Melinda BARABAS, Georgeta BOANEA, Andrei Bogdan RUS, Virgil DOBROTA

Technical University of Cluj-Napoca

Congestion Control Based on Distributed Statistical QoS-Aware Routing Management

***Abstract**. In this paper a distributed routing management solution is described that takes into consideration statistical Quality of Service (QoS) information about the state of network links. The goal is to offer dynamic metrics to the routing algorithm based on the current resource utilization, i.e. the routing metrics will depend on the traffic load and the delay measured on individual links in real-time. Identification of congested areas allows situation-aware path selection, enhancing the global capacity of the network by determining optimal routes.*

***Streszczenie.** W artykule przedstawiono sposób zarządzania routowaniem, w którym wzięto pod uwagę parameter QoS na temat stanu linków w sieci. Głównym celem było zastosowanie dynamicznej metryki do algorytmu routowania, opartej na informacji o bieżącym wykorzystaniu zasobów. Identyfikacja rejonów przeciążenia pozwala na polepszenie działania i pojemności sieci poprzez optymalizację ścieżki routowania. (**Kontrola przeciążenia w routowaniu QoS**).*

Keywords: Congestion control, QoS-aware routing, self-management, traffic-aware shortest paths.
Słowa kluczowe: kontrola przeciążenia, routowanie QoS, samo zarządzanie, najoptymalniejsza ścieżka dostępu.

Introduction

Initially, it was believed that routing management is not necessary because of the adaptive routing protocols, such as OSPF (Open Shortest Path First), which can self-manage and are able to react to failures and changes in the network. However, efficient traffic engineering is not possible if the routing process does not take into account real-time traffic conditions.

Designing a routing strategy is difficult because it depends on a number of variables and parameters that are sometimes uncertain. The complexity is also increased by the diversity and the highly dynamic nature of services that have to be provided by IP networks. In addition, the routing policy must be adaptive in order to cope with topological changes and traffic condition fluctuations.

According to [1], routing management comprises different functionalities necessary for the efficient operation of the routing process. Its main objective is to increase the availability of the network, while ensuring the performance requirements of current and future connections. Specific issues of interest are: detecting congestion on different links, monitoring packet losses, load balancing, configuring the parameters of the routing algorithm, determining dynamic routing metrics based on existing traffic on each link, etc.

The centralized management paradigm employed in today's computer networks is not suitable for large-scale network topologies and for services provided in the future Internet. The main problem of legacy technologies lies in the incremental adding of new features, leading to an increased complexity of the system [2]. Thus, new approaches are needed that rely on management functionalities integrated into the network architecture in the design phase. Reference [3] summarizes the state of worldwide research in the domain of network management and proposes design requirements for future solutions. New trends focus on self-managing, scalable, intelligent networks which do not require manual interventions by an administrator, as presented in [4].

In this paper, an autonomous system is proposed that enables routing management based on network status awareness. Thereby, statistical QoS information is made available to the routing algorithm, helping it to avoid congested links and to determine optimal routes. The resource monitoring functions necessary for the QoS-aware routing process are not concentrated in a central device, but distributed among the network elements.

The remainder of the paper is organized as follows. In Section 2 existing traffic control techniques are presented and a congestion control scheme based on QoS-aware routing management is proposed. Section 3 presents the designed local management. 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.

Congestion Control through QoS-Aware Routing

QoS-aware routing has become a great challenge in computer networks because the Best Effort (BE) delivery provided by legacy IP networks cannot guaranty the Quality of Services even if the load on links is less than 50% [5]. Thereby, the BE model is suitable only for services insensitive to delay and packet drops. It is inadequate for real-time and multimedia applications where packet retransmission is not a viable option, for example in the case of VoIP (Voice over IP), video- and teleconference, VoD (Video on Demand), IPTV (Internet Protocol Television), etc. Cisco forecasts that the sum of all forms of video transmissions will reach 90% of global consumer traffic by 2015 [6]. Thus, assuring traffic delivery in terms of bandwidth, delay and packet loss becomes more important.

At the moment, traffic control can be achieved either through *preventive* or through *reactive* functions. The preventive approach relies on resource reservation before transmission, thus avoiding the likelihood of congestion. An example would be the IntServ (Integrated Services) QoS assurance mechanism [5]. The disadvantage of this approach is that reserving resources in advance can lead to inefficient use of link capacity. On the other hand, reactive congestion control mechanisms are based either on queue management algorithms (dropping packets randomly or based on priority schemes) or on TCP (Transmission Control Protocol) congestion avoidance algorithms. From the end-user perspective, none of the solutions is optimal because they lead to packet losses or reduce the transfer rate, affecting especially the Quality of Experience (QoE) of multimedia streams.

Legacy routing protocols implement a simple form of link quality assessment by periodically sending Hello messages. But this method offers only limited capabilities in terms of identifying congested links. For example, in the case of OSPF, 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.

The authors of [7] show that congestion control schemes based on load factor can improve network performance in terms of utilization, packet loss rate and delay. Accurate representation of network load levels can lead to a more efficient indication of congestions information to neighbouring nodes.

