

14.02.06

## Deliverable DJ1.2.3

# Network Metric Report



### Deliverable DJ1.2.3

|                        |   |
|------------------------|---|
| Contractual Date:      | 31/01/06                                      |
| Actual Date:           | 14/02/06                                      |
| Contract Number:       | 511082  |
| Instrument type:       | Integrated Infrastructure Initiative (I3)     |
| Activity:              | JRA1 – Performance Monitoring and Measurement |
| Work Item:             | 2   |
| Nature of Deliverable: | R - Report                                    |
| Dissemination Level    | PU - Public                                   |
| Lead Partner           |   |
| Document Code          | GN2-05-265v4                                  |

**Authors:** Maurizio Molina (DANTE) Andy Van Maele (Belnet) Igor Velimirovic (CARNET) Andreas Hanemann (DFN), Andreas Solberg (UNINETT), Athanassios Liakopoulos (GRNET), Martin Swany (Un. Of Delaware, Internet2), Steven Van den Berghe (Belnet), David Schmitz (DFN)

### Abstract

This report identifies the network performance metrics of potential interest for the users of the perfSONAR monitoring framework (developed by Joint Research Activity 1 of the GÉANT2 project), and describes methods for composing them. In the first part, the document provides a categorized list of these metrics and presents the methodologies that have to be followed in order to measure them with the necessary accuracy. In the remaining of the document composition of metrics is described, using three different composition methods: aggregation in time, aggregation in space and concatenation in space.

# Table of Contents

|       |   |    |
|-------|---|----|
| 0     | Executive Summary                         | v  |
| 1     | Introduction                              | 1  |
| 2     | Network metric definition                 | 2  |
| 2.1   | Performance Metrics                       | 3  |
| 2.1.1 | Availability                              | 4  |
| 2.1.2 | Loss and errors                           | 6  |
| 2.1.3 | Delay                                     | 9  |
| 2.1.4 | Bandwidth                                 | 11 |
| 2.2   | Miscellaneous Metrics                     | 14 |
| 2.2.1 | Device specific metrics                   | 14 |
| 2.2.2 | Flow monitoring metrics                   | 15 |
| 2.2.3 | Routing metrics                           | 16 |
| 2.3   | Additional information related to metrics | 17 |
| 3     | Composition of network metrics – General  | 19 |
| 3.1   | Terminology                               | 20 |
| 3.2   | Composition types                         | 21 |
| 4     | Composition of network metrics - Details  | 26 |
| 4.1   | Aggregation in time                       | 26 |
| 4.1.1 | From Singleton to Singleton               | 27 |
| 4.1.2 | From Sample to Sample                     | 27 |
| 4.1.3 | From Sample to $\Delta T$ metrics         | 27 |

|            |  |    |
|------------|--|----|
| 4.1.4      | Further time aggregation of $\Delta T$ metrics     | 33 |
| 4.1.5      | Histogram composition                              | 37 |
| 4.2        | Aggregation in space                               | 37 |
| 4.2.1      | Possible aggregation scopes                        | 37 |
| 4.2.2      | From Singleton to Singleton                        | 42 |
| 4.2.3      | From Sample to Sample                              | 42 |
| 4.2.4      | From Sample to $\Delta S$ metrics                  | 43 |
| 4.2.5      | Further space aggregation of $\Delta S$ metrics    | 49 |
| 4.2.6      | Histogram composition                              | 49 |
| 4.3        | Concatenation in space                             | 49 |
| 4.3.1      | Rationale  | 50 |
| 4.3.2      | Problems   | 51 |
| 4.3.3      | From Singleton to Singleton                        | 52 |
| 4.3.4      | From Sample to Sample and From Sample to Statistic | 52 |
| 4.3.5      | From Statistic to Statistic                        | 53 |
| 4.3.6      | Histogram Composition                              | 56 |
| 4.4        | Summary table of network metric composition        | 57 |
| 4.5        | Cascaded Time and Space composition                | 58 |
| 5          | Conclusions  | 59 |
| 6          | Future work  | 60 |
| 7          | Acronyms   | 62 |
| 8          | References   | 64 |
| Appendix A | Formulas of draft revised Rec Y.1541               | 66 |

## Table of Figures

|  |    |
|--|----|
| Figure 2-1 The Availability is the ratio of the average (MUT) over the average (MDT) in $\Delta T$ | 6  |
| Figure 3-1 Aggregation in time   | 22 |
| Figure 3-2 Aggregation in space: average domain's delay  | 23 |
| Figure 3-3 Aggregation in space: maximum domain's edge-to-edge delay                               | 23 |
| Figure 3-4 Concatenation in space  | 24 |
| Figure 4-2 Aggregation in time – an example  | 30 |
| Table 4-1: Applicability of the function F for different metrics                                   | 32 |
| Figure 4-3: Further order aggregate functions  | 34 |
| Table 4-2: Applicability of the further order aggregation function G                               | 36 |
| Figure 4-4 Aggregation of paths on the basis of their end or intermediate points                   | 39 |
| Figure 4-5 Geographical location aggregation   | 41 |
| Figure 4-6 Aggregating network-wide link utilisation   | 45 |
| Table 4-3: aggregation in space: from sample metrics to $\Delta S$ metrics.                        | 48 |
| Figure 4-7 Routing constraints for concatenation in space  | 50 |
| Figure 4-8 Measurement points deployment for concatenation in space                                | 51 |

## 0 Executive Summary

The Joint Research Activity 1 (JRA1) working group within the GÉANT2 project designed a multi-domain network performance measurement system [13] to provide both network engineers and end users performance data relative to the GÉANT2 network. That system, currently in the development and test implementation phase with the name “perfSONAR” (Performance focused Service Oriented Network monitoring ARchitecture), is designed to let the users configure measurements, collect and store a measurement results, perform post processing operations on the collected data and finally visualize the result of the analysis.

This deliverable brings two major contributions in relation to this activity: it defines which network performance metrics are relevant to be measured, and it defines the post processing operations that are most useful from both a system and an end user’s perspective.

The first part of the deliverable is thus devoted to the description and categorization of the most relevant network performance metrics. This was done by both surveying data usually made available by the network equipments, or the most widely used network monitoring tools, and by considering the answers to a questionnaire [12] about the common practice of network monitoring within the NRENS (National Research and Education Networks). This work led to the classification of metrics into two main sets. The ones referred to as “Basic performance metrics” are describing the performances of a network as seen by a user and are divided in four broad categories: availability, loss and errors, delay and bandwidth. These metrics have frequently a commonly agreed definition and measurement practice. The ones referred to as “Miscellaneous metrics” are on the contrary more device specific, connected with routing or ad hoc monitoring protocols that may or may not be active on the network equipment. These metrics are frequently non-standard, and are of limited use for a user of the network, but are useful for the network engineers for understanding the possible causes of a degradation of one or more Basic performance metrics. This first part of the deliverable also considers which kind of additional information (or metadata) is useful to associate to measurements in order to ease their post processing and interpretation.

The second part of the deliverable addresses the methodologies for post processing (or composing) measurement data. There are several reasons why the composition of raw measured data is needed:

- to provide values representative of an extended period of time or of a larger part of the network;
- to get a quick overview of the network or of a network segment status;
- to deduce desirable information which is not practically obtainable with a direct measurement.

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

Three different composition methods are presented: aggregation in time, aggregation in space and concatenation in space. Aggregation in time is the post processing of measurement time series of high granularity to provide new summary values representing longer periods of time. Aggregation in space provides a value to represent a metric summary value in a network region, or for a whole category of applications. These values can be of interest for planning or cost assessment purposes. Concatenation in space emulates a measurement on a longer path using measurement values on contiguous segments of the path. It is mainly useful for scalability reasons (instead of setting up a full mesh of end-to-end measurements, a limited number of partial results are exploited).

Both in the first part (metric definition) and in the second (metric composition), the approach taken is to combine the formally rigorous description with practical considerations. The first part is meant to be primarily useful for the developers of the perfSONAR system, so that they can include tools capable of providing the most relevant performance data and adapt them, if needed, to a commonly agreed metric definition. The second part is meant to be useful for both developers and users of the perfSONAR system. The formers should get a better understanding of which basic post processing capabilities (Transformation Services, see [13]) are more useful and should be made available along with measurement tools; the latter should be able to better plan the measurement analysis to perform once the data is available. In particular, we tried to anticipate and clarify possible doubts the end user may have, like “is this type of post processing statistically correct? And if yes, what is it useful for?” This document is providing relevant information enabling better understanding and usage of the performance measurement data, and facilitating network management, problem debugging and application tuning.

While this deliverable we set forth a framework for a common understanding, within JRA1, of the meaning of network performance metrics and of their post-processing (or *composition*) methodologies, a number of open issues must still be addressed for turning it into a “cook book” of set of practical guidelines for developers and adopters of the JRA1’s perfSONAR measurement framework. A list of these issues is given in the “Future Work” (Section 0). The detailed plan for the follow up of this activity in Y3 and Y4 of the GÉANT2 project is still in discussion.

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

# 1 Introduction

The JRA1 activity within the GÉANT2 project addresses the performance monitoring of European Research Networks. For this purpose, a variety of network metrics must be collected and stored, and DJ1.2.1 [13] defines an abstract architecture for this purpose, that is currently being implemented with the name of perfSONAR. Some metrics of interest have already been listed in DJ1.1.1 [12], taking into account the answers to a questionnaire that has been circulated among NRENs, selected projects using the European Research Network infrastructure and experiences of some end users. The first purpose of this document is to develop this work on network metrics, giving a precise definition of them, listing measurement procedures and consideration on their accuracy. All this is addressed in Section 2.

Nevertheless, the measurement and collection of metrics may not be sufficient. As an example of a metric consider, for instance, to the network delay that can be directly measured by deploying active test nodes in the network. The availability of these results can however not be sufficient, and post processing steps are often required to gain additional insight into network performance, or for storage and/or visualisation reasons. This process of deriving (or composing) metrics is addressed in the remainder of the document. Precisely, for the definition of composed metrics three ways of composing have to be distinguished: Aggregation in time, aggregation in space, and concatenation in space (see Sections 3 and 4).

Section 5 provides a quick summary and outlines the aspects that were not possible to address in this deliverable but that would be relevant to consider in future work.

## 2 Network metric definition

This chapter presents the set of network performance metrics of interest to the users of the JRA1 monitoring infrastructure. Identified metrics are first described, the specific measurement methodology is then proposed and accuracy issues discussed.

The metrics presented here were identified by potential users of the JRA1 measurement framework during the requirement and evaluation analysis phase that resulted in [12]. Users participating in the surveys belonged to the following categories: NREN's NOC members, Performance Enhancement Response Team (PERT) [17], research project or other activities end users.

A *metric* is an entity that allows us to describe the performance, the reliability and the operational state of a network or its network elements. It is a “formal description” of the services or the operational conditions in the network. A metric should not depend on the methodology used to measure it and measurement results must use standard units of measurements, e.g. *bits per second (bps)*. Examples of metrics are *one-way-delay*, *jitter*, *packet loss*, etc. A *measurement* result is the outcome of a test that assesses the network performance. One or more measurement results are used to calculate the *value of a metric*, e.g. multiple one-way-delay measurements are used to calculate the average value of one-way-delay. Accuracy of the measurements and of the derived value of a metric are influenced by the measurement methodology. Accuracy is expressed with as an *error fraction*  $\varepsilon$  relative to the measured or derived metric values. A confidence level  $\alpha$  may be associated with an error percentage, meaning that one is confident that with probability  $\alpha$  the true value of the metric falls in the interval [measured\_value- $\varepsilon$ , measured\_value+ $\varepsilon$ ].

The metrics assess the performance of the network in the layer 3 (network layer, or “IP” layer) of the ISO/OSI model, if not otherwise specified.

The identified metrics are classified into two major categories; *performance metrics* and *miscellaneous metrics*. The *performance metrics* are further divided into the following four general groups:

- **Availability:** Group of metrics that assess how robust the network is, i.e. the percentage of time the network is running without any problem impacting the availability of services. It can also be referred to specific network elements (e.g. a link or a node), and in that case it will measure the percentage of time they are running without failure.

- **Loss and errors:** Group of metrics that are indicative of the network congestion conditions and/or transmission errors and/or equipment malfunctioning. They usually measure the fraction of packets lost in a network due to buffer overflows or other reasons, or the fraction of eroded bits or packets.
- **Delay:** Group of metrics that also assess the network congestion conditions or effect of routing changes. They measure the delay and delay variation of the packets transferred by a network.
- **Bandwidth:** Group of metrics that assess the amount of data that a user can transfer through the network in a time unit, both dependent and independent from the existing network conditions.

The *miscellaneous metrics* are further divided in the following groups:

- **Device specific:** Metrics that assess diverse characteristics of network elements, such as the CPU load, etc.
- **Flow:** Metrics related with flow-based information usually collected by the routers themselves.
- **Routing:** Metrics related with the information exchanged among routers required for forwarding traffic through the network. This exchange happens through the so-called routing protocol (e.g. OSPF, BGP).

## 2.1 Performance Metrics

A set of basic performance metrics is defined in this section. The metrics are classified into groups, and, for each specific metric within the group, a separate definition is given. Also, a methodology for measuring each metric is presented and accuracy requirements are discussed.

It should be noted that the metrics below are often used to assess the performance guarantees provided to the parts of traffic belonging to different *Quality of Service (QoS)* classes. In this case, even if the same metrics are used, the measurement methodology and accuracy may significantly differ. For example, an elastic application using the *Best Effort* service may only need a rough estimation of packet loss or delay. On the contrary, a real-time multimedia application using a high-priority service, such as the *Premium IP* service offered in GÉANT, may require accurate measurements of one-way delay and packet loss patterns. In summary, if the achieved accuracy is too low for the application wanting to utilise the measurement, the information is not useful at all.

Performance measurements can be collected via different methods. Without getting into too much detail, measurements can be classified as *active* or *passive*. In active measurements, artificial traffic is exchanged among the monitoring nodes, while in passive measurements the network traffic in transit or a sample of it is examined by the monitoring nodes. Measurement information can be collected requesting the measurements from the monitoring node (*pull model*) or the monitoring node can regularly (or under specific conditions) publish the information (*push model*).

## 2.1.1 Availability

The *availability* metric, in IP networks, has been traditionally referred to the measure of the time percentage during which a link or L3 node (a router), is fully functional (usually referred as “up” state), as opposite to being non-functional (usually referred as “down” state). The concept of availability can however be extended to other types of devices or even to network services (2.1.1.2). Finally, availability can be referred to a whole network, but this is a form of derived metric and is thus addressed in 4.2.4.4 and 4.3.5.3.

### 2.1.1.1 Availability of a router or a link

#### Definition

The percentage of time in a defined time period during which link or a router is in the “up” state. For a link, being in the “up” state means to be able to transfer information in one or both direction (if bi-directional). A router is in the “up” state if its loop back interface is reachable<sup>1</sup>

#### Measurement methodology

##### Link

The link availability can be measured at the physical (layer 1), the data link (layer 2) and the network (layer 3) layers of the OSI reference framework.

There are two different models to obtain availability measurements; *pull* and *push*. In the pull model, regular query messages towards the network elements attached to the link gather information of the link state. In the push model, the network elements attached to a link generate traps when the link state changes.

The physical and data link layer availability measurements may be performed via either the pull or the push model. In the pull model, SNMP query messages are usually generated by a Network Management System (NMS) and sent in regular intervals to the network elements attached to a link<sup>2</sup>. Specific SNMP MIB variables define the state of the link at the different OSI layers. Based on this information, the network elements attached to a link generate SNMP reply messages to the NMS. Another pull method is to use a CLI and parse the command output to extract the information. In the push model, the network elements asynchronously generate traps to the NMS whenever a link state is changed. Availability measurements via SNMP queries are usually performed in regular time intervals, such as five-minute intervals.

The network layer availability of a link may also be tested (always according to the pull model) using *ICMP* based tools, such as *ping*. The NMS generates in regular intervals *ping* packets to the elements attached to a link and waits for the corresponding reply. If there is no reply in a specific timeout period, the link state is considered as “down”

<sup>1</sup> If there is no in-band connectivity (no access to the loop back) the router may still be reached for configuration/troubleshooting out of band (e.g. PSTN/ISDN dial up). But as long as it does not deliver any transfer service, it is considered not available

<sup>2</sup> Some network equipment, e.g. some SDH switches, may not be able to respond to SNMP queries. In this case, the only way to obtain from the equipment information about the attached link is to use the equipment’s proprietary interface

## Router

All SNMP queries are made to the loop back address of the router. If there is no response it can be assumed that the router is down

### Units of measurement and accuracy

Availability measurements of an element are typically expressed as the percentage time the element operated without failures in a specific observation period of time, e.g. a “network link was operational at the layer 2 for 99.9% of the time in one year period”. The measurement value does not include any units.

The use of SNMP queries or traps imposes inaccuracies in the measurement. For example, a link may be in “up” state at layers 1 and 2 but in “down” state in layer 3. As SNMP messages require layer 3 connectivity to be established between the network element and the monitoring node, layer 1 and 2 state information may be unavailable for long periods of time. Similar inaccuracy problems may arise when traps are used for collecting availability information, Traps may be lost while they are forwarded from a network element towards the monitoring node and there is no “acknowledgement” mechanism for identifying the loss.

In the pull model, polling rate impacts accuracy. Higher polling rates increase the achieved accuracy (provided that no other network problems exist) but increase the network overheads in terms of bandwidth consumption and CPU usage. Therefore, the polling rate should be chosen after taking of account overheads, needed accuracy, possibility of network failures, etc.

### Derived measurements

The availability  $A$  does not give the frequency of link state changes from state “up” to “down” and vice versa. Based on the information collected in a monitoring node, usually via SNMP queries or traps, the *Mean Up Time (MUT)*, *Mean Down Time (MDT)* and *Mean Time Between Failures (MTBF)* are calculated using the following formulas:

$$MUT = \frac{A \times \Delta T}{N}$$
$$MDT = \frac{(1 - A) \times \Delta T}{N}$$
$$MTBF = \frac{\Delta T}{N}$$

where  $A$  is the measured availability of an element in a specific observation period ( $\Delta T$ ) and  $N$  is the number of failures that took place during  $\Delta T$  (see Figure 2-1). Note that MUT, MDT and MTBF can only be defined when there is *at least* one failure in the observation period.

