

04-05-2018

D8.6 Network Monitoring / Performance Verification Architecture Production Service

Deliverable D8.6

Contractual Date:	30-04-2018
Actual Date:	04-05-2018
Grant Agreement No.:	731122
Work Package/Activity:	8/JRA2
Task Item:	Task 2
Nature of Deliverable:	OTHER
Dissemination Level:	PU (Public)
Lead Partner:	UoB/AMRES
Document ID:	GN4-2-18-333C1B
Authors:	Pavle Vuletić (UoB/AMRES), Bartosz Bosak (PSNC), Pascal Merindol (Renater), Marinos Dimolianis (GRNET), Henrik Wessing (Nordunet), David Schmitz (DFN), Jerry Sobieski (NORDUnet)

© GÉANT Association on behalf of the GN4-2 project.

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 731122 (GN4-2).

Abstract

This document describes the implementation and organisation of the Network Monitoring and Performance Verification Service. The first part of the document describes the goal of the Network Monitoring and Performance Verification service and the second part of the document focuses on the service operations and improvement.

Table of Contents

Executive Summary	1
1 Introduction	2
2 PVM Service Description	3
3 Service Benefits	4
4 Service Users	4
4.1 Network Service Providers	4
4.2 Network Service Users	5
4.3 Other OSS/BSS Components	5
5 Network Monitoring/Performance Verification Architecture	6
5.1 General Design Principles	6
5.2 PVM Modes of Operation	7
6 PVM Implementation Description	11
6.1 Key PVMv1.0 Components	12
6.2 Network Service Environment	18
6.3 PVMv1.0 Installation Organisation	19
7 PVM Service Operations	20
7.1 Operations and Support Teams	20
7.2 Incident, Problem and Event Management	21
7.3 Request Fulfilment	21
7.4 Facilities Management	21
7.5 Access Management	21
7.6 Technical Support	22
7.7 Supporting Infrastructure	22
8 PVM Continual Service Improvement	22
9 Service Metrics	22
10 PVM Service Roadmap	23
11 Conclusions	23
References	25

Table of Figures

Figure 4.1: Automated test initiation information and process flow	6
Figure 5.1: Mode 1 Architecture	8
Figure 5.2: Mode 2 Fault localisation principles of operation	9
Figure 6.1: Pilot implementation of the PVMv1.0 system	12
Figure 6.2: Example MC screens: Service test dashboard and TMF-compliant service test specifications for the in-production tests	13
Figure 6.3: Example MC screens: PVM device dashboard	14
Figure 6.4: Example of delay and jitter test screens for service 65278 between MA2 and MA3 devices	15
Figure 6.5: Packet information sent from PC to the MCorr	16
Figure 6.6: Sorted and ordered set of data about packets at MCorr	17
Figure 6.7: Summary per-segment statistic information sent from MCorr to MRR	18
Figure 6.8: The topology of the PVM 1.0 installation	19

Executive Summary

The GÉANT Performance Verification and Monitoring (PVM) service provides support for the service assurance processes involved in network service management: Service Quality and Performance Management. It provides a set of functionalities in network service performance monitoring and verification that does not currently exist in the available tools that measure network performance. PVM has a unique capability to simultaneously monitor multiple network services, regardless of the underlying network technology and equipment vendor, and can localise the performance degradation when it happens, separately, in each of the network services.

PVM is not a user-facing service in itself. It is a supporting service to all network services that one provider or a consortium of providers could have. Therefore, its users can be network service operators, or network service users who want to verify the quality of the services they use, as well as other network management software components over the standard-based interfaces. PVM provides automated, pre-production service continuity tests and later in-production, continual, key performance indicator monitoring and tracking, which can be observed by human operators, or can be sent to the other management components in case of performance degradation, which is seen as parameter-threshold violation. Fault localisation is a unique feature that required innovative technical solutions in the design and development phase. It significantly facilitates fault-to-resolution process flows, especially in multi-domain environments.

PVM is being developed as a part of the GÉANT Network 4, Phase 2 (GN4-2) project within the Joint Research Activity 2 (JRA2), Network Services Development. This document provides more information about the PVM service, existing installations, technical description of the system and service description, especially requirements for the efficient service operations, and future roadmap.

1 Introduction

The goal of JRA2 Task 4 (T4) is to develop a generalised, but comprehensive, network monitoring capability that will measure the performance of GÉANT network services, provide almost real-time feedback to network operations personnel or users, determine whether those services are performing to specification, and if not, initiate an automated analysis to localise the fault, and notify the appropriate agent to take corrective action. This monitoring capability, is referred to as performance verification and monitoring (PVM). This document relates to version 1.0 of the PVM system and related service.

The set of GÉANT network services for which JRA2, T4 provides a service monitoring capability includes various types of VPN based on legacy technologies: MPLS-based L2 and L3, point-to-point and multipoint network services, Ethernet-based services, services based on configuration, and activation software developed in a National Research and Education Network (NREN) environment (e.g. NSI-based circuits). However, the JRA2 T4 solution can be used for some future services, such as: services composed of chained service segments (SFC), all the services which are provided on top of the shared network infrastructure like network slices, NFV-based services and those established by SDN.

The environment in which these services are operated is assumed to be multi-domain (different administrative domains) and multi-vendor (various domains use equipment bought from different vendors), with multiple services multiplexed over the same physical links. PVMv1.0 tracks key performance indicators for each service instance/service user separately for various network services: Ethernet-based and IP-based network service metrics as defined in Metro Ethernet Forum [[MEF10.3](#)] and ITU-T [[Y.1540](#)] documents, respectively. At the moment there are no tools or systems which support the set of features listed above.

Target users of the PVMv1.0 system are network service operators, but also knowledgeable network service users who want to verify the quality of services being provided to them before the service is put into production and after that, during its operation. The system provides real-time insight into the key network service performance indicators in form of dashboards and temporal parameter graphs. It is also possible to track key SLA parameters and create periodic SLA reports. PVMv1.0 can be used as a stand-alone tool but is also ready to be integrated into the overall OSS/BSS architecture of GÉANT network services. The integration is achieved using standard TMF interfaces for fetching the information from the service and customer inventories, providing the information to the other components and sending alarms to the other OSS/BSS components if any of the parameters go beyond the predefined thresholds.

PVMv1.0 is based upon the Advanced Service Monitoring/Performance Verification Architecture (ASM/PVM) created by JRA2 T4 in the early stages of the GN4-2 project (Milestone M8.4). The system is built reusing the existing open-source and available components as much as possible and integrating

them into the functional unit with minimal amount of component development from scratch. Such an approach allows easy maintenance of the system, even in environments that have an unpredictable human resource situation and discontinuities in system development. Monitoring tools that aim to monitor service performance previously developed by GÉANT (e.g. perfSONAR) were built with a different use-case in mind and lack some important features like: the ability to monitor various network services at the same time, ability to localise the performance degradation in the network and standard-based interfaces towards other network management support components. However, the development of the PVM is built on the experience and architectural solutions from these tools.

This document focuses on the PVM system as an IT service and defines the key actors and processes in PVM service operations. The document is organised as follows: Section 2 provides a description of the PVM service, followed by PVM service benefits, and users are defined in Section 3 and 4. Section 5 and 6 describes the architecture and provides a technical description of the PVM solution. The remaining sections address service operations and continual service improvement phases in the service lifecycle. Key service metrics and service roadmap for the remaining part of the project are described in the last section.

2 PVM Service Description

PVM is a supporting service to all network services that one provider, or a consortium of providers, could have. PVM supports Service Quality and Service Performance Management processes in the eTOM Assurance process area, especially the following functions:

- Pre-production service continuity tests with standard-based automated invocation and feedback results. These tests are invoked after the network service is preliminary provisioned, to test whether the network service provides the required connectivity and automates this process.
- Continuous monitoring of key standard-based, network-service-performance parameters, such as delay, jitter, loss or availability for each service instance once the network service is in full production.
- Dashboards for quick insight into the network service health and temporal graphs of all key performance metrics for in-depth analysis.
- Ability to detect network service performance degradation end-to-end and localise the faults and segments where the performance is below the required level.
- Ability to send alarm notifications towards human operators and other OSS/BSS systems when there is performance degradation and some parameter is above the threshold.
- Ability to detect network element configuration errors that can happen in any virtual network.
- Periodic service level agreement (SLA) report calculation.
- Automated standard-based monitoring invocation, which enables integration with other OSS/BSS components.