Starting from the limitations of link state routing protocols and the drawbacks of congestion control mechanisms, a distributed routing management system is proposed which relies on statistical QoS-aware information regarding the utilization of link resources. The idea is to handle the congestion at the network layer in a simple and efficient manner. Instead of eliminating low priority packets or reducing the transmission rate, we choose to reroute streams affected by congestion.

As it is demonstrated in this paper, the proposed approach leads to a greater end-user satisfaction because it can improve the quality of transmissions affected by congestion. However, it does not fully eliminate the possibility of short-term packet drops.

Statistical QoS-Aware Routing Management System

The main objective of the self-managing system described herein is to integrate the monitoring process as an intrinsic part of the network architecture. The routing management system is characterized by the following features: *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 idea is to separate the link monitoring and the update of network state information from the routing itself, as illustrated in Fig. 1. This requires a clean-slate approach, rethinking the entire routing process. The routing management system takes over some functionalities traditionally belonging to the routing protocol, such as: communication with neighbouring nodes, topology discovery, connectivity verification, link monitoring, etc. In this way, the routing algorithm can focus on the determining paths and maintaining the routing tables.

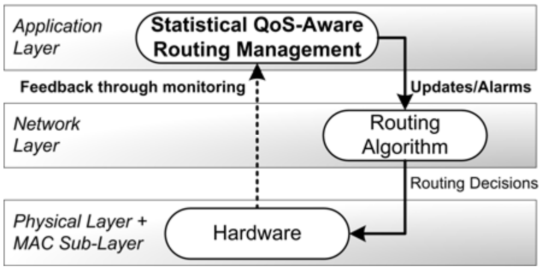


Fig.1. Separating the link monitoring from the routing process (new functionalities represented with bold)

The information gathered by the routing management system describing the current state of the network can be reused for other purposes. Another advantage of this approach is that the routing management will not depend on a particular routing algorithm. The management system provides information about the network topology and the available link resources to the routing algorithm, without imposing the way in which routing metrics should be used or paths should be determined. The feedback regarding the routing decisions is obtained through hardware monitoring. By this, the level of automation, the flexibility and the manageability of the system can be increased significantly.

Two types of QoS parameters are used for congestion detection:

- Available Transfer Rate (ATR), obtained by passive monitoring;
- One Way Delay (OWD) measured through active monitoring (i.e. by injecting probes into the routing queues). It consists of processing-, queuing-, transmission- and propagation delay.

These QoS parameters are measured in real-time at the Physical Layer and MAC Sub-Layer, for each direction of each link, as described in [8] and [9].

Starting from the real-time measurements, the following statistical indicators are computed:

- Simple Moving Average (SMA): calculated every second for an interval of $N = 10$ seconds in order to estimate the value of the monitored parameter:

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

where: t – current time, x – value of the parameter.

- Standard Deviation (SD) σ: used in order to identify events/changes in the traffic conditions:

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

Statistical indicators are needed to eliminate the fluctuations of the monitored values. For example, if the measurements oscillate around a fixed threshold used to detect congestion, this leads to false positive and negative alarms. Thus, if we want to catch the true evolution of the available transfer rate around the threshold, we must look at longer intervals, although the monitoring takes place every second. The statistical indicators calculated for the QoS parameters represent the link state information used for QoS-aware routing. Based in this, traffic-aware shortest paths can be computed for optimal network performance.

The routing management system is composed of management entities running on each node and collecting measurement data through hardware monitoring. The goal is to determine the traffic load and the delay on each unidirectional 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 administers the routing table of that node. Interaction between neighbouring 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 of dynamically adapting to external events, minimizing the need for human intervention.

Designing the Management Entities

The proposed system consists of so called Local Management Entities (LME). In this section the functionalities and the operation mode of the implemented entities are presented, as well as the interaction between them. The 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.

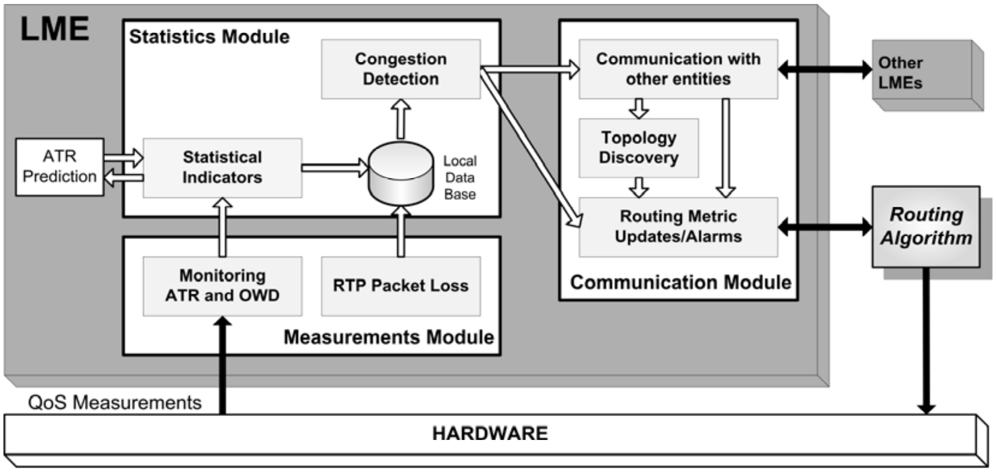


Fig.2. Internal structure of a Local Management Entity

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 neighbouring nodes.