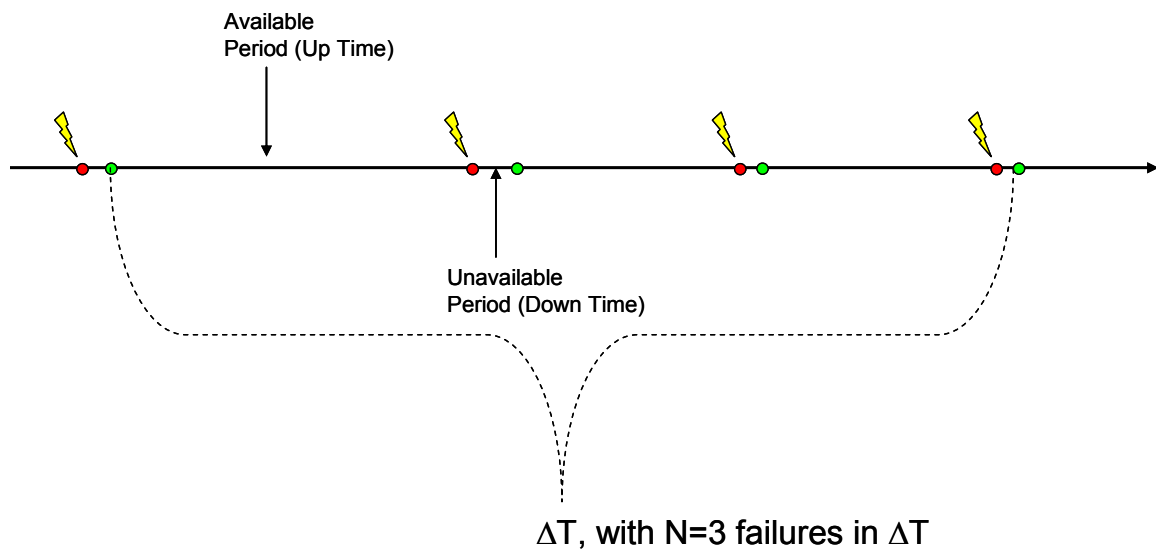


Figure 2-1 The Availability is the ratio of the average (MUT) over the average (MDT) in  $\Delta T$

### 2.1.1.2 General availability of systems and services

The availability metric is applicable to all systems, subsystems or services that exhibit “up” and “down” state. The following list provides some examples for which availability metrics can be measured:

- Ethernet switches, servers, transmission systems, etc
- Network element subsystems: cards or interfaces attached to routers, disks in data servers, etc.
- Application services: Web servers, Sendmail servers, DNS servers, NTP servers, etc.

It should be noted that special care must be taken to define the “up” and “down” states of the monitored system under. For example, a web server availability may be defined as the percentage of time that a web server replies to a request within a predefined interval, e.g. in 15 seconds. This means that if the server’s reply arrives at the monitoring node delayed more than 15 seconds, the state of the services is considered as “down”.

## 2.1.2 Loss and errors

The loss and error metrics describe how much information transmitted from a source is not delivered or received with errors to the corresponding receiver as compared to the total amount of information. Loss and error measurements are usually counted in percentage of errored bits or dropped packets.

In this document, the following metrics are considered: *Bit Error Rate (BER)*, *Error metrics in SDH* and *Packet Loss*.

*BER* can be typically measured at the physical layer. Error metrics in SDH are associated with data link layer (layer 2) while the *packet loss metric* is linked with the network layer (layer 3).

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

### 2.1.2.1 Bit Error Rate (BER)

**Definition:**

The ratio between the number of bits received with errors and total number of transmitted bits in a specific time period.

**Measurement methodology:**

*BER* can be measured by specialized equipment that transmit over the network a pseudo-random bit sequence, which is received at the opposite end of the tested link and compared to a sequence of bits computed in the same way. Such measurement method provides very accurate results but can not be easily applied in an operational network.

**Units of measurement and accuracy:**

Sometimes, only the order of magnitude of BER is reported, instead of the full measurement (e.g.  $10^{-6}$  instead of  $1.56 \times 10^{-6}$ ). The measured BER over optical fibres varies in the range of  $10^{-13}$  to  $10^{-15}$ . Consequently, the measurement equipment has to exhibit similar measurement accuracy.

### 2.1.2.2 Error metrics in SONET/SDH transmission

**Definition:**

Error metrics in SONET/SDH transmission assess the periods of time that errors were detected during the transmission of data among SONET/SDH systems. Metrics are associated with relevant MIB objects defined in RFC 3592 [18], such as:

- errored seconds
- severely-errored seconds
- severely errored framing seconds
- unavailable seconds

Only the above MIB objects are considered in this document.

**Measurement methodology:**

Coding and error correction techniques allow network elements to identify with significant accuracy the erroneous (or not delivered) chunks of information, e.g. bits or code characters, transmitted at the physical and data link layers.

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

Errors are identified and counted by the SONET/SDH transmission systems during their ordinary operation. Error information is available through NMS and/or vendor proprietary interfaces. The accuracy of the measurements depends on the SONET/SDH technology and it is not further discussed in this document.

#### Units of measurement and accuracy:

Measurements unit for the MIB objects mentioned previously is the *second*, and this is also the accuracy.

### 2.1.2.3 One-way / Two-way Packet Loss

#### Definition:

The ratio between the number of packets considered as lost at the destination node and the total number of packets transmitted by the source node. As stated in RFC2680 [4], a packet transmitted by a source node towards a destination node is considered lost if it does not reach the destination node within a certain time period. If the source and the destination nodes are distinct, the metric is called *one-way packet loss*. If the source and the destination nodes are the same, the metric is called *two-way packet loss*. In this case, a node is needed to “bounce” monitoring packets back to the source, and what is tested is the loss on the path from the source node to this bouncing node and back.

Some of the main reasons of the packet losses are the following:

- Errors in layers 1 & 2 causing CRC operations at the layer 3 to fail. In such case, the received packet is discarded.
- Congestion in the network links or equipment switching fabric that cause buffer overflows.
- Protocol instability or configuration errors that cause packets to be misrouted or forwarded into loops. In such cases, packets are discarded when their TTL expires.

#### Measurement methodology:

The *packet loss* metric is measured on the network layer. Consequently, for each measurement one has to define explicitly the network layer parameters applied during the measurements, e.g. source/destination node IP addresses, packet size, packet Type of Service value, time-to-live value, etc. Also, as discussed in the definition, a packet is considered as lost only if it is not delivered to the destination node within a specific time period. Consequently, a timeout value has to be chosen prior performing measurements. If not defined, the timeout interval is considered an extremely large time value (e.g. 255 s).

*One-way packet loss* measurements across a link or a path may be performed with either active or passive measurement methodologies. In active measurements, artificial traffic is inserted in the network by a sender and collected by a receiver at the end of the path. By measuring the packet loss of the artificial traffic, estimations of the packet loss of real traffic can be made. Passive measurements can be done correlating data extracted from packet headers at two measurement points. In contrast to active measurements, passive measurements provide

measurements of packet loss on real (and not artificial) traffic, and are therefore more reliable. On the other side, they are more difficult to be implemented in wide scale (see e.g. [11]).

*Two-way packet loss* measurements across a link or a path may only be performed with the active measurements. Tools based on the *ICMP* protocol, such as *ping*, generate artificial traffic towards a destination node, which upon reception of the *ICMP* messages generates *ICMP* replies. By measuring the packet loss of the artificial traffic, estimations of the packet loss of real traffic can be made.

JRA1 measurement points (MPs) may support both *one-way packet loss* and *two-way packet loss* measurements. In most cases, active measurements will be deployed.

### Units of measurement and accuracy:

*Packet loss* metric is represented as the ratio of the number of lost packets to the overall number of packets generated in a given time period (see RFC2680 [4]) and, thus, no measurement units are used. As an example, the *packet loss* between the two nodes is expressed as a% in a x-minute period.

The achieved accuracy of a specific packet loss measurement test may vary significantly in different networks. However, high priority classes of service, e.g. the Premium IP service, usually exhibit extremely low packet loss. Thus, measurement tests must ensure that achieved accuracy is adequate to assess the committed service level. This may mean sending a lot of test packets or carrying out the tests for a long time.

## 2.1.3 Delay

The *delay* metrics assesses the time that a chunk of information, e.g. a packet, takes to be transported from a source node to a destination node.

In this document, we will concentrate on delay measurement at the network layer (layer 3). There are three different instances of delay metrics presented in the next paragraphs: *One-Way Delay (OWD)*, *IP Packet Delay Variation (IPDV)* (sometimes called "jitter"), and *Round Trip Time (RTT)*.

The delay that a packet experiences in a network is affected by four different factors; queuing delay, switching delay, transmission delay and propagation delay, but the metrics we describe and the associated measurement methodologies do not distinguish between them.

### 2.1.3.1 One Way Delay (OWD)

#### Definition:

*One-Way Delay* is the time between the occurrence of the first bit of a packet on the first observation point, e.g. transmitting monitor interface, and the occurrence of the last bit of a packet on the second observation point (see RFC 2679.[3]).

#### Measurement methodology:

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

Active or passive measurements can be used to measure the OWD. In active measurements, monitoring packets are time-stamped as they “enter” the network at the source node. The receiver node compares the time-stamp time information carried in each monitoring packets with the time that each packet was received and calculates the delay.

In passive measurements, monitoring nodes collect header information from real traffic passing observation points. The collected packet header information is time-stamped in observation points. Data from different observation nodes are then correlated to identify the OWD of the packet passed through them. This approach has been proposed in [11]. Passive monitoring methods exhibit many implementation and scalability limitations in high speed network environments.

### Units of measurement and accuracy:

Delay measurement unit is the *second* and a valid value is always positive or infinite. In geographically distributed network delays are usually of the order of *milliseconds*, thus measurement results must at least support this precision.

Achieved accuracy in delay measurements is affected by a lot of different parameters. Time synchronization among the observation points is probably the most difficult one to be achieved. For accurate synchronisation the clocks on observation points are synchronised by GPS receivers and, thus, can achieve an accuracy in the order to tens of microseconds. Alternatively, the Network Time Protocol (NTP) can be used, but in this case the accuracy depends on the RTT between the NTP servers and the measurement point (and this can be as high as several tenths of milliseconds and, worse than that, unpredictable). Other parameters that impact the delay measurements are packet loss, packet fragmentation, inaccuracies in the time-stamping process by monitoring nodes.

### 2.1.3.2 IP Packet Delay Variation

#### Definition:

Given a stream of at least two packets crossing observation point A and observation point B, we define *Inter-Packet Delay Variation IPDV* as the difference in the OWD of a selected pair of packets in the streaming (see RFC 3393 [6]). The ITU-T (rec. Y.1540 [22]) has a slightly different definition of IPDV: the IPDV is the difference between the 99,9 percentile of the OWD and a reference delay. The reference delay can be chosen in different ways, but the advised choice is to select the minimum path delay. Note that if the reference delay is the one of the previous packet, the ITU-T definition coincides with the IETF one.

#### Measurement methodology:

IPDV measurements may be performed by using a pre-defined train of packets between two observation nodes (active monitoring). The receiving node knows exactly the traffic profile characteristic of the monitoring traffic generated by the source node, i.e. the exact packet rate, the packet size distribution, the intervals between the packets, etc. The receiver node timestamps the incoming packets and using the information regarding the packet train produced by the source, it can accurately estimate the jitter exhibited among each pair of packets. Alternatively, if the traffic profile characteristic of the monitoring traffic is unknown, the source node may timestamp the packets while transmitting them. This will allow the receiver node to measure the IPDV for a series of monitoring of packets. However, both methodologies do not require exact time synchronisation between source and receiver nodes (but they are sensitive to clock drifts, see sec. 5 of [6]).

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

### Units of measurement and accuracy:

IPDV measurement unit is the *second* and a valid value is either positive or negative. In geographically distributed networks delay variations are usually of the order of *milliseconds*, thus measurement results must at least support this precision.

The time calculation for IPDV measurements is based on the clock information of the receiver monitoring node only (i.e., there's no need of clock synchronisation). Consequently, there are no synchronisation related inaccuracies in the measurements, provided that the source node generated packets according to an agreed traffic profile or time stamped the packets.

#### 2.1.3.3 Round Trip Time (RTT)

##### Definition:

*Round Trip Time (RTT)* is considered the period between the time instance a request packet is sent by a source node and the time instance a response packet is received from the destination node (see RFC 2681 [5]). Note that response packet must be sent by the destination node as soon as the request packet is received.

##### Measurement methodology:

RTT is usually measured via the *ping* tool, which is widely available in end systems and network elements, e.g. routers. The *ping* tool uses ICMP, so firewalls have to permit ICMP packets. In order to overcome firewall restrictions, various tools perform RTT measurements using unrestricted UDP ports. However, such tools are not widely implemented in the network.

In most tools that perform RTT measurements (including *ping*), a train of packets with evenly distributed inter-departure times is generated and the RTT is measured for each packet. *ping*, beyond single measurement instances, may also provide the Minimum/Average/Maximum/Standard Deviation values of the train of packets.

##### Units of measurement and accuracy:

Round Trip Time measurement unit is the *second* and a valid value is always positive or infinite. RTT measurements are based on a single clock, therefore no clock synchronisation is required. As for OWD and IPDV, in geographically distributed networks a precision of the order of 1 millisecond or lower is desirable.

As ICMP packets are usually considered as lower priority traffic by network elements, the RTT measurements with ICMP usually give an upper bound of the delay that real traffic will exhibit in the network.

#### 2.1.4 Bandwidth

There are many different proposed definitions for the *bandwidth* metric in the literature. Generally, bandwidth corresponds to the total amount of data (as seen by a certain ISO/OSI layer) that is possible to send over a network in a certain period of time, measured in bits per second (bps). Therefore, bandwidth should always be referred to an

ISO/OSI layer, as each layer is encapsulated within headers of the lower layers, and this reduces the “available” bandwidth for upper layers.

In this document, the *bandwidth* metric is related only with the layer 3 of ISO/OSI model, i.e. bandwidth is referred to the ability of transferring whole IP packets (IP header and payload are not distinguished).

There are four metrics discussed in this section: *capacity*, *utilisation*, and *available bandwidth* and *achievable bandwidth*. These metrics are both defined for a single link and for a path, i.e. a sequence of links and hops.

#### 2.1.4.1 Capacity of a link / path

**Definition:**

*Capacity of a link (path)* is defined as the maximum amount of data (IP packet's bits) per unit time that a link (path) can transport when there is no competing traffic.

**Measurement methodology:**

The link capacity information is usually gathered from attached routers to the link either via SNMP queries or via remote shell queries at the *command line interface* (CLI).

Path capacity is defined as the minimum capacity of the links that comprise a path.

**Units of measurement and accuracy:**

*Capacity* measurement unit is the *bits per second (bps)* and a legitimate value is positive. In high-speed networks, capacity is usually presented in *giga* ( $10^9$ ) or *mega* ( $10^6$ ) *bits per second*.

The capacity at the layer 3 varies according to the average IP packet size, i.e. the size of IP header plus the size of the payload. As IP packets are encapsulated by lower layer headers, these overheads may depend on IP packet size, and this influences the capacity in layer 3. Therefore, capacity in layer 3 should be estimated using a realistic distribution (or at least a realistic average) of packet size. See e.g. [20].

#### 2.1.4.2 Link Utilisation

**Definition:**

The *link utilisation* metric is a measure of the amount of capacity used by IP packets (both header and payload) over a specific time window (period).

**Measurement methodology:**

The link utilisation is usually calculated from traffic counters that routers update. Counter information are collected via SNMP queries on the router's MIB (that reflects the counters values) and calculations are performed for specific

period of time (typically, the average between two subsequent readings). The information can also be retrieve via remote shell queries at the *command line interface* (CLI).

**Units of measurement and accuracy:**

*Utilisation* is represented as a ratio of the capacity occupied by traffic to the overall link capacity in layer 3. The time window of the measurements is usually large (typically 5-15 minutes) in order to smooth effects due to short term traffic bursts and limit the load on the routers necessary to reply to SNMP queries.

### 2.1.4.3 Available bandwidth of a link

**Definition:**

The *available bandwidth of a link* is defined as the maximum amount of data (IP packets' bits) per time unit that can be transferred over a link exhibiting a certain utilisation level.

**Measurement methodology:**

*Available bandwidth* is usually calculated by subtracting the link utilisation from the link capacity.

**Units of measurement and accuracy:**

*Available bandwidth* measurement unit is the *bits per second (bps)* and a legitimate value is positive or zero. In high-speed networks, capacity is usually presented in *giga ( $10^9$ )* or *mega ( $10^6$ ) bits per second*.

*Available bandwidth* measurement can also be expressed as the percentage of link capacity that is not utilised.

### 2.1.4.4 Achievable bandwidth on a path

**Definition:**

The *achievable bandwidth on a path* is defined as the maximum amount of data (IP packets' bits) per time unit that can be transferred through a path consisting of multiple links, each of them exhibiting a specific utilisation level.

**Measurement methodology:**

Achievable bandwidth on a path can only be measured by performing specific tests using attached end-nodes at the each end of the path. Therefore, the measurement methodology is dependent on the available test equipment.

JRA1 will use the *iperf* tool to perform measurements of the achievable bandwidth on a path.

**Units of measurement and accuracy:**

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

*Achievable bandwidth* measurement unit is the *bits per second (bps)* and a legitimate value is positive. In high-speed networks, achievable bandwidth may be presented in *giga* ( $10^9$ ) or *mega* ( $10^6$ ) *bits per second*.

## 2.2 Miscellaneous Metrics

A monitoring infrastructure's purpose is to estimate the performance that traffic experiences as it is forwarded through the network. In case of degradation of end-to-end transport services, the monitoring infrastructure should provide the network administrator with tools to quickly determine the cause of the degradation.

There are network operation parameters that are not directly related with the performance guarantees of packet transfer services but that can potentially impact the level of services offered in a network. An example of such parameters is the CPU load in routers. High CPU load usually increase the packet switching delay and jitter and, in some cases, causes packet drops. The monitoring infrastructure should, therefore, observe these network operation parameters.

We categorise the *miscellaneous metrics* as follows:

1. Device specific metrics
2. Flow monitoring metrics
3. Routing metrics

### 2.2.1 Device specific metrics

#### Definition:

*Device specific metrics* are often used to describe operational parameters of layer 3 network devices, such as IP/MPLS routers. The list of devices can be further extended to include devices operating at the physical or data link layers, such as L2 switches, SONET/SDH transmission equipment, etc, or at the application layer, such as network application server.

Some examples of *device specific metrics* are:

- Current state of a network device or a group of devices (different from simply the “up” and “down” states indicating the availability);
- CPU and memory utilisation of a device;
- Number of forwarded IP packets, number of non-forwarded IP packets, forwarded rate, received IP packets, etc. (for routers and switches, per interface);
- Device temperature.

### Measurement methodology:

SNMP queries should often be used for retrieving information from the network devices in order to estimate the above metrics. In some cases SNMP traps may be generated autonomously from the network devices. The information can also be retrieve via remote shell queries at the *command line interface* (CLI).

### Units of measurement and accuracy:

The units of measurement are specific for each metric. For example:

- percentage for CPU utilisation (average, maximum, minimum over period of time);
- number of IP packets for metrics like forwarded and not forwarded IP packets, received IP packets, etc ;
- number of interface errors
- Celsius or Fahrenheit degrees for temperature

## 2.2.2 Flow monitoring metrics

Analysis of IP traffic characteristics using flow related information is widely performed in today's networks. Flow related information can be used for:

- Monitoring transferred traffic over the network to, for example, estimate the percentage of traffic related with a specific application or protocol.
- Network planning and traffic engineering to, for example, monitor traffic growth or measure traffic exchanged between domains.
- Accounting and billing to, for example, count how much traffic a subscriber injected into the network for a specific traffic class in a specific time period.
- Security analysis to, for example, detect anomalies in the traffic patterns due to Denial of Service (DoS) attacks or viruses.

