A Monitoring and Measurement Architecture for Traffic Engineered IP Networks

Aolghasem (Hamid) Asgari[1], Steven Van den Berghe[2], Christian Jacquenet[3], Panos Trimintzios[4], Richard Egan[1], Danny Goderis[5], Leonidas Georgiadis[6], Eleni Mykoniati[7], Panos Georgatsos[8], and David Griffin[9].

Abstract - With the constantly growing usage of the Internet, the need for deployment of value-added IP services has recently yielded a dramatic investigation effort in the field of IP traffic engineering. Traffic engineering can be defined as a collection of techniques that will allow service providers to use the network resources as efficiently as possible, according to the different quality levels that are associated with the range of services they provide. The monitoring activity can play an important role for assisting the operation of traffic engineered networks. This paper explains how to obtain and manage the measurement information which is required by traffic engineering algorithms in dimensioning the network, dynamic resource allocation and route management, and in-service verification of traffic and performance characteristics of value-added IP services. The paper focuses on the description of a monitoring and measurement architecture that is designed within the context of the TEQUILA project, partly funded by the European Commission within the context of IST development program.

***Keywords**: Monitoring, Measurements, Quality of Service (QoS), Service Level Specification (SLS), Traffic Engineering (TE), Per Hop Behaviour (PHB), Label Switch Path (LSP).*

1 Introduction

TEQUILA stands for "Traffic Engineering for QUality of service in the Internet, at LArge scale". The main objective of TEQUILA is to specify, develop, and validate a system that will be capable of dynamically negotiate, invoke, and provision the resources associated to the deployment of Quality of Service (QoS) based IP service offerings over the Internet. The TEQUILA system is to provide service guarantees through planning, dimensioning and dynamic control of traffic management techniques based upon the Differentiated Services (DiffServ) architecture [1] in a flexible policy-driven manner. The TEQUILA system is composed of a set of elementary blocks that comprise traffic engineering management capabilities [2]. It relies upon the use of classical IP routing protocols for the establishment of IP routes, as well as the use of the Multi-Protocol Label Switching (MPLS) technique [3], for the establishment of Label Switched Paths (LSPs) that are expected to comply with the QoS requirements specified by the customers. In TEQUILA, QoS refers to a service offering where one or more traffic/performance parameters (i.e., throughput, delay, loss, jitter) are quantified [2].

Monitoring and measurement architectures are becoming increasingly important for providing QoS and service assurance. The Internet has been delivering single class best-effort IP service without traffic and performance guarantees. The measurement functions in current best-effort networks mostly have a diagnostic role. They evaluate the current status of the network, or analyse the network behaviour during a certain time period, and report their findings to a management system. The measurement information is normally collected per-traffic flow basis for accounting and per-link basis mainly for diagnostic purposes. When adding traffic engineering to the network, the algorithms used will need an overview of the network status for their dynamic reactions. The measurement functionality that delivers this status is viewed as operational measurements. *Rondo* is an existing automated control system using a monitoring sub-system designed to manage congestion in MPLS traffic-engineered networks in near real time [4].

Traffic forwarded in QoS-enabled networks (i.e. IP networks that are composed of DiffServ-capable routers that process traffic according to a QoS policy) might encounter a differentiation into

[1] Thales Research Limited, Worton Drive, Reading RG2 0SB, UK. Email: Hamid.Asgari/Richard.Egan@uk.thalesgroup.com.

[2] IMEC, St. Pietersnieuwsstraat 41, B-9000 Ghent, Belgium, Email: steven.vandenberghe@intec.rug.ac.be.

[3] France Telecom R & D, 42, rue des Coutures, BP 6243, 14066 Caen Cedex 04, France, Email: christian.jacquenet@francetelecom.fr.

[4] Centre for Com. Systems Research (CCSR), University of Surrey, Guildford, Surrey GU2 7XH, U.K, Email: p.trimintzios@eim.surrey.ac.uk.

