Routing Management Based on Statistical Cross-Layer QoS Information Regarding Link Status

Melinda Barabas, Georgeta Boanea, Andrei Bogdan Rus, Virgil Dobrota
Communications Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
Email: {Melinda.Barabas, Georgeta.Boanea, Bogdan.Rus, Virgil.Dobrota}@com.utcluj.ro

***Abstract*—This paper presents the design principles and the practical implementation of a routing management solution which takes into account statistical cross-layer Quality of Service information regarding the state of the network. Link monitoring and communication between neighboring nodes is realized by the routing management system, while the routing protocol deals with routing tables and packet forwarding. We propose two types of management entities: *a)* local management entity (LME) responsible for monitoring directly connected incoming links and detecting traffic changes and *b)* domain management entity (DME) which captures the global state of the network and informs routers to avoid congested areas. The goal is to offer physical layer information to the routing algorithm, i.e. the metrics will depend on the traffic load and the delay measured on individual links. By identifying congested areas and anomalies, we can ensure a situation-aware path selection, improving the global capacity of the network by determining optimal routes.**

***Index Terms*—congestion control; cross-layer QoS; routing management; self-management; traffic-aware shortest paths.**

I. INTRODUCTION

Routing management is defined by different types of management functions necessary for the efficient operation of the routing process. A significant part of this consists of the monitoring procedures which help to understand what is going on inside the network. Traditional management relies on human supervision and monitoring, as well as manual interventions, in order to ensure that the system operates as desired. Traffic can be manually rerouted, but only after an extended service outage. But new networks and services are becoming more complex and thus new approaches are needed. The authors of [1] summarize the state of worldwide research in the domain of management and propose design requirements for Future Internet management solutions. New trends focus on increasing the level of automation, developing scalable and intelligent self-management systems which do not require human intervention, as presented in [2].

The role of management for the different routing mechanisms is discussed in [3]. Initially, it was believed that routing management was not necessary because the adaptive routing protocols, such as OSPF (Open Shortest Path First), can react to failures and changes in the network. But efficient traffic engineering is not possible if routing protocols do not take into consideration the real-time traffic conditions.

In the case of OSPF, the most widely used routing protocol in large networks, link connectivity is verified via Hello messages sent every 10 seconds. If no Hello message is received from a neighbor for 40 seconds, it is marked as being down. Thus, OSPF is not able to solve the problem of congestion detection. If Hello messages are missing because of congestion, apart from the fact that in the majority of cases congestion is unidirectional, OSPF eliminates the possibility of using the affected link in both directions. In addition, all links through that router are also marked as down and will not be used. Thereby, the initial traffic routed through this node will be transmitted on other links, reducing the load balancing in the network and increasing the possibility that other congested areas appear. Another problem is the fact that if a link flaps constantly due to congestion, but at least 1 out of every 4 Hello messages is received, OSPF does not detect the cause of missing packets.

To solve the problems mentioned above, a routing management system is proposed which offers cross-layer QoS information to the routing algorithm, helping it to avoid congested links and determine optimal routes.

The remainder of the paper is organized as follows. In Section 2 the main objectives of a routing management system are identified and the characteristics of the proposed system are described. Section 3 presents the designed management entities and the interaction between them. In Section 4 the practical testbed used for performance evaluation is described, followed by the experimental results in Section 5. Finally, Section 6 concludes the paper and discusses future work.

II. ROUTING MANAGEMENT SYSTEM

QoS-aware routing has become a great challenge in computer networks. It refers to guaranteeing traffic delivery in terms of bandwidth, delay and packet loss. Traditional QoS models are presented in [4]. Usually, QoS-aware routing algorithms, like the one described in [5], assume source routing and allocate bandwidth in advance, according to traffic parameters. Besides resource admission control, traffic shaping and dropping can also be used in order to meet QoS requirements. These solutions have some disadvantages, for example the waste of bandwidth on the entire path reserved in advance or packet losses due to special packet drop policies.