A flow was traditionally defined as a sequence of packets exhibiting the same *5-tuple set* composed of the source and destination IP address, the protocol over IP and the source and destination transport ports. There are multiple implementations available in commercial products that are able to identify different flows passing through a network router, the most widely deployed one being Cisco Netflow v5 (and the most recent being Netflow v9). These implementations allow the network administrators to collect flow information based on multiple criteria, such as specific source and/or destination IP addresses, specific or groups of TCP or UDP port, etc. Collected information include packet and byte counters related with individual flows, start and end of flow time stamp, etc.

The IETF IP Flow Information Export (IPFIX) working group is standardising a protocol for efficient export of flow information. This new protocol is basically extending the Netflow v9 protocol, developed by Cisco Systems. The IPFIX requirement document (RFC 3917, [19]) states that “A Flow is defined as a set of IP packets passing an Observation Point in the network during a certain time interval; all packets belonging to a particular Flow have a set of common properties”. In the same document, in section 4, the list of mandatory properties for distinguishing flows is provided: IP protocol version, source IP address, destination IP address, protocol type (TCP, UDP, ICMP, ...), Transport Header Fields, MPLS Label, DiffServ Code Point. This list of parameters is an extended *n-tuple parameter set*, which can be used for distinguishing flows.

According to the requirement questionnaire [12], the most relevant information that can be derived from flow metrics is:

- Busiest IP source-IP destination traffic interests (Pkts/s or Bytes/s), for DoS detection;
- AS (autonomous System) traffic matrix, for network planning;
- Number of flows per IP source/ IP destination, to detect scanning activity;
- Per port traffic, to evaluate traffic percentage of specific services using well-known ports.

As mentioned previously, the collection of flow related information is usually performed by production routers: while incoming packets are forwarded towards the next hop, specific counters inside the router are updated. At regular intervals, flow related information is transferred from the router towards specific *collectors*. The latter are responsible for analysing and archiving the data. The final set of flow information stored in the collectors may be incomplete due to errors in the router operation or due to packet losses while exporting the data. Proprietary industry standards (Netflow) or IETF ones (IPFIX) describe the formats and protocols to transferring the flow records from routers to collectors.

### 2.2.3 Routing metrics

Recording routing related information, i.e. how traffic is forwarded between the network nodes in a specific time instant, can ease the off-line performance and troubleshooting analysis, as it allows the network engineers to correlate performance degradation events with network failures or topology changes. Other routing information metrics, such as the size of the full routing table, are significant for the ordinary network operation: the short-term fluctuations of the size of routing table may reveal severe network problems, such as instability of some of network links or errors in the applied routing policies.

Routing tables may contain information such as the link utilisation, alternative routes to a destination address, preference weights for each specific link, etc. Each administrative domain maintains inter- and intra-domain routing information, which is used for creating the forwarding table –usually called the forwarding information base (FIB)- in routers, that contains all the needed information to forward incoming packets.

Routing information can be used for discovering the network topology at a specific time instant. From the performance measurements perspective, routing information can be used for discovering the path between any pair

of measuring nodes (an alternative method is to use simple management tools, such as traceroute). Furthermore, assuming that network engineers do not often change the routing configuration of the routers, monitoring routing updates can be used to estimate the availability of the network links. Analysing the routing update messages provides better insight of the network than polling the network interfaces regularly. In large network topologies, routing updates analysis may reduce network management overheads.

Routing information may also give accurate information of the link reservations inside an MPLS-enabled domain. Reservations on network links usually change as more connections are established or terminated via signalling, such as RSVP or LDP: routing updates of the OSPF-TE and ISIS-TE propagate the information about aggregate reservations to all network nodes. Thus, tracking this information can be useful to understand the availability of resources in a network.

The discussion about the potential use of routing information within the perfSONAR system, however, has just started at the time of this writing.

## 2.3 Additional information related to metrics

A metric may be further related to information that specifies the conditions under which measurements were performed. Consistently with [24], we call this information metadata. *“... Metadata describes the type of measurement data, the entity or entities being measured and the particular parameters of the measurement.... the data itself is simply a timestamp and a vector, or array, of values”.*

In other words, whatever is not a metric value and a timestamp can be considered Metadata. However, sometimes metadata may be treated as a metric itself. For example, the link capacity (defined in 2.1.4.1 as a metric) may become metadata information while aggregating in space OWD measurements (see 4.2.4.1).

Following is a non-exhaustive list of what can be considered as metadata:

- Start of time measurement period
- Time resolution (e.g. the period an average refers to)
- Number of days in a given month or other period
- Whether the year is a leap year or not
- UTM coordinates [16] of a network point of presence (PoP)
- Which router, link or interface the measurement relates to.
- Physical location of the equipment, e.g. city, room number etc.
- Administrative interface IP address

- In case of active measurements, information about the traffic used for the test: e.g. IP packet size, IP version, ToS, Protocol over IP.

### 3 Composition of network metrics – General

The deployment of a measurement infrastructure and the collection of elementary measurements are not enough to understand and keep under control the network's behaviour. Network measurements need in general to be post-processed to be useful for the several tasks of network engineering and management. The first step of this post processing is often a process of "composition" of single measurements or measurement sets into other ones. The reasons for doing so are briefly introduced here.

A first reason, mainly applicable to network engineering, is scalability. Due to the number of network elements in large networks like GÉANT and the connected NRENs, it is impossible to perform measurements from each element to all others. It is necessary to select a set of links of special interest and to perform the measurements there. Examples for this are active measurements of one-way delay, jitter, and loss which are performed by IPPM measurement boxes from DFN Erlangen. If there are boxes deployed at locations A, B, and C and measurements are performed between A and B as well as between B and C, an important question is if we can we also provide information about the corresponding metrics for the connection A to C without doing a specific, additional measurement?

Another reason may be data reduction (opposite need with respect to the previous one, where "more" data is generated...). This is of interest for network planners and managers. Let us assume that there is network domain in which delay measurements are performed among a subset of its elements. A network manager might ask whether there is a problem with the network delay in general. Therefore, it would be desirable to obtain a single value giving a summary indication of the network delay. This value has to be calculated from the elementary delay measurements. For example, it may be computed as the weighed average over the link's delays, where the weight is determined by the traffic carried over the links.

Moreover, metric composition can be helpful to provide, from raw measurement data, some tangible, well-understood and agreed upon information about the elements taking place in the services guarantees provided by a network. Such information can be used in the SLA/SLS contracts among a provide and its customers

Finally, another important reason for generating derived information from measurement results is to perform trend analysis. For doing so, a single value for an hour, a day or, a month is computed from the basic measurements which are scheduled e.g. every five minutes. In doing so, trends can be more easily witnessed, like an increasing usage of a backbone link which might require the installation of alternative links or the rerouting of some network flows.

## 3.1 Terminology

This section introduces the terminology used in the subsequent parts of this document. Our principal goal is to extend the terminology defined in the RFC 2330 “Framework for IP Performance Metrics” [1]. Therefore, we avoid repeating definitions, but we provide some comment in order to present GN2-JRA1 concepts more clearly. In few cases, where terms used in this document deviate from the terminology of [1], differences are explicitly stated and justification of our choice is given. Novel terminology with respect to [1] is emphasised in this section using *italics*.

According to [1], a single observation of a metric is called a singleton metric, a group of singleton metrics is called a sample metric and a statistic computed over a sample metric is called a statistic metric.

In network operations, there are however some metric post-processing practices that lead to intermediate or final results not falling in any of the categories listed above. According to [1], these would be called derived metrics: “Derived metrics may be defined purely in terms of another metrics”.

Derived metrics may be obtained by applying a statistical operation to the elements of the samples, e.g. averaging them, and in this case they are clearly a statistics metric. However, a derived metric could also be obtained through some operation (statistical or not) and not have any practical significance, but just be of help for a further operation. In this case, we call these intermediate results “*help metrics*”.

In this document, we try to be more specific about what derived metrics have practical interest, and for that we do not use the term “derived” metric, but rather “composed” metric. In other words, metric derivation is a generic operation, while metric composition is an operation leading to a result having some established meaning, that makes sense to render through a visualization system. Note that [1] uses almost interchangeably “derived” and “composed” metric, so by using “composed” metrics we are not inventing new terminology, but just give to it a more precise meaning, as we explained.

There are three types of composition that are considered in this document, but before detailing them (in 3.2) we need to introduce some more concepts.

The *physical and logical metric scope* qualifies the physical or logical entity to which a metric refers to<sup>3</sup>. Examples of physical metric scopes are:

- the observation point, if the measurements are taken on a single observation point, such a router interface;
- the network path, if measurements are taken between two different observation points one or more hops away (we also distinguish between loose and strict paths: only a pair of endpoints identifies loose paths, while strict paths are identified by all the hops along the path, even if measurements are taken among the endpoints only)

Examples of logical metric scopes are:

---

<sup>3</sup> So, metric scope is similar to the concept of “metadata” introduced in 2.3. We use the word scope to indicate the abstract concept (e.g. a geographic location), that will be represented by some metadata in a data structure.

- the IP level properties of the packets involved in the measurement (also defined in [1] as “packet type P distinguisher”), like the IP addresses, the transport protocol, the ToS fields, the packet size, etc;
- the transport level properties like the UDP or TCP ports;
- the properties derived from the packet’s treatment in a router, like the input and output interface, or the previous and next AS.

In formal terms, we can say that the above list of properties forms an unordered list of variables that describes the context the metric refers to. We call it “*metric distinction tuple*”. The metric scope can be one to one associated with a “metric distinction tuple”.

Most measurement values are further associated with metadata, usually related to time. The *metric estimation time instant* identifies a time which the metric can be attributed to. The *metric estimation time window* identifies a time window, defined by specific starting/ending instants, which a metric can be attributed to. Data of the same metric could either be time-normalized according to this time window or not. In any case it should be clear whether the metric value is time-normalized or not. An example is the number of total octets sent over a link during a certain time window, which can be given as an absolute value or as an average bit rate if divided by the time duration. Normally, the time-normalized value is presented to users (because it’s more intuitive), while the non-time-normalized value is stored in tools for internal purposes.

Note that if a metric is reported along with the metric estimation time instant, it does not necessarily mean that it is the result of a single observation (singleton). It may also be a statistic metric computed over a sample for which the given collection time is used as a conventional reference. On the contrary, if a metric is reported along with a time window, it means that it is a statistic metric over the specified time window, or a singleton metric that is calculated over a specific time period, e.g. derived from a counter reading.

The *metric nature* broadly qualifies the metric. Examples of metric nature are One-Way Delay (OWD), Packet Loss, Round Trip Time (RTT), (see sec. 2.1 and 2.2).

The *metric type* fully qualifies the metrics. That is, it indicates at least the metric nature, and then it *can* indicate one or more of the further attributes that the metric may have: if it is a singleton, sample or statistic (and which statistic, e.g. “mean”, or “95<sup>th</sup> percentile”), what is its scope, and if it is a composed metric what composition was applied.

## 3.2 Composition types

There are three types of composition considered in this document.

Firstly, *aggregation in time* is defined as the composition of metrics with the same type and scope obtained in different time instants or time windows. For example, starting from a time series of One-Way Delay measurements on a certain network path obtained in 5-minute periods and averaging groups of 12 consecutive values, a time series measurement with a coarser resolution is obtained (Figure 3-1). There are often several possibilities to define such an aggregated value. For example, the average (or a percentile) of delay values could be chosen. The main reason for doing time aggregation is to reduce the amount of data that has to be stored, and make the visualisation, the spotting of regular cycles and/or growing or decreasing trends easier. Note that in [1], the term “temporal composition” is introduced, but with a different meaning than the one given here to aggregation in time. The temporal

composition considered there refers to methodologies to predict future metrics on the basis of past observations, exploiting the time correlation that certain metrics can exhibit. We do not consider this type of composition here, though it may be the subject of future work.

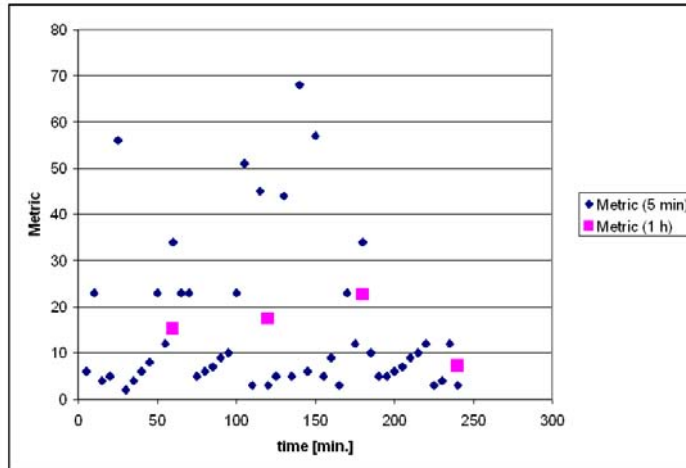
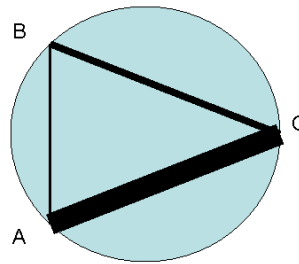


Figure 3-1 Aggregation in time

Secondly, *aggregation in space* is defined as the composition of metrics of the same type but with different scope. This composition may involve weighting the contributions of the several input metrics. For example, if we want to compose the average OWD of Origin-Destination pairs of a network domain (thus where the inputs are already “statistics” metrics) it makes sense to weight each metric according to the traffic carried on the relative OD pair:

$$OWD = F_1 * OWD_1 + F_2 * OWD_2 + \dots + F_n * OWD_n, \text{ where } F_k = \frac{Link\_Load_k}{\sum_i Link\_Load_i}$$



|        | Delay  | Load      |
|--------|--|-----------|
| A-B    | 24.5 ms  | 1 Gbit/s  |
| B-C    | 7.8 ms   | 3 Gbit/s  |
| A-C    | 4 ms   | 9 Gbit/s  |
| Domain | $1/13 * 24.5 + 3/13 * 7.8 + 9/13 * 4 = 6.4$ ms | 13 Gbit/s |

Figure 3-2 Aggregation in space: average domain's delay

If an “average domain's delay” defined as above were available for the majority of domains offering commercial services, a customer could decide to switch to a provider offering better services, or to change the balance of the traffic it sends towards its downstream domains, in case of multi-homing.

Another example of metric that could be aggregated in space, is the maximum edge-to-edge delay across a single domain. Assume that a service provider wants to advertise the maximum delay that transit traffic will experience while passing through his/her domain. As there are multiple edge-to-edge paths across a domain, shown with different coloured arrows in the following figure, the service provider has to either advertise a list of delays each corresponding to a specific edge-to-edge path, or advertise a maximum value. The latter approach is more scalable and simplifies the advertisement of measurement information via inter-domain protocols, such as BGP. Similar operations can be applied to other metrics, e.g. “maximum edge-to-edge packet loss”, etc.

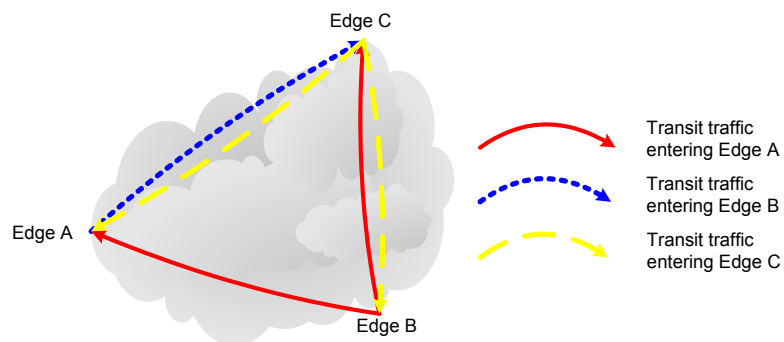


Figure 3-3 Aggregation in space: maximum domain's edge-to-edge delay

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

The purpose of *aggregation in space* is to obtain a summary view of the behaviour of large network sections, or in general of coarser aggregates. The metric collection time instant, i.e. the metric collection time window of measured metrics is not considered in space aggregation. We assume that either it is consistent for all the composed metrics, e.g. we compose a set of average delays all referred to the same time window, or in the case when time window of each composed metric does not affect the aggregation result.

Thirdly, the *concatenation in space* is defined as the composition of metrics of same type but with a different, non-overlapping, physical spatial scope, so that the resulting metric is representative of what the metric would be if directly obtained with a direct measurement over the sequence of the several spatial scopes. An example is the sum of OWDs of different edge-to-edge domain's delays, where the intermediate edge points are close to each other or happen to be the same. In this way, we can for example in Figure 3-4 estimate  $OWD_{AC}$  starting from the knowledge of  $OWD_{AB}$  and  $OWD_{BC}$ . Differently from aggregation in space, all path's segments contribute equivalently to the composed metric, independently of the load on them. Concatenation in space is not only useful for actively measured metrics, but also for passively measured ones. For example, it is currently the only way to provide accurate IP available bandwidth along a path. The purpose of *concatenation in space* is to emulate a measurement along a network path using measurement values for the parts of the network path (path sections). A full mesh of *regularly* scheduled measurements between all measurement probes is in fact not likely to happen because of the N-square problem, and because of the diversity of tools used. Even if it was possible to perform *occasionally* on-demand tests among whatever probe couple, the results obtained could not be compared against reference historical values.

Note that in [1] concatenation in space is called “spatial composition”. We think that our proposed term concatenation in space is a more intuitive description, and thus we will use it throughout the document. We avoided on purpose assigning a meaning to spatial composition in this document, to avoid confusion.

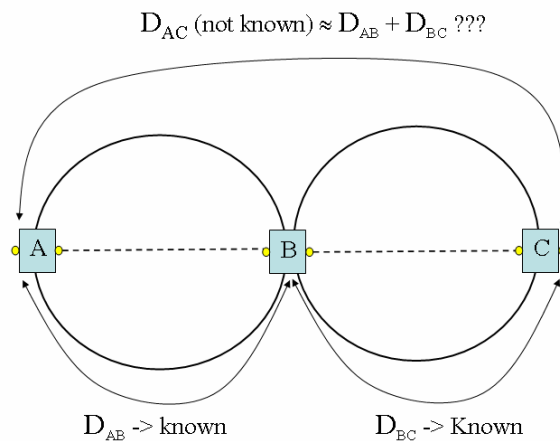


Figure 3-4 Concatenation in space

In the next section we investigate in detail the previously described types of composition.

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

## Network Metric Report

Composition of network metrics – General



The operation of post-processing a set of composed metrics (alone or in conjunction with help metrics) to get another result is called *further composition of metrics*.

