# Table of Contents – Part II

## Green Computing and Health Care Informatics

- Lot-Size Planning with Non-linear Cost Functions Supporting Environmental Sustainability ........................................... 1  
  Markus Beck and Guenther Schmidt

- Electronic Health Record (Dossier Medical Personnel) as a Major Tool to Improve Healthcare in France: An Approach through the Situational Semantic .......................................................... 7  
  Christian Beccer

- Data Mining Technique for Medical Diagnosis Using a New Smooth Support Vector Machine ........................................... 15  
  Santi Wuhan Pornani, Jasmi Mohamad Zain, and Abdullah Emhong

- Rule Induction as a Technique in Genomic Analysis for Cancer .......................................................... 28  
  M. Adib, Md. Moharref Hossain Sarkar, S. Syed Ahmed, Eendir Arava, and Fazool Siddiqui

- Clustering Analysis for Vasculitis Diseases .......................................................... 36  
  Pınar Yıldırım, Çınar Çeken, Kağan Çeken, and Mehmet R. Tohun

- Analysis on the Characteristics of Electroencephalogram (EEG) and the Duration of Acupuncture Efficacy Depending on the Stimulation at the Acupuncture Points .......................................................... 46  
  Jong-Heon Shin and Dae-Hyun Park

## Web Services Architecture, Modeling and Design

- Web Services Platform Based Context-Aware Services across Home .......................................................... 60  
  Jin-Neng Wu and Yu-Chung Chou

- Web Services Testing Approaches: A Survey and a Classification .......................................................... 70  
  Mohamad L. Ladm

- Benefits of Semantics on Web Service Composition from a Complex Network Perspective .......................................................... 80  
  Chantal Cherifi, Vincent Llobet, and Jean-François Seraucicci

- Development Tool for End-to-End QoS Sensitive Frameworks and Technologies .......................................................... 91  
  Bogdan Iancu, Adrian Pricalea, and Vasile Tudor Dateret
Development Tool for End-to-End QoS Sensitive Frameworks and Technologies

Bogdan Iancu, Adrian Peculea, and Vasile Teodor Dadarlat

Technical University of Cluj-Napoca, Computer Science Department,
26 G. Baritiu Street, 400227 Cluj-Napoca, Romania
(Bogdan.Iancu,Adrian.Peculea,Vasile.Dadarlat}@cs.utcluj.ro

Abstract. This paper proposes a software development tool for the
development and testing of end-to-end QoS sensitive frameworks and
technologies. The development tool is a modular software system, which
allows researchers to assess the performances and functionalities of diffe-
rent QoS frameworks and technologies using an experimental methodol-
y, rather than using simulation environments. We describe and analyse
the behavior of the proposed development tool by testing, in a compar-
avative manner, two end-to-end QoS frameworks: On-Demand QoS Path
framework (ODP) and Self-Adaptive Bandwidth Reconfiguration QoS
framework (SAR).

Keywords: development tool, end-to-end QoS, experimental
methodology

1 Introduction

Classical computer networks use best-effort model, a model which assumes that
the main task of the network is to provide data to the destination. Transported
traffic types have been diversified with the development of computer networks,
and several types present some constraints on the transport parameters such as
minimum bandwidth, delay, delay variation or data loss. Transport procedures
affect traffic flows, which require traffic characterization and setting requirements
for quality of service. To allow applications using these types of traffic to com-
municate in the network, while respecting the imposed constraints, the services
are defined with the specific traffic parameters that must be followed and quality
of service is implemented (Quality of Service – QoS) to enable compliance with
three parameters.

For providing QoS over IP networks, the network must perform two tasks [1]
[2]. The first task is to differentiate between types of traffic or service. This task is
performed at user-network and network-network interfaces and includes marking
techniques and packet classification. The second task is to differentiate between
traffic classes. The task is performed by the network, including traffic selection
(Traffic Policing), active management of queues (Active Queue Management -
AQM) and packets planning and modeling. This paper presents the design and

© Springer-Verlag Berlin Heidelberg 2010.
implementation of a development tool for testing and testing end-to-end QoS sensitive frameworks and technologies. This effort is motivated by the need of an experimental methodology that allows researchers to assess the performances and functionalities of different QoS frameworks and technologies.