In this paper we present the design principles and the practical implementation of a distributed routing management system. The main objectives of the approach described herein are the following: *a)* real-time monitoring of network parameters; *b)* detecting events, anomalies and congestion in order to estimate the state of the network; *c)* communicating with the routing protocol in order to recalculate the routes and *d)* self-management. Through these functionalities QoS-awareness can be achieved.

The main characteristic of our routing solution consists of the idea of separating the link monitoring and the update of network state information from the routing itself. The routing management system takes over the monitoring functionalities and all the communication between neighboring nodes, while the routing protocol deals with the routing decisions and packet forwarding. Hence, the information gathered by the routing management system regarding the state of the network can be reused for other purposes.

The routing management system is composed of management entities which collect measurement data through monitoring. Link status information is gathered through cross-layering and is used for QoS-aware routing. The goal is to determine the traffic load and the delay on each link in order to offer an overview of current network resource availability. Based on the topology and the network state, the routing algorithm running on a given node determines the routing table of that node. Interaction between neighboring nodes is carried out just by the routing management system; no other routing messages are exchanged. The proposed routing management solution represents a situation-aware self-managing system which is capable to dynamically adapt to external events, minimizing the need for human intervention.

The solution presented in this paper relies on cross-layering techniques which allow communication between non-neighboring layers of the OSI model. The implemented cross-layer QoS mechanism is described in detail in [6] and [7]. Information derived from the physical layer is passed to the upper layers to assist the routing management system in determining the operation modes of routing protocols. Two types of QoS parameters are used, reflecting link status at the Physical Layer and MAC Sub-layer: *a) available transfer rate* (ATR) which is obtained through passive monitoring and *b) one-way delay* (OWD) which is measured in real-time for each direction of each link through an active monitoring approach (i.e. by injecting probes). OWD consists of processing-, queuing-, transmission- and propagation delay. When routers receive the link state information (ATR and OWD), they can compute the traffic-aware shortest paths for optimal network performance.

III. DESIGNING THE MANAGEMENT ENTITIES

Two types of management entities were defined: *Local Management Entity* (LME) and *Domain Management Entity* (DME). In this section we describe the functionalities and the operation mode of the implemented entities, as well as the interaction between them.

Our solution is a software application running on Linux-based machines. Management, measurement and routing applications are written in C++ and Qt under Fedora operating system, using MySQL database with mysql++ connector. Management entities located on different nodes communicate through XML messages, storing the network status information in local databases.

A. Local Management Entity

Local Management Entities are software applications located on every node of the administered system. These capture the local network state by observing, collecting and interpreting data and by communicating with neighboring nodes.

The main functionalities of such an entity are the following:

- *Starting the cross-layer measurement applications for monitoring ATR and OWD on the directly connected inbound links.* Resource monitoring is made periodically, second by second.
- *Using statistical indicators to detect changes on individual links.* These can indicate not just the real-time values, but also the trend of the evolution for a specific parameter. At the moment we are using two different indicators: *simple moving average* (SMA) calculated every second for an interval of $N = 10$ seconds in order to estimate the value of the monitored parameter, and *standard deviation* (σ) for identifying events/changes of the traffic conditions.

$$SMA(t) = \frac{1}{N}\sum_{i=0}^{N-1} x(t-i)\ . \tag{1}$$

$$\sigma(t) = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}\left(x(t-i) - SMA(t)\right)^2}\ . \tag{2}$$

- *Extracting the current status of the local network from the collected cross-layer QoS information.* The local network refers to the directly connected incoming links.
- *Communicating with other LMEs and DME over UDP or TCP connections by sending XML messages.*

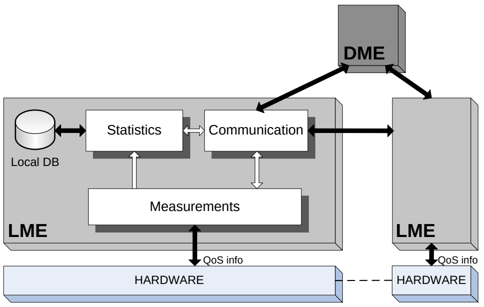

Fig. 1. Local Management Entity

Fig. 1 shows the main building blocks of the LME. The *measurement module* manages the monitoring applications.