The main functionalities of such an entity are the following:

- Starting the QoS measurement applications for monitoring ATR and OWD on the directly connected inbound links. Resource monitoring is done periodically, second by second.
- Using the statistical indicators SMA and SD 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.
- Extracting the current status of the local network from the collected QoS information. The local network refers to the directly connected incoming links.
- Communicating with other entities over UDP connections by sending XML messages.

Fig. 2 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. There also exists the possibility to monitor RTP packet losses. The *statistics module* calculates the SMA and the standard variation of the monitored network parameters and identifies changes in the link status and detects congestions. 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, discovering the topology and interacting with the routing algorithm.

LME starts monitoring the local links as it detects the corresponding neighbouring 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 other LMEs are announced that an event occurred.

Because $N = 10$ in equation (1), congestion can be detected after at least 10 seconds. This value was chosen as a compromise between the possibility of generating false alarms and the reaction time of the system.

A detailed description of the routing management system can be found in [10].

B. Interaction between Entities

As mentioned before, management entities interact by sending XML messages. In Fig. 3 we can see the communication between local management entities. This represents a very simple scenario with the purpose of illustrating the message-passing between LMEs.

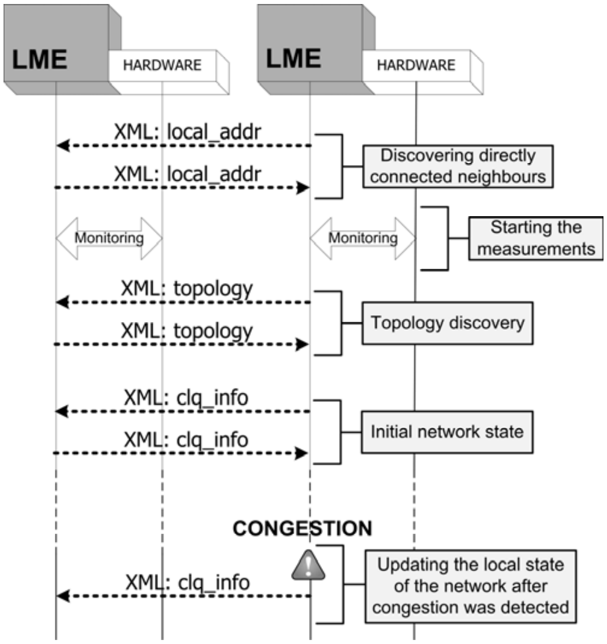
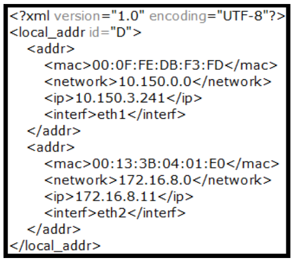


Fig.3. Interaction between LMEs

We can identify the following steps in the presented message flow:

1) *Detecting directly connected neighbouring nodes*: The LMEs periodically advertise their local addresses in order to detect directly connected nodes and to start monitoring the QoS parameters on the corresponding incoming links. Fig. 4 illustrates the structure of a message (*local_addr*) used for identifying neighbours. This XML message is only sent to directly connected routers and it contains: the MAC address, the network address, the IP address and the name for all local interfaces. The id attribute specifies the local identifier of the node.



```
<?xml version="1.0" encoding="UTF-8"?>
<local_addr id="D">
  <addr>
    <mac>00:0F:FE:DB:F3:FD</mac>
    <network>10.150.0.0</network>
    <ip>10.150.3.241</ip>
    <interf>eth1</interf>
  </addr>
  <addr>
    <mac>00:13:3B:04:01:E0</mac>
    <network>172.16.8.0</network>
    <ip>172.16.8.11</ip>
    <interf>eth2</interf>
  </addr>
</local_addr>
```

Fig.4. XML message: *local_addr* – used for identifying neighbouring nodes

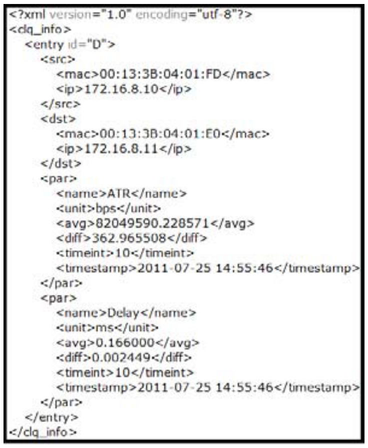
2) *Starting the cross-layer QoS measurements*: After detecting the direct neighbours, the measurement applications are started. Each LME will monitor the ATR and the OWD on incoming links. Passive monitoring is used for ATR, and active monitoring for.

3) *Discovering the topology*: The structure of XML messages used for topology detection is presented in Fig. 5. Each LME sends such a message to every neