support passive one-way delay. We performed further investigation on this tool. NetMate is a measurement probe with a variety of supported passively measured metrics beyond one-way delay. Although this probe calculates packet identifiers, the actual computation for the cases of one-way delay and one-way loss is not done. This meter is currently under further development (Version 0.8). For future development, this meter could be integrated into the 6QM platform. NetMate's implementation of the packet classifier based on Recursive Flow Classification appears particularly advantageous. As a conclusion this tool is just a meter. It does not comprise the whole meter and server infrastructure to perform the desired QoS measurement that provided in 6QM.
- Generally speaking inter-domain is not very developed yet this is certainly due to the lack of clear and standardized framework for such a purpose.
- The table shows only few systems with active and passive techniques.
- Concerning the security, the comparison is difficult because we do not have a clear information about the security provided by the other solutions.

Comparisons:

- In this table, only 6QM provides the complete environment for passive QoS measurement.

- Unlike 6QM tool, for the other tools, the functions in active and passive are disjoint. Typically, delay is estimated by active and flow information or bandwidth usage by passive methods. 6QM can measure QoS actively and passively.
- Except 6QM tool, no other system combining passive and active for continuous monitoring have been identified.
- 6QM provides a prototype inter-domain component.
- It is not mentioned on the table but 6QM is software based not hardware based as QoSmetrix, this will certainly enable to have a cheaper deployment. This issue is especially important when thinking about end-to-end measurements, as the number of measurement points can be high. As a conclusion the target is slightly different for both tools.

9. Summary and Conclusions

The document presented selected topics for prospective developments of the 6QM measurement platform. Among the main points, this document pointed to advisable adaptations to make the 6QM measurement platform useful in an inter-domain scenario. The document also commends to implement data consolidation mechanisms and architectural modification to improve scalability and security.

Moreover some of the identified subjects, such as inter-domain measurement, security and additional metric have been partially addressed in the prototype during the extension period of the 6QM project.

In concluding this summary, we have to note that the original specification presented the base for an extensible system. However, this revised specifications presented new ideas and proposed functions indicating that there is room for development beyond the 6QM project frame. Moreover the deliverable D3.4 addressing the guidelines for further research will provide some other topics that could lead to further study in the field of measurement.

10. References

[BELL] S. Bellovin, “Guidelines for Mandating the Use of IPsec”, IETF internet draft, draft-bellovin-useipsec-03.txt.

[Boschi2004] Elisa Boschi, Salvatore D’Antonio, Giorgio Ventre, «Inter-domain Communication and Data Exchange», In Proceedings of 2nd International Workshop on Inter-Domain Performance and Simulation (IPS 2004), Budapest, Hungary, March 2004.

[diameter] Diameter home page: http://www.diameter.org

[DuGeGr02] Trajectory Engine: A Backend for Trajectory Sampling, N.G. Duffield, A. Gerber, M. Grossglauser, IEEE Network Operations and Management Symposium 2002, Florence, Italy, April 15-19, 2002

[DuGr00] Nick Duffield, Matthias Grossglauser: “Trajectory Sampling for Direct Traffic Observation”, Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, 28th August- 1st September 2000

[Intermon] INTERMON: Advanced architecture for INTER-domain quality of service MONitoring, modelling and visualisation, IST Project IST-2001-34123, http://www.ist-intermon.org/

[IPFIX-ARCH] Sadasivan, G, Brownlee, N. "Architecture Model for IP Flow Information Export" draft-ietf-ipfix-arch-04.txt", October 2004

[IPFIX-AS] Zseby, T, Boschi, E, Penno, R, Brownlee, N, Claise, B, "IPFIX Applicability", draft-ietf-ipfix-as-03.txt, October 2004

[IPFIX-INFO] Calato, P, Meyer, J, Quittek, J, "Information Model for IP Flow Information Export" draft-ietf-ipfix-info-06, October 2004

[IPFIX-PROTO] Claise, B.

[IPFIX-REQ] Quittek, J, Zseby, T, Claise, B, Zander, S,"Requirements for IP Flow Information Export" RFC 3917.

[MOME] Carsten Schmoll et Al., “D11- State of Interoperability”, MOME project Deliverable, 30 June 2004.

[P-Arch] Nick Duffield, “A Framework for Packet Selection and Reporting”, IETF internet draft, draft-ietf-psamp-framework-08.txt.

[pathChirp] Vinay Ribeiro et Al., “pathChirp: Efficient Available Bandwidth Estimation for Network Paths”, `assive and Active Measurement Workshop, 2003.

[pathload] Manish Jain et Al., “End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput

[PoMB04] G. Pohl, L. Mark, E.Boschi, “Use of IPFIX for export of per-packet information“, July 2004, IETF Internet Draft (work in progress), draft-pohl-perpktinfo-00.txt.

[PSP] T. Zseby et Al., “Sampling and Filtering Techniques for IP Packet Selection”, IETF internet draft, draft-ietf-psamp-sample-tech-04.txt.

[RRDt] RRDTool website: http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/

[SSLA] T. Zseby , “Deployment of Sampling Methods for SLA Validation with Non-Intrusive measurements”, Proceedings of Passive and Active Measurement Workshop (PAM 2002), Fort Collins, CO, USA, March 25-26, 2002

[Yama04] Lidia Yamamoto, «Automated Negotiation for On-Demand Inter-Domain Performance Monitoring», In Proceedings of 2nd International Workshop on Inter-Domain Performance and Simulation (IPS 2004), Budapest, Hungary, March 2004.