[5] Alcatel – Network Strategy Group, Francis Wellisplein 1, 2018 Antwerp Belgium, Email: danny.goderis@alcatel.be.

[6] School of Elect. & Computer Eng., Aristotle Uni. of Thessaloniki, PO Box 435, 54006, Thessaloniki, Greece, Email: leonid@eng.auth.gr.

[7] Dept. of Elect. Engineering & Computer Science, National Technical University of Athens, Greece, Email: mykel@telecom.ntua.gr.

[8] A.T.T.S. Dept., Algosystems S.A., 4, Sardeon str., 171 21 N.Smyrni, Athens, Greece, Email: pgeorgat@algo.com.gr.

[9] Dept. of Electronic & Elect. Eng., University College London, Torrington Place, London WC1E 7JE, UK, Email: D.Griffin@ee.ucl.ac.uk.

several service types/classes. As the network attempts to offer several service types (e.g., real-time, best-effort services, etc. [2]) by employing traffic engineering mechanisms, service monitoring is becoming increasingly important for providing end-to-end QoS and service assurance. Traffic belonging to each service type has certain requirements and exhibits certain behaviour. Having only a single measurement result (e.g., round-trip/one-way delay performed between a given source-destination pair) is not adequate for explaining all traffic belonging to different service types. Therefore in QoS-enabled networks, measurement information needs to be collected in finer granularity e.g., per service type.

This paper[10] is organised as follows. Section 2 describes some requirements that need to be taken into account when developing a measurement architecture for use in traffic engineered IP networks. Section 3 explains the scalability issues for such an architecture. Section 4 briefly describes the TEQUILA functional architecture and its monitoring and measurement requirements. The monitoring and measurement architecture in TEQUILA, the monitoring components and their relationship with other functional components are explained in section 5. In section 6, we propose how the measurements at the node, network, and service levels should be performed. Section 7 summarises and addresses the ongoing work of monitoring and measurement in TEQUILA.

2 Monitoring Requirements for Traffic-Engineered Networks and Services

Traffic engineering is achieved through capacity and routing management [5]. These two are realised with the calculation, selection and installation of a set of routes and queue management parameters, throughout the network in order to accommodate as many customer requests as possible, while at the same time satisfying their QoS requirements and optimising the use of network resources. The traffic engineering functions require observing the state of the network through a monitoring system and applying control actions to drive the network to a desired state.

Monitoring and measurement determines the operational state of a network and can assist the traffic engineering algorithms in the optimisation and dimensioning of the network by providing feedback data about the status of network resources. This data can be used by traffic engineering mechanisms to automatically react and adaptively optimise network performance. Consequently, monitoring no longer has only diagnostic role but also it turns into an important tool for assisting the network operation and providing service auditing for both traditional and value-added services. As a result, monitoring architecture should provide information for:

- Assisting traffic engineering in allocating resources (e.g., to queues and paths over which routes will be established) efficiently and effectively. The capability to obtain statistics at the QoS-enabled route level is so important that it should be considered an essential requirement for traffic engineering.
- Assisting traffic engineering in dimensioning the network for any short or long term changes required in the network configuration set-up.
- Verifying whether the QoS performance guarantees (negotiated between a customer and a provider) committed in Service Level Specifications (SLSs) is in fact being met. This requires an in-service verification of traffic and performance characteristics per customer service. SLS is a set of technical parameters and their values, which together define the service, offered to a traffic stream by a DiffServ domain. SLSs can differ depending on the type of services offered and different SLS types have different QoS indicators that require processing of different types of information [6].

