Measurement-Based Policy-Driven QoS Management in Converged Networks

S. Y. Yerima, G.P. Parr, S. McCLean, P. J. Morrow
School of Computing and Information Engineering,
University of Ulster, Northern Ireland
Email: {s.yerima, gp.parr, si.mcclean, pj.morrow}@ulster.ac.uk

Abstract— **Policy-based management is considered an effective approach to address the challenges of resource management in large complex networks. Within the IU-ATC QoS Frameworks project, a policy-based network management framework, CNQF (Converged Networks QoS Framework) is being developed aimed at providing context-aware, end-to-end QoS control and resource management in converged next generation networks. CNQF is designed to provide homogeneous, transparent QoS control over heterogeneous access technologies by means of distributed functional entities that co-ordinate the resources of the transport network through policy-driven decisions. In this paper, we present a measurement-based evaluation of policy-driven QoS management based on CNQF architecture, with real traffic flows on an experimental testbed. A Java based implementation of the CNQF Resource Management Subsystem is deployed on the testbed and results of the experiments validate the framework operation for policy-based QoS management of real traffic flows.**

Keywords- QoS; Policy-based network management; converged networks; resource management; network and service management; DiffServe

I. INTRODUCTION

As fixed and wireless access networks converge towards common IP-based transport in next generation networks, the need to provide unified control and management infrastructure to support transparent, end-to-end, QoS-enabled service provision becomes even more crucial. The inadequacy of traditional best-effort model of the IP-based transport to meet QoS of real-time applications such as VoIP, streaming video, IPTV and other real-time multimedia applications is intended to be addressed by IP QoS architectures such as IETF DiffServe and IntServe models. However, with the convergence of heterogeneous access technologies such as WiMAX, PONs, xDSL etc., towards interconnection by a common IP core network (CN), the interoperability of often incompatible QoS mechanisms in the different access segments with the CN IP-based QoS mechanisms is crucial to enabling end-to-end QoS provisioning in the transport plane.

Equivalently, unified management and control plane functionalities are needed to co-ordinate the converged end-to-end transport infrastructure, adding to the complexity of configuration, control and management operations needed to support transparent service provisioning. Policy-based network management (PBNM) is emerging as a promising approach to address these challenges. PBNM eases the management of complex networks through automated and distributed structures using centralised policies. PBNM systems generally use hierarchical structures and will facilitate the management of Next Generation Networks [1]. Furthermore, PBNM provides the means for application transparency across existing and emerging access technologies which permit applications to be transport layer-agnostic when deployed [2].

To this end, we have proposed a policy-based network management framework for converged networks end-to-end QoS control and resource management, CNQF (Converged Networks QoS Framework) [3]. CNQF is designed to provide homogeneous, unified, end-to-end QoS management over heterogeneous access technologies, together with scalable, context-aware adaptive QoS control using policy-based paradigms aligned with IETF, TISPAN and 3GPP policy architectures.

CNQF consists of distributed Policy Decision Points (PDPs) which are responsible for various management and control decisions driven by high-level declarative policies and enforced at policy enforcement points (PEPs) such as routers, switches, and gateways in the converged network transport plane. Our previous work in [3], presented CNQF framework architecture and exemplified use case scenarios for context-driven QoS adaptation. This paper focuses on measurement-based evaluation of policy-driven QoS management based on CNQF architecture with real traffic flows on a testbed where a Java based prototype of the CNQF Resource Management Subsystem is deployed. The rest of the paper is organised as follows. Section II reviews related work. Section III summarises the CNQF framework architecture and description of its constituent entities. Section IV describes the CNQF testbed configuration while section V presents the experiments for policy-based QoS management validation. The paper is concluded in section VI.