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

## 4 Composition of network metrics - Details

In Section 3.2 we defined three ways of composing metrics, i.e. aggregation in time, aggregation in space, and concatenation in space. In this section further elaboration is presented, giving indications and examples that can serve as guidelines to the implementers of the JRA1 measurement framework and its users.

For each of the three composition types, we chose to consider the three metric categories described in [1], i.e. singletons, samples and statistics, and illustrate composition *starting* from each of these categories. This follows the natural flow of what happens in real systems: singletons are collected to form samples, samples are processed to form statistics, and statistics may be further processed to form summary indicators of network performance. The description is abstract, but references in several concrete examples the metrics defined in Section 2.

However, sometimes users may be focused on a single metric category (e.g. delays) and would like to be able to quickly refer to a “cookbook” listing all the composition possibilities for that metric. We present this alternative metric-centric view of composition in Section 4.4.

How to combine (or better put in a cascade) the three composition types is briefly discussed in 4.5.

### 4.1 Aggregation in time

As defined in Section 3.2, aggregation in time is the composition of two or more values of metrics of the same metric nature and scope, but collected in different time instants or time windows.

We follow the approach of categorising the possible time aggregations on the basis of what the starting and ending objects of the aggregations are (e.g. from a sample of singletons to a statistics metrics, or from a sample of statistic metrics resulting from a previous composition to another statistics metric, etc.). Then, within each aggregation category, we elaborate to which metric it applies (e.g. delays, losses, etc) and which aggregation functions (e.g. statistic functions) can be used.

### 4.1.1 From Singleton to Singleton

Two or more measured instances of a singleton metric of the same type and scope are, by definition, considered as measured instances of a sample metric. Therefore, aggregation in time of singleton metrics does not exist by definition. Aggregation in time of these measured instances of a sample metric is discussed in the next paragraph.

### 4.1.2 From Sample to Sample

Simply enlarging a sample by taking samples together or taking subsamples from a sample cannot be regarded as a type of aggregation in time; it is rather preliminary to it. The only relevant thing is that possible meta-data information associated to samples' singletons may have to be updated.

### 4.1.3 From Sample to $\Delta T$ metrics

We call " $\Delta T$  metric" a value which is representative of all the sample values falling in a time interval  $\Delta T$ . In the following subsections we first elaborate on how  $\Delta T$  can be chosen (Subsections 4.1.3.1 and 4.1.3.2), then on what is the functional relationship between the samples and the summary value in  $\Delta T$  (Subsections 4.1.3.3 and 4.1.3.4). In Subsection 4.1.3.5 we summarise it all by providing an overview table.

#### 4.1.3.1 Fixed time window

Suppose we have a sample  $S$  composed of two parameters  $\langle \text{time}, \text{metric} \rangle$  with values  $\langle T_i, M_i \rangle$ , taken in the time window  $(T_0, T_f)$ . Assume that we subdivide the time window  $(T_0, T_f)$  into  $N$  consecutive time intervals  $(T_{a,k-1}, T_{a,k})$ . The values  $\langle T_i, M_i \rangle$  falling in this time interval, form a sub-sample  $S_k$  of the original sample  $S$ .

For simplicity we initially consider the case where we subdivide the sample  $S$  into consecutive equally sized time intervals of duration  $\Delta T$ , which are adjacent in time (we call these *fixed time windows*). Therefore:

$$T_{a,k} - T_{a,k-1} = \Delta T, \text{ for } k = 1, \dots, N$$

$$T_{a,k} = T_0 + \Delta T * k/N, \text{ especially } T_{a,N} = T_f$$

(Note that the time interval limits  $T_{a,k}$  are in general not related with the singleton collection instant  $T_i$ )

The duration of  $\Delta T$  specifies the *resolution* of aggregation. When fixed time windows are aligned to the resolution, we call them *slotted time windows*, e.g. when the resolution is 10 minutes and the slots start at wall clock time X:00, X:10, X:20, X:30 ...

We now define the aggregated value of the first order  $V_k$  of the time interval  $(T_{a,k-1}, T_{a,k})$  as

$$V_k = F(M_i), \text{ for each } \langle T_i, M_i \rangle \text{ with } T_{a,k-1} \leq T_i < T_{a,k}$$

where  $F$  is the composition function applied to  $S_k$ , and  $M_i$  is the set of singleton values in the interval  $(T_{a,k-1}, T_{a,k})$ .

This leads, for each time interval in which we divided the original collection window, to a new metric  $\langle$  time interval, aggregated value  $\rangle$  that we name “ $\Delta T$ -metric” and that we formally indicate as  $\langle (T_{a,k-1}, T_{a,k}), V_k \rangle$ .

The  $\Delta T$ -metrics  $\langle (T_{a,k-1}, T_{a,k}), V_k \rangle$ , with  $k = 1, \dots, N$ , can be taken together to form a  $\Delta T$ -sample. How to further apply composition functions on a  $\Delta T$ -sample is described in Section 4.1.4.

The choice of  $\Delta T$  can range, depending on the aggregation purpose, from very small time scales to very large ones (e.g. one year). From the questionnaire [12], it appears that NRENs do not use a common set of values, (although some values are more commonly used than others), and this makes the comparison of the data unnecessarily complicated. A convergence towards a common set of time windows would ease the comparison of the measured data of different NRENs. We therefore report hereafter the most common values NRENs declare to be using, recommending to align to this set:

- 100 milliseconds
- 1 second
- 1 minute
- 5 minutes
- 15 minutes
- 1 hour
- 1 day
- 1 week
- 1 month
- 1 year

#### 4.1.3.2 Free time window and time instants

In the case the time intervals in which we subdivide the time window  $(T_0, T_f)$  are not equally sized, the time interval are called *free time windows*. An example of measurements where the singletons are related to free time windows are flow related measurements such as the ones provided by Netflow: each flow has a start time, and duration. As the start and duration of the flows cannot be forced to fit into a slotted time window, we cannot deal with them directly in the same way as described in the previous Subsection.

However, when samples are not aligned, they may lead to misleading statistical values. Therefore, it is advised that all data from free time windows is converted into slotted time windows whenever possible. The critical issue is how to deal with metric values with duration spanning two or more slotted time windows (see **Figure 4-1**). There are three approaches:

1. Attribute the measurements to the time slot in which the start time occur. Will shift the data spanning two or more slotted time windows to an earlier time.
2. Attribute the measurements to the time slot in which the end time occur. Will shift the data spanning two or more slotted time windows to a later time.

3. Distribute the amount of the metric values to several slots, weighted by the part of the metric duration that falls within the slot. Will have a smoothing effect on aggregated data.

Approach 3 is probably the most correct, but also the most difficult to implement. In cases where only a very small percentage of the metric occurrences spans multiple slots (thus the time-shift effect is negligible), it is easier to apply approaches 1 or 2.

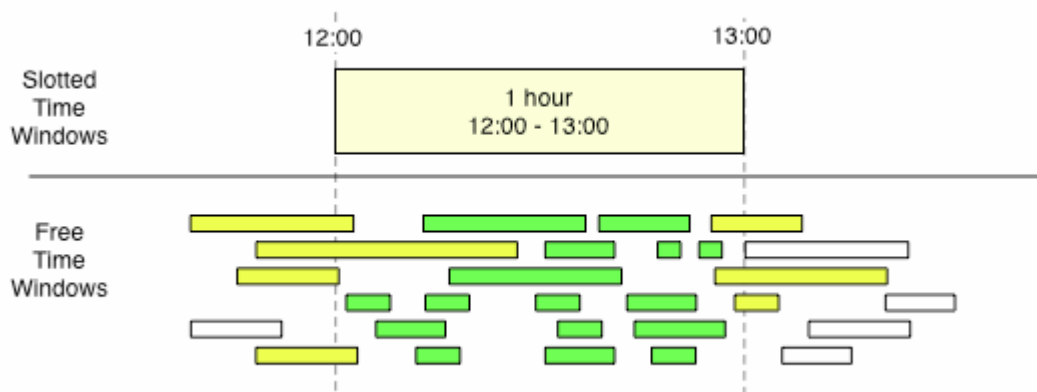


Figure 4-1: Aggregation from free time windows to a slotted time window

Depending on the composition function  $F$ , there may be two types of  $\Delta T$ -metrics:  $\Delta T$ -statistics-metric and  $\Delta T$ -help-metric.

#### 4.1.3.3 $\Delta T$ -statistics-metric

If the  $F$  function is a statistical function (such as average, min, max, percentile, etc.) we say that the result values are a  $\Delta T$ -statistics-metric. This terminology is indeed coherent with [1], where a statistic computed over a sample metric is called a statistic metric (and the sample, in this case, is the subsample  $S_k$ ). By attaching the  $\Delta T$  prefix we emphasise that the statistical function is applied to a sample of singletons collected within a time window of size  $\Delta T$ .

Starting from a sample  $S$  with duration  $T_f - T_0$ , all the  $\Delta T$  such that  $T_f - T_0$  is an integer multiple of  $\Delta T$  can in principle be chosen as an aggregation time slot. However, it is advisable to choose a value within the ones listed at the end of Subsection 4.1.3.1. The only constraint is that each  $\Delta T$  should contain enough singletons to lead to a trustworthy value for the applied statistic function  $F$ . What does “enough” means depends whether a rigorous statistical approach is used or not. If a rigorous method is used, two parameters need to be defined: the error  $\varepsilon$  of the obtained value and its level of confidence  $\alpha$  (Typical values are, for example,  $\varepsilon=0.01$  and  $\alpha=0.95$ ). Once these parameters are fixed, statistical formulas can be applied to verify that, given the sample, the estimation of the applied statistic function  $F$  has an error  $<\varepsilon$  with a level of confidence  $>\alpha$ . Therefore, in general it cannot be said “a priori” if the number of samples is sufficient, although this can be tuned with an iterative approach, i.e. analyse past measurement to have an estimate of the needed number of samples, obtain an estimate of  $\varepsilon$  and  $\alpha$  with actual data, and review the choice of the needed samples. For example, [26] gives some formulas for finding the minimum sample set for estimating delay quantiles. In most practical cases, however, this rigorous approach can not be followed and may be replaced by rules of thumb, which can have a justification in their specific context.

### Delay-average

If  $dT_i$  is the generic delay singleton in a  $\Delta T$  time interval, the composition function  $F$  can be the average of  $dT_i$  in each  $\Delta T$  time interval. The average calculated in one time interval is called a  $\Delta T$ -delay-average<sup>4</sup>. Averages of consecutive time intervals lead to a  $\Delta T$ -delay-average sample.

Example:

We have a sample metric  $\langle 10:00:06, 10\text{ms} \rangle$ ,  $\langle 10:00:24, 9\text{ms} \rangle$ ,  $\langle 10:00:34, 7\text{ms} \rangle$ ,  $\langle 10:00:42, 9\text{ms} \rangle$ ,  $\langle 10:00:55, 10\text{ms} \rangle$ ,  $\langle 10:01:04, 11\text{ms} \rangle$ ,  $\langle 10:01:14, 12\text{ms} \rangle$ ,  $\langle 10:01:34, 8\text{ms} \rangle$ ,  $\langle 10:01:56, 9\text{ms} \rangle$ .  $\Delta T$  is set to 1 minute, and the slotted time windows are (10:00:00, 10:01:00) and (10:01:00, 10:02:00). The 1min-average statistic singleton for the first time slot is  $(10+9+7+9+10) / 5 = 9\text{ms}$ , the 1min-average statistic singleton for the second time slot is  $((11+12+8+9) / 4) = 10\text{ms}$ .

These values together form a 1min-average statistic sample:  $\langle (10:00:00, 10:01:00), 9\text{ms} \rangle$ ,  $\langle (10:01:00, 10:02:00), 10\text{ms} \rangle$ .

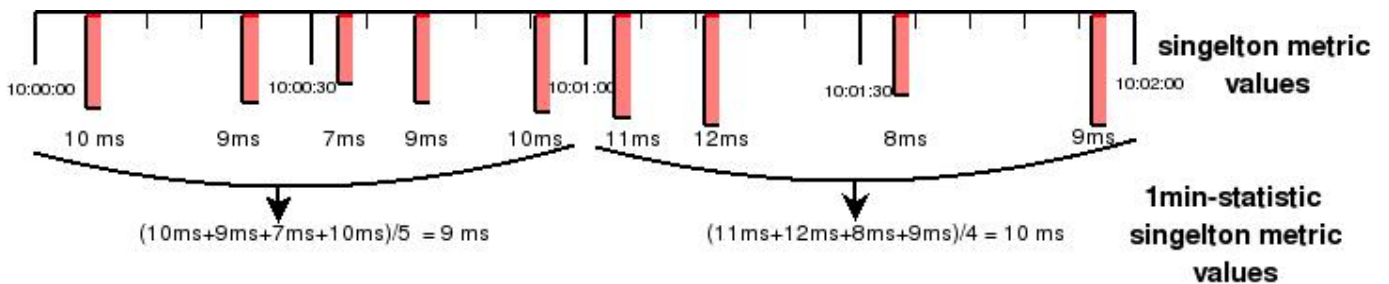


Figure 4-2 Aggregation in time – an example

### Delay-median

The median is the value such that an equal number of samples are less than and greater than the value (for an odd sample size), or the average of the two central values (for an even sample size)

In many cases median is a much more useful aggregate function than the average. This holds true, in particular, when data has a long-tailed distribution. This includes OWD and round-trip-time measurements. In long-tailed distributions, a few measurement values are significantly larger than the average. For example, suppose that most of OWD measurements are around of 15 ms, with a few values larger than 1s. The absolute value of these outliers may heavily impact the average, while the impact on the median will be very limited. Therefore, the median can be better choice for non-symmetrical distributions from a statistical point of view.

### Delay-min-max

<sup>4</sup> It should be  $\Delta T$ -statistics-delay-average, but since it is clear that average is a statistic function, we omit it.

The composition function  $F$  can be the minimum or maximum delay  $dT_i$  of the time interval. The min/max calculated in one time interval is called a  $\Delta T$ -delay-min/max. Min/max of consecutive time intervals leads to a  $\Delta T$ -delay-min/max sample.

#### Delay-X-percentiles

The composition function  $F$  takes the  $X$ -percentile out of the values of  $dT_i$  in the time slot  $(T_{a,k-1}, T_{a,k})$ . Common values for  $X$  are  $X=2.5$  and  $X=97.5$ . The 2.5 and 97.5 percentiles are e.g. used in the RIPE measurements [21].

The same reasoning given for median applies to percentiles, as they are an alternative for the min/max statistics. In a small sample from a long-tailed distribution, the maximum value will generally not be of much interest, since the result is volatile. A 97.5-percentile metric may provide much more interesting data.

#### 4.1.3.4 $\Delta T$ -help metrics

The composition function  $F$  does not always need to be a statistical function. When other mathematical functions are applied to the values of the sample within the time slot  $(T_{a,k-1}, T_{a,k})$ , they can lead to other  $\Delta T$ -metrics called  $\Delta T$ -help-metrics. E.g. the sum of squares can be stored, to later on calculate the standard deviation, and the raw data aren't needed any more. Help metrics, do not have interest *per se*, but it is necessary to store them for performing operations (e.g. step wise time aggregation of statistics) that would otherwise be unfeasible or computationally very intensive.

#### 4.1.3.5 $\Delta T$ -metrics: applicability to metrics

The function  $F$  can be a mathematical function, and can be applied to different metrics. An overview is given in Table 4-1. In general, we remark that:

- The 2.5 percentile is a better indication for the minimum, as it doesn't take extreme values into account.
- The 97.5 percentile is a better indication for the maximum, as it doesn't take extreme values into account.
- In general, the Standard Deviation/RMSD (Root Mean Square Deviation) can give an indication on the spread of the measured values.
- If the IPDV average is not zero, this indicates that the delay over time is increasing or decreasing. In the latter case, this would not be a problem, and the worse that can happen is that the delays converge to the minimum possible. In the former, it would on the contrary indicate that the network is unstable, as the delays are continuously growing.
- Being a distribution centred around zero, for IPDV it is interesting to concentrate on the values that significantly deviate from zero, both on the positive and on the negative side
- The notation "50 Perc. <" means the 50<sup>th</sup> Percentile of the negative IPDV values. The notation "50 Perc. >" means the 50<sup>th</sup> Percentile of the positive IPDV values

Table 4-1: Applicability of the function F for different metrics

| <b>Metric</b>            | <b>F</b>  | <b>Useful</b>   | <b>Comments</b>   |
|--------------------------|---|---|---|
| <b>Delay metrics</b>     | When aggregating in time, there are significant differences between OWD/RTT and IPDV. That's why they have been considered separately hereafter |   |   |
| OWD / RTT                | Average   | Y   |   |
|                          | Min/max   | Limited   | Better to use 2.5 and 97.5 Percentiles  |
|                          | Standard Deviation / RMSD   | Y   |   |
|                          | median  | Y   | Does not suffer from extreme values the same way as the average   |
|                          | 2.5 Percentile  | Y   | But likely to be close to the minimum value   |
|                          | 97.5 Percentile   | Y   |   |
| IPDV                     | Average   | Y   | Should be zero  |
|                          | Min/max   | Limited   |   |
|                          | Standard Deviation / RMS  | Y   |   |
|                          | 50 Perc. < / 50 Perc >  | Y   | This value is the median on the negative resp. positive side of the IPDV distribution. These values hence give "which IPDV to expect. |
|                          | 97.5 Perc. < / 97.5 Perc >  | Y   | Good indication for the maximum IPDV to expect (positive or negative)   |
| <b>Loss metrics</b>      | Average   | Y   |   |
|                          | Min/max   | N   | Since losses tend to occur in bursts, Max risks to be very high when a burst occurs   |
|                          | median  | Y   |   |
|                          | 2.5 Percentile  | Y   | Good indication for the minimum loss, without taking extreme values into account  |
|                          | 97.5 Percentile   | Y   | Good indication for the maximum loss, without taking extreme values into account  |
| <b>Bandwidth metrics</b> |   |   |   |
| Link Utilisation         | Average   | Y   |   |
|                          | Min   | Limited   | If time window is very small, the min would be 0 as Internet traffic is bursty  |
|                          | Max   | Limited   | If time window is very small, the max would be the link capacity, as Internet traffic is bursty                                       |
|                          | median  | Y   | If there is one busy hour, this influences the average. The median then represents the utilisation during non-busy moments.           |
|                          | 2.5 Percentile  | Limited   | Likely to be very low   |
|                          | 97.5 Percentile   | Y   | Indicates the utilisation during busy moments.  |
|                          | Available Bandwidth   | Being link capacity – utilisation, same considerations as above apply, with 2.5 and 97.5 percentiles reversed |   |
| Capacity                 | Rather static. Not suitable for applying statistical considerations   |   |   |
| Achievable bandwidth     | Not likely that tests are performed regularly, their time aggregation is probably of limited utility  |   |   |
| <b>Availability</b>      | Average   | Y   |   |
|                          | Min/max   | Limited   |   |
|                          | Percentiles   | Y   |   |

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

#### 4.1.4 Further time aggregation of $\Delta T$ metrics

As mentioned in Section 4.1.3,  $\Delta T$ -statistics metrics or  $\Delta T$ -help metrics can form samples that can be further aggregated. When for example  $\Delta T$ -statistics-metric with  $\Delta T = 1$  day are calculated, these values could be used to calculate  $\Delta T$ -statistics-metric with  $\Delta T = 1$  month. We call this *further* order aggregation, as it builds further on the results of the first step of aggregation made via the function  $F$  instead of using the raw data set. What distinguishes *first* and *further* order aggregation is that while (by definition) all statistic computations (e.g. averages, variances, max/min, percentiles) are applicable to first order aggregation, their applicability to this second order aggregation may not always make sense.