It receives the ATR and OWD measurements through cross-layering techniques and makes these values available to the statistics block. The *statistics module* calculates the SMA and the standard variation of the monitored network parameters and identifies changes in the link status. This module is also responsible for storing the status information in a local database. The *communication block* is responsible for transmitting messages to other entities.

The flowchart diagram in Fig. 2 depicts the operation of such an entity. LME starts monitoring the local links as it detects the corresponding neighboring nodes. The SMA and the standard deviation are calculated for a certain time period, this is done every second. The statistical indicators are compared with predefined threshold values ($th1$ and $th2$). If $SMA < th1$ or $\sigma > th2$, the DME is announced that an event occurred. Because $N = 10$ in equation (1), congestion can be detected after at least 10 seconds. We chose this value as a compromise between the possibility of generating false alarms and the reaction time of the system.

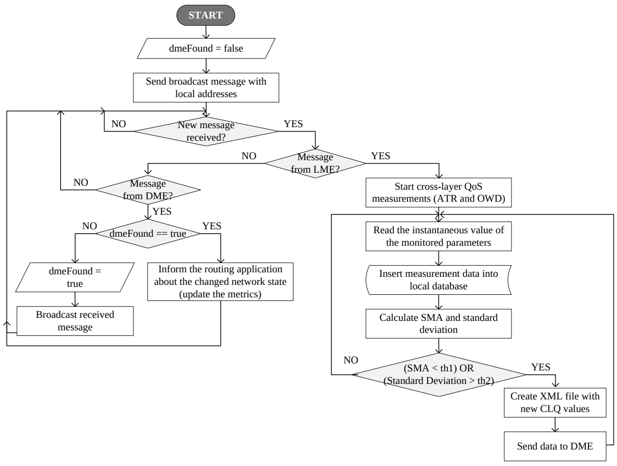

Fig. 2. Operation mode of LME

B. Domain Management Entity

To address the problem of scalability, we consider that the network is divided into domains. A domain management entity corresponds to a domain (i.e. a group of routers), and is dealing with problems that require a global vision of the network (e.g. routing metrics). In the case of routing management, the system has to provide the same information regarding the network state in all nodes. Otherwise the routes could not be determined correctly. This means that we have to centralize this information and send it simultaneously to the routers.

The modules of the DME are illustrated in Fig. 3. Through the *communication module*, the DME interacts with LMEs. The *statistics module* is responsible for storing the status information in a local database. The domain management entity was designed to have a module for *network state prediction*, but this is not yet implemented practically. We intend to use artificial neural networks (NN) for time series forecasting, allowing us to detect congestions before they appear. Thus, the routing management system would be able to take actions in order to avoid these affected areas, reducing packet losses.

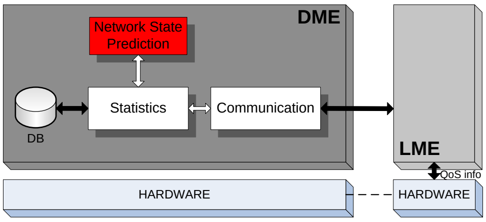

Fig. 3. Domain Management Entity

The DME can be located on any of the nodes of the domain. The existence of such an entity is not just motivated by the need to centralize data, but by performing functions exceeding the scope of LMEs. It is able to perform more complex operations that require additional resources, such as prediction. We chose this approach because forecasting is a resource consuming task in the means of memory and processing power. Most nodes in a network have limited capabilities, thus adding prediction functionalities in every single node is not feasible.

C. Interaction between Entities

As mentioned before, different management entities interact by sending XML messages. In Fig. 4 we can see the communication between the different types of entities. This represents a very simple scenario with the purpose of illustrating the message-passing between entities.