The design of the monitoring system is scalable and the number of monitoring components does not grow with the number of network service instances due to smart network virtualisation use on the monitoring agents which are used simultaneously for multiple service instances.

3 Service Benefits

Performance verification has been recognised as a mandatory part of network monitoring operations since the first models developed for fault, configuration, accounting, performance and security (FCAPS) verified the operation of the network and its services. There are now many tools that monitor network services using various approaches. However, GÉANT's PVM system offers some unique features that are non-existent in the market at the moment:

- PVM is vendor-independent.
- PVM is not tailored for a single network service but can be used for any L2 or L3 service.
- PVM offers unique fault localisation capabilities.
- PVM can be integrated with the other OSS/BSS components using standard-based interfaces, which allows better process automation.
- The PVM system is tailored for virtualised network environments and is ready for new network-slicing technologies and chained services.

4 Service Users

PVM users are:

- Network service providers, which offer monitored network services.
- Network service users.
- Other OSS/BSS components.

The following sections represent the three user groups and individual use cases.

4.1 Network Service Providers

As previously described, the service assurance process area and service quality management are key elements and inseparable parts of network service operations. Network service providers must monitor the network services they provide in order to be able to verify the health of the products they are offering and the underlying infrastructure. This also supports the expansion of the service portfolio and increases customer satisfaction. Since the PVM system is able to monitor each customer and its service instance separately, it can be used to detect configuration errors in some of the service instances. As the proposed PVM system is vendor and service independent, it enables the use of a single system for different types of network services. Further, the system is able to track SLA and contractual obligations in order to create SLA reports. The capability to detect faults can significantly decrease fault-to-resolution process flow times.

4.1.1 GÉANT Community as a Service Provider

The GÉANT community is a unique network service provider environment in which multiple providers provide network services together (e.g. MDVPN service which connects customer endpoints in different NREs via the GÉANT network). The PVM system recognises the specifics of multi-domain operations and allows modes of operation that support both the export of data out of the single administrative domain, or a domain's opt-out of the data export. Further, especially if monitoring zones¹ are full administrative domains, the PVM fault localisation feature allows a fast and easy way to detect a domain with performance problems, and so shortens the time to resolve the issues.

4.2 Network Service Users

Network service users with strict requirements about the performance parameters (like delay, jitter or packet loss) regarding the network services they use need to be able to verify that the user requirements as defined in the Service Level Agreement (SLA) are actually met by the network. The PVM system allows read-only access to the user, which can continuously monitor the overall health of their service through the dashboard or specific parameters.

4.3 Other OSS/BSS Components

Users of the PVM system are other operations support system/business support system (OSS/BSS) components in a well-organised, automated, network service management software portfolio. Typical process flows are:

- **Initiating service testing and monitoring** (Figure 4.1): In this process flow, the service provisioning component requests either a pre-production test or service monitoring using the standard TMF Service Test API for a specific service instance supplying references to the Service Test Specification and respective service instance from the PVM (step 1 in Figure 4.1).

The PVM component then uses the reference to the Service Test Specification, and queries, the local or external Test-API compliant service (step 2) to get a set of predefined settings needed to configure a specific type of a test (e.g. capture method, basic metrics and thresholds)

Based on the data collected by that moment, the PVM component gathers the required details of the specific service instance from the Service and Resource Inventory using Service Inventory API (step 3). The details include service termination points and other technical details needed for setting up the service (VLANs, IP addresses and so on).

In addition, SLA data and threshold values can be gathered from the inventory, or dedicated SLA component. With this information and the information stored in the internal database (step 4) PVM is able to automatically setup all components of the system in a way which allows

¹ The concept of the monitoring zone is described in more details in Chapter 5.2.2.

service instance monitoring (steps 5 and 6). All the components of the system described in Figure 4.1 are described further in Section 5 and 6.

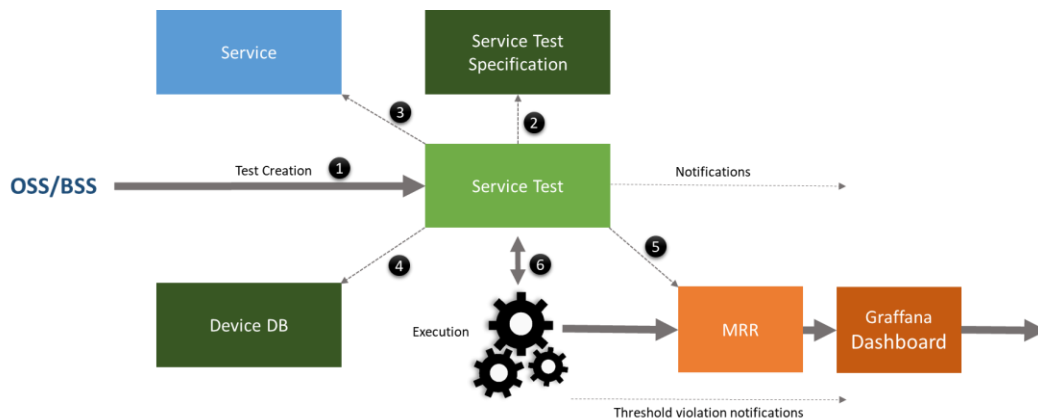


Figure 4.1: Automated test initiation information and process flow

- Alert in response to a KPI-threshold violation:** In this process flow, the PVM system tracks all performance indicators and compares them with predefined threshold values. If any of the values crosses the threshold, an alert is created and sent. Standard alerts create a trouble ticket using the TMF Trouble Ticket API, however, JRA2 T4 decided to use more lightweight approach of sending formatted emails to the trouble-ticketing system and/or Slack messages, as this does not require changing the existing trouble-ticketing system. Other criteria for sending an alert to reduce the number of alerts can also be configured, such as crossing a given threshold value in n consecutive tests.

5 Network Monitoring/Performance Verification Architecture

5.1 General Design Principles

In the initial stages of the GN4-2 project, JRA2 T4 analysed various types of network services and the way network performance metrics, such as delay in L3 MPLS VPN or in Ethernet circuits, can be standardised, especially the localisation of performance faults for each service user in these environments. As part of the research summarised in Milestone M8.4, there is no single network performance measuring platform or even methodology that is suitable for all types of network services within the scope of PVM and with similar features (i.e. with fault localisation capabilities) as those developed in JRA2 T4.

On one side, passive reading data from the network elements is not suitable for delay and jitter measurements, and there are still no good vendor-independent tools which can monitor these parameters for each network service separately. On the other side, active probing can provide end-to-end metrics, but fault localisation requires insight into the performance metrics at the intermediary

points in the network (borders of the so-called monitoring zones) for each separate network service instance.

This data cannot be gathered from the active probes, but instead, requires gathering data from the packets that cross the network service instance path. The approach, which consists of a combination of active and passive monitoring is called hybrid methods [[RFC7799](#)] and is used in PVM.

Since PVM architecture is not created from the point of view of a network equipment vendor with access to network elements and its software stack,

Unlike equipment vendors, with access to the network element software stack, it is not possible to change or implement the system onto network elements. Therefore, PVM passive observing at intermediate points in the network involves capturing packets, which are either gathered on a mirror port of a network element or from a tap element deployed on a link.

Depending on the type of the network service, in some favourable cases, performance metrics can be deduced by simply capturing packets belonging to the specific service instance: e.g. MPLS L3VPN, Kompella and Martini MPLS L2VPNs have inner labels which allow the distinction of packets belonging to different users/service instances, while for the other services like Network Service Interface (NSI)-based on E-Line and OpenNSA, or EoMPLS Virtual Private Wire Service (VPWS), this is not possible since the labels are changed at each hop. For the latter services (NSI and E-Line) it is required to capture user traffic combined with specially crafted active probes which carry service identifiers so that per-service monitoring is possible. In order to allow various network services to be monitored by the same system, PVMv1.0 has three different modes of operation which are described in the next chapter.