The remaining of the paper is organized as follows: Section 2 outlines the main end-to-end QoS frameworks, the existing development and testing methodologies and presents the motivation for our work. Section 3 describes the architecture and the functioning of the proposed development tool and Section 4 presents the implementation and testing, using the proposed development tool, of two end-to-end QoS frameworks: On-Demand QoS Path framework (ODP) [3] and a novel QoS framework called Self-Adaptive bandwidth Reconfiguration QoS framework (SMR) [4], and discusses the results and performance analysis. Section 5 concludes the paper.

2 Background

Quality of Service frameworks in computer networks are sets of mechanisms that run over multiple network equipment with the aim of ensuring network services.

Integrated Services [5] have developed a novel architecture for resource allocation, which aims to meet the requirements of real-time applications. The basic idea is reserving resources for each stream. The purpose of the Integrated Services is based on maintaining IP datagram network model, while reserving resources for real-time applications. The Integrated Services architecture uses a set of mechanisms and protocols for explicit reservation of resources in the Internet. Before transmitting packets, applications reserve resources along the route. Integrated services have standardized two service models: guaranteed service and controlled load service. In addition to these two services there is also a best-effort service which does not guarantee quality of service. For the Integrated Services, the RSVP protocol (Resource Reservation Protocol) was developed by the IETF as a resource reservation setup protocol for the Internet.

Differentiated Services architecture is a simple architecture that provides several levels of service. If in the Integrated Services architecture resources are allocated to the individual flows, Differentiated Services model [5] divides traffic into a small number of classes and allocates resources based on these classes. In the Differentiated Services approach traffic is divided into a few groups called forwarding classes. Forwarding class part of the package is coded in the field of the IP packet header. Each forwarding class represents a predefined forwarding treatment in terms of exclusion priority and bandwidth allocation. Individual classes represent aggregated traffic. In the Differentiated Services network, network border nodes (boundary nodes or edge nodes) and the nodes within the network (internal nodes or core nodes) have different responsibilities. Edge nodes have two functions: packet classification and traffic conditioning. Core nodes forward packets based on forwarding classes from the packet header.

Two common methods for assessing the performance of a large scale end-to-end QoS framework are used in computer networks: simulation methodology and experimental methodology.
Firstly, a simulation (analytical) methodology assumes the usage of a software application that is able to provide support for simulation of different protocols, technologies or processing methods. Some popular network simulators, that are used in academic, commercial and industrial communities, are OPNET Modeler, NS-2 [7] and OMNeT++ [8]. The main advantage of these tools is the possibility to test the behavior of different technologies, without having physical devices. A disadvantage of this type of methodology however, is that one can never be sure of the resulting behavior of the system; the only way to be sure is to try it.

Secondly, the experimental methodology assumes that one has to determine the behavior of a system experimentally—exploring the behavior, making hypotheses about the behavior [9] and testing them on physical testbeds. The intended behavior of a system is only a hypothesis about the system's behavior, thus it must be checked by experimentation [10]. This methodology have proven an accuracy close to that of real cases (the application of technologies in large networks WAN) [11].

In order to develop and test different QoS technologies and frameworks we have proposed, designed and implemented a software system for the development of QoS sensitive frameworks and technologies which allows researchers to assess the performance and functionality of different QoS technologies using an experimental methodology.

3 The Development Tool

The software system is primarily designed to assess the performance and functionality of different QoS sensitive frameworks using an experimental approach.
The developed system allows the conversion of a computer into a software router, thus achieving rapid, modular and extensive test configuration, while giving its users a reconfigurable set of traffic statistics.

The development tool for QoS sensitive frameworks and technologies uses a component-based architecture (Fig. 1) consisting of an inter-router communication protocol and the following modules: a network interface module used for handling network interfaces and network packets, a routing module and a test QoS framework module which are the core of the development tool, and a user interface module.

Also, a benchmarking system [12] was developed and integrated with the development tool. The benchmarking allows the possibility to define and store complex traffic patterns that can be rechaged for making further measurements to test various QoS techniques based on the same traffic characteristics.

3.1 Traffic Classification

The application defines two types of routers: the edge router (border router) and core router (interior router). Edge routers make the decision of acceptance or rejection for each input flow, map the traffic flows to their corresponding classes and forward packets belonging to admitted network flows. Core routers recognize traffic classes and provide service differentiation based on those classes.