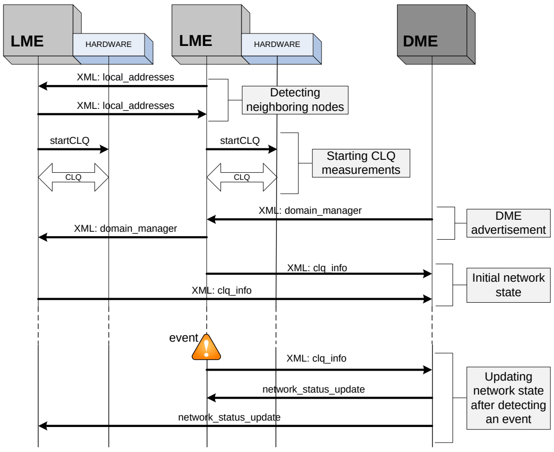

Fig. 4. Interaction between LMEs and DME

We can identify five major steps in the presented message flow:

1) *Detecting neighboring nodes:* The LMEs periodically advertise their local addresses in order to detect directly connected nodes (this also includes topology discovery) and to start monitoring the corresponding links.
2) *Starting the cross-layer QoS measurements:* After detecting the direct neighbors, the measurement applications are started. Each LME will monitor the ATR and the OWD on incoming links. Passive monitoring is

used for ATR, and active monitoring for OWD (i.e. by sending probes to the neighboring nodes).

3) *Advertising the presence of DME:* DME periodically advertises its presence by sending XML messages which contain its IP address and the destination port.
4) *Determining initial network state:* When LMEs detect the presence of the DME, they will send an XML message with initial link status information. In this way, the domain manager can have an initial global view of the network topology and the status of the network.
5) *Updating network state after event detection:* The network status updates are event-driven. Through the congestion discovery mechanism, a LME detects changes in traffic conditions on a monitored link and informs the DME. DME sends the global state of the network to LMEs at maximum 2 hop distance from the affected link, limiting the amount of signaling and the number of routers which will have to make changes to the routing table. TCP connections are used for this type of interaction in order to ensure that the messages are received.

Routers will update their routing tables according to actual traffic conditions indicated by DME, selecting the route with the largest ATR and the lowest delay. When selecting the path, the routing protocol uses the idea of the Ford-Fulkerson algorithm, i.e. as long as there is no congestion, the current paths are not changed. By using this technique, we avoid frequent oscillation of the routing process, which could cause out-of-order packet arrivals and performance degradation.

But if a link is found to be congested, not all traffic will be rerouted because otherwise the link would remain unused. At a given moment only loss-sensitive streams (e.g. video, audio) belonging to a *source–destination* pair will be transfered to a different path. For this, traffic flows affected by packet losses and delays have to be identified. The network state is reevaluated and if the congestion persists other stream will be also re-routed. This is possible because we foresee a *multipath routing solution* to be used. This mechanism allows for balancing the traffic load sent between a source–destination pair on different links.

IV. PERFORMANCE EVALUATION

For performance evaluation we selected the test architecture presented in Fig. 5. We chose this network topology because it offers sufficient paths between the source and destination nodes, but at the same time it is simple enough to allow practical implementation using physical machines. Beside the source and destination, the testbed contains six routers (R1, R2, R3, R4, R5 and R6) running on Linux-based machines with Fedora operating system. The computers have Intel Core 2 Duo CPUs running at 2.66 GHz and 4GB of memory. On each machine runs a QoS-aware multipath routing application and the LME application. In addition, on R5 there is a domain management entity running.

The experimental parameters are shown in Table I.

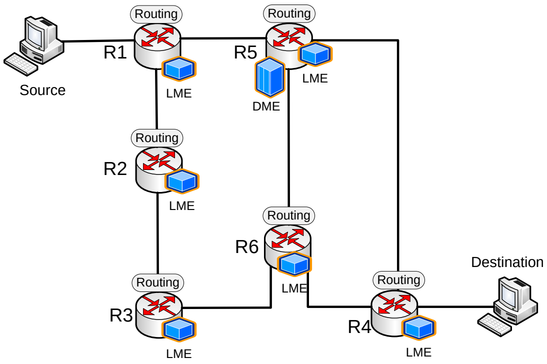

Fig. 5. Practical testbed

TABLE I
EXPERIMENTAL PARAMETERS

Parameter	Value
Number of routers	6
Background traffic generator	iperf
Multimedia traffic generator	VLC
Experiment duration	3 minutes
Congestion appearance	after 1 minute

To demonstrate the beneficial effects of cross-layer QoS-aware routing, we chose to use video streams in the experiments because the Internet is experiencing a substantial growth of video traffic. Providing good video quality is a major problem since video traffic is both massive and intolerant to packet loss or latency. According to Cisco VNI (Visual Network Index) forecasts, the sum of all forms of video will exceed 91% of global consumer traffic by 2014 [8].