5.2 PVM Modes of Operation

PVM system works in three different modes of operation:

1. Mode 1: PVM provides only end-to-end network service performance metrics, and no-fault localisation capability. Metrics provided are typical standard metrics described in ITU-T [Y.1540] and MEF specifications [MEF10.3] (delay, jitter, loss, availability, reliability). Mode 1 uses active probing.
2. Mode 2: PVM provides the same end-to-end performance metrics as Mode 1, plus the possibility to detect network segment, which contributes to the performance degradation (fault localisation). This is achieved by capturing **probe packets** at some strategically chosen points in the network (borders of the monitoring zones) and correlating the timestamp information for the same packet as it flows through the network.
3. Mode 3: PVM provides the same end-to-end performance metrics as Mode 1 and 2, plus the possibility to detect the network segment which contributes to the performance degradation by capturing the **user's packets** at some strategically chosen points in the network. Capturing user's traffic in this way enables different types of fine-grained traffic analyses: per address,

per flow, etc. A lightweight version of Mode 3 can be chosen with passive capturing of user traffic only at the edges, however, this fails to provide sufficient t per-segment insight.

Modes 2 and 3 require capturing active probe packets in Mode 2, or a user's traffic in Mode 3 at a strategically chosen set of points in the network. As capturing traffic on its way through the network may raise some privacy and security concerns, a user can choose the mode of operation depending on his/her needs and readiness to accept packet capturing as a part of the monitoring methodology. Generally, it is not possible to enable fault localisation in the network services without any implication of the intermediary points in the network, or in this case, packet capturing. In all modes, the PVM system is designed to work in multiple domains, respecting the principles of controlled/restricted export of monitoring information.

The three modes are described further in the following sections.

5.2.1 Mode 1

Mode 1 assumes sending probe packets between Monitoring Agents (MA), which sit on the edges of the network services (Figure 5.1).

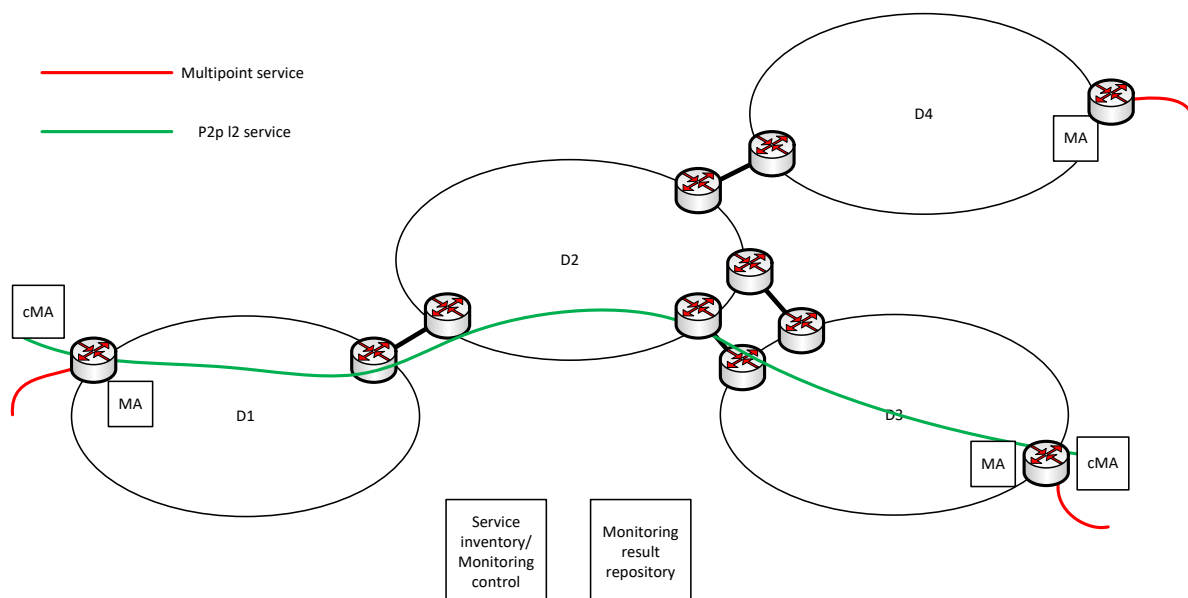


Figure 5.1: Mode 1 Architecture

Monitoring agents (MA) are physical or virtual devices, typically connected to the Provider Edge (PE) network devices (routers/switches or virtual devices to which customers devices are connected). Each MA can be multihomed: connected simultaneously to multiple service instances, which exist on the particular PE. Such an approach significantly improves the scalability of the solution. For a moderate (several tens, up to one hundred) number of network services, a single MA can be used at each PE point. MAs use a standard-based OWAMP [OWAMP] protocol for measuring performance parameters, and send separate OWAMP probes over each network service instance. This way, each network service is separately monitored, and performance parameters are obtained per-service instance. If there is any network service that does not allow injecting probe packets into the network service

instance at the PE device, there is an option to use customer MA (cMA) devices in the customers network. cMA have a very small footprint (virtual machine or a single board computer) and fully automated configuration.

5.2.2 Mode 2

Mode 2 provides the same end-to-end performance metrics as Mode 1, plus the possibility to detect the network segment that contributes to the performance degradation (fault localisation). ETSI defined a process of fault localisation in virtualised environments, which starts with service sectionalisation and per-section testing [ETSI]. In PVM fault localisation is achieved by capturing probe packets at some strategically chosen points in the network and correlating the packet timestamp information for the same packet as it flows through the network. This concept appeared in some recent research and commercial systems (e.g. Ericsson Diamond), which defined the concept of the monitoring zones as subsets of network elements where the performance of the zone can be estimated by capturing packet ingress and egress to the zone and analysing key performance indicators from these captures [Diamond]. The same problem was addressed by the IETF, but with the assumption that the intermediate network elements are going to implement the detection of the blocks of the packet which are alternatively marked in order to allow delay and loss measurements on intermediate monitoring zones [ALTMARK].

The density of the packet capturing points (or the size of the monitoring zone) depends on the desired granularity of the fault localisation and the total cost of the solution. The density or size of the monitoring zone can range from very small monitoring zones, which consist of a single network segment, where packets are captured at each network element along the packet path and performance parameters are calculated per network segment, to the monitoring zones, which correspond to the whole network (e.g. NREN network).

In the latter case, if multi-domain network services are being used, it is possible to detect which domain contributed the most to the performance degradation of the service. Such a feature is particularly useful for debugging multi-domain services, as appropriate corrective action can be requested from the domain responsible for the performance degradation.

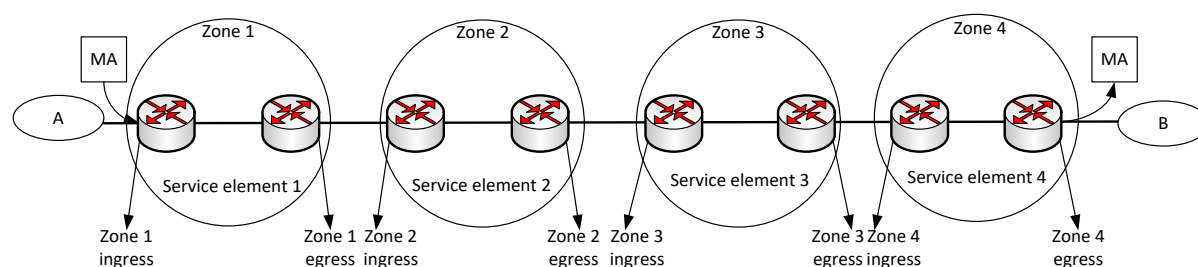


Figure 5.2: Mode 2 Fault localisation principles of operation

PVM methodology of fault localisation is described using an example shown in Figure 5.2. This figure shows a network path (e.g. network service as aL2 point-to-point connection between A and B), which consists of four monitoring zones. The two MA devices inject probe traffic into the point-to-point connection. Probe packets are then captured at the borders of each monitoring zone and timestamped.

Once the probe packet is captured at the ingress or egress point of the zone, the packet capturing (PC) device timestamps it, computes its fingerprint (a pseudo-unique hash or a deterministic subset of header fields) to enable the detection of the same packet at distinct locations on their flow through the network, detects the service it belongs to, and sends the correlator (MCorr) densely packed reports with the summary information of each packet (timestamp, hash, service test ID). MCorr gathers the report data from multiple PCs and then looks for the same packet captured in different PCs by searching for the specific packet fingerprint. It then calculates per-zone monitoring performances by analysing the timestamps from different PCs.