When aggregating in time, this further order aggregation means e.g. that when building a “daily view”, we use the already aggregated data for the 24 hours of the day (instead of starting from the original data, probably collected with a higher granularity, e.g. each 5 min). When building a “weekly view” we use the already aggregated data for the 7 days of the week, etc. **Figure 4-3** summarizes this concept that is called “step wise time aggregation”. Note the presence of what we called “help metrics”, i.e. intermediate values that are not interesting per se, but that is necessary to store for performing step wise time aggregation of statistics that would otherwise be unfeasible (in the example, the step wise computation of sample variance).

The motivation behind performing step wise time aggregation is more effective computation. When aggregating to day from hour rather than from original data, supposing a measurement granularity of 5 min, the data set that you have to access and use for computations is 12 times smaller. In further aggregation to week and month the difference is obviously even more noticeable.

Step wise aggregation however cannot be applied to all statistics: Median and  $X$ -percentiles, for example, are impossible to exactly calculate without access to the full data set of metric’s singletons (only an estimation would be possible). **Figure 4-3** shows a practical example of how median and percentiles can be substituted with min, max and average functions beyond hour-aggregation. This is an example of a compromise between full statistical correctness and scalability.

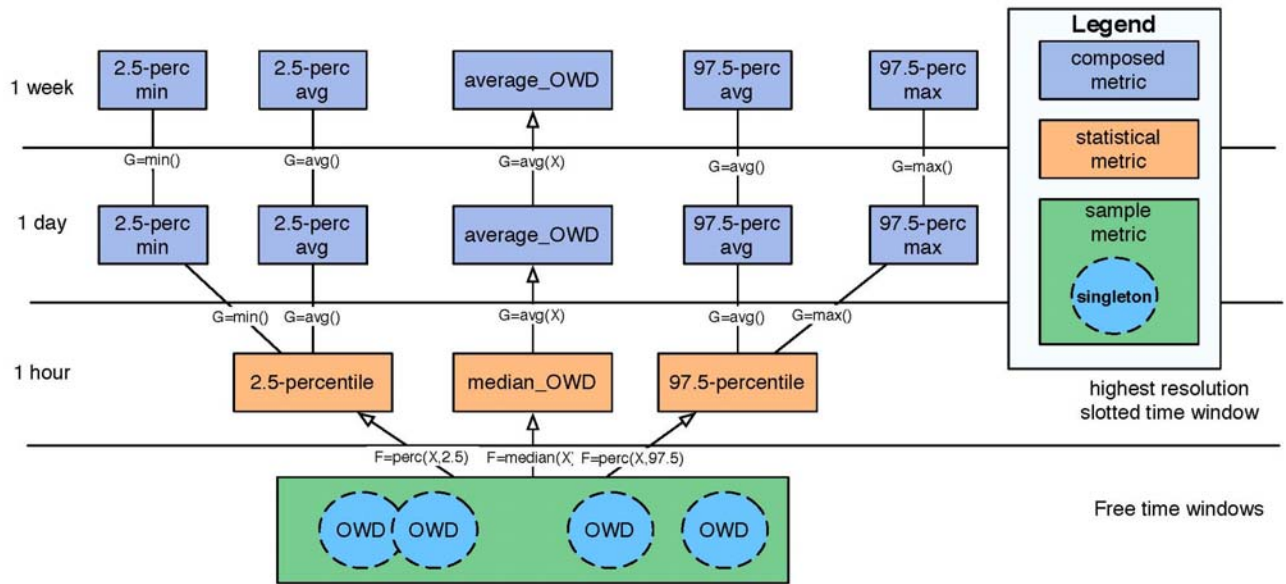


Figure 4-3: Further order aggregate functions

Let us now express more formally the concepts introduced above:

We have N consecutive time slots  $(T_{a,k-1}, T_{a,k})$  in the time slot  $(T_{a,m-1}, T_{a,m})$ , each associated to value  $V_k$ , resulting from a previous aggregation. The values  $\langle T_{a,k-1}, T_{a,k}, V_k \rangle$  form thus a sample metric associated with the time slot  $T_{a,m-1}, T_{a,m}$ . A new aggregated value of the second degree  $W_m$  can be defined:

$$W_m = G(V_k), \text{ for each } \langle T_{a,k-1}, T_{a,k}, V_k \rangle \text{ with } T_{a,m-1} \leq T_{a,k} < T_{a,m}$$

where G is a new composition function and  $V_k$  indicates each  $\Delta T$  metric value of the sample S falling in the interval  $(T_{a,m-1}, T_{a,m})$ .

This leads to a new metric  $\langle \text{time interval, derived value} \rangle == \langle (T_{a,m-1}, T_{a,m}), W_m \rangle$  that can be regarded as a new  $N\Delta T$ -statistic metric. When referring to it, the name of the composition function G shall be mentioned, as well as the steps of resolution (indicating this is an aggregation of the second order).

When statistics of free time windows are available, these can be combined as well. Typically, a weighting will be applied to the values of the statistics, depending of the duration of the free time window.

$$\text{Value}_{\text{normalised}} = ( \text{duration time window } m ) / ( \text{duration all time windows} ) \times \text{Value}_{\text{time window } m}$$

Let's see now some concrete examples of further order aggregation, when the first aggregation is, respectively, an average, a min/max or a percentile. The examples are not exhaustive. More combinations are considered in the summary Table 4-2.

Average

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

When aggregating to a coarser time window, the composition function  $G$  can be the same function as  $F$ , but now applied to the average values  $V_k$  instead of the  $M_i$ .

When we have several consecutive time windows with each their average delay  $\mu_m$ , the average of the metric over the duration of all the time windows can be calculated as the weighted average of the averages  $\mu_m$ . The weight is determined by the duration of a time window, compared to the total time of all the time windows.

Example:

If we have a 1min-statistic sample, already resulting from averaging raw data samples on 1 min slotted time windows:  $\langle(10:00:00, 10:01:00), 9\text{ms}\rangle$ ,  $\langle(10:01:00, 10:02:00), 10\text{ms}\rangle$ ,  $\langle(10:02:00, 10:03:00), 13\text{ms}\rangle$ ,  $\langle(10:03:00, 10:04:00), 11\text{ms}\rangle$ ,  $\langle(10:04:00, 10:05:00), 12\text{ms}\rangle$ , then these statistic values can be aggregated to one new 5 min-statistic  $\langle 10:00:00, 10:05:00), 11\text{ms} \rangle$ .

### Min-max

When aggregating to a coarser time window, the composition function  $G$  can be the same function as  $F$ , but now applied to the min/max values  $V_k$  instead of the  $M_i$ .

The minimum of the minimum value of each consecutive time window is the minimum of the entire time span, regardless of the duration of the time windows. The maximum of the maximum value of each consecutive time window is the maximum of the entire time span, regardless of the duration of the time windows

### X-Percentiles:

When aggregating to a coarser time window, the composition function  $G$  cannot be the same function as  $F$  once the original data have been discarded. In general, there is no closed mathematical formula to perform a composition of the second order on X-Percentiles.

We can deduct some information about quartiles. E.g. if we have two samples with known median  $Md_1$  and  $Md_2$ , we can join the two samples together. The new Median  $Md_n$  cannot be deducted. At most, we can say that  $Q1_n \leq \text{Min}(Md_1, Md_2)$   $Q3_n \geq \text{Max}(Md_1, Md_2)$ . But what can we do with this information?

Example: We have a sample of values [7, 8, 10, 12, 12] and another sample [8, 8, 9, 9, 10]. The median  $Md_1$  is 10, the median  $Md_2$  is 9. If we join both samples, we get [7, 8, 8, 8, 9, 9, 10, 10, 12, 12]. The median is now 9, but how can we deduct this from  $Md_1$  and  $Md_2$ ? We can just state that  $Q1_n$  will be smaller or equal than 9, and  $Q3_n$  will be greater than or equal to 10.

In general, the composition function  $G$  will perform a certain operation on the known values of the percentiles. Other statistics may be used, e.g. taking the average or the min/max values of the 1min-2.5percentiles over one hour.

An overview of the applicability of the function  $G$  (which can be several mathematical functions) on different metrics that are already the result of a previous statistical aggregation is presented in Table 4-2.

Table 4-2: Applicability of the further order aggregation function G

| <u><math>\Delta T</math>-Metric (F-first aggr. function)</u> | <u>G (second aggr. function)</u> | <u>Useful</u> | <u>Comments</u>   |
|--|----------------------------------|---------------|---|
| Average (OWD, IPDV, Loss, other)                             | Average                          | Y             | Averaging the averages leads to the correct overall average. For IPDV it must converge to zero when aggregation time increases, otherwise network is unstable!                                |
|  | Min/max                          | Y             | Selects the busiest / least busy time slot  |
|  | X-Percentile                     | Limited       | May make sense if you want to make a study of the average values over $\Delta T$ instead of the loss, delay, ... values themselves, i.e. observe performances with a coarser time granularity |
| Median (OWD, IPDV, Loss, other)                              | Average                          | Y             | A good compromise, with the advantages of the median, without the scalability issues.   |
|  | Min/max                          | Y             | Selects the busiest / least busy time slot  |
|  | X-Percentile                     | Limited       | May make sense if you want to make a study of the median values instead of the loss, delay, ... values themselves   |
| Min/max (OWD, IPDV, Loss, Other)                             | Min/max                          | Y             | The min/max of the min/max leads to the correct overall min/max   |
|  | Average                          | Y             | Leads to a smoothing of the original min/max  |
|  | Standard Deviation / RMS         | Y             | Gives an idea on the spread over time of the min/max values   |
|  | X-Percentile                     | Limited       | May make sense if you want to make a study of the min/max values instead of the loss, delay, ... values themselves  |
| 2.5-Percentile (OWD)   | Average                          | Y             | Smooths the overall 2.5 percentile, and can be seen as the "average minimum"  |
|  | Min                              | Y             | The 2.5-Percentile can be seen as the minimum without aberrations; taking the minimum of all these values gives the overall minimum without aberrations.                                      |
|  | Max                              | N             |   |
| 97.5-Percentile (OWD)  | X-Percentile                     | N             | Has no relationship with the <i>original</i> percentiles, thus it would be misleading   |
|  | Average                          | Y             | Smooths the overall 97.5 percentile, and can be seen as the "average maximum"   |
|  | Min                              | N             |   |
|  | Max                              | Y             | The 97.5-Percentile can be seen as the maximum without aberrations; taking the maximum of all these values gives the overall maximum without aberrations.                                     |
|  | X-Percentile                     | N             | Has no relationship with the <i>original</i> percentiles, thus it would be misleading   |

### 4.1.5 Histogram composition

A histogram is a measure of the frequency with which metric values fall within a finite number of “bins”. The union of bins must cover the whole possible range of the metric values. A density histogram is a normalized histogram that can be an experimental instance of the statistical distribution of the random variable representing the metric.

When aggregating the same metric in time, if a histogram of the metric values is known for each time slot, a possibility is to combine two or more histograms. This does not lead to the loss of any information, and all statistics can be correctly computed on the resulting histogram. However, this technique has of course scalability limitations.

There are two issues to consider when combining histograms in this way. One is the bin width and the other is the sample size. First, if the bin widths in two histograms are not compatible, then combining them must take this into account. It is possible that the histogram with the finer binning can be recomputed to match the bin sizes in the other histogram. If this is not possible, then one histogram must be modified with possible loss of information. Using simple linear interpolation, it is possible to reconfigure a histogram, but without accuracy guarantees. The next issue with combining histograms is with the sample size. If two frequency histograms are combined with simple addition of corresponding bins, then the resulting histogram will be biased toward the larger sample size. This can be ameliorated by constructing density histograms, i.e. histograms with integral equal to 1.

## 4.2 Aggregation in space

As described in Section 3.2, aggregation in space is the composition of metrics collected in the same time or time windows (temporal scope), but different in physical or/and logical scope.

### 4.2.1 Possible aggregation scopes

The first step is thus to precisely define what scopes have a practical value for aggregation purposes. We foresee the following elementary physical aggregation scopes:

- *Path end point or path intermediate points*
- *Devices*

And the following physical/administrative scopes on which path end points and devices can be further aggregated:

- *Geographical locations*
- *Administrative domains*

In addition,

- *packet header fields*

represent another (logical) scope for space aggregation, “orthogonal” to the mentioned ones.

The space aggregation can be cascaded in multiple ways. For example, someone may firstly aggregate OWD statistics of several paths but relative to a single transport protocol, and then aggregate the results over the several possible protocols. In this document, we do not elaborate further the several possible cascading options.

In the rest of this Subsection (4.2.1) we give more details on the mentioned aggregation scopes. In the other Subsections (4.2.2 and the following ones) we consider how the several performance metrics and their statistics can be aggregated in space. As for the time aggregation, we follow the approach of categorizing the possible space aggregations on the basis of what the starting and ending objects of the aggregations are (e.g. from a sample of singletons to a statistics metrics, or from a sample of statistic metrics resulting from a previous composition to another statistics metric, etc.). Then, within each aggregation category, we elaborate on which metric it can be applied (e.g. delays, losses, etc) and what are the aggregation functions (e.g. statistic functions) that can be used.

#### 4.2.1.1 Path end or intermediate points

Some of the metrics in Section 2 refer to the performances experienced by packets along a *path*, i.e. couple of specific monitoring endpoints (A and B). *Links* are a particular case of paths, where between the end points packets do not traverse any other L3 (IP) device. As such, they do not deserve any special classification, but the term link is frequently used in the following text as well.

We foresee the following three main types of aggregation of metrics relative to a set of paths (See Figure 4-4: tiny links represent physical connectivity, thick ones the path taken by the traffic):

- aggregation for source or destination end point is possible where multiple paths exists with either the same source or destination end point. An example is aggregating all paths with *host A* as destination;
- aggregation by midpoint (for loose paths only) is possible when multiple paths cross the same L3 point. An example is aggregation of all paths crossing *router X*;
- aggregation for source and destination end point is possible when measurements are available for a set of loose paths having the same endpoints but differing for one or more intermediate hops (e.g. because load balancing is applied). Recalling the definition of loose and strict paths given in section 2 (only a pair of endpoints identifies a loose path, while a strict path is identified by all the hops along the path) we can say that this type of aggregation transforms a set of metrics relative to several loose paths in a single one logically relative to a strict path.

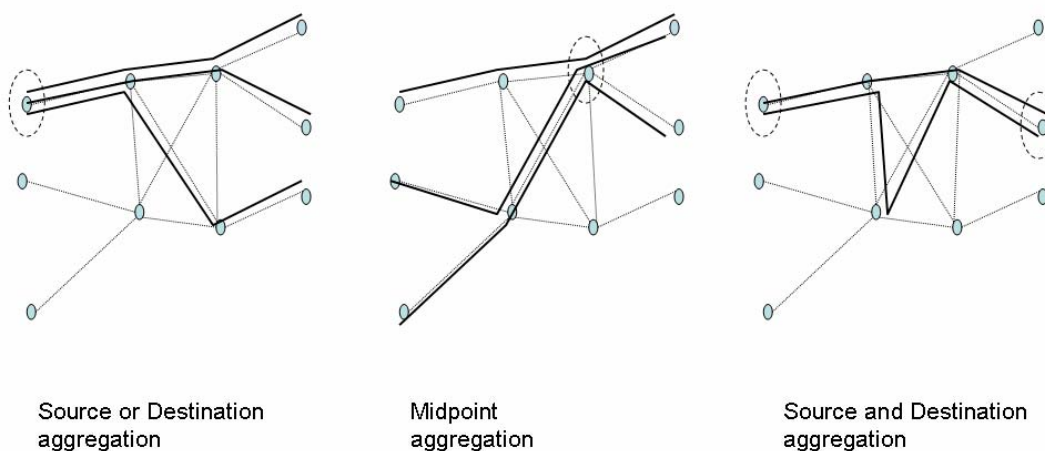


Figure 4-4 Aggregation of paths on the basis of their end or intermediate points

The described aggregations are particularly useful when there is some concern that performances may be bad for all the traffic coming from//going to a specific destination, or crossing a specific router, or being exchanged by two specific sources/destinations

When aggregating performance figures of a set of paths, some sort of weighting may be applied: for example, multiply each path’s metric value by the relative weight of the traffic carried on the path.

When aggregating metrics on a set of paths, some of them may have overlapping links. Links crossed by several paths are likely to dominate the performance figure of the aggregation.

The path aggregation we described assumes the availability of measurement explicitly taken among *couples* of end points. This is different from the aggregation of passive measurements collected at a *single* measurement point on the basis of the IP header source / destination address pair of measured packets. The latter is a typical type of aggregation for flow related metrics, as we describe it in Subsection 4.2.1.5 below.

If an end or intermediate point can be associated to a physical location or administrative domain, aggregated path metrics are good candidates for further space aggregation (see Subsections 4.2.1.3 and 4.2.1.4).

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

#### 4.2.1.2 Devices

A set of metric values of the same type and relative to the same device are aggregated together. The metrics that can be aggregated per device are mostly (but not only) the device specific metrics described in Section 2.2.1. Some examples are the following:

- Temperature from each sensor on the device.
- Load on all CPUs on the device.
- Statistics for all interfaces on the device. (for example, interface errors or flow metrics)

For flow metrics, a common aggregation operation is the grouping of flow records produced by the different interfaces of a router. This is a preliminary operation that falls in the category of “sample to sample” aggregation (see Section 4.2.3): the new sample groups the information for all the flows crossing the router. The further aggregation of these flow records on the basis of the specific characteristics of the flows (e.g. their source and destination IP addresses) is another operation, which is considered in Subsection 4.2.1.5.

A special type of device aggregation is where metrics are aggregated not per “physical” device, but based on device specific meta-data. An example can be; “Average CPU load of all Cisco 3000-series routers in the network”.

If a device can be associated to a physical location or administrative domain, device aggregated metrics are good candidates for further aggregation (see Subsections 4.2.1.3 and 4.2.1.4).

#### 4.2.1.3 Geographical locations

A link has a source and destination, both related to an interface, and interfaces are related to devices. If the device specific meta-data physical location (e.g. in the form of UTM coordinates) is available, location aggregation is useful (Figure 4-5). Locations groups are hierarchical structured groups of physical locations. A physical location can be attributed to one or more group in the hierarchy, e.g. a room, building, campus, city, region or country.