For the test scenario, we assume that MPEG-4 VBR (Variable Bit Rate) video flow is sent over RTP/UDP/IP between the source-destination pair. The video stream has an average bitrate of 1.5 Mbps and it is transmitted for a period of three minutes, by a VLC client running on the source. The IP packets sent by the VLC client contain 1356 bytes (RTP/UDP/IP headers: 40 bytes; multimedia payload: 1316 bytes). The main parameters of the video stream are presented in Table II.

TABLE II
VIDEO STREAM PARAMETERS

	Parameter	Value
Encapsulation		MP4
Video	Codec	H.264
	Resolution	624×352
	Frame rate	25
Audio	Codec	AAC
	Sample rate	48 kHz
	Channels	Stereo

To introduce congestion between R5 and R4, we used `ethtool` to limit the nominal rate of the link to 100 Mbps (the actually observed transfer rate at the network layer is less than this theoretical maximum). After this, we generated background traffic between R5 and R4 using `iperf`: 10 UDP streams, with the total transfer rate of 100 Mbps.

The goal of the tests carried out is to demonstrate the capability of the routing management system to reduce the negative effects of congested links by providing statistical cross-layer QoS information regarding link status to the routing application. We performed experiments in order to compare the behavior of the following routing approaches in case of congestion on one or more links: *Case 1)* OSPF (Open Shortest Path First) and *Case 2)* QoS-aware routing application communicating with the routing management system.

The evaluation of the performance of each routing solution is based on the parameters of the received video stream. We take into consideration the number of lost packets, determined by examining the Sequence Number field in the RTP (Real-time Transport Protocol) header and observing if there is a gap in the counter. Besides the percentage of lost packets, other two video quality metrics are also used for performance evaluation, namely: *a)* magnitude of loss events and *b)* discontinuity counter. The magnitude of loss events expresses the number of packets that were dropped at each loss event (note that a magnitude of 0 means the packet arrived successfully at the destination). The discontinuity counter measures discontinuity events, characterizing the frequency with which discontinuities were detected, i.e. the number of times loss was detected.

V. EXPERIMENTAL RESULTS

If there is no congestion in the network, both approaches use the same path for routing the packages (source–R1–R5–R4–destination) and no packets are lost during the streaming.

In the experiments, we suppose that after 1 minute of streaming, link R5–R4 starts to be affected by congestion due to background traffic generated with the `iperf` network testing tool. As a result, ATR drops below the required rate to transmit the stream and the OWD from node R5 to R4 increases.

Case 1:

Fig. 6 illustrates the testbed for *Case 1*. After we introduce congestion between R5 and R4, OSPF does not modify the path between the two nodes because it does not take into consideration the physical state of the links. As an effect, we observe packet losses at the destination node, the quality of experience being very poor. In this case 6461 packets where missing at the destination out of 21980 transmitted packets, i.e. 29.39% of the transmitted packets was lost.

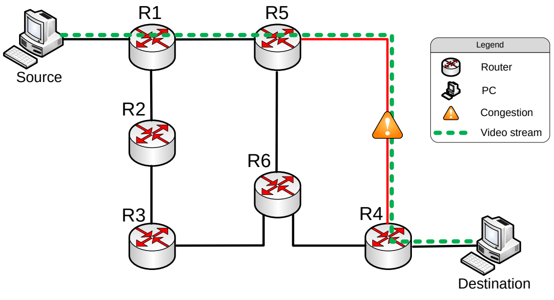

Fig. 6. Case 1 – Testbed

In Fig. 7 the magnitude of loss events is presented. The average magnitude of loss events is 0.416 and the maximum value is 16. The discontinuity counter is equal to 4413.