Probe packets are not sent as-is with the full payload from the PCs to the correlator. This way, the amount of information sent for each packet is reduced to a fraction of the original size of the packet (to at worst 10% of the original size). This is especially important for the environments with small zones where one packet can be captured many times on its way through the network. Packet capturing and sending a summary to the correlator in Mode 2 does not significantly increase the amount of traffic in the network, as it only gathers the active probe packets, which are sent in rates of about several tenths per minute, per service instance. Also, since only probe packets are being captured, there are no concerns about privacy violations or unauthorised access to the contents of the user's traffic.

One significant challenge is determining that a captured probe packet belongs to a particular network service instance. As previously stated, for some services such as MPLS L3VPNs, there is an inner label that is unique for one service instance, and at each packet capturing point, packets belonging to one service instance will have the same inner label. However, there are network services like the OpenNSA-based E-Line, which use a single label, changed at each hop. Such services make it impossible to rely on the labels for the detection of the service instance a packet belongs to. Therefore JRA2 T4 developed an extension of the OWAMP protocol and software suite, which injects the service ID into the OWAMP probe packets. This extension is fully compliant with RFC4656, as it allows the existence of padding in the probe packets which can be filled with arbitrary content of varying length [RFC4656]. This way each MA encodes the service ID and other relevant information into the OWAMP probe packet and achieves the automated detection of the service instance for any type of network service, even for those that consist of multiple chained network service types.

Figure 5.2 shows a trivial topology, which has only one path between A and B. In reality, there are multiple, possible paths between two endpoints and topology can change due to the faults of some network components. While Mode 1 inherently supports changes in service paths since probe packets take the same route as user's traffic and are observed at the service end points, the PVM system is able to detect path changes and give per-segment performance indicators in cases of path changes in Mode 2 as well.

As previously discussed, the use of packet capturing as a key prerequisite for fault localisation raises some privacy and security concerns. However, each domain manages the entire infrastructure it owns and can deploy a policy on its network elements that restricts mirroring traffic only to active probes and excludes user traffic from the data capture. If there are domains which are still reluctant to use packet capturing in their domain, they can opt-out from the fault localisation in their domain by simply not installing the PCs. If the neighbouring domains use fault localisation, in that case it would still be possible to detect faults in the domain which opted out.

5.2.3 Mode 3

The operation of Mode 3 is similar to the operation of Mode 2, except that there is no active probe traffic, but user traffic is captured at the borders of the monitoring zones. This mode of operation brings additional stress to the packet-capturing devices, requires dedicated capturing hardware on high-speed links, significantly increases the amount of monitoring traffic from the capture devices to the correlator, and exposes the content of the user's traffic to the monitoring system. Moreover, it is required to work jointly with the Mode 2 to get necessary information about the Service Test ID for each instance.

Alternatively, Mode 3 is the only method that provides full insight into the performance of user traffic without the interference of active probe packets and enables fine-grained analyses such as those that provide the performance of a particular user, application or protocol. However, due to the aforementioned negative effects of capturing user's traffic, Mode 3 is seen as a method which is not going to be used while the service is in production, but in testing and debugging phases. Mode 1 and 2 can be used for both pre-production and in-production tests.

5.2.4 PVM v1.0 Capabilities

The first version of the PVM system v1.0 supports Modes 1 and 2.

6 PVM Implementation Description

The initial PVM v1.0 implementation focuses on testing per-network service instance performance monitoring and verification and fault-localisation in multi-domain network environments with paths that can be dynamically changed. Figure 6.1 represents all the components of the pilot implementation, which are described in the following text. All components are implemented in GÉANT/GTSv5 environment.

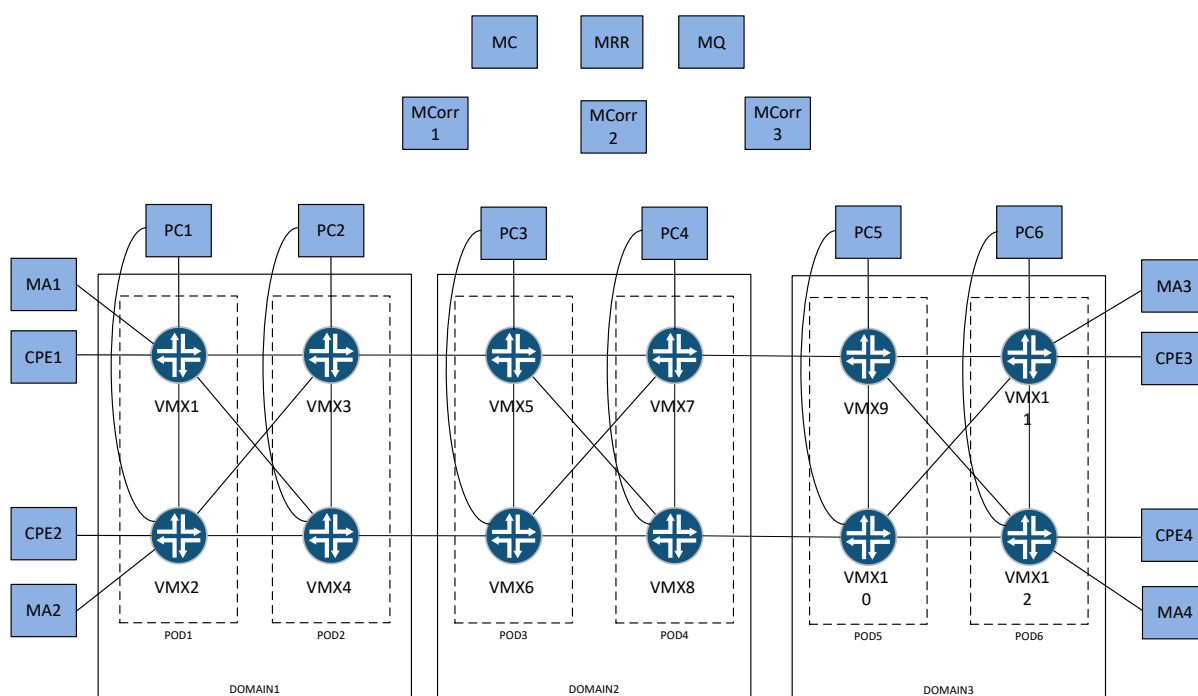


Figure 6.1: Pilot implementation of the PVMv1.0 system

The topology described in Figure 6.1 allows the dynamic provisioning and decommissioning of MPLS L2 and L3 network services in the same technological setup as the GÉANT multi-domain environment (three domains can be seen as two NRENs and GÉANT in between) with the routers which have the same set of capabilities as those in GÉANT network.