A useful operation is to aggregate paths having physical location of their source and destination belonging to the same group. For example, all the links from a room to room, or a city to city. Storing the aggregated data at each hierarchy level may seem a waste of space, but can be useful because the aggregates are subjects to step-wise aggregation. That means, e.g. that the aggregated room pairs of two cities can be used when aggregating at the city level. By aggregating data to such groups, we will obtain metrics (e.g. the traffic between two cities) that persist in time. These aggregated metrics will not be disrupted when routers are updated, new IP address spaces are used, backup or load sharing links added, provided all the appropriate interfaces / devices are dynamically assigned to the correct group. These aggregations, especially for the groups corresponding to geographically big aggregates (city, country) is more useful for planning purposes than for real time performance monitoring.

Another kind of location groups are ingress and egress groups, which includes all interfaces of a group with an endpoint outside the location. For example, all the outgoing interfaces of a campus.

Also device specific metrics can be aggregated the same way, e.g. on a room, building, campus, city, region or country level.

Location aggregation can also be weighted by meta-data, e.g. by the number of students per university.

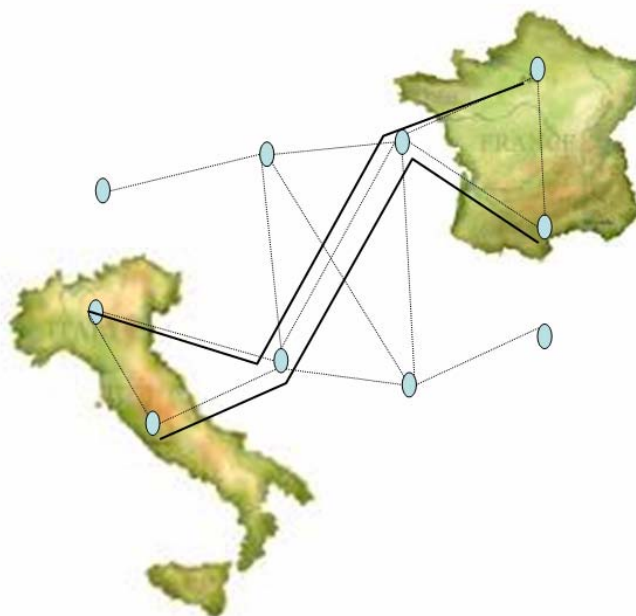


Figure 4-5 Geographical location aggregation

#### 4.2.1.4 Administrative Domains

An administrative domain can be defined as a collection of links and devices that are operated by the same authority. Administrative domains are conceptually identical to the geographical locations defined in the previous section.. The only difference is that the grouping of path end points interfaces is based on administrative properties rather than on geographical ones.

#### 4.2.1.5 Packet header fields

This is a logical aggregation scope. For most of the practical purposes the aggregations that make sense are the following ones:

- Protocol aggregation

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

- Application aggregation
- Subnetwork aggregation

The *protocol aggregation* exploits the “protocol over IP” field of the IP header. The most common protocols are UDP, TCP, ICMP. This aggregation happens when a certain metric (e.g. the OWD) is known separately for each protocol (e.g. because active experiments are made with packets of UDP, TCP or ICMP type) and a summary value is desired.

The *application* aggregation exploits the source or destination ports of the IP header. Several applications are associated to so-called “well known ports” (e.g. http: port 80, ftp port 23, etc.). This aggregation happens when a certain metric (e.g. the OWD) is known separately for each application and a summary value is desired.

The *subnetwork* aggregation exploits the source or destination address of IP headers. The practical utility is when a certain metric is known separately for different couples of (masked or unmasked) src/dest IP addresses and a summary value is desired. Note that depending on the network routing, this aggregation might actually better fit in the “path” aggregation category described above (depending on the src or dst address the end points of a path may be different). However, in some cases it may be useful to aggregate (and compute statistics on) metrics corresponding to packets traversing a common path portions but coming from/going to different subnetworks. For example, a large variance in the OWD of a sample where the path in the core network is the same may reveal that some end campus networks are experiencing problems, or have under-dimensioned access links.

Note that for flow metrics, defining a new flow in a coarser way starting from several more granular flows is a very specific case of aggregation on IP protocol fields, where the applied function is the simple sum of the components flows. Looking at coarser aggregates (like PoP to PoP traffic) is an operation useful for traffic engineering and network planning purposes.

#### 4.2.2 From Singleton to Singleton

Two or more measured instances of a singleton metric of the same nature but different scope can be considered as measured instances of the sample metric. Therefore, aggregation in space of singleton metrics does not exist by definition. The aggregation in space of several singletons is the aggregation of a sample and is discussed hereafter.

#### 4.2.3 From Sample to Sample

Simply enlarging a sample cannot be regarded as an aggregation; it is rather preliminary to it. A typical example is a flow collector that receives and archives flow metrics coming from multiple interfaces. The only relevant thing is that possible meta-data information associated to samples’ singletons may have to be updated.

## 4.2.4 From Sample to $\Delta S$ metrics

To keep a notation similar to the time aggregation case, we say that the aggregation in space starting from a sample metric leads to a  $\Delta S$  metric. In this section, for the several metrics identified in section 2, we investigate what scopes it makes sense to aggregate, and which specific aggregation functions to use.

For space aggregation the starting sample, in many cases of practical interest, may not be a set of singletons but already a set of time-aggregated metrics (e.g. the average OWD from point A to point B over a 5 minutes interval). For space aggregation purposes, however, a time-aggregated value can be considered as a singleton. The only thing to worry about is if it is necessary to associate to each aggregated instance a weight (e.g. the relative traffic of a link compared to the other links considered for the space aggregation). We assume that this additional information (see Section 2.3) is always available.

### 4.2.4.1 Delay (OWD, RTT)

Delay, being intrinsically defined between a couple of points, i.e. on a physical link or path, can be aggregated in space in one of the ways described in Subsection 4.2.1.1.

Average and median are good aggregation functions, helpful to track the delay behaviour in a network over time. To be able to calculate percentiles, a high number of monitored paths is needed (recall that we talk about *space* aggregation, i.e. a single value per each path). However, when delays over a lot of monitored paths are available, the 97.5-percentile gives a good indication of the higher values without taking into account the extremes. The minimum and 2.5-percentile have limited interest, because it mostly will be dominated by the link with the lowest propagation + transmission delay and with empty queues. This delay will be pretty much constant and does not provide any particularly interesting information.

When computing the average, it is necessary to weight the delays with the amount of traffic carried on that path or link to obtain a correct overall figure.

Another way to aggregate delay measurement is on the basis of IP header fields. For example, if measurement are performed on a given link with several different values in the ToS field, it is interesting to compute the average, minimum and maximum values over the several ToS fields and compare them.

### 4.2.4.2 IP Delay Variation (IPDV)

The reasoning for aggregation of IPDV is mostly the same as for delay. The computation of the minimum value of the IPDV of a set of paths makes no sense. In contrast, the computation of the maximum of the IPDV of a set of paths is of special interest. For example, when setting up a multi-user two-way videoconference, the aggregated maximum value of IPDV for the paths between them will be useful for determining the size of the needed receiver buffer.

Note that for IPDV, being a distribution centred around zero, the minimum/maximum and percentiles of interest are referred to the two halves of the distribution, see comments in Subsection 4.1.3.5 and Table 4-1.

#### 4.2.4.3 Packet losses

Packet losses, being intrinsically defined between a couple of points, i.e. on a physical link or path, can be aggregated in space in one of the ways described in Subsection 4.2.1.1.

As regards the aggregation functions to apply, the only statistics of practical relevance are average and maximum. The minimum loss value of a set of paths is likely to be zero, and rarely percentiles are computed in association with losses, since for losses the interest is more in their order of magnitude rather than in their detailed value.

When endpoints are placed at the edge of the network, thus closer to end-users, the average aggregated packet loss rate will be a good indicator for the packet loss experienced by the users.

When aggregating packet loss metrics, it is necessary to weight the packet loss with the amount of the traffic carried on that path or link to obtain a correct overall figure.

A common practice is to perform delay and packet loss measurements with different ToS values. In this case the aggregated average and maximum packet loss rates are useful for comparison. This is a logical aggregation of the IP header fields.

#### 4.2.4.4 Availability

Link availability can be monitored by looking at the state of the router interfaces, while path availability is best monitored through active tests (e.g. ping). The average availability of a set of links/paths is probably the only interesting figure in this case, even if it is just a qualitative parameter: a network may be fully connected even if some links are unavailable, thanks to rerouting strategies. When computing the averages, it makes sense to weight the availability with the capacity of the links. Weighting with carried traffic does not make sense, because when a link is unavailable it will not carry any traffic. The maximum and minimum aggregates of availability do not give very interesting information. In a large network there will often be links that dominates these aggregates by either being constantly down or being near constantly up.

Availability of single components of a device can be aggregated at the device level, and then the devices can be aggregated in geographical or administrative domains. The average availability is probably the only interesting figure in this case, even if it is just a qualitative parameter: a city may be available, i.e. reachable, even if some of its routers are not!

#### 4.2.4.5 Bandwidth

##### Utilisation

On a single link, utilization is defined as the carried traffic over the link capacity. By simple extrapolation, we may define the total utilization on a set of (independent) links as the sum of carried traffic over the sum of link capacities. This is equivalent to say that the total utilisation  $U_{tot}$  on a set of links is the sum of the utilisation on the single links  $u_i$ , each weighted by the ratio of the link capacity  $C_i$  over the total sum of link capacities:

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

$$U_{tot} = \sum_i u_i \frac{C_i}{\sum_j C_j}$$

However, if we try to apply the same formula to a generic network, where the link set may not be independent (i.e. the same traffic is carried over multiple links), we may experience problems. Let's refer to the simple example of the picture below.

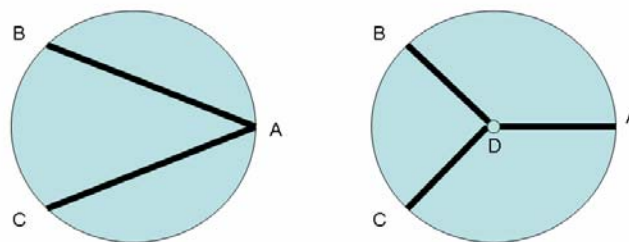


Figure 4-6 Aggregating network-wide link utilisation

Two different networks connect three end points, and they are composed of two and three links, respectively. Assume that all the links have the same capacity  $C$ , and that the carried traffic between AB and AC is  $C_{AB}$  and  $C_{AC}$ , respectively. In the second network, link AD carries a total of  $C_{TOT} = C_{AB} + C_{AC}$  traffic

If we want to aggregate, as suggested, link utilization on the whole networks we would obtain:

$$U_{tot} = \frac{C_{AB}}{C} \frac{C}{2C} + \frac{C_{AC}}{C} \frac{C}{2C} = \frac{C_{AB} + C_{AC}}{C} \quad \text{For the first network}$$

$$U_{tot} = \frac{C_{AB} + C_{AC}}{C} \frac{C}{3C} + \frac{C_{AB}}{C} \frac{C}{3C} + \frac{C_{AC}}{C} \frac{C}{3C} = \frac{2C_{AB} + 2C_{AC}}{3C} \quad \text{For the second network}$$

If  $C_{AB} = C_{AC} = C_{TOT}/2$ , this simplifies to  $U_{TOT} = C_{TOT}/C$  for the first network and  $U_{TOT} = 4C_{TOT}/3C$  for the second.

So, the second network would appear *more* utilized than the first one, even if this contrast with the intuition because it carries the same traffic but has more overall capacity. In fact, if we could ideally directly “measure” the carried traffic between the networks’ end points ( $C_{AB}$  and  $C_{AC}$ ), and directly calculate the network utilization dividing  $C_{AB} + C_{AC} = C_{TOT}$

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

by the network capacity, we would obtain again  $C_{TOT}/C$  for the first network, but  $2C_{TOT}/3C$  for the second. This result (second network is less utilized) is in accordance with intuition.

Clearly, the discrepancy between what we would ideally define for network utilization and what we can infer from link by link measurement is given by traffic that is carried between network end points over multiple hops. Unfortunately, there's no easy way out of it, if the link utilization measurements is the only available information<sup>5</sup>.

From the previous example it is clear that aggregating network-wide link utilisation as proposed *should not* be used to compare two different networks, at least without any knowledge of the network topology and routing. However, it can still be used on the same network for the following purposes:

- compare utilisation figures taken in different time instants (e.g. to assess traffic trends)
- assess the effect of routing changes or temporary reroutings (the overall network utilization, with the same input traffic, should grow if the routing strategies are worse, should decrease if they are better)

In particular for the second purpose, a single summary average value is not enough, because it doesn't give any idea of the spread of utilisation values on the network links. A routing strategy that produces a heavy unbalanced utilisation on the network links is not good, even if the overall network utilisation is low. For that reason, on large networks (with hundredths of links), it would be useful to couple the average utilisation figure described above with the spread between a low and a high percentile.

Aggregation of the utilization is a very good candidate for hierarchical geographical location aggregation, as described in section 4.2.1.3, and is mostly useful for capacity planning purposes, especially if it is cross-compared with similar geographical aggregation of delay metrics. High delays in a highly utilized network portion can suggest the need of capacity increases, while high delays in a lowly utilized network segments rather indicate problems not related to capacity (e.g. equipment misconfiguration or under dimensioning of the equipment processing power).

### Available bandwidth

Since it is the complement of utilization, similar considerations as above apply.

### Capacity

Capacity is often considered an attribute to the network topology rather a variable metric. But another way to look at it, since the capacities of the links are regularly upgraded, is to consider it as a slowly varying metric. In this case it is useful to perform do space aggregation on this metric. If we aggregate all link capacities in the network, the sum of the aggregate will track the growth of the network during time. The average link capacity could also be interesting to track the network history. Aggregating on a geographical basis as described before (e.g. considering the aggregate capacity of the set of links connecting two cities or two countries) the capacity evolution can be tracked. This aggregation is still valid not only for link upgrades among existing points, but also when new links are added or old ones are removed, and pairing it with aggregated performance data (e.g. delays among the same set of geographical

---

<sup>5</sup> If link utilization is coupled with routing information, an estimation of the Origin-Destination traffic matrix can be computed, but this is a complex problem still considered in research, see e.g. [23]

aggregations) it can be understood if the network evolution really helped to enhance performances, and on the basis of that corrections to the planning process can be made.

#### Achievable bandwidth

We do not foresee any practical usage of space aggregation of achievable bandwidth.

#### 4.2.4.6 Device Specific Metrics

Device specific metrics is a generic category including all possible data related to routers. Most common are metrics on temperature, CPU load and memory usage retrieved from the routers using SNMP or via remote shell queries at the *command line interface* (CLI).

Averaging CPU load and memory usage for a set of devices will give indication when a network event causes a correlated increase in memory and CPU usage across multiple devices. Maximum CPU load can be useful, while the minimum is probably not (likely to be very low on at least 1 device).

However, if the averaged set is composed by inhomogeneous devices, the obtained value is often only qualitative.

#### 4.2.4.7 Flow Metrics

Since flow metrics include a value for number of packets and bytes in each flow, along with the start and end time of the flow, they can be used to build utilisation statistics similar to the ones obtained by regularly reading flow counters on routers' interfaces. Even better, flow metrics are more detailed in the sense that they can be distinguished with a metric distinction tuple, e.g. based on IP header fields (some examples are given in Section 2.2).

Space aggregation from interfaces to city, campus or organization and to all interfaces gives interesting statistics (hierarchical space aggregation as described in Subsection 4.2.1.3). Aggregating the flow metric distinguished by field TCP/UDP port field to get averages over all interfaces in the network will give a good overview of services in use on the whole network. Aggregating across multiple interfaces is problematic because the same flows will be included from multiple interfaces. Still, the aggregates can be useful in troubleshooting to track the routing of flows, but how to efficiently do the matching of identical flows from different interfaces is still a matter of research.

Examples of IP header field aggregates that are interesting for flow metrics include:

- Destination interface of flow on device
- Protocol above IP
- IP Source / Destination Address
- Source / Destination AS
- Transport layer (TCP/UDP) source / destination port number

#### 4.2.4.8 Summary

Table 4-3 summarises the discussed space aggregations. Aggregations not having a practical usage are not reported

Table 4-3: aggregation in space: from sample metrics to  $\Delta S$  metrics.

| <b>Metric</b>  | <b>Aggregation Scope</b>                                      | <b>F</b>              | <b>Useful</b>  | <b>Comments</b>  |
|--|---|-----------------------|--|--|
| <b>Delay metrics</b>                                     | Path / geographic locations / Administrative domains          | Average               | Y  | Useful summary value for set of links / path. Must be weighted.  |
|  |   | Min                   | Limited  | Minimum is not interesting for IPDV.   |
|  |   | Max                   | Y  | Especially interesting for IPDV.   |
|  |   | Std.dev. / RMS        | Limited  | Gives a rough idea of the spread of performances, but not particularly useful.   |
|  |   | 2.5-perc              | May be (if many links)   | Not particular for delay, may be interesting IPDV.   |
|  |   | 50-perc / Median      | Y  | Does not suffer from extreme values the same way as average.   |
|  | 97.5-perc   | Y (if many links)     | Good indication of the high values without taking the extreme values into account. |  |
| IP Header fields   | Average   | Y                     | Useful for comparison with min / max   |  |
|  | Min / Max   | Y                     | Useful to verify ToS effect  |  |
| <b>Path or link packet loss metrics</b>                  | Path / geographic locations / Administrative domains          | Average               | Y  | Useful summary value for set of links / path. Must be weighted.  |
|  |   | Min / Max             | Limited  | Min will probably be 0 for most of the time for most of the paths  |
|  | IP Header fields  | Average               | Y  | Useful for comparison with min / max   |
|  |   | Min / Max             |  | Useful to verify ToS effect  |
| <b>Availability</b>                                      | Path / device / geographic locations / Administrative domains | Average               | Y  | Useful qualitative indication, but cannot be used to infer real availability of the aggregates, because of redundancy/rerouting  |
|  |   | Min / Max             | Limited  | Not very useful, since often 0 or 1  |
|  | IP Header field   | All                   | N/A  | IP header cannot distinguish these metric types.   |
| <b>Bandwidth metrics (excluded achievable bandwidth)</b> | Path / geographic locations / Administrative domains          | Average / Sum         | May be   | Indicate average interface speed or utilisation and can be used to measure network growth. More useful if coupled with delay and losses metrics on the same aggregates |
|  |   | Min / Max             | Y  | Indicate span in interface speeds / utilisation for monitored network.   |
|  |   | 2.5 to 97.5 perc span | Y (if many links)  | Compare with average to check if network utilization is unbalanced. Useful for network engineering   |
| <b>Flow metrics</b>                                      | IP Header fields  | All                   | Y  | Actual fields includes: IP address, AS, destination interface, IP protocol and ToS.  |
| <b>Device specific metrics</b>                           | Device / geographic locations / Administrative domains        | Average               | Y/May be   | Useful summary value for set of devices. If devices are inhomogeneous indication is more qualitative   |
|  |   | Max                   | Y  | Extreme values can lead to faults in the future  |
|  |   | Min                   | Limited  | Likely that conditions (CPU load, temperature are normal on at least 1 device)   |

### 4.2.5 Further space aggregation of $\Delta S$ metrics

Further aggregation is not as applicable and useful to space as it is to time. However, there are some cases where it is of interest to perform combined aggregations. For example, after aggregating router temperatures to room averages, select the maximum room average temperature on campus. Another example is creating a histogram of link utilisation percentage between cities.