II. RELATED WORK AND MOTIVATION

Policy-based architectures have been defined within the standards bodies; for example, Telecoms and Internet Converged Services and Protocols for Advanced Networks (TISPAN) technical committee of the European Telecommunications Standards Institute (ETSI) has defined a Resource and Admission Control Subsystem (RACS) consisting of a Service-based Policy Decision Function

This work is funded by the EPSRC-DST India-U.K. Advanced Technology Centre of Excellence in Next Generation Networks, Systems and Services (IU-ATC) (www.iu-atc.com) under grant EP/G051674/1

(SPDF) and Access Resource Admission Control Function (A-RACF) [4]. Both of these interact with Policy Enforcement Points (PEPs) in the underlying networks. In that regard, the architecture shares similarity with the IETF policy model which specified a policy enforcement point (PEP) and a Policy Decision Point (PDP) as part of its architecture [5]. Similarly, Third Generation Partnership Project (3GPP) defined a Policy Decision Function (PDF) in their Release 5/6 policy framework [6]. Although the policy frameworks within the standards defined function blocks, interfaces and protocols to facilitate interoperability of standards-compliant products, no details are provided on how the various functionalities would be implemented. Our work aims to bridge this gap by not only leveraging the aligned policy architecture to develop scalable solutions for management and control of converged networks, but also introducing important novel extensions such as context-aware functionality support, in order to provide value-added intelligence to the PBNM system.

Also, policy-based management systems for resource management in different contexts such as tactical networks, VPNs, MPLS-enabled networks, virtualization environments etc., have been proposed in such works as [1], [7-10]. An important distinguishing feature of CNQF design from these policy-based systems is the incorporation of context management functionality as a building block of the PBNM architecture thus enabling value-added intelligence to provide adaptive policy-driven decisions within the PBNM. Furthermore, unlike the aforementioned systems, CNQF is aimed at providing policy-based infrastructure to support resource management across fixed-wireless access and core network domains in converged next-generation networks.

III. CNQF Architecture And Entities

CNQF is made up of three logical subsystems: Resource Management Subsystem (RMS); Context Management and Adaptation Subsystem (CAS); and Measurement and Monitoring Subsystem (MMS). Together, these three subsystems provide the policy-based infrastructure to enable closed-loop, scalable, end-to-end QoS control and resource management in converged next generation networks.

A. Resource Management Subsystem (RMS)

The primary role of the RMS is providing allocation, co-ordination and control of the resources in the end-to-end transport layer in accordance with customer Serve Level Agreements (SLAs). RMS is made up of distributed instances of peer resource brokers (RB) which act as Policy Decision Points (PDPs) within each (access, and core) network present on the end-to-end transport plane (see Fig. 1).

A CNQF RB may be a Fixed Access Resource Broker (FARB), Wireless Access Resource Broker (WARB) or Core Network Resource Broker (CNRB), each performing similar policy-based resource management roles depending on the type of network. For example, in a wireless access network such as WiMAX, a Wireless Access Resource Broker (WARB) and will be responsible for policies that co-ordinate the resources in the wireless access portion of the converged network. Such policies will include local and global admission control and bandwidth management policies. Likewise, for wired access networks such as Optical networks or xDSL, a Fixed Access Resource Brokers (FARB) will act as a PDP to perform a similar role. For the core network(s), a Core Network Resource broker (CNRB) is responsible for policy-driven decisions in the CN. The CNRB also interfaces with the various WARBs and FARBs of the wireless/fixed access networks connected to the core network being overseen by the CNRB. This is important because CNRB will handle resource brokerage between the access RBs allowing for scalable management of resources on an end-to-end basis.

In the CNQF architecture, each RB communicates with one or more Resource controllers (RCs) which are the logical management and control entities responsible for low level (re)configuration at the Policy Enforcement Points (PEPs) in the transport plane. The PEPs are at the network entities such as gateway nodes, access routers, edge routers etc. where the PDP (i.e. RBs) policy decisions are enforced. Thus each RB (WARB, FARB, CNRB) is interfaced with one or more corresponding RCs (FARC, WARC, CNRC) which perform different configuration and control functions depending on where the PEPs are located on the transport plane. For instance, in the CN, an RC located in the edge router (PEP) may be responsible for packet marking (e.g. DiffServe Code Points, DSCP marking in a DiffServe domain) in response to CNRB policy decisions. While in the wireless access network, an RC may be responsible for configuration of gateway nodes to map layer 2 QoS parameters (e.g. WiMAX QoS classes) to layer 3 IP QoS parameters (e.g. DiffServe DSCPs).