6.1 Key PVM v1.0 Components

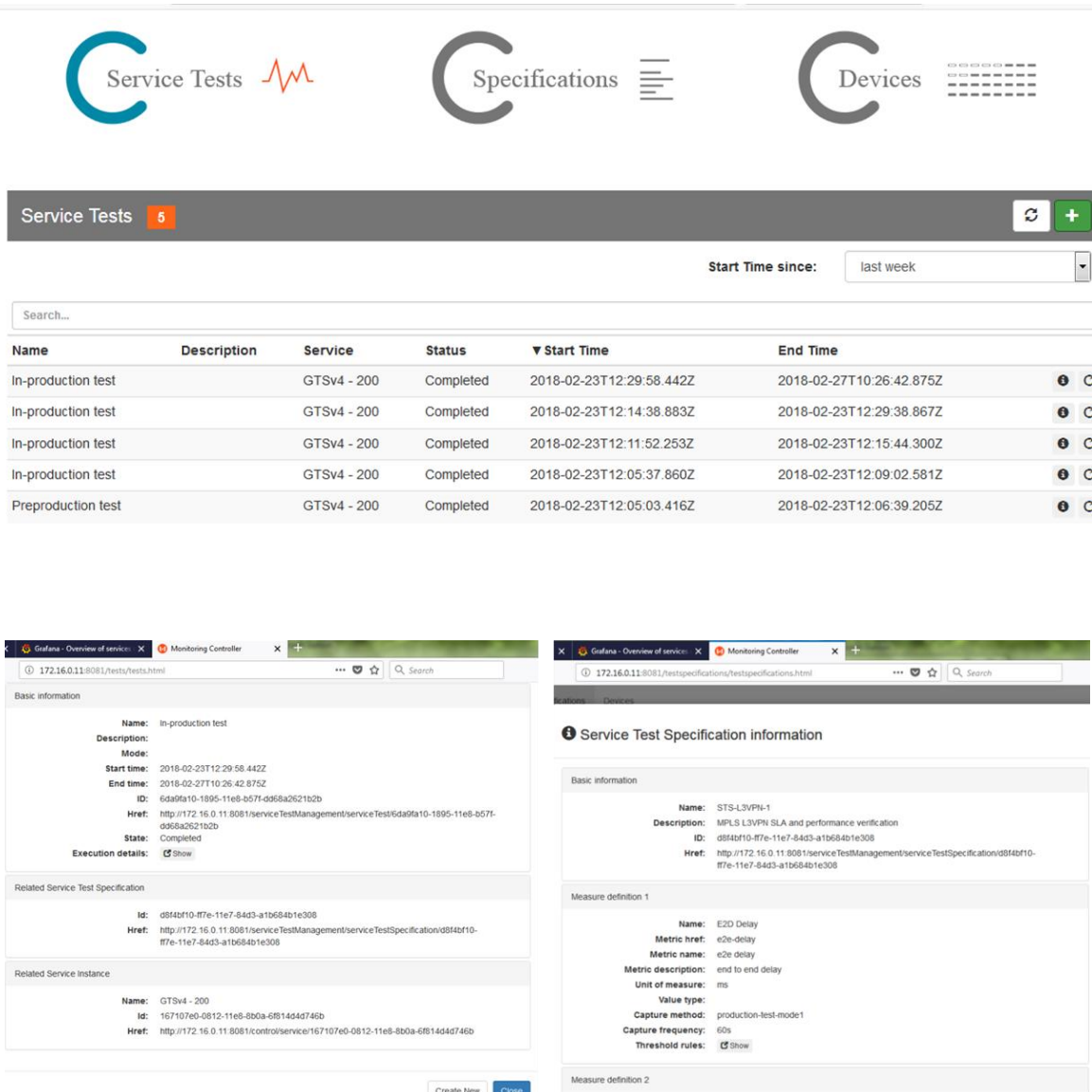
6.1.1 Message Queue

PVM components use Advanced Message Queuing Protocol (AMQP) message protocol for the internal communication, in particular, rabbitMQ software. It supports various communication patterns between the components and message delivery guarantees that even if some components that read the data fail during the operations. This way the task was relieved of the need to define its own protocol or to use some heavyweight APIs for inter-component communication.

6.1.2 Monitoring Controller

The top-level component in the architecture of PVM is Monitoring Controller (MC). It is a REST TMF-compliant service, which couples external components with the PVM ecosystem and manages all network monitoring tests executed in the PVM environment. The MC is built upon three basic classes of objects, namely: Service Test, Service Test Specification and Device. The first two classes are adopted from the TMF Test API specification [TestAPI]. Service Tests are a representation of the actual monitoring actions performed for network services in the PVM environment. Service Test

Specifications define parameters for Service Tests, e.g. an expected mode of operation or thresholds for monitored characteristics. Whenever a new service needs to be monitored, an interested party defines a new Service Test using a selected Service Test Specification. The Devices (database of PVM components) are required for internal configuration. They represent specific elements of the PVM infrastructure (particularly monitoring agents (MAs) that should be configured in order to perform monitoring actions. A set of devices that need the configuration is selected for a given service instance during the Service Test preparation phase. After that the data relating to the Service Test is pushed to the Monitoring Result Repository (MRR) using the defined REST interface. The actual configuration of Devices in MC is invoked with the help of Ansible automation tool using a set of predefined Ansible playbooks (Ansible roles).



The screenshot displays the Monitoring Controller (MC) interface with three main navigation tabs: Service Tests, Specifications, and Devices. The Service Tests tab is active, showing a dashboard with a search bar, a table of service tests, and detailed views for a specific test and its specification.

Name	Description	Service	Status	Start Time	End Time
In-production test		GTSv4 - 200	Completed	2018-02-23T12:29:58.442Z	2018-02-27T10:26:42.875Z
In-production test		GTSv4 - 200	Completed	2018-02-23T12:14:38.883Z	2018-02-23T12:29:38.867Z
In-production test		GTSv4 - 200	Completed	2018-02-23T12:11:52.253Z	2018-02-23T12:15:44.300Z
In-production test		GTSv4 - 200	Completed	2018-02-23T12:05:37.860Z	2018-02-23T12:09:02.581Z
Preproduction test		GTSv4 - 200	Completed	2018-02-23T12:05:03.416Z	2018-02-23T12:06:39.205Z

The detailed view for the 'In-production test' shows the following information:

- Name:** in-production test
- Description:**
- Mode:**
- Start time:** 2018-02-23T12:29:58.442Z
- End time:** 2018-02-27T10:26:42.875Z
- ID:** 6da9a10-1895-11e8-b571-d958a2621b2b
- Href:** http://172.16.0.11:8081/serviceTestManagement/serviceTest/6da9a10-1895-11e8-b571-d958a2621b2b
- State:** Completed
- Execution details:** Show

The 'Service Test Specification information' view shows:

- Name:** STS-L3VFN-1
- Description:** MPLS L3VFN SLA and performance verification
- ID:** d84df10-f7fe-11e7-84d3-a1b654b1e308
- Href:** http://172.16.0.11:8081/serviceTestManagement/serviceTestSpecification/d84df10-f7fe-11e7-84d3-a1b654b1e308

The 'Measure definition 1' view shows:

- Name:** E2D Delay
- Metric href:** e2e-delay
- Metric name:** e2e delay
- Metric description:** end to end delay
- Unit of measure:** ms
- Value type:**
- Capture method:** production-test-mode1
- Capture frequency:** 60s
- Threshold rules:** Show

Figure 6.2: Example MC screens: Service test dashboard and TMF-compliant service test specifications for the in-production tests

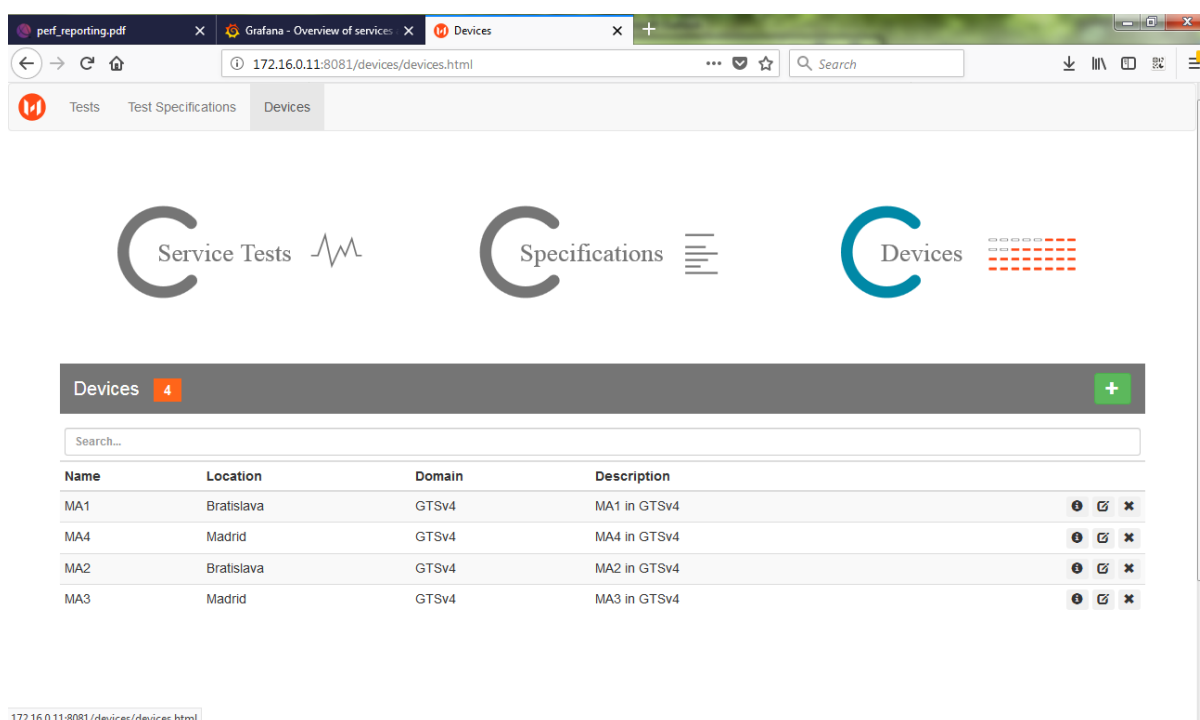


Figure 6.3: Example MC screens: PVM device dashboard

Along with the MC service, there is also a complementary Monitoring Controller web portal developed on top of AngularJS and Bootstrap technologies. It allows using functionality of MC service to create and manage Service Test Specifications and Devices as well as to conduct Service Tests.