3 Scalability of the Monitoring Architecture

The monitoring architecture must be able to scale with the size and the speed of the network as it evolves. In order to have a scalable solution for monitoring and measurement, we propose the following approaches:

I. Defining the monitoring process granularity

In a DiffServ environment, the measurement methodology must be aware of different service types. Traffic-engineered networks such as the one assumed in TEQUILA rely on forming IP routes (shortest paths) or explicit paths (LSPs) from ingress to egress points through the core network. IP routes/explicit paths allow control over routing of traffic flows requiring specific QoS within the TEQUILA domain. IP engineered routes/LSP tunnels are used to carry aggregate user traffic belonging to several SLSs having similar performance requirements. In addition, traffic

[10] An extended version of this paper and other TEQUILA's published papers can be found at http://www.ist-tequila.org/.

engineering algorithms do not need to operate at small scale of individual packets as collecting packet-level micro-flow related statistics would be prohibitively expensive and unnecessary. Instead, observation is performed over all packets but statistics are gathered at the aggregated macro-flow level. Hence, the monitoring process functions based on the configured classes of service handling of the data streams (i.e., Hop Behaviours - PHBs) and the scope of offered services between ingress and egress points (i.e., traffic-engineered routes).

II. Minimising the measurement transmission overhead by processing data close to the source

To support the dynamic operation of the network, the measurement architecture must be able to capture the operational status of the network without degrading network performance and without generating a large amount of data, causing unnecessary overhead. In order to transmit the data efficiently to the components, it is important to process the raw data close to its source, which necessitate a distributed data collection system, and to condense the collected data by summarisation. Two forms of data summarisation are considered i.e., event notification and statistics that are explained in the section 6.1.

III. Using aggregate performance measurements combined with per SLS traffic measurements

The granularity of measurements can be related to SLSs since every SLS might not need to be monitored in the same way. Ideally, an SLS belonging to a premium class might need measurement results with higher frequency. Monitoring SLSs at different levels of granularity using different sampling frequencies make the measurement architecture far more complex. Instead, monitoring every customer SLS is scalable and feasible provided aggregate network performance measurements (e.g., delay, loss, jitter) are used combined with per SLS ingress/egress traffic measurements (e.g., throughput). As several SLSs may use a single IP route/LSP, single performance monitoring action is enough to satisfy the requirement of these SLSs.

IV. Hop-by-hop vs edge-to-edge measurements

The scope of measurements is an important aspect of measurement architecture. Two approaches can be taken for performing performance measurements. The measurements are carried out either between two edge nodes for edge-to-edge measurements or between two neighbouring nodes for hop-by-hop measurements. The first method provides edge-to-edge measurement results but it might not scale well if there are a huge number of IP routes/LSPs in the network that each ingress node might need to inject test traffic to its associated routes. The second method overcomes the scalability problem by adding the hop-by-hop results and calculating an edge-to-edge measurement result. As multiple routes may be related to a single PHB by sharing a physical link, a single test traffic sent to quantify the behaviour of a given PHB satisfies the performance monitoring requirements of these routes using that link. This results in significant reduction of test traffic in the network. With the hop-by-hop method, the status of every individual link will be known, but inaccuracy will be introduced due to the non-synchronised individual hop-by-hop measurements and concatenating these discrete measurements to estimate per-route edge-to-edge measurement values. Depending on the type of SLS that has been subscribed by the customers, this method may be appropriate for estimating the performance measurements of routes used e.g., for best-effort traffic.

4 The TEQUILA Architecture and its Monitoring Requirements

The TEQUILA project addresses the following areas: the customer demands through SLSs, the protocols and mechanisms for dynamically negotiating, monitoring and enforcing SLSs, and the QoS-related technologies required for meeting these customer demands (SLS enforcement), including the provisioning, management and intra- and inter-domain traffic engineering schemes to ensure that the network can cope with the contracted SLSs - within domains, and in the Internet at large [2], [7]. Figure 1 shows the functional architecture of the TEQUILA system.