B. Monitoring and Measurement Subsystem (MMS)

In order to facilitate *closed-looped, adaptive measurement-based QoS control*, CNQF architecture incorporates a Measurement and Monitoring subsystem (MMS). Without the MMS, CNQF may be limited to providing open loop QoS provisioning based on, for example, pre-determined end-to-end resource allocation derived from a priori SLA negotiations. But with MMS in the loop, fine-grained resource allocation and adaptive QoS control is enabled through the feedback of measurement data to the RMS. The MMS consists of network monitoring entities (NMs) located at the PEPs for monitoring and measurement collection.

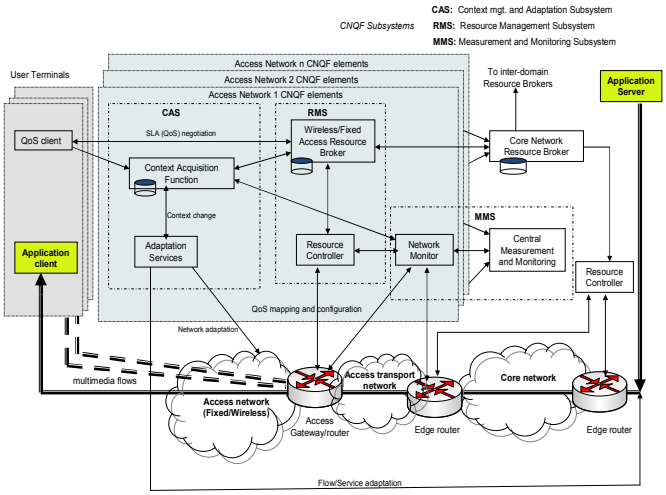

Fig. 1. CNQF operational entities in a converged network [3].

The NMs are low level function blocks within the MMS which can perform both active and passive measurements on request, from RCs within the RMS or from other function blocks in CNQF that may require measurement data to drive their policies. The NMs interface with a central measurements server which serves as aggregating entity for the entire MMS. The aggregated data could provide high level summaries that could be used to gauge the health of the network through visualisation interfaces on a centralised management station.

C. *Context Management and Adaptation Subsystem (CAS)*

Policy-based network management infrastructures, such as CNQF, stand to benefit from the use of *context information* to drive policies/policy adaptation. This is because context information equips the management system with increased intelligence and ability to adapt service provision, resource allocation, and QoS control in a more flexible and efficient manner. It also gives more autonomy to the system to respond to highly dynamic operational conditions. For example, resource allocation may be made responsive to different user contexts such as location, time, device capability, battery capacity etc. Through *context-awareness*, the PBNM system may apply different resource management policies to different 'contexts'. For example a user may receive different bandwidth allocations in different locations if the network is aware of the user's location (context) and is able to allocate location-dependent usage through context-aware policies.

As shown in Fig. 1, CNQF provides context-aware functionality through its Context Management and Adaptation Subsystem (CAS). CAS consists of distributed Context Acquisition Function blocks (CAF) instantiated in each access network. The CAFs are PDPs that execute context-aware or context-driven policies within the CNQF system. Each CAF elements has associated Adaptation Servers (ADs) which are function blocks that configure/reconfigure PEPs that are directly affected by context-driven policy decisions in the CAF. Entities characterised by context within the PBNM system could be physical objects e.g. a user device, router, switch, GGSN node, physical link, wireless channel; or could be a virtual object such as MPLS path, or a VPN tunnel.