Within the PVA environment, MC is deployed on a dedicated machine with preinstalled Ansible ($\geq 2.4.2$), MongoDB ($\geq 3.6.1$) and NodeJS ($\geq 8.9.4.1$) software packages. In order to enable remote execution of the Ansible's tasks, it is important to enable SSH communication from the MC host to all Devices.

6.1.3 Monitoring Result Repository

The Monitoring Result Repository (MRR) is the main component for gathering, storing and displaying monitoring data as it arises from Monitoring Agents (MAs), the Monitoring Controller (MC) and the Monitoring Correlator (MCorr). In operation Mode 1, data obtained from MAs regarding network metrics (delay, jitter, loss) is collected and stored with the corresponding Service ID and Service Test Specification ID. In operation Mode 2, the MRR acquires the data related to a Service Test of a monitored service from the MCorr (via the message queue MQ). The MC provides the Monitoring Result Repository (MRR) with SLA thresholds for active service tests. To this end, dashboards depicting all the relevant performance metrics and RAG indicators for threshold violation are displayed. In Mode 1 network metrics related to a service between MAs are presented and used as hyperlinks to temporal graphs, where a more detailed view is supplied. In Mode 2, the presentation of temporal graphs of network metrics is used, grouped by domain. Dynamic dashboards and alerting capabilities offered by Grafana (and its plugins) are exploited to fit the needs for presentation and threshold violation of active services.



Figure 6.4: Example of delay and jitter test screens for service 65278 between MA2 and MA3 devices

MRR leverages Grafana v4.5.0 (Data Visualisation) and its integration with InfluxDB v1.5.0 (Time-Series Database), as also Python (both 2.7 and 3.5) and JavaScript custom scripting for data consumption (pika/influxdb python's rabbitMQ/InfluxDB client) and visualisation enhancements. The implementation of a REST API was necessary for communication with MC using TinyDB and HUG (as per REST use described in Section 6.1.2).

6.1.4 Measurement Agents

Measurement Agents (MAs) are Linux virtual machines used in Mode 1 and Mode 2 network service monitoring. MA devices are used as sources and sinks of active probe traffic which is being used to measure key performance indicators (delay, jitter, loss) end to end. Each MA device can be in multiple VPNs simultaneously, using VLANs in separate network namespaces on the connection towards the respective vMX router. This way it is possible to inject the probe traffic into all the user VPNs and monitor the performance of the user service instances.

MA devices require the installation of the modified one-way ping (OWAMP) code, capable to store Service Test identifier and other relevant information in the probe packets. This approach enables per-service instance fault localization in Mode 2 through capturing probe packets encoded with specific Service Test ID. Dependencies for the modified OWAMP are the same as for normal OWAMP: I2util. MC configures MAs upon receiving the request to monitor new service instance using ansible playbooks which are filled with a set of dynamic data obtained from the service and resource inventories. MA devices also require Python 3.5+ because of the Pika package required for sending messages with results towards the MRR through the AMQP message queue (MQ).

6.1.5 Packet Capturers

In the testbed described in Figure 6.1, packet capturers (PCs) are Linux virtual machines used in Mode 2 (and Mode 3 network service monitoring). They have interfaces attached to mirror ports of the vMX routers. PCs filter particular Mode 2 probe packets from the traffic tapped at these interfaces, as well as any user traffic corresponding to a user's overlay network service in Mode 3.

The capturing is based on extended Berkeley Packet Filter (eBPF) using a recent Linux kernel (at least v4.10). It is implemented in Python + C (for eBPF code itself) based on bcc-0.3.0 (including Python wrapper library). For Mode 2 each captured Mode 2 packet is directly forwarded towards the MQ using the Pika package.

6.1.6 Monitoring Correlator

The Monitoring Correlator (MCorr) is the central component of the per segment verification capabilities, a feature that enables inter-domain fault localisation. It is developed from scratch in Python3 (with Pika and NumPy libraries). Standard performance metrics such as delay, loss and jitter are computed at least at the domain granularity (one monitoring zone equals one administrative domain). For all monitored network services, if edge routers of all traversed inter-domain paths deploy PC, the MCorr is able to provide such useful indications to troubleshoot the network service.

The MCorr correlates all reports about captured packets from all packet capturers (PCs) for all monitored network services and sends the per-zone performance data to the MRR. It uses a simple sampling method that subdivides the monitoring time in short periodic chunks of constant time size s (in seconds). The MCorr considers that a packet is lost if it does not retrieve its trace in a moving window of n consecutive chunks ($n > 1$) in particular, in any report of the last PC on the packet path assigned to a given flow/network service. For Mode 2, this last PC is the destination MA, while, for Mode 3, it is the PC closest to the network service ending point known by the PVM system.

If the packet is lost, its statistics are computed only between the PCs that have detected the packet (at least the source MA and/or the PC close to the network service ingress point). To verify the performance of the path (made of monitoring zones) of a given packet belonging to a given network service for a given Service Test ID, the MCorr uses its unique packet identifier $P-id$ as the index of PC reports. This identifier consists of a packet fingerprint based on a hash calculation on the packet (header and payload) or a deterministically unique value such as a sequence number (in particular, for Mode 2). Packet capture information is sent from PC via the RabbitMQ queue that is processed by a parsing function of the MCorr. Reports coming to MCorr from the PCs have the format described in Figure 6.5.

```
P-id (Packet ID: an unique packet digest per n chunk),
F-id (Flow ID: IP src-dst addresses),
MA-id (internal labels or global unique IPs for src-dst of probe packets),
Service-test-id (an internal identifier describing the monitored network
service),
PC-id (an internal label or global unique IP),
Timestamp (having great accuracy possibly)
```

Figure 6.5: Packet information sent from PC to the MCorr

In order to extract standard metrics about the performance of each network service, the MCorr computes and process this spatiotemporal dataset in three steps:

1. It spatially structures the data considering the entry/exit termination points of each network service (detects the path of a directed flow). The MCorr sorts the packet reports it retrieves for each PC to compute ordered per flow and per segments statistics at the chunk scale. This sorting process uses two kinds of information: with Mode 2, entry/exit points can be provided within the reports, while in other cases (Mode 3 or intermediate segments), for a given flow, the order of PCs can be deduced from packet timestamps which have sufficient accuracy to Determine the path from the increasing ordered set of timestamps for each *P-id* of a given flow in a given chunk (this order should be consistent for all packets belonging to a given flow).

```

Ci: (the temporal bounds of chunk I, i.e., [t,t+s])

Service-id1:
  F-id1 (src-dst): (the first tunnel of first service)
    MAsrc1: (TFirst), (the starting index)
    PC1.1: (T1), (first ranked PC)
    PC1.2: (T2), (second ranked PC)
    ...,
    MAdst1: (TLast); (the ending index)

  F-id2 (src-dst): (the second tunnel of first service)
    MAsrc2: (TFirst), (the starting index)
    PC2.1: (T1), (first ranked PC)
    PC2.2: (T2), (second ranked PC)
    ...,
    MAdst2: (TLast); (the ending index)
  ...
Service-id2:
  ...

```

Figure 6.6: Sorted and ordered set of data about packets at MCorr

2. Having accurate entry/exit point information (which is known in Mode 2) significantly eases the indexing and basic metric computation. A monitored flow should necessarily be forwarded through those termination capturing points, i.e., all paths from (F-id (IP src)) to (F-id (IP dst)) enters at the ingress MA (or the first static PC in mode 3) and ends at egress MA (or the last static PC in Mode 3) during the entire flow lifetime. For chunk *i*, *C_i*, the MCorr parses all reports with timestamps which belong to the time window $[i, i+n]$ of *n* consecutive chunks to build a new data structure that is organised at the Service-Test-id / F-id granularity (Figure 6.6). In this structure, for each network service/flow, packets belonging to each network service/flow are sorted in the order of their timestamps.