### 4.2.6 Histogram composition

Nothing conceptually different from aggregation in space. See 4.1.5.

## 4.3 Concatenation in space

Concatenation in space means deducing a metric value on a path knowing only the metric values on a set of non-overlapping sections of the path, that together constitute the full path. Without loss of generality, in the following we will often refer to a path whose end points are A and C, split in two path sections A-B and B-C, where B is an intermediate point in the path. To concatenate the measurements, it is required that:

- The routing of packets sent directly from A to C is the same as the routing of packets sent from A to B up to the intermediate point B, and of packets sent from B to C from the intermediate point B up to the final point C (see Figure 4-7 below).
- Packets on the several path sections must have the same ToS

Note that the first requirement does not mean that all the packets must take *exactly* the same paths on intermediate points between A, B and C: load balancing (at L2 or L3) *may* be performed, but point B must be an intermediate “regrouping” point for all packets from A to C. If the load balancing applies to all packets sent from A to B (including those sent from A to C and *passing* through B), then the statistical reasoning we present in the following still applies, because the load balancing can be seen as another component of the variation in performances experienced by the packets, like the ones introduced by queuing or processing. However, some active measurement tools used for measuring the performances on the path sections may not have the capability of “well” emulating the real traffic so that the load balancing on the test traffic reflects the real one (e.g. using several different source addresses). But this limitation impacts the accuracy of the measurements themselves, and not only their concatenation.

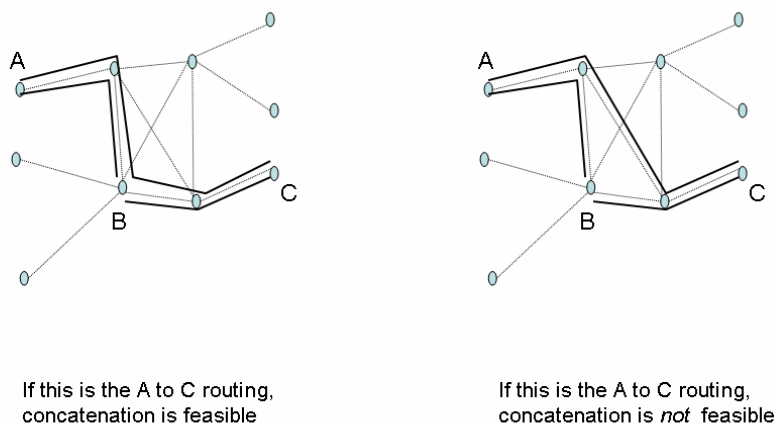


Figure 4-7 Routing constraints for concatenation in space

As regards the measurement time interval, it makes sense to concatenate measurement in space if the measurements taken on the path sections are “close in time” enough so that the network conditions on the path sections are similar. We are not referring to short term conditions, like the one due to buffering in network elements, but to longer term ones. To give a practical counter-example, it probably does not make sense to try to infer the delay on AC, starting from a set of measurement taken on a busy hour from A to B and on a non-busy hour from B to C.

### 4.3.1 Rationale

Deducing metric values through concatenation in space (as opposed to directly take end-to-end measurements) allows better scalability and re-usability of a measurement infrastructure, particularly in a multi-domain environment where the number of end points is large. JRA1, in the specification of the measurement architectural framework [13], envisaged a scenario where some measurements are taken regularly and in a coordinated way between selected domain end points, giving thus the possibility to get a first hint on end-to-end performances without explicitly setting up those measurements. Dedicated end-to-end measurements may be set up when the composition indicates large performance deviations from expected values, or when end-to-end problems are reported and the composition is either not available or does not provide enough reliability.

Beyond scalability considerations, there are also administrative ones. While the architecture defined in DJ1.2.1 [13] explicitly includes the possibility to establish measurements to-from end points belonging to administrative domains different from the home domain of the user/network engineer setting them up, these will be subject to restrictions and

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

local authorisation policies. As a result, a user may be denied an end to end measurement not having the rights to configure one of the end points or because the number of measurements “for external users” on it has exceeded a threshold. Allowing external users to access already existing edge-to-edge measurements results (for performing the composition) is on the contrary of minor concern.

### 4.3.2 Problems

Concatenating measurement values in space is conceptually simple for some metrics, while for others it can be very challenging.

For bandwidth related metrics, loss and error metrics, availability and device specific metrics, it is relatively straightforward. We will give some concatenation rules and formulas in Section 4.3.5, though most of them arise just from simple common sense reasoning

For all the delay-related metrics, on the contrary, concatenation in space is not trivial because there are two potential sources of inaccuracies. The first is that making an (active) measurement means diverting traffic to (or injecting traffic from) some device external to the normal routing path that non-test traffic would follow. These devices are in general connected to the routers via a “measurement network” (typically a switch) that should introduce negligible additional delays with regard to the delays on the path under measurement (Figure 4-8). Nevertheless, the measurement network in B is skipped by packets sent directly from A to C, while it is crossed by packets from A to B and packets from B to C.

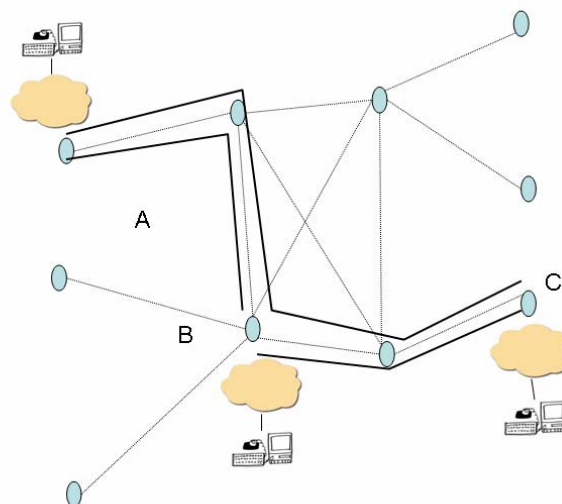


Figure 4-8 Measurement points deployment for concatenation in space

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

The second is that with existing measurement equipment, it is difficult to *widely* perform measurement pairs that are precisely dependent and synchronised with each other. That is, one cannot use the arrival of a measurement packet at B to trigger the start of a measurement packet B→C. Therefore, one must live with the limitation that the measurement results to concatenate are relative to experiments performed on A→B and B→C at different times. This leads to the need of using composition formulas that *assume the independence* between the performances along the two paths. Statistically speaking, this means that the measurement results of A→B and B→C must be considered instances of two independent random variables. We will give in Sections 4.3.5 and 4.3.6 details about what operations can be done to compose metrics under this “independence” hypothesis.

### 4.3.3 From Singleton to Singleton

In the aggregation in time and aggregation in space cases we argued that composing singletons did not make sense “by definition” because when we compose more than one singleton we are actually dealing with a sample (see Sections 4.1.1 and 4.2.2).

On the contrary, for concatenation in space we can argue that two metric values of a singleton metric (e.g. the OWD) on A→B and B→C represent actually just a “partition” of the corresponding A→C singleton metric’s value (if it were available). Therefore, a singleton A→C metric can in principle exist as result of a concatenation operation: in the OWD case it would be the sum of the two OWDs, in the loss case it would be the result of an OR operation (if 1 indicates a loss and 0 the absence of loss on A→B and B→C,  $\text{loss}[A\rightarrow C] = \text{loss}[A\rightarrow B] \text{ OR } \text{loss}[B\rightarrow C]$ ).

However, this is difficult to obtain in practice, because it would require the two measurements to be synchronised in such a way that the arrival of the first measurement packet in B triggers the generation of the second one. Therefore, if it is not possible to claim that the composition of two measurement instances (A→B and B→C) gives *exactly* the same result as if a measurement packet A→C had left A at the same time instant, it is necessary to consider them as part of their respective samples, and compose them statistically as it is described in Sections 4.3.5 and 4.3.6.

### 4.3.4 From Sample to Sample and From Sample to Statistic

For the aggregation in time and aggregation in space case, we argued that composing a sample was only the trivial operation of building a larger dataset out of two ones. For concatenation in space, on the contrary, the operation of putting together the samples would be *incorrect*. In fact, if one has two measurement samples, the first of M values for the A→B OWD, and the second of N values for the B→C OWD, the resulting sample of M+N values is clearly *not* representative of the delay A→C.

For the same reason there is no an equivalent, in the concatenation in space case, of what is described in Sections 4.1.3 and 4.2.4 (Sample to  $\Delta T$  metrics and Sample to  $\Delta S$  metrics). For concatenation in space, it only makes sense to speak about concatenation of *statistics* relative to the several sections of a path. This is addressed hereafter.

### 4.3.5 From Statistic to Statistic

For bandwidth related metrics, loss and error metrics, availability and device specific metrics, concatenation of statistics is relatively straightforward. For all the delay-related metrics, on the contrary, concatenation in space of statistics is not trivial.

#### 4.3.5.1 Bandwidth related metrics

For the bandwidth related metrics described in Section 2.1.4 (apart from the achievable bandwidth) the concatenation result will simply reflect the minimum of the measured metric values on the single path sections. For the achievable bandwidth, concatenating results taking the minimum is correct *only* if referring to UDP traffic. For TCP traffic, the RTT plays a fundamental role and it is not correct to infer any conclusion about the A→C achievable bandwidth given experiments on A→B and B→C.

#### 4.3.5.2 Loss and errors metrics

For the loss and errors metrics defined in Section 2.1.2, composing metrics does not lead to conceptual difficulties either. For one-way packet loss, the following simple formula can be applied:

$$\underline{\text{Loss rate path}} = 1 - \{(1 - \text{loss rate sec1}) (1 - \text{loss rate sec2}) \dots (1 - \text{loss rate sectN})\}$$

(An identical formula applies to errors).

Developing the products in the formula, it's easy to see that the result of the composition is approximately

$$\underline{\text{Loss rate path}} = \text{loss rate sec1} + \text{loss rate sec2} + \dots + \text{loss rate secN} + \text{higher order terms}$$

The higher order terms can be neglected if the single loss rates are small and the number N of composed sections is  $\ll$  than the inverse of average order of magnitude of losses (i.e.  $N \ll \text{avg}(1/\text{loss\_rate})$ ). For example, if  $\text{avg}(1/\text{loss\_rate}) = 1000$ , N should not exceed 100.

For two way packet losses, there is the additional complication that the correct formula to compute the loss on the path would be

$$\underline{\text{Loss rate path}_{2\text{way}}} = 1 - \{(1 - \text{loss rate sec1}_{\text{forward}}) (1 - \text{loss rate sec2}_{\text{forward}}) \dots (1 - \text{loss rate sectN}_{\text{forward}}) \dots (1 - \text{loss rate secN}_{\text{backward}}) (1 - \text{loss rate secN-1}_{\text{backward}}) \dots (1 - \text{loss rate sect1}_{\text{backward}})\}$$

but for each section, the forward and backward component of the two way packet loss may not be available, so some hypothesis should be made on how to split the two way value into two components. It is easy to note, however, that whatever the splitting, if we choose to ignore high order terms the product boils down to

$$\underline{\text{Loss rate path}_{2\text{way}}} = \text{loss rate sec1}_{\text{forward}} + \text{loss rate sec2}_{\text{forward}} + \dots + \text{loss rate sectN}_{\text{forward}} + \text{loss rate secN}_{\text{backward}} + \text{loss rate secN-1}_{\text{backward}} + \dots + \text{loss rate sect1}_{\text{backward}}$$

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

Independently of the chosen splitting, this is trivially

$$\underline{\text{Loss rate path}_{2\text{way}} = \text{loss rate sec1}_{2\text{way}} + \text{loss rate sec2}_{2\text{way}} + \text{loss rate secN}_{2\text{way}}}$$

#### 4.3.5.3 Availability

As regards the availability, a path is available if all the sections of the paths are available. Assuming independency of the availability of the single sections, we have therefore that

$$\text{Path\_availability} = (\text{availability\_sec1}) (\text{availability\_sec2}) \dots (\text{availability\_secN})$$

If e.g. a path is composed by three sections whose availability is, singularly, 99.5%, the path availability will be 98.5074%.

However, we remark that deducing path availability with the formula above will be in most cases incorrect. In fact, IP layer rerouting should normally divert traffic around a failed section. Therefore, a path may be available even if a single section is not. But this actually brings us out of the scope of concatenation in space: as we postulated, concatenation in space assumes that the end points of the sections composing a path are always traversed by packets following the end-to-end path, which is of course not true in case of rerouting (see Figure 4-7). The sum of the availabilities of the path sections give however a useful indication: the frequency of rerouting that happens on the path to guarantee its end-to-end survivability (and rerouting has often a transient negative impact on path's performances).

#### 4.3.5.4 Device specific metrics

For the device specific metrics listed in Section 2.2.1, concatenation in space is nothing different from aggregation in space where the scope of the aggregation is a path.

#### 4.3.5.5 Delay related metrics

We already described in Section 4.3.2 what the two sources of inaccuracy potentially affecting the composition operation for delay-related metrics are. For the first one, represented by the path diversion from the actual one that measurement packets on the path's sections must take (see e.g. Figure 4-3) one must essentially ensure that the estimated delay introduced (or removed) by the diversion is small compared to the measured delays of the A→B and B→C legs.

As regards the second, we already explained that it is necessary to make an independence hypothesis between the delays on A→B and B→C. We further detail here what sort of compositions can be done under this hypothesis. Following considerations apply to both OWD (Subsection 2.1.3.1), Packet Delay Variation (Subsection 2.1.3.2) and Round Trip Time (Subsection 2.1.3.3), as they are all additive metrics, in the sense that

$$\text{Metric}_{AC} = \text{Metric}_{AB} + \text{Metric}_{BC}$$

#### Average

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

Concatenating the average of delay metrics as a simple sum, i.e.

$$\text{avg}(\text{Metric}_{AC}) = \text{avg}(\text{Metric}_{AB}) + \text{avg}(\text{Metric}_{BC})$$

is always correct. This is a simple consequence of the fact that the average of the sum of two random variables is always the sum of the two averages, even if the variables are correlated. Therefore, the above result holds also if the metric values on the two legs *are* actually dependent.

### Variance

Concatenating the variance of delay metrics as a simple sum, i.e.

$$\text{var}(\text{Metric}_{AC}) = \text{var}(\text{Metric}_{AB}) + \text{var}(\text{Metric}_{BC})$$

is correct only under the independency hypothesis, because the general formula is

$$\text{var}(\text{Metric}_{AC}) = \text{var}(\text{Metric}_{AB}) + \text{var}(\text{Metric}_{BC}) + 2 \text{cov}(\text{Metric}_{AB}, \text{Metric}_{BC})$$

and the last term is zero only in the independence case.

### X-Percentiles

The concatenation of X percentiles is much more complicate because it requires the knowledge of the full distribution of the two (or more) components. If these distributions are known, always under the independency hypothesis, the distribution along the full path is the *convolution* of the two distributions.

$$\text{Metric}_{AC} = \text{Metric}_{AB} \otimes \text{Metric}_{BC}$$

Where the convolution, indicated with  $\otimes$ , is a mathematically defined operation that involves integration [15], and therefore can have close forms only if the integrals are solvable<sup>6</sup>.

Despite several studies, performed particularly in relation with ATM networks, commonly accepted and standardized procedures for concatenating delay metric percentiles in space do not exist. This is due to the difficulty of reaching closed formulas even with simple traffic modelling assumptions. However, in the ITU-T revised draft version of recommendation. Y.1541 [14] presents a methodology to compose the 99.9 percentiles of delay distributions on concatenated paths. The given formulas are simple and just require the knowledge of the mean, the variance and of the third moments of the distribution (the third moment of a distribution gives an indication about how asymmetric the distribution is). The modelling assumptions behind these formulas are not detailed in [14], however it seems that the hypothesis is that the delay distributions are “almost” following a normal distribution, but with a correction factor linked to the third moment of the distribution. In fact, if all the distributions are normal (thus with a zero third moment), the formulas simply give the 99.9 percentile of the sum of a set of normal distributions, which can be computed in a

<sup>6</sup> In practical cases the mathematical formula of the distribution may be not known, but an experimental density histogram is available. The convolution operation can then be implemented numerically. It is not difficult, but it has  $N^2$  complexity, where N is the number of the histogram's bins

closed form. According to the editor of [14], the Recommendation document is closed to be approved by the ITU-T. The relevant section of the recommendation and the mentioned formulas are reported in 8Appendix A.

Note that the ITU-T defines as IPDV what in fact is the 99.9 percentile of the OWD distribution, with its origin shifted on the delay value of the packet with the minimum delay. The IETF, in RFC 3393 [6], defines IPDV as the difference in delay of a packet with a reference one. If the reference packet is the packet with the minimum delay on the path, then the 99.9 percentile of the IPDV (in the IETF sense) corresponds to the IPDV (in the ITU-T sense). But the IETF definition allows other choices for the reference packet. A common one is for example to use the previous packet as the reference. This leads to an IPDV that can reflect the “instantaneous” behaviour of a path (a sequence of positive IPDV values indicate a continuous increase of the delay of a path). However, the proposed composition method is general and would be also applicable to the concatenation of the 99.9 percentile of the IPDV in the IETF sense, with the reference packet chosen in whatever way.

### Max-Min

The maximum of a delay metric is not such a representative statistic, and normally percentiles are preferable. However, if the maximum OWD, RTT or IPDV of a concatenation of sections has to be estimated, it is clearly *wrong* to assume that it is the sum of the maximum values on the composed sections. A heuristic approach, but without any theoretical or experimental background behind it, could be the following: select the maximum of the maximum values of the sections and add to it the average values of the other sections.

As regards the minimum, in principle no conclusion could be driven by the single minimum values on the several sections. However, if the single minimum values form a relevant peak in each of the distributions (e.g. 10% of the delay values of a section), under an independency hypothesis it can be estimated if the probability of crossing all the sections with such a minimum delay is not negligible. For example, if there are four sections each with 10% of the OWD values concentrated on they relative minimum (say  $m_1$ ,  $m_2$ ,  $m_3$  and  $m_4$ ), the probability of crossing the whole sections with a delay of  $m_1+m_2+m_3+m_4$  is 0.01%. This has to be compared with the number of packet in transit during the whole observation window to state if a good number of packets had a chance to cross the path with such a minimum delay. If yes, this value is assumed to be as the minimum delay for the concatenation.