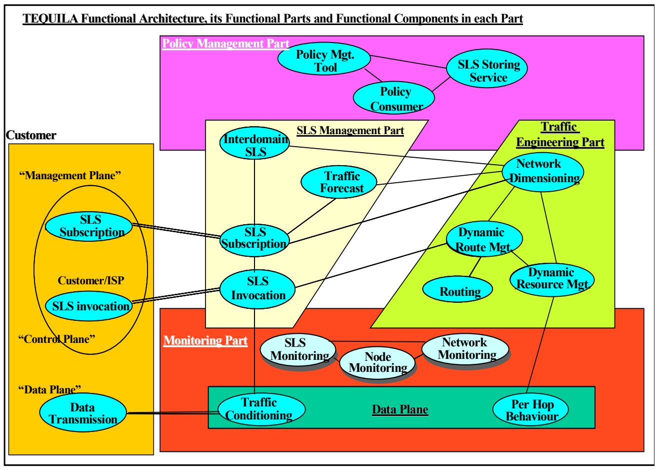

Figure 1: TEQUILA Functional Architecture specifying distinct functional parts and components.

TEQUILA architecture has the following main functional parts: SLS Management, Traffic Engineering, Policy Management, and Monitoring in addition to Data Plane functionality. The SLS Management is responsible for subscribing and negotiating SLSs with customer, a customer being possibly a service provider. It also performs admission control for the traffic associated/ depicted to/in the invoked SLSs. Traffic Forecast component of SLS Management is responsible for mapping and aggregating traffic demands of multiple SLSs having an ingress node and a set of egress nodes requiring a certain QoS and forming a Traffic Matrix. The Traffic Matrix is then used by the Network Dimensioning (ND) component of Traffic Engineering part.

The Traffic Engineering part of the architecture is responsible for dimensioning the network according to the projected demands, and for establishing and dynamically maintaining the network configuration that has been selected to meet the SLS demand. ND is in general centralised and is responsible for mapping the Traffic Matrix onto the network resources by computing a set of optimal routes (by maintaining the link metrics or by setting explicit paths) in order to accommodate the forecasted traffic demands subject to resource restrictions, load trends, QoS requirements, and policy directive and constraints. Dynamic Route Management (DRtM) is a distributed component located at the routers, responsible for managing the routing processes (including LSP set-up in the case of MPLS, traffic re-routing, multi-path load balancing, and issuing alarms to ND) in the network according to the guidelines provided by ND on routing traffic. Dynamic Resource Management (DRsM) is distributed, with an instance attached to each router. It is responsible for ensuring the link capacity is appropriately distributed among a limited number of PHBs sharing the link by setting buffer and scheduling parameters associated with the interface attached to the link, according to ND directives, constraints, and rules.

Policy Management gives the ability to the administrators to define high-level policies, which are translated into policy objects, using a well-defined traffic engineering policy information model and stored in the policy repository. Policies are refined into the more detailed actions reflecting the hierarchical TEQUILA architecture.

4.1 Monitoring & Measurement Requirements in TEQUILA

In TEQUILA, the following parts and components are interested in the measurement information:

- The SLS Management part including:
  - ➢ Traffic Forecast for optimising the forecasted traffic related to SLS instances as a basis for long-term network configuration. Monitoring is also to provide analysed traffic and performance information for long-term planning in order to optimise the network and to avoid undesirable network conditions. The analysed information might include traffic growth patterns and congestion issues.
  - ➢ SLS Invocation that may use current SLS loads for SLS admission control of new flows.
- The Traffic Engineering part including:
  - ➢ ND for calculating a new dimensioning of the resources if any part of the network is not able to meet performance objectives.
  - ➢ DRtM for taking appropriate engineering actions on setting up new routes, modifying existing routes, load-balancing among routes, and re-routing of traffic for optimisation purposes or work around congestion.
  - ➢ DRsM for performing node-level optimisations on resource reservations (bandwidth assignment and buffer management) to combat localised congestion.

  It should be noted that Traffic Engineering components operate in different time scales ranging from weeks through days for ND, or hours through minutes for DRtM, and minutes through seconds for DRsM.
- Policy Management part for getting notifications, triggering events which signal the enforcement of specific policies, or the inability to enforce a policy (policy run-time conflicts) which trigger alarms to the administrator.
- The SLS Monitoring component of the Monitoring part for monitoring the continuity and quality of services, auditing services, and reporting.