As mentioned earlier, PBNM systems typically employ hierarchical structures. The CNQF system can be layered into the hierarchical structure depicted in Fig. 2. At the highest level various tools that provide a centralised policy management are present such as GUI interfaces for high level policy entry, editing and validation; visualization tools for network-wide status monitoring; and central high-level policy repositories. The policy decision layer comprises of the various PDPs such as WARB, FARB, and CNRBs which are centrally managed within the policy management layer. The policy enforcement layer comprises of distributed resource controllers directly interfacing with PEPs in the transport network. The policy communication layer contains the communication protocols that allow for PDP-PEP communication. This layer also provides for policy translation between the two policy element layers so that high-level policies can be mapped to equivalent low level policies/policy actions performed in the PEPs.

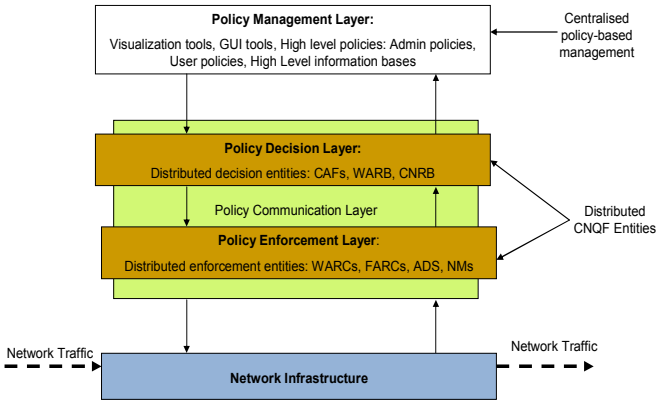

Fig 2. CNQF logical layered architecture.

Based on the CNQF architecture, a prototype of the Resource Management Subsystem has been implemented using Java. The current implementation allows for RB-RC communication via Java RMI (Remote Method Invocation). This allows the RBs to process high level policies at the CNQF management station whilst invoking equivalent policy actions on the RC co-located with the PEPs i.e. the edge and core routers on our testbed (Fig. 3.). The testbed setup and measurement-based tests undertaken with real traffic to validate the CNQF RMS functionality are described in the next sections.

IV. CNQF TESTBED CONFIGURATION

A. *Linux-based testbed implementation*

The testbed used for development of CNQF and evaluation of its functionalities is implemented using Linux systems. The configuration is shown in Fig. 3. We set up several test experiment to validate the RMS functionality for IP QoS management using the testbed configuration depicted in Fig. 3. The testbed consists of two Linux-based edge routers and a Linux-based core router. These elements constitute the PEPs each having an instance of CNQF RC that interacts with the Linux router kernel to set various parameters that enable configuration/reconfiguration of QoS management strategies implemented according to the higher-level policies of the RBs. The Linux TC (traffic control) utility in the kernel provides commands for implementing packet marking, classification, queuing disciplines, and policing of flows. In our testbed, the RCs employ TC commands for low level configuration which have equivalent mappings to high level *policy actions* in the RB. The testbed elements include:

- *CNQF management station*: this houses the RMS PDP i.e. CNRB implemented in Java which invokes policy actions via Java RMI to enforce configuration policies at the PEPs (routers) via the RCs.
- *Edge routers A and B*: are Linux PCs configured as edge routers with TC utility installed to allow the configuration of the router interface(s) for ingress packet marking, and for egress classification, queuing and policing via RC's response to CNRB policy decisions.

- *Core router*: a Linux system with TC utility installed to allow for configuration of packet classifiers and filters through CNQF policies also via an RC.
- *Traffic generators*: The Ntools [11] traffic generator is used to generate multi-client traffic with different flow characteristics including constant bit rate (CBR), On-Off traffic, and variable bit rate (VBR) traffic. VLC client/server is used for streaming MPEG video traffic.

TABLE I. TESTBED ELEMENTS

Elements	Description
CNQF Management Station	CNQF application with RB elements running on Ubuntu l0.0.4 Linux host.
Edge router A	Ubuntu l0.0.4 Linux host running on 2.66 GHz Intel Xeon PC, 3.0GB RAM
Core router	Ubuntu l0.0.4 Linux host running on 2.66 GHz Intel Xeon PC, 3.0GB RAM
Edge router B	Ubuntu l0.0.4 Linux host running on 2.66 GHz Intel Xeon PC, 3.0GB RAM
Multi client Traffic generator	Ntools traffic generator running on laptop and emulating several clients generating real traffic.
Traffic sink end hosts	End systems where traffic statistics are collected.
Video client	VLC: Open source player, server used to stream real video traffic on the testbed.