### 4.3.6 Histogram Composition

The histogram composition in the case of concatenation in space is conceptually very different from the histogram composition for the aggregation in time and space case (Sections 4.1.5 and 4.2.6).

In the other two cases, histogram composition was indeed a “merging” operation requiring only taking care about bin-width coherence and relative weight. In histogram composition for concatenation in space, on the contrary, one must *not* take into account the relative weight of traffic on the several paths for which the histograms are composed. Then, histogram composition is simply the numerical implementation of the convolution operation described above.

## 4.4 Summary table of network metric composition

For ease of reference, we summarise in Table 4-1 below the requirements, the usability and the most relevant operations that can be performed for the three different composition strategies and for the network performance metrics we defined in 2.1. The table's content has been justified in previous sections. Note that for aggregation in time, we refer only to first order aggregation (thus to the content of Table 4-1, and not to Table 4-2).

Table 4-4 - Summary of requirements, usability and most relevant operations for the three metric composition strategies

|                                 | <b>Aggregation in time</b><br>Aggregate measurements of the same scope and type performed in different time windows or time instants.   | <b>Aggregation in space</b><br>Aggregate measurements of the same type but of different (physical or logical) scope.   | <b>Concatenation in space</b><br>Concatenate measurements of the same type performed on consecutive paths   |
|---------------------------------|---|--|---|
| <b>Definition</b>               |   |  |   |
| <b>Usability</b>                | Reduce the amount of collected data, observe trends.  | Provide a summary metric value for a group of network elements or links in a domain.   | Combine the results from multiple measurements in order to estimate the e2e performances for a longer path. |
| <b>Requirements</b>             | <p>NA</p> <p>NA</p> <p>Measurements should be performed with the same type-packets, e.g. size, ToS, etc. (For space aggregation, this applies to <i>physical</i> space aggregation only. <i>Local</i> space aggregation is by definition over packet properties!)</p> <p>Measurements should be collected during all the time widow. Otherwise, measurements have to be weighted.</p> <p>NA</p> | <p>NA</p> <p>Measurements should be performed in the same timeframe</p> <p>Measurements should be weighted according to the link characteristics (e.g. capacity, utilisation) and their significance</p> <p>Measurements should have comparable accuracy.</p> <p>Operations should be performed over an adequate data set.</p> | <p>Measurements should be taken in consecutive links.</p>   |
| <b>Most relevant Operations</b> |   |  |   |
| <b><u>OWD, RTT</u></b>          | Average, percentiles  | Average, maximum, percentiles  | Average Percentile (but difficult to compute exacty)  |
| <b>IPDV</b>                     | Average, percentiles  | Average, maximum, percentiles  | Average Percentile (but difficult to compute exacty)  |
| <b>Packet Loss</b>              | Average, median, percentiles  | minimum,maximum (ToS effect)   | Average   |
| <b>Available Bandwidth</b>      | Average, minimum, maximum   | Average, minimum, maximum, percentiles   | minimum   |
| <b>Utilisation</b>              | Average, median   | Average, minimum, maximum, percentiles   | NA  |
| <b>Capacity</b>                 | NA (capacity is a slowly varying "metric")  | Average, minimum, maximum  | minimum   |
| <b>Achievable bandwidth</b>     | NA (Not likely that tests are performed regularly)  | NA   | NA  |
| <b>Availability</b>             | Average   | Average  | Average   |

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

## 4.5 Cascaded Time and Space composition

For practical purposes, it is sometimes necessary to perform in sequence (cascade) two of the composition operations mentioned before.

Time aggregation can always be followed by aggregation in space or concatenation in space. It means first aggregating data to a coarser time resolution and after that apply the space composition operation.

Space aggregation can always be followed by aggregation in time. It means first aggregating on a coarser physical or logical scope measurements taken in the same time instants (or windows), and then apply the time aggregation to move to a coarser time resolution. Cascading space aggregation with space composition might on the contrary not make sense if, as a result of the space aggregation, the resulting metric cannot be related any more to a loose or strict path, as defined in Section 3.

Concatenation in space can always be followed by aggregation in time. It means first applying the concatenation operation on measurements set relative to time contiguous (but non-overlapping) intervals and then time aggregate the results to a coarser time resolution. Concatenation in space can also always be followed by aggregation in space, provided that a coherent physical or logical aggregation scope can be defined for the paths resulting from the concatenation in space operation.

There are no general rules for preferring a cascade sequence, to another (e.g. perform aggregation in time before performing aggregation in space or vice versa), but in general the result *depends* on the composition order. In some particular cases however (e.g. if the composition operation is a simple average for the two stages of the cascade, or if it is a max or a min selection), the result will be independent of the order.

## 5 Conclusions

In this deliverable we set forth a framework for a common understanding, within JRA1, of the meaning of network performance metrics and of their post-processing (or *composition*) methodologies.

For the metric definition, we referred both to standards definitions (IETF, ITU-T) but also to common practice of collection and analysis of network metrics in operational networks. We based this work on the replies to a questionnaire circulated within the NREs and on the personal operational experience of some of the contributors to the document.

For the metric composition methodologies, we could only partly rely on standards, since the IETF has so far only marginally addresses the issue, and in the ITU-T the most relevant work is still in progress. Therefore, we classified the possible network metric compositions in three possible categories, called aggregation in time, aggregation in space, and concatenation in space. Then, we investigated for all the possible combinations of a metric and a composition category which are the operations (e.g. statistical operations) that are mostly useful for extracting summary parameters truly representative of network performances. This investigation was based both on abstract reasoning and on common operational practice, and was summarized in a few tables in Section 4, which can be used by the reader as a quick reference.

Part of this classification work was submitted as an Internet Draft [25] to the IETF IPPM Working Group, where some of the contributors to this document stimulated the discussion on network metric composition. The proposal of undertaking work in this area was well received, and some other participants to IPPM have submitted their own Internet Drafts as well. We have the intention to continue with our contributions to the IETF, on the basis of this document and of the follow up of this activity. This is of course well in line with the JRA1's goal of extending performance monitoring beyond the borders of a single domain.

## 6 Future work

While this deliverable sets an initial framework about metric definition and composition, a number of open issues must be addressed for turning it into a “cook book” of set of practical guidelines for developers and adopters of the JRA1’s perfSONAR measurement framework.

The main point to be addressed is the measurement accuracy. The measurements of network metrics, as any other measurements, are affected by errors, and these errors affect computed statistics as well as the results of the metric compositions we described. NREN Network Engineers are of course aware of this issue, but procedures to bound measurement errors (if ever used) are far from being harmonised. This is of course unacceptable in a multi-domain network monitoring environment, if measurement results have to be shared across domains or to be used to check SLAs/SLSs. Specifically, we should give quantitative indications about how to perform active tests (or collect passive measurements) so that the test’s results can be trusted. Concrete examples can be:

- how many OWD samples per minute do we need to collect if we want the error of the estimated OWD average to be bounded with a given level of confidence?
- Similarly, how frequently should we collect link utilization?
- Or how can we estimate the accuracy of Netflow results, when Netflow data are the result of packet sampling with a given sampling rate?
- How frequently can iperf test be done to have a reliable estimate of the achievable bandwidth without injecting too much test traffic into the network?
- How justified is the independency hypothesis for the OWD concatenation of percentiles? For this, we already collected some experimental data on a path from Germany to Italy (Erlangen→Frankfurt→Rome), and we are going to apply the ITU-T proposed formulas (see Appendix A) for composing percentiles on the Erlangen→Frankfurt and Frankfurt→Rome sections, and compare the result with the percentiles obtained with a direct measurement on the full path.

Another point that we need to address is how to practically implement the metric composition types that we described. The summary tables already contain some guidelines, but perfSONAR developers and users would benefit for more details, starting from the raw measurements of specific tools and describing step by step what needs to be one.

Moreover, the specification and/or sample code implementation of some of the most important statistical operations and metric composition operations described here would be very beneficial to perfSONAR developers. The

“Transformation Service”, whose functionality is exactly to manipulate basic measurement results, is in fact one of the services described in the perfSONAR architecture (see [13]) and its detailed developments has still to be undertaken.

We pointed also out in 2.2.3 the potential high benefit of recording and post processing routing information (for troubleshooting and topology discovery). The discussion about what information to consider and how to record it has however just started at the time of this writing.

Another issue of potential interest is the extension of the temporal composition analysis the prediction of future metrics on the basis of past observations, exploiting the time correlation that certain metrics can exhibit. When applied to bandwidth metrics, these predictions could be an alternative to resource consuming achievable bandwidth tests.

Finally, layer 1 and layer 2 network performance measurements and their composition, was only briefly mentioned in this document. Only recently JRA1 has started, jointly with JRA3, to address this issue. As soon as it becomes clearer what performance metrics at those layers are mostly useful in the GÉANT2 network, it will be beneficial to give a formal description of them, their measurement methods and of their composition, similarly to what has been done in this document for layer 3 network performance metrics.

The detailed plan for the follow up of this activity in Y3 and Y4 of the GÉANT2 project is still in discussion.

As we do not see this deliverable as the conclusion of work in the better understanding and usage of network performance metric within JRA1, we will of course welcome any feedback from readers of this document, both within and outside the GÉANT2 community.

## 7 Acronyms

|           |  |
|-----------|--|
| ACM       | Association for Computing Machinery  |
| API       | Application Programming Interface  |
| AS        | Autonomous System  |
| ATM       | Asynchronous Transfer Mode   |
| BER       | Bit Error Rate   |
| BGP       | Border Gateway Protocol  |
| CLI       | Command Line Interface   |
| CPU       | Central Processing Unit  |
| CRC       | Cyclic Redundancy Check  |
| DNS       | Distributed Name Service   |
| DoS       | Denial of Service  |
| EGEE      | Enabling Grids for E-science   |
| GPS       | Global Positioning System  |
| ICMP      | Internet Control Message Protocol  |
| IETF      | Internet Engineering Task Force  |
| IP        | Internet Protocol  |
| IPDV      | IP Delay Variation   |
| IPER      | IP Error Rate  |
| IPPM      | Internet Protocol Performance Metrics (IETF Working Group)                       |
| IPTD      | IP Transfer Delay  |
| ISIS TE   | ISIS w/ Traffic Engineering extensions   |
| ITU-T     | International Telecommunication Union – Telecommunication Standardization Sector |
| JRA1      | Joint Research Activity 1  |
| MIB       | Management Information Base  |
| MPLS      | Multi Protocol Label Switching   |
| NAT       | Network Address Translation  |
| NMS       | Network Management System  |
| NOC       | Network Operation Centre   |
| NREN      | National Research and Educational Network  |
| NTP       | Network Time Protocol  |
| OSPF      | Open Shortest Path First   |
| OSPF-TE   | Open Shortest Path First – w/ Traffic Engineering extensions                     |
| perfSONAR | Performance focused Service Oriented Network monitoring ARchitecture             |
| OWD       | One Way Delay  |
| PERT      | Performance Emergency Response Team  |
| PoP       | Point of Presence  |
| RFC       | Request For Comment  |
| RIPE      | Réseaux IP Européens   |
| RMSD      | Root Mean Square Deviation   |
| RTT       | Round Trip Time  |
| SDH       | Synchronous Digital Hierarchy  |
| SNMP      | Simple Network Management Protocol   |

## Network Metric Report

### Acronyms



|         |   |
|---------|---|
| SONET   | Synchronous Optical NETwork                           |
| SLA/SLS | Service Level Agreement / Service Level Specification |
| TCP     | Transport Control Protocol                            |
| ToS     | Type of Service                                       |
| TTL     | Time To Live  |
| UDP     | User Datagram Protocol                                |
| UNI     | User Network Interface                                |
| UTM     | Universal Transverse Mercator                         |

|                     |              |
|---------------------|--------------|
| Project:            | GN2          |
| Deliverable Number: | DJ1.2.3      |
| Date of Issue:      | 14/02/06     |
| EC Contract No.:    | 511082       |
| Document Code:      | GN2-05-265v4 |

## 8 References

- [1] V. Paxson et al. - RFC 2330 - Framework for IP Performance Metrics - <http://www.ietf.org/rfc/rfc2330.txt>
- [2] J. Mahdavi, V. Paxson – RFC 2678 - IPPM Metrics for Measuring Connectivity – <http://www.ietf.org/rfc/rfc2678.txt>
- [3] G. Almes, S. Kalidindi, M. Zekauskas – RFC 2679 - A One-way Delay Metric for IPPM - <http://www.ietf.org/rfc/rfc2679.txt>
- [4] G. Almes, S. Kalidindi, M. Zekauskas – RFC 2680 - A One-way Packet Loss Metric for IPPM - <http://www.ietf.org/rfc/rfc2680.txt>
- [5] G. Almes, S. Kalidindi, M. Zekauskas - RFC 2681 - A Round-trip Delay Metric for IPPM - <http://www.ietf.org/rfc/rfc2681.txt>
- [6] C.Demichelis, P.Chimento – RFC 3393 - IP Packet Delay Variation Metric for IPPM - <http://www.ietf.org/rfc/rfc3393.txt>
- [7] GGF NM-WG proposed recommendation "A Hierarchy of Network Performance Characteristics for Grid Applications and Services", May 24th 2004 - <http://www.gridforum.org/documents/GWD-R/GFD-R.023.pdf>
- [8] EGEE project definition the basic set of network performance metrics and composite measurements required by GRID middleware - <https://edms.cern.ch/document/475908/1>
- [9] IP Performance Metrics (ippm) IETF working group - <http://www.ietf.org/html.charters/ippm-charter.html>
- [10] IP Flow Information Export (ipfix) IETF working group - <http://www.ietf.org/html.charters/ipfix-charter.html>
- [11] N. G. Duffield, M. Grossglauser, "Trajectory Sampling for Direct Traffic Observation", IEEE/ACM Transactions on Networking, 9(3):280-292, June 2001
- [12] DJ1.1.1: Requirements Report on the Design of the Measurement System (GÉANT2 deliverable, Feb 05)
- [13] DJ.1.2.1: General Framework Design – Single Domain (GÉANT2 deliverable, Mar 05)
- [14] Al Morton et al.: Draft ITU-T revised rec. Y.1541: Network Performance Objectives for IP-based Services
- [15] <http://mathworld.wolfram.com/Convolution.html>
- [16] The Universal Transverse Mercator projection and grid system <http://www.maptools.com/UsingUTM/UTMdetails.html>
- [17] PERT – Performance Enhancement and Response Team <http://www.geant2.net/server/show/conWebDoc.1061>
- [18] K. Tesink - RFC 3592 - Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type <http://www.ietf.org/rfc/rfc3592.txt>
- [19] J. Quittek – RFC 3917 - Requirements for IP Flow Information Export (IPFIX) <http://www.ietf.org/rfc/rfc3917.txt>
- [20] Internet2 Netflow reports <http://netflow.internet2.edu/weekly/20031117/#packsizes>
- [21] RIPE Test Traffic Measurements Service <http://www.ripe.net/test-traffic/>
- [22] ITU-T Recommendation Y.1540 (2002), Internet protocol data communication service – IP packet transfer and availability performance parameters
- [23] S.Vaton et al. "Network Tomography: an iterative Bayesian Approach" ITC 18, Berlin 2003
- [24] D.J1.2.2: Base services detailed design (GÉANT2 deliverable)
- [25] S. Van den Berghe et al. - Temporal Aggregation of Metrics – Internet Draft <http://www.ietf.org/internet-drafts/draft-svdberg-ippm-temporal-00.txt>
- [26] B.Y. Choi, R. Cruz, S.Moon - Practical Delay Monitoring for ISPs, Conext 2005, Toulouse (France) <http://www.sce.umkc.edu/~choiby/papers/conext05.pdf>

# Acknowledgements

We would like to thank Christian Cinetto, Sven Ubik, Harris Laskaridis and Nicolas Simar for their work on Milestone Mj1.2.1 “Choice of metrics” that served as the basis for Section 2. We would also like to thank Stephan Kraft and Roland Karch for the fruitful discussions on concatenation in space, and Al Morton for providing draft versions of ITU-T recommendation Y.1541.

## Appendix A Formulas of draft revised Rec Y.1541

Hereafter we report the section of draft revised Rec Y.1541 that addresses the concatenation in space of IPDV (remember that in ITU-T terminology IPDV is a percentile of OWD, thus this is actually a reasoning about composing OWD percentiles).

The relationship for estimating the UNI-UNI Delay Variation (IPDV) performance from the Network Section values must recognize their sub-additive nature and is difficult to estimate accurately without considerable information about the individual delay distributions. If, for example, characterizations of independent delay distributions are known or measured, they may be convolved to estimate the combined distribution. This detailed information will seldom be shared among operators, and may not be available in the form of a continuous distribution. As a result, the UNI-UNI IPDV estimation may have accuracy limitations. Since study continues in this area, the estimation relationship given below has been specified on a provisional basis, and this clause may change in the future based on new findings or real operational experience.

The provisional relationship for combining IPDV values is given below.

The problem under consideration can be stated as follows: estimate the quantile  $t$  of the UNI-UNI delay  $T$  as defined by the condition

$$\Pr(T < t) = p .$$

### Step 1

Measure the mean and variance for the delay for each of  $n$  Network Sections. Estimate the mean and variance of the UNI-UNI delay by summing the means and variances of the component distributions.

$$\mu = \sum_{k=1}^n \mu_k$$

$$\sigma^2 = \sum_{k=1}^n \sigma_k^2$$

## Step 2

Measure the quantiles for each delay component at the probability of interest,  $p = 0.999$ . Estimate the corresponding skewness and third moment using the formula shown below, where  $x_{0.999} = 3.090$  is the value satisfying  $\Phi(x_{0.999}) = 0.999$  where  $\Phi$  denotes the standard normal (mean 0, variance 1) distribution function.

$$\gamma_k = 6 \cdot \frac{x_p - \frac{t_k - \mu_k}{\sigma_k}}{1 - x_p^2}$$

$$\omega_k = \gamma_k \cdot \sigma_k^{3/2}$$

Assuming independence of the delay distributions, the third moment of the UNI-UNI delay is just the sum of the Network Section third moments.

$$\omega = \omega_1 + \omega_2 + \omega_3 + \dots = \sum_{k=1}^n \omega_k$$

The UNI-UNI skewness is computed by dividing by  $\sigma^{3/2}$  as shown below.

$$\gamma = \frac{\omega}{\sigma^{3/2}}$$

## Step 3

The estimate of the 99.9-th percentile ( $p = 0.999$ ) of UNI-UNI delay  $t$  as follows.

$$t = \mu + \sigma \cdot \left\{ x_p - \frac{\gamma}{6} (1 - x_p^2) \right\}$$

where  $x_p = x_{0.999} = 3.090$ .

As stated earlier, the nature of the IPDV objective is the upper bound on the  $1-10^{-3}$  quantile of IPTD minus the minimum IPTD (i.e., the distribution of IPDV is normalized to the minimum IPTD). The units of IPDV values are seconds, with resolution of at least 1 microsecond. If lesser resolution is available in a value, the unused digits shall be set to zero.