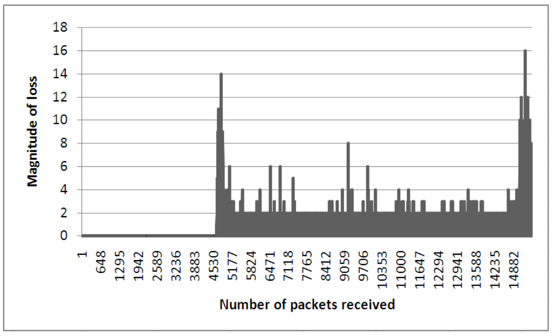

Fig. 7. Case 1 – Magnitude of loss events

Case 2:

In this case, the congestion is detected by the local management entity located on router R4. The QoS-aware routing solution is notified by the routing management application about the congestion and it determines the new route from source to destination node, based on the updated routing metrics. In this scenario, presented in Fig. 8, the new path which is written into the routing tables will be source–R1–R5–R6–R4–destination. The quality of the streaming is affected just for a short period of time, as a result of congestion detection and router reconfigurations. The percentage of lost packets was 3.37%; this means that only 741 packets got lost out of 21980.

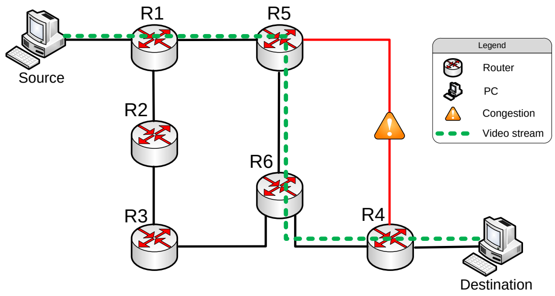

Fig. 8. Case 2 – Testbed

Table III depicts the number of packets sent and received in the two cases analyzed earlier.

TABLE III
COMPARISON BETWEEN THE TWO CASES

	Case 1	**Case 2**
Sent packets	21980	21980
Received packets	15519	21239
Lost packets	6461	741
Percentage of lost packets	29.39%	3.37%
Average magnitude of loss	0.416	0.035
Maximum magnitude of loss	16	11
Discontinuity counter	4413	273

Fig. 9 illustrates the magnitude of loss event in *Case 2*. It can be observed that losses appear just for a short period of time, until the stream is rerouted. In this case, the average magnitude of loss events is 0.035, with a maximum value of 11. The discontinuity counter is 741.

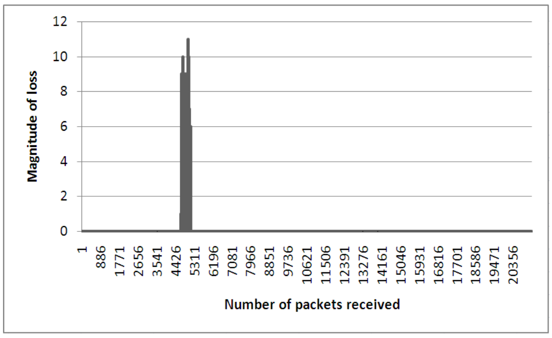

Fig. 9. Case 2 – Magnitude of loss events

In Fig. 10 the evolution of the success ratio is shown over the experiment duration. The success ratio at a given moment is determined by the ratio of the number of packets received successfully and the total number of packets sent. In *Case 1* the success ratio is falling steadily upon the appearance of congestion, reaching a minimum of 70.61% at the end of the experiment. In *Case 2* the success ratio drops for a short time to 87.4%, but after that it recovers, increasing to the final value of 96.63%.