B. *Traffic QoS management configuration with CNQF*

As described in section III, CNQF PDPs process high-level policies to drive decision making which triggers the actions to be taken in response to policy conditions within the policy rules. Recall that the RC is the element responsible for QoS mapping and configuration of the policy enforcement points i.e. edge routers and core routers in our testbed. Hence the RC contains the logic to configure the parameters using the Linux TC commands. The configuration is triggered by messages received from the RB via Java RMI. Future implementation could be extended to support policy provisioning protocols such as COPS (Common Open Policy Service).

In our tests we consider the validation of our CNQF RMS in implementing DiffServe IP QoS management. The parameters needed for initial configuration of the routers with DiffServe capability are supplied to the RB via a user interface in the CNQF application. These include the required DSCP to be employed for packet marking at the edge routers and the queuing disciplines for implementing the DiffServe Per Hop Behaviours (PHB). These parameters are then used within policy rules which invoke corresponding policy actions within the RCs that utilise TC commands to implement the configuration settings within the routers. Policy rules to map specific flows to the different DiffServe classes are then supplied to the CNRB for example:

Policy rule 1: *If src.ip_address == 192.168.20.10 MARK packets with DSCP ==0x2e*

This will invoke functions within the RC which will run the corresponding TC commands required to mark and filter packets from the specified address with DSCP 0x2e (i.e. DiffServe Expedited Forwarding EF class) :

tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32 match ip src 192.168.20.10/32 flow id 1:1

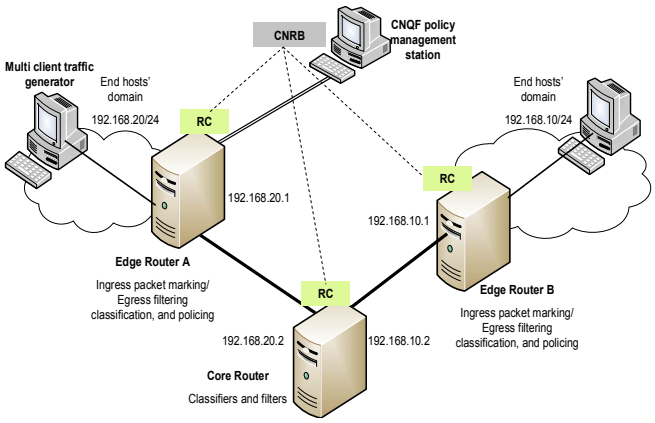

Fig. 3. CNQF development and evaluation testbed.

Where *flow id 1:1* corresponds to the filter that directs the packets matching ip source address 192.168.20.10 to the DSCP marker to be marked with the code point 0x2e. Further information on Linux TC utility can be found in [12].

V. VALIDATION TESTS

Tests conducted to verify the operation of DiffServe QoS management policies in the CNQF CNRB are presented in this section. The tesbed is connected as shown in Fig. 3. The traffic generator emulates injection of flows from different networks by using different IP addresses. Several flows were injected into the network through the edge router A. The edge to edge link from Router A to B is 100Mbits/s. The parameters of the flows are given in Table II. Initially, there was no configuration applied to the edge and core routers and effect on traffic is observed. Afterwards, DiffServe configuration is applied via CNRB policies and the corresponding configurations were applied to the edge and core routers to mark and classify specific flows with different DSCPs.