Traffic classification is performed by the edge routers. Traffic is classified based on two criteria: transport layer protocol and port number. The most significant three bits of TOS field (Type of Service) from the IP header will be used for marking the packets. Thus one can define eight classes of traffic. Each class is assigned a queue where the packets to be forwarded are stored. The use of three bits for marking the packets is motivated by the fact that for the 802.1Q encapsulation also three bits are defined for service classes, thus traffic marking is uniform.

3.2 Communication Protocol

The inter-router communication protocol allows the communication between the software routers, in order to discover new routers and to exchange topological information. The protocol is used for the construction of the network topology database, called LSDK (Link-State Database). Six types of custom ICMP control messages were defined: Advertise, Acknowledge, InfldLSDK, NewLSDK, Delay and DelayTimer.

The Advertise and Acknowledge messages are used to build a router's network topology database - Link-State Database. Upon the receiving of the first acknowledge packet, a newly started router must perform a synchronization step. This is useful to compute accurate delays - one of the metrics used by the routing application. Only synchronized routers can participate in further actions.

To compute the delay metric two types of messages are used: Delay and DelayTimer. In order to exchange topological information, InfldLSDK and NewLSDK messages are used. The first message is sent by the neighboring routers to a new
router and contains their current LSDBs and the second message is used by all
routers to announce any alteration in the current topology.
Based on the constructed LSDB, routers are able to recognize, automatically,
their role in the test network: core routers or edge routers.
The communication protocol can be further extended to accommodate the
tested frameworks' specific control messages, using the same format as the above
described messages.

3.3 Network Interface Module
The network interface module avoids the functions of SharpPcap framework [13]
—a accessible standalone application, which is a wrapper over Wireshark [14], the
industry-standard tool for link-layer network access in Windows environments.
By bypassing the protocol stack, Wireshark functions allow applications to cap-
ture and transmit network packets. The network interface module interfaces with
SharpPcap framework, providing specific tasks, such as network adapters detec-
tion, functions to manipulate a selected network adapter, traffic packets captur-
ing, retransmission, and analysis functionalities.

3.4 Routing Module and Tested QoS Framework Module
The routing module provides the system the methods needed to classify traffic
and to construct the LSDB, using the communication protocol described above.
Based on the LSDB information, the default routing table is computed based on
Dijkstra's algorithm using the delay metric.
Core routers recognize traffic classes and provide service differentiation based
on those classes, avoiding the default routing table.
Edge routers are used to support end-to-end QoS, having the role, in addition
to the core routers, to make admission control and routing decisions, depending
on the QoS sensitive framework being tested. Thus, the Routing module is in-
terfaced with the Tested QoS Framework module, making available the LSDB
information. The system user has the option to choose between the default rout-
ing table calculated from the LSDB using Dijkstra algorithm, a static routing
table based on configuration files, a class-based routing table or other routing
algorithms, according to the QoS sensitive framework that is intended to be
developed and tested.
 Also different methods of admission control, management, identification and
packet forwarding, and new control messages can be integrated, specific to the
end-to-end QoS sensitive frameworks that are being tested.
The statistics component allows the development tool to store traffic statistics
such as the number of accepted / rejected flows, the number and type of control
messages used by the evaluated frameworks. The generated statistics will to de-
terdue the performance of the admission control method (admitted vs. rejected
flows) and the extend of the network overload due to the control messages of the
tested framework. Also, the extension of the statistics set that are intended to
be considered is permitted.
4 Experimental Results

We analyze the behavior of the proposed development tool by testing, in a comparative manner, two end-to-end QoS frameworks: On-Demand QoS Framework (ODF) and Self-Adaptive Bandwidth Recomputation QoS Framework (SAR). The SAR framework was developed using the development tool, by enhancing the ODP framework.

A test network composed of three edge routers (RG1, RG2 and RG3) and three served networks (H1, H2 and H3) was built using the proposed development tool. Each physical line has a capacity of 10Mbps full duplex. Routers use class based static routing. Static routing was chosen because it was intended to test the frameworks in identical conditions.

The modularity of the software system allows adapting the development tool to run the tested framework just by changing of modifying few components. The network interface module and the routing module were implemented in the same manner.