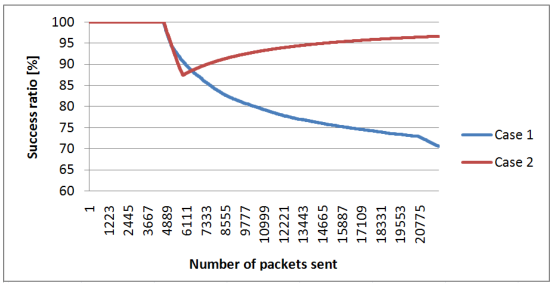

Fig. 10. Success ratio over experiment duration

The percentage of lost packets is 29.39% when using OSPF, whereas in the case of our solution it drops to 3.37%. As a conclusion, we can say that due to routing management a new route can be selected which avoids the congested link and ensures the lowest OWD. Thereby, the percentage of lost packets can be reduced significantly which results in a better quality of experience. Thus, the routing management system can complete the routing algorithm, offering a more reliable media streaming. Not only the media streaming benefits from this rerouting, but also the other transmissions on the affected link because in this way the congestion can be alleviated or can disappear. This comes with the cost of introducing additional traffic overhead necessary for transmitting management information throughout the network.

Regarding the average hardware resources, the LME application uses 0.3% of the CPU and 0.1% of the memory of the PC, while the DME application uses 0.8% of the CPU and 0.1% of the memory.

VI. CONCLUSIONS AND FUTURE WORK

Summarizing the presented routing management system, we can tell that this is a self-managing system. This means that after setting the management policies in the configuration files and starting the applications on every node, no further human interaction is needed.

The proposed routing management solution takes into account statistical cross-layer QoS information regarding the state of the network. These are distributed throughout the network and are used by the routing protocol as routing metrics. Thus, the global capacity of the network can be improved by avoiding congested areas. The idea was to separate the monitoring functions from the actual routing, making the statistical network status information reusable. The routing protocols can focus on path determination and updating the routing tables.

This is a preliminary implementation, covering only a part of the possible functionalities of a complete routing management system and offering a proof-of-concept. As future work, we intend to evaluate the effect of prediction in terms of the compromise between the utilized resources and the improved global capacity of the network. We envisage reducing the packet losses through congestion prediction as opposed to congestion detection used in the present implementation.

ACKNOWLEDGMENT

This work was partially funded by the PRODOC program ”Project of Doctoral Studies Development in Advanced Technologies” (POSDRU/6/1.5/S/5 ID 7676).

REFERENCES

[1] Sung-Su Kim, Y. J. Won, , Mi-Jung Choi, J. W.-K. Hong, J. Strassner, *Towards management of the Future Internet*, 11th IFIP/IEEE International Symposium on Integrated Network Management. Long Island, USA, 2009.

[2] R. Roth, F. Wolff, T. Zseby (eds.), *D-4.1 Definition of scenarios and use cases.* FP7-ICT-2007-1-216041-4WARD – Architecture and Design for the Future Internet, 2009.

[3] D. Medhi, *Routing Management in the PSTN and the Internet: A Historical Perspective*, Journal of Network and Systems Management, Volume 15, pp. 503–523. Springer, 2007.

[4] M. S. H. Lipu, M. N. -U. Haque, T. F. Karim, M. L. Rahman, M. H. Morshed, , *Quality of service model selection criteria and traffic controlling algorithm of a native IP Network*, International Conference on Educational and Information Technology ICEIT 2010, pp. V3-144 – V3-148. Chongqing, China, 2010.

[5] E. Marin-Tordera, X. Masip-Bruin, S. Sanchez-Lopez, J. Domingo-Pascual, A. Orda, *The Prediction Approach in QoS Routing*, IEEE International Conference on Communications ICC 2006, pp. 1020–1025. Istanbul, Turkey, 2006.

[6] A. B. Rus, M. Barabas, G. Boanea, Zs. Kiss, Zs. Polgar, V. Dobrota, *Cross-Layer QoS and Its Application in Congestion Control*, 17th IEEE Workshop on Local and Metropolitan Area Networks LANMAN 2010. Long Branch, USA, 2010.

[7] L. M. Correia, H. Abramowicz, M. Johnsson, K. Wuenstel (editors), *CLQ-Based Testbed used for Generic Path*, Chapter 12 ”Prototype Implementations”, pp. 271-276, Architecture and Design for the Future Internet. 4WARD Project. Series: Signals and Communication Technology. 1st Edition, Springer Science + Business Media, 2011.

[8] *Cisco Visual Networking Index: Forecast and Methodology, 2009–2014*, White Paper, 2010.