TABLE II. TRAFFIC FLOW PARAMETERS

Traffic type	Flow no.	Parameters
MPEG video	Flow 1	1Mbits/s VBR video stream
CBR	Flow 2	1Mbits/s CBR UDP traffic
VBR	Flows 3-20	Exponentially distributed inter-arrival times: 0.2ms Packet size: 1518 bytes
On-OFF	Flows 21-30	On: 3s Exponentially distributed packet inter-arrivals: 5ms Packet size: 1518 bytes; Off :3s

Due to space limitation, only selected test results are presented. Figs. 4 (a) and (b) depict bitrate measurements of an MPEG video streamed form the VLC player at source and destination points respectively. Background traffic consisting of CBR, VBR and on-off flows configured as shown in Table II, is also injected to edge router A. The bottom plot of 4(b) shows the impairing effect of the background traffic as the video arrives at the destination with much reduced bitrate (400 kbps on average). The top plot of 4(b) shows the same video arrived at the destination with similar bit rate pattern to the source measurements in 4(a). This is the case where the CNRB policy rules were applied to mark the video packets with EF DSCP in the edge routers and apply the PHB in the core router so that video packets were scheduled with higher priority at the egress interfaces.

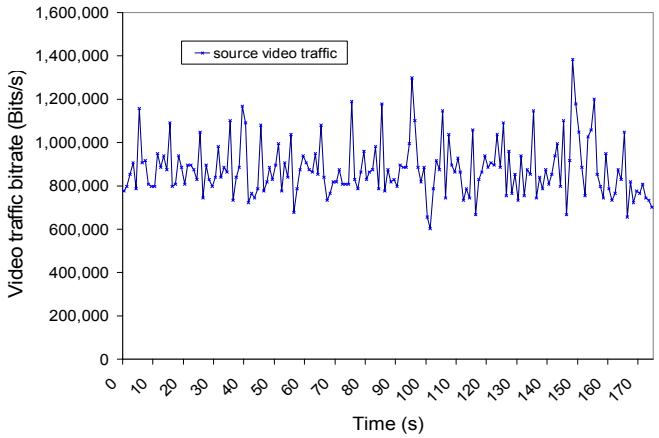

Fig. 4(a). Video traffic measurement at source.

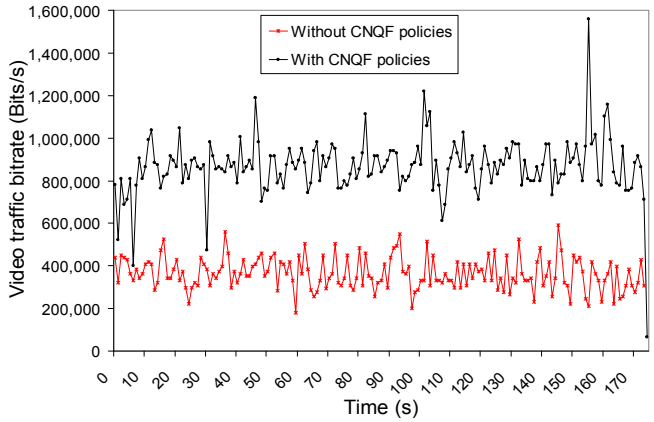

Fig. 4(b). Video traffic measurement at destination, with CNQF QoS management policies applied and without CNQF policies.

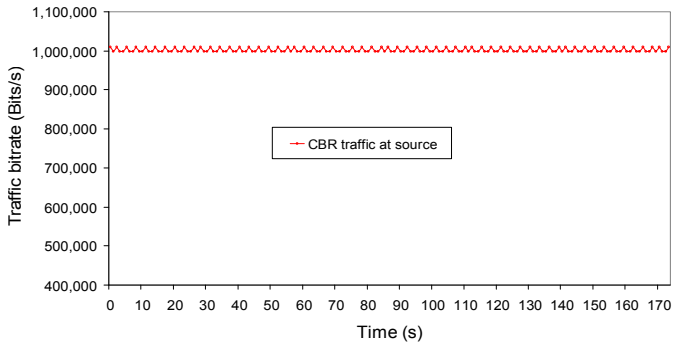

Fig. 5(a). CBR traffic measurement at source.