5 Monitoring & Measurement Architecture in TEQUILA

The monitoring architecture of TEQUILA includes the following components:

1) Node Monitoring (NodeMon) responsible for node related measurements
2) Network Monitoring (NetMon) responsible for edge-to-edge performance monitoring and any required network-wide post-processing based on statistical functions

3) SLS Monitoring (SLSMon) responsible for customer related service monitoring
4) Monitoring Repository (MonRep) for storing configuration information and measurement data
5) Monitoring GUI (MonGUI) for displaying measurement results.

Monitoring part in TEQUILA, its components, interfaces to other components, the interface technologies and protocols are shown in Figure 2. The next sections explain the monitoring components and their functions.

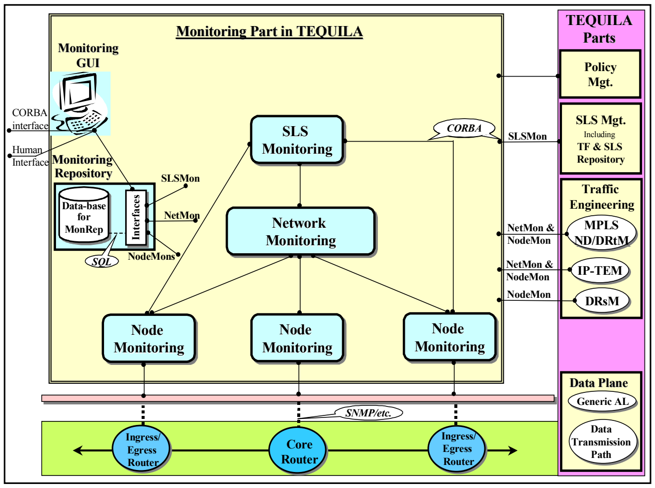

Figure 2: TEQUILA Monitoring Architecture and its interactions with other parts.

In general, the monitoring functions are split into four phases:

Request: Every component that requires monitoring information must register to one of the NodeMon, NetMon, or SLSMon requesting monitoring actions by indicating what measurement data it wants to be notified about.

Configuration: NetMon will decide which NodeMons are needed to be at the basis of any measurements and it configures them. SLSMon performs some configurations on NodeMons located at ingress/egress points.

Execution: NodeMons perform the measurements on basis of these configurations. Other available data such as metering information may also be used. NodeMons also perform some basic measurement processing. NetMon/SLSMon will further aggregate and process NodeMon measurements if it is necessary.

Reporting and exception: The analysed measured data and events are sent back to the registered components.

5.1 Node Monitoring

A diverse variety of measurement data is needed in order to perform network and customer service performance and traffic monitoring. The variety of data, the necessary processing and the magnitude of the raw data make a distributed data collection system more practical. Processing and aggregating the raw data into accurate and reliable statistics and reducing the amount of data near its source in order to transmit the data efficiently to the components of the system is key to automating the dynamic operation of the network. Hence, the Node Monitoring is distributed across the network i.e., one per Network Element (NE).

NodeMon allows other components to request monitoring and measurement actions. NodeMon includes the following functions: *Configuration and Monitoring* function handles registration requests and initiates measurements on NEs (i.e., routers) for both diagnostic and operational monitoring and sets thresholds. NodeMons receives the configuration information with entries that each defines a variable, polling/sampling period, and threshold parameters. A local *Data Collector* (Reader) collects measurement results from either meters/probes located at NEs or active monitoring agents. A probe is a generic term for a dedicated machine or a software agent that measures data moving through the network or injects test traffic in the stream to take its measurements. Probes present the data they collect in a variety of ways. The job of data collector is to regularise and re-abstracts various types of measured data in a structural way. A *Local Performance Analyser* performs some short-term basic evaluation of results such as averaging. It also performs threshold crossing detection and notification. The process data is passed to the components and it is also stored in the monitoring data store (part of MonRep).