The comparison between the sets of timestamps can be implemented in several ways, e.g. a partial order instead of a total, to finely model path changes with a better granularity than the chunk time period. For a given flow, the existence of a consistent ordering can be checked and computed efficiently, if necessary (to scale for Mode 3).

This step is challenging when path changes occur, with some anomalies, i.e. with routing loops (a graph with cycles should be computed) or if multi-path capabilities are enabled with a per-packet, round-robin scheduling. After detecting packets belonging to some network service

and sorting it in time-increasing order, the MCorr performs multiple statistical analysis for each tunnel and service. It provides the average, the median, minimum and maximum value for each standard performance metric for any monitoring zone (segment) including the E2E.

```

Ci:
  Service-id:
    F-id:
      E-2-E (MAsrc-MAdst, delays (avg, med, min,...), loss (rate), jitter (avg,...),...),
      Seg1 (MAsrc-PC1, delays (avg, med, min), loss (rate), jitter (avg, med, min)),
      Seg2 (PC1-PC2, delays (avg, med, min), loss (rate), jitter (avg, med, min)),
      ...
      Segk (PCK-MAdst, delays (avg, med, min), loss (rate), jitter (avg, med, min));
...

```

Figure 6.7: Summary per-segment statistic information sent from MCorr to MRR

3. Finally, the MCorr sends all these results for chunk i ($C_i := \text{timestamps } (t, t+s)$) in a JSON format to the MRR (again via the MQ but with a refined JSON feed (Figure 6.7) that processes this aggregated data with Influx DB (time series manipulation) and Grafana (metrics visualisation). An outgoing chunk thus consists of a new summarised distribution for s seconds for all monitored tunnels.

6.1.7 vMX Routers

vMX routers are virtual machines with Juniper virtual vMX router software installed. These routers have the same set of features as actual routers of Juniper MX series and allow the implementation of all types of VPNs. The test environment has L2 and L3 MPLS VPNs installed, both point-to-point and multipoint.

6.1.8 Customer Edge Devices

Customer edge devices (CE) are Linux virtual machines which act as Customer Edge devices in VPN scenarios. Each CE device can be in multiple VPNs simultaneously, using VLANs in separate network namespaces on the connection towards the respective vMX router. This way it is possible to generate traffic which will pass the network through different VPNs, simulating the traffic of multiple users in the network.

6.2 Network Service Environment

The testbed described above will have multiple instances of various types of VPNs installed:

- Multipoint L3VPN
- Point-to-point L3VPN
- Multipoint L2VPN (VPLS)
- Point-to-point L2VPN (VPWS)

This way it will be possible to test the behaviour of the system in the environment with multiple service instances with various packet encapsulation types and different network service topologies.

Furthermore, since there are redundant paths in the network between the vMX routers, it will be possible to shut down some of the links and create situations in which the path between the customer endpoints is changed, and performance parameters affected (e.g. increased delay and jitter). This way, it will be possible to test fault localisation capability in realistic conditions.

6.3 PVM v1.0 Installation Organisation

PVM version 1.0 is installed in version 5 of the GÉANT Testbeds Service (GTS) network in six different GTS Points of Distribution (PoDs) across Europe in the following order: London, Bratislava, Paris, Milan, Hamburg and Prague (Figure 6.5). The domains are grouped in the following way:

- Domain 1: London and Bratislava
- Domain 2: Paris and Milan
- Domain 3: Hamburg and Prague

Such network allows testing the system on a network with real distances and delays, an environment which is completely identical to the GÉANT production network.

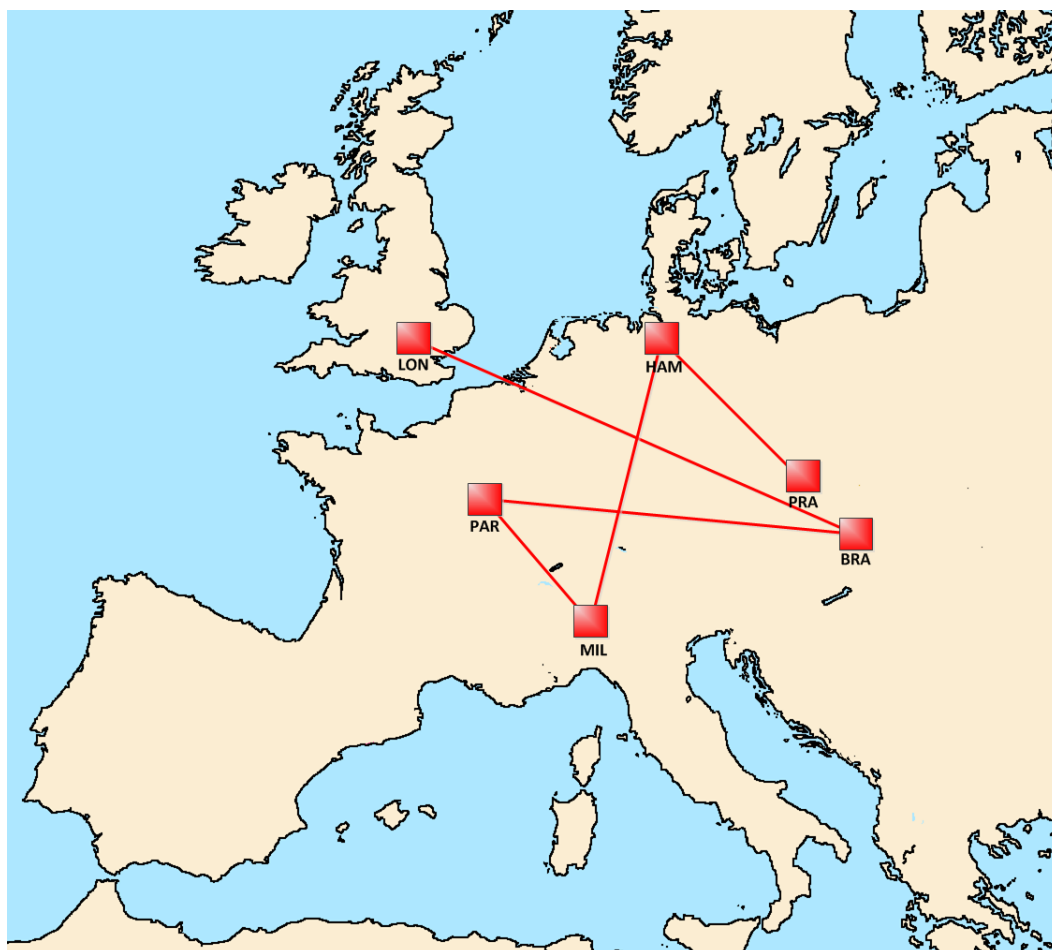


Figure 6.8: The topology of the PVM 1.0 installation

It is possible to verify that PVMv1.0 supports the following features:

- Monitoring network service instances of various types (MPLS L2, L3, point-to-point and multipoint) end to end.
- Monitoring network service end-to-end in situations when the path between the endpoints is changed.
- Ability to detect network segments which cause the performance degradation – fault localisation.
- Automated pre-production and in-production test invocation and decommission.
- Automated generation of service dashboard and temporal graph operation.
- Alarm generation in case of performance parameter threshold violation.
- The operation of the PVM service overall, reliability and availability.

The testbed can be accessed by the JRA2 T4 task members who operate the PVM system, other JRA2 activity members who can observe and assess the performance data obtained in the GTS environment, SA1 and SA2 activity members who are the target users of the system in the support of the GÉANT network service portfolio.

7 PVM Service Operations

The objective of the PVM Service Operations is to make sure that PVM service is delivered effectively and efficiently. The Service Operation lifecycle includes the fulfilling of user requests, resolving service failures, fixing problems, as well as carrying out routine operational tasks. These processes are described in more detail later in this section.

7.1 Operations and Support Teams

There are four different actors that support the PVM service:

- An IT team supports each of the domains where the PVM system is installed. The IT teams are responsible for the operation of the underlying infrastructure (servers, virtual machines, network elements, etc.) where the PVM is installed.
- An Operations team takes care of the PVM service in production, i.e. manages, maintains and monitors the installed components of the PVM system, as well as provides support for the users. It can be contacted at pvm-ops@lists.GÉANT.org. This team is involved with Service Activities of the GÉANT project and will follow the procedures defined by the PVM business development and service management and developers team.
- A PVM Developer team translates the input from the operations, business development and service management teams into future software and service versions, and also adapts the PVM system to the regular upgrades of the underlying software infrastructure. The team can be contacted at pvm-dev@lists.GÉANT.org, primarily for technical and implementation-specific topics. This team is part of the Joint Research Activities of the GÉANT project.