The only modifications were made to the tested QoS module. The admission control and the routing component are specific to the tested framework and are described in [3] [4]. Both for ODF and SAR frameworks the Edge-to-Edge approach was chosen. Admission control and resource allocations are controlled at edge routers level. Also the communication protocol was extended to accommodate the frameworks specific control messages. The tests were intended as performance comparison for ODF and SAR frameworks. The statistics module collects data, at each edge router, concerning the following parameters: the number of admitted/rejected flows in the network - to determine network resource efficiency and the number of control messages transmitted - to establish the degree of additional load added to the network - parameter that affects the frameworks scalability.

The traffic classes and the traffic patterns were defined as follows. Four traffic classes were considered, class 1 having the highest priority and class four, the lowest. Class 1 accommodates the control messages of the tested frameworks. Class 2 is composed of UDP traffic with destination port range from 2000 to 2099 and class 3 of UDP traffic with destination port range from 3000 to 3099. Class 4 is the best-effort class. Two traffic patterns were defined. For each flow of the two traffic patterns a reserved bandwidth equal to 0.5Mbps was considered.

In the first test, flows from class 1 and 3 were injected into the network, using the following distribution: 60 flows are injected into the network, each flow transmitting for 3 minutes and 58 seconds. Each flow starts its transmission with a delay of 2 seconds from the previous flow. Thus, the network is loaded with 0.5Mbps every 2 seconds and a balanced distribution of traffic, from and to the served networks, is ensured.

In the second test, flows from one class 1 and 3, were injected into the network, using the same distribution. The results of the tests show that the number of admitted flows by SAR framework is greater than ODF framework. SAR framework shows a 65% increase in the number of admitted flows, which demonstrates a more efficient use of network resources.
Fig. 2. SAR vs. ODP: a. test I results; b. test II results

For both test scenarios the number of control messages transmitted was the same, thus SAR proves as efficient as ODP at this criterion. However the performance analysis of admitted vs. rejected flows, in both test scenarios, shows the superiority of SAR framework.

5 Conclusions

A software system for developing and testing of QoS sensitive frameworks and methodologies was proposed. The main advantage of this system is that it allows testing and validation of the functionalities and performances of QoS technologies and frameworks, in a complex environment, with accuracy close to that of real ones, using an experimental methodology.

The development tool, through its modular software solution, allows adapting the development tool to run various end-to-end QoS frameworks, just by changing or modification of components. Also, the system allows to save traffic statistics, for instance, the number of flows accepted / rejected, or the number and type of control messages used by the evaluated frameworks, and to view these statistics in digital format. Thus, the system allows in-depth analysis of the frameworks behavior.

The proposed development tool allowed the elaboration of a new end-to-end QoS sensitive framework called SAR and also allowed comparative testing with ODP framework, using a network of programmable routers and hosts.
Acknowledgments. This work was supported by the PNII-IDEI 328/2007 "QAF - Quality of Service Aware Frameworks for Networks and Middleware" research project within the framework "National Research, Development and Innovation Programme" initiated by The National University Research Council Romania (CNCSIS - UEFISCUS).

References

Communications in Computer and Information Science

The CCIS series is devoted to the publication of proceedings of computer science conferences. Its aim is to efficiently disseminate original research results in informatics in printed and electronic form. While the focus is on publication of peer-reviewed full papers presenting mature work, inclusion of reviewed short papers and abstracts reporting on work in progress is welcome, too. Besides globally relevant meetings with internationally representative program committees guaranteeing a strict peer-reviewing and paper selection process, conferences run by societies or of high regional or national relevance are considered for publication as well.

The topical scope of CCIS spans the entire spectrum of informatics ranging from foundational topics in the theory of computing to information and communications science and technology and a broad variety of interdisciplinary application fields.

Publication in CCIS is free of charge. No royalties are paid, however, CCIS volume editors receive 25 complimentary copies of the proceedings. CCIS proceedings can be published in time for distribution at conferences or as post-proceedings, as printed books and/or electronically as CDs, furthermore CCIS proceedings are included in the CCIS electronic book series hosted in the SpringerLink digital library.

The language of publication is exclusively English. Authors publishing in CCIS have to sign the Springer CCIS copyright transfer form, however, they are free to use their material published in CCIS for substantially changed, more elaborate subsequent publications elsewhere. For the preparation of the camera-ready papers/flows, authors have to strictly adhere to the Springer CCIS Authors’ Instructions and are strongly encouraged to use the CCIS LaTeX style files or templates.

Detailed information on CCIS can be found at www.springer.com

ISSN 1865-0929
ISBN 978-3-642-14305-2