5.2 Network Monitoring

Network Monitoring is in general, centralised and it utilises network-wide information collected by NodeMons. NetMon instructs the NodeMons to measure the performance and traffic parameters and builds a physical and logical network view (i.e., the view of the routes that have been established over the network) based on measurement information collected for links, nodes, PHBs, and route statistics. NetMon includes the following functions: *Configuration and Monitoring* function handles monitoring registration requests and configures the NodeMons including threshold setting. NetMon needs to know the network logical configuration, which changes as the ND re-dimensions the network or DRtM re-routes the traffic to alleviate congestion. *Data Collector* accesses MonRep to get measurement data and may notify other components about threshold crossing detected by NodeMons if necessary. *Performance Analyser* aggregates and performs longer-term in-depth statistical analysis on measurement data including trend analysis. The data produced by such analysis is stored in the monitoring repository and the appropriate processed data is forwarded to the interested components.

5.3 SLS Monitoring

SLS Monitoring is centralised, since it must keep track of the compliance of the level of service provided to the customer SLS instances, by analysing information provided by NetMon and ingress/egress NodeMons. SLS Management notifies SLSMon and requests the creation of any necessary monitor instances when an SLS is invoked. SLSMon acts as a client to NodeMons and NetMon. SLSMon retrieves SLS related information from SLS repository. When a SLS is invoked, a specific route will be used for the traffic related to this SLS. SLSMon needs to receive performance-related information (one-way delay and loss on this specific route from NetMon and traffic-related information (throughput) specific to this SLS from ingress/egress NodeMon.

SLSMon includes the following functions: *Configuration and Monitoring* function handles activation/deactivation process issued by SLS Management and configures the ingress/egress NodeMons. A *Data Collector* accesses MonRep for measurement results collected by ingress/egress NodeMons and NetMon and combines the data for each individual SLS. Each contracted SLS's performance and traffic related values are checked against measurement data through a Contract Checker of an SLS Manager to determine whether any violations occur and then generate reports. SLS Manager is also responsible to activate the Report Generator. Necessary Reports are provided to both the customer and the management.

5.4 Monitoring Repository and Monitoring GUI

The MonRep consists of two major parts for data cataloguing, a "data store" having a database functionality for storing the possibly large amounts of data for monitoring components and an "information store" for storing smaller amounts of configuration type information. Measurement data is stored in a "data store" for possible later analysis via the GUI, or performance analysers.

MonGUI presents a user interface allowing human operators to request graphical views of monitoring statistics extracted from the monitoring data store. It also exposes an interface to allow other components to request display of statistics. MonGUI might be used in a Network Operations Centre.

6 Node, Network, & Service Level Measurements

6.1 Measurement Data

Monitoring can occur at different levels of abstraction. Measurements can be used to derive packet level characteristics, application level characteristics, user/customer level characteristics, traffic aggregate characteristics, node level characteristics, network wide characteristics, etc. In TEQUILA, the monitoring framework focuses on deriving all except packet and application level measurements and gathers the network-layer measurements. The network-layer measurements include: one-way delay and packet loss ratio in the granularity of PHBs and routes, the traffic load on PHB basis, the throughput on a link/route basis, and the aggregate traffic load for SLSs and macro-flows. Two forms of measurement data are considered: event notification and statistics:

- ❑ ***Events:*** Basic raw measurement data is taken in short-time scales from variables in the monitoring probes. The measurement data is compared with two previously configured thresholds (the upper mark and the normal mark). If the measurement data is found to cross the upper threshold value, the relevant functional component is informed. Depending on the measurement time-scales, event notification might be postponed on instantaneous upper threshold crossings until successive/frequent threshold crossings are observed and realised that the problem persisted for a specified time interval. Upon event notification on upper threshold crossing, further triggers are not delivered until the measurement data returns to normal when the relevant component is notified. Threshold detection implies asynchronous notification of the event. This event notification method is employed to reduce a large amount of data frequently passed from monitoring to other functional components. It is also designed to insure that transient spikes do not contribute to changes unless they occur frequently. The granularity of event notifications is for PHBs and IP routes/LSPs.