- A Business Development and Service Management team tracks the usage of the established service, gathers feedback and input from the user groups and actively takes part in the future-service roadmaps. The team's focus is on adapting service to new emerging technologies and spreading the system to new set of users. The team is available at pvm-team@lists.GÉANT.org.

7.2 Incident, Problem and Event Management

These processes handle various incidents, problems and events that might occur during the service operation. The goal of the incident and problem management processes is to return the service into the fully operational state as soon as possible, while the event management monitors the health of the service, classifies events when they occur upon the impact they might have to service operations and alarms the appropriate actors, when needed. The operations team is responsible to receive and manage incident and problem reports and to continuously monitor the health of the PVM service. If corrective action beyond the capability and procedures of the operations team is needed, incidents, problems and events are escalated to the PVM Developers team.

7.3 Request Fulfilment

Users might have requests for customisation and minor changes in the way PVM system operates like e.g. changing the layout of the service dashboard, changing the destination of the alarms, or changing the way service endpoints are displayed. Requests are received by the Operations team, and changes are handled by the PVM developers team, as well as testing the customised system.

7.4 Facilities Management

Facilities Management manages the physical and virtual environment where the PVM components are located. GÉANT IT team and/or NREN IT teams are responsible for facilities management in their respective domains. Any plans for major changes in the status of the underlying infrastructure have to be reported to the PVM operations and developers' team in order to enable a smooth transition to the new environment.

7.5 Access Management

The access management process grants authorised users the right to use PVM service, while preventing access to non-authorised users. Access restrictions are built in the PVM system.

7.6 Technical Support

Technical support for the users is organised in two levels. First-level support, which assumes instruction for users on how to deploy the system and use it in the most efficient way, is handled by the PVM Operations team. For the more complex issues, technical support requests will be escalated to the PVM developers team.

7.7 Supporting Infrastructure

Based on the service architecture and technical description presented in the previous sections, the following supporting infrastructure was used for the PVM service:

- The read-only access to the infrastructure monitoring systems so that the Operations team can check the health of virtual machines, systems and services that host PVM components,
- An issue-tracking system for incident, problem, event handling and technical support cases.

All of these components are established, validated and tested before the service transitions into production.

8 PVM Continual Service Improvement

The PVM system consists of several components, which are developed on top of the existing hardware and software infrastructure, and it also relies on several well-known open-source tools. The IT infrastructure lifecycle consists of updating, upgrading or changing the underlying infrastructure and software. Since such changes can have the impact on the PVM system, for each change, the PVM Developers team should be notified in order to verify whether the changes can have the impact on the PVM system. In cases when the changes to the infrastructure affect the PVM system, the PVM Developers team has to make changes in line with the existing PVM software.

Other types of changes include system improvements in order to accommodate new network services or to add new features (e.g. new alarm types, new graphs and dashboards, etc.). These changes have to be made according to the ITIL Continual Service Improvement (CSI) recommendations and need approval by the Business Development and Service Management team, which will have the role of CSI Manager.

9 Service Metrics

PVM usage depends entirely on the usage of the network services it supports. Therefore, an initial set of key metrics of the PVM service are designed so that they assess the quality of the provided service and include:

- Service metrics:

- Service availability: Monitoring selected services deployed over testbed, ensuring operational 99.5% of the time without errors of the PVMv1.0 system during the 30 days period: no stops in data gathering and key metric displaying due to the outages of PVMv1.0 system.
- Service scalability: System capable to scale up to 20 service instances simultaneously being monitored without flaws and system degradation.
- Process metrics:
 - Adding new monitoring instance, decommissioning old monitoring instance processes within 10 minutes after the request for the services which have automated network service provisioning and data in service and resource inventories.
 - Fault-to-detection process: Localising faults within 15 minutes after the problem occurs.

10 PVM Service Roadmap

In the remaining project period, the focus is on the transition of the PVM v1.0 into production. PVM v1.0 supports Modes 1 and 2, and is installed in the testbed, as described in Section 6. During the last year of the project, By the end of 2018, PVM v1.0 will also be deployed in the GÉANT core network, on the same servers where the production personae system is installed. The plan for the last year of the project is that the PVM integration with the Service Provider Architecture (SPA) OSS/BSS code in the GTS laboratory in Prague is also planned in the upcoming months. The test will verify automated processes of setting up the network circuit together with pre-production continuity tests and later in production performance verification. The development of the PVM feature set will not stop with the PVM v1.0 release. PVM v1.1, which will support in addition to the existing functionalities: Mode 3, standard based alarming towards other OSS/BSS components and SLA report production will be released by the end of the project.

11 Conclusions

PVM is a new development in GN4-2, which is driven by the need for new types of network services, for a monitoring solution that allows for monitoring the performance of each separate network service instance installed on top of the shared network infrastructure. PVM is developed with the network service operations in mind and in a way which allows the integration of the system in the overall network management software architecture, but also gives the interface for the human operators to observe the health of the services.

PVM is built using the experience gained on previous projects, such as the work on service quality management and other monitoring mechanisms and tools like personae. It is a unique tool, offering scalable network services monitoring regardless of the underlying network technology. It monitors network services simultaneously, based on multiple different technologies, and regardless of the network equipment vendor or the number of network service instances in the network. Fault localisation is a unique capability which significantly decreases the time to resolve the faults in the network, especially in complex multi-domain environments such as those in GÉANT/NRENs.

PVM is not a typical customer-facing service, but rather, a supporting service for network services, and an essential part of the network service management portfolio. This document maps the transition from the development into production, which includes the preparation of the production infrastructure, as well as the establishment of the supporting and operations teams.

References

- [ALTMARK] Alternate-Marking Method for Passive and Hybrid Performance Monitoring, RFC 8321, January 2018.
- [Diamond] Ming Xia, Meral Shirazipour, Heikki Mahkonen, Ravi Manghirmalani and Attila Takacs, Resource Optimization for Service Chain Monitoring in Software-Defined Networks
<https://pdfs.semanticscholar.org/0119/099638d68a0836d55d7de0dfc00891571876.pdf>
- [ETSI] Network Functions Virtualisation (NFV); Assurance; Report on Active Monitoring and Failure Detection, v1.1.1, April 2016.
http://www.etsi.org/deliver/etsi_gs/NFV-REL/001_099/004/01.01.01_60/gs_nfv-rel004v010101p.pdf
- [MEF10.3] MEF 10.3 Technical specification – Ethernet Services Attributes Phase 3, October 2013:
https://www.mef.net/Assets/Technical_Specifications/PDF/MEF_10.3.pdf
<https://github.com/NORDUnet/openssa>
- [OpenNSA]
- [OWAMP] Shalunov S., Teitelbaum B., Karp A., Boote J., Zekauskas M., A One-way Active Measurement Protocol (OWAMP), IETF RFC 4656, September 2006.
<https://tools.ietf.org/html/rfc4656>
- [RFC4656]
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016,
<https://www.rfc-editor.org/info/rfc7799>
- [TestAPI] Framework Specification, Service Test Management, API REST Specification, TMForum, April 2016
- [Y.1540] Y.1540 - Internet protocol aspects – Quality of service and network performance – July 2016 - <https://www.itu.int/rec/T-REC-Y.1540/en>

Glossary

AMQP	Advanced Message Queuing Protocol
cMA	customer Monitoring Agent
CoS	Class of Service
CPE	Customer Premises Equipment
E2E	End to End
EoMPLS	Ethernet over MPLS
L3VPN	Layer 3 Virtual Private Network
MA	Monitoring Agent
MPLS	MultiProtocol Label Switching
MQ	Message Queue
NFV	Network Function Virtualisation
MC	Monitoring Controller
MCorr	Monitoring Correlator
MP	Monitoring Portal
MQ	Message Queue
MRR	Monitoring Result Repository
PC	Packet Capturer
PW	PseudoWire
SDN	Software Defined Networking
SFC	Service Function Chaining
SI	Service Inventory
SLA	Service Level Agreement
vMX	Juniper virtual router
VPLS	Virtual Private LAN Segment
VPWS	Virtual Private Wire Service