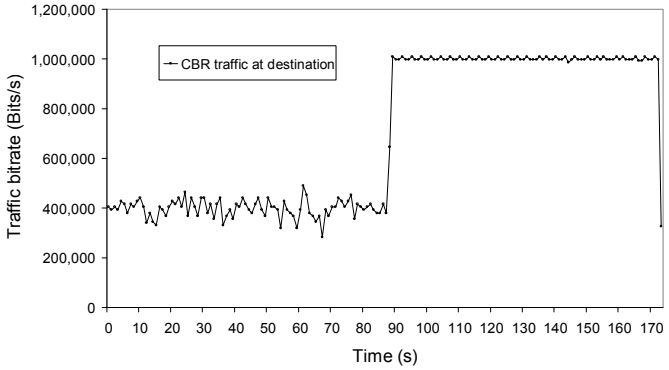

Fig. 5(b). CBR traffic measurement at destination, without CNQF policies and with CNQF policies applied at time 90 seconds.

In a separate test, the 1 Mbps CBR flow was measured at source and destination (Fig. 5(a) and (b)). The other flows from Table II were applied as background traffic and initially without CNQF QoS configuration policies. Fig. 5(b) shows a noticeable impairment to the CBR traffic at the destination, until at time 90s when the CNRB QoS management policies are applied to configure the network and mark the CBR traffic with EF DSCP. The consequent noticeable improvement can be seen from the 90s point onwards.

VI. Conclusion And Future Work

CNQF is designed to provide an infrastructure for policy-based management of converged networks through the various functional elements that make up its subsystems. Its design allows for closed-loop, scalable, context-aware and adaptive end-to-end QoS control in converged networks. In this paper we presented measurement-based evaluation of QoS management based on a Java prototype implementation of the CNQF Resource Management Subsystem. Its ability to facilitate DiffServe QoS management is demonstrated by test results obtained from a Linux-based testbed. In our future work we would investigate context-aware adaptive QoS management scenarios in converged networks based on CNQF architecture.

References

[1] V. G. Oziany, R. Good, N. Carrilho, N. Ventura, " XML-Driven Framework for Policy-Based QoS Management of IMS networks", in Proc. IEEE GLOBECOM 2008, New Orleans, USA, Nov. 2008.

[2] N. Mistry "The importance of policy-based resource control in future networks". Nortel Technical Journal, issue 4, 2006.

[3] S. Y. Yerima, G. P. Parr, C. Peoples, S. McClean, P. J. Morrow, "A Framework for Context-Driven End-to-End QoS Control in Converged Networks" in Proc. 6th International Conference on Network and Service Management, IEEE CNSM 2010, Niagara Falls, Canada.

[4] ETSI ES 282 003: Telecoms and Internet converged Services and Protocols for Advanced Networking (TISPAN); Resource and Admission Control Subsystem (RACS); Functional Architecture, June 2006.

[5] IETF Policy Framework Working Group web page: http://www.ietf.org/wg/concluded/policy.html

[6] 3GPP TS 23.207: 3rd Generation Partnership Project; End-to-end Quality of Service (QoS) concept and architecture, (Rel. 9) Dec. 2009.

[7] B. C. Kim et al., "A QoS Framework Design Based on Diffserve and SNMP for Tactical Networks" in. Proc. IEEE MILCOM '08, San Diego, California, Nov. 2008, pp. 1-7.

[8] P. Nanda and A. Simmonds, "A Scalable Architecture Supporting QoS Guarantees Using Traffic Engineering and Policy Based Routing in the Internet" International Journal of Communications, Networks and System Sciences, 2009. pp 583-590.

[9] N. Carrilho and N. Ventura, "Policy-Based Management of a DiffServe Network Using XML Technologies," in Proc. Third International Conference on Web Information Systems and Technologies (WEBIST 2007), Mar. 2007

[10] X. Guo et al. "A Policy-Based Network Management System for IP VPN" International Conference on Communication Technology Proceedings (ICCT 2003), Volume 2, April 2003.

[11] Ntools Traffic Generator for Linux: http://norvegh.com/ntools/

[12] Linux Advanced Routing And Traffic Control: http://lartc.org/