- ***Statistics:*** The measurement data is aggregated into summarising statistics in order to have a scalable system. Summarisation is usually done by integrating the measurement data over a pre-specified period. The granularity of summarisation periods must be suitably chosen based on the requirements of the interested component. In addition to some basic functions by NodeMons, more complex traffic analysis is performed in longer intervals by NetMon. The granularity of statistics range from PHB and route level to the aggregated flow levels for customer service monitoring.

6.2 Measurement Methods

There are two types of methods to perform measurements. *Active measurements* inject test traffic into networks based on a scheduled sampling in order to observe network behaviour. Normally, active measurement tools require co-operation from both end-points of the measurement and they need to have a continuous session as long as the active measurement is required between two nodes. In addition and specifically in the case of measuring one-way delay, both end-points require to be synchronised. Therefore, the deployment of Network Time Protocol (NTP) or Global Positioning System (GPS) for synchronisation of end-points is required. It should be noted that NTP accuracy depends on the network conditions and it could provide poor level of precision. GPS provides high precision but its deployment in all routers and more specifically on edge routers makes it an expensive solution.

In contrast, *passive measurements* observe actual traffic without injecting extra traffic into the network. While passive measurement does not require co-operation from end-points, it requires continuous collection of data and must monitor the full load on the link which can be problematic on high-speed links. In both cases, the quality of analysed information depends on the granularity and integrity of collected data.

Measurement data is polled from passive monitoring probes with a request/reply mechanism whereas whenever the measurement data is available it is pushed asynchronously from active monitoring probes. The choice of measurement sampling method and sampling interval determines the level of measurement accuracy (within a given confidence interval) and reliability. A sampling interval is defined as the read-out periods for passive measurements or the average time interval over which two test packets are separated. Periodic sampling and random sampling methods are usually used for passive and active measurements respectively. Since the passive measurement data is available at any given time, the sampling interval can be properly specified. The amount of test traffic generated by active measurements methods will be increased in QoS-enabled networks where several PHBs per node and large numbers of routes need to be monitored. The amount of test traffic (traffic load and packet rate) on each network link, sent to measure one-way delay and packet loss during a specified time interval, will depend on the number of IP routes/LSPs crossing the link, the number PHBs attached to the link interface, the number of different test packet sizes used for route and PHB related measurements, the length of these packets, and the statistical average of sampling intervals used. The proper selection of sampling intervals in both methods and summarisation periods is for further study.

6.3 Engineering Aspects

TEQUILA uses an object-oriented approach for monitoring architecture. The monitoring architecture is realised as a set of Java classes. The monitoring architecture defines a set of CORBA (Common Object Request Broker Architecture) interfaces to internal monitoring components for communicating with one another and to external components. All of the CORBA interfaces are implemented using the Java 2 platform. Measurement results are passed to NodeMons via a Generic Adaptation Layer (GAL), and NEs use COPS (Common Open Policy Service [8]), SNMP (Simple Network Management Protocol), etc. to communicate with the GAL.

Two types of routers are used in the testbeds: commercial (Cisco) and PC-based (Linux) routers. For passive measurements such as throughput, load and packet discards, MIBs (Management Information Bases), PIBs (Policy Information Bases), and metering information from traffic conditioners (whichever is possible) are used to poll the data. The availability of required passive measurements is limited in commercial routers. Cisco routers collect byte counts for physical network interfaces and virtual interfaces using MIB-2 [9]. In Cisco routers, LSPs are represented as logical interfaces or tunnel interfaces at the head-end routers. Therefore, data derived from the ingress router of each LSP has an interface definition in MIB-2 that can be used by ingress/egress NodeMon. Packet forwarding data is available for LSPs at the intermediate routers by using the Command Line Interface (CLI). Linux routers are configured to provide any required per-PHB and per IP route/LSP passive measurements.

In TEQUILA, both edge-to-edge and hop-by-hop approaches are used for performing active measurements. Ideally, PHB-based delay measurements must be implemented in NEs, which is not currently available in commercial routers. Hop-by-hop measurements are used to estimate a PHB-based delay. This is a practical approach but at the expense of introducing some inaccuracy. Active test traffic is sent between neighbouring hops for estimating PHB-based delays. This introduces inaccuracy as it includes the test packet processing at the originator including PHB function, packet transmission delay onto the link, propagation delay, and packet processing delay at the next hop. The inaccuracy level is reduced by subtracting some of these fixed delays from the measured value. This hop-by-hop method requires the ability of forcing test packets to pass through particular PHBs, calculating the measurement (e.g., delay) by the test traffic receiver, and sending the result back to the originator. One-Way Delay measurement Protocol (OWDP [10]) is used with some modifications to measure one-way delay and packet loss either hop-by-hop or edge-to-edge.

7 Summary and On-going Work

Engineering large IP networks introduces fundamental challenges that stem from the dynamic nature of user behaviour. Careful engineering of the network is important, since the network dimensioning and routing management have significant implications on resource efficiency and user performance [5]. In this paper, we propose monitoring and measurement architecture for node, traffic-engineered network, and service monitoring. This is aimed at facilitating route calculation and optimisation, user service auditing, and traffic forecasting. We also present scalable methodologies for event monitoring and measurement statistics to be used for network operation and in-service verification of traffic and performance characteristics of offered services. Our on-going work focuses on the TEQUILA system implementation and examines the practical effectiveness of monitoring on traffic engineering algorithms and traffic forecasting in both simulation and testbeds environments. This also enables us to investigate the scalability of the monitoring architecture in real-time data processing of the events and statistics according to the current state of the network.

Acknowledgements

This work was undertaken in the Information Society Technologies (IST) TEQUILA project, which is partially founded by the Commission of the European Union. The authors would also like to thank the rest of the TEQUILA colleagues who have contributed to the ideas presented here.

References

[1] S. Blake, D. Black, et al., "*An Architecture for Differentiated Services*", RFC-2475, Dec. 1998.

[2] D. Goderis (ed.), "*Functional Architecture and Top Level Design*", TEQUILA Deliverable D1.1, September 2000.

[3] E. Rosen, A. Viswanathan, R. Callon, "*Multiprotocol Label Switching Architecture*", FRC-3031, January 2001.

[4] James L. Alberi, Ta Chen, et al., "*Using Real-Time Measurements in Support of Real-Time Network Management*", PAM2001, Amsterdam April 2001.

[5] D. Awduche *et al.* "*A Framework for Internet Traffic Engineering*", Internet Draft, Expires: November 2001.

[6] D. Goderis, et al., "*Service Level Specification Semantics, Parameters and negotiation requirements"*, Internet Draft: draft-tequila-sls-01.txt, Expires December 2001.

[7] P. Trimintzios, I. Andrikopoulos, G. Pavlou, et al., "*A Management and Control Architecture for Providing IP Differentiated Services in MPLS-based Networks*", IEEE Com. Magazine, vol. 39, No. 5, pp. 80-88, May 2001.

[8] D. Durham Ed., J. Boyle, R. Cohen, et al., "*The COPS (Common Open Policy Service) Protocol*", RFC-2748, January 2000.

[9] K. McCloghrie, M. Rose, "*Management Information Base for Network Management of TCP/IP-based Internets: MIB-II*", RFC-1213, March 1991.

[10] S. Shalunov, B. Teitelbaum, M. Zekauskas, "*A One-way Delay Measurement Protocol*", Internet Draft, Expires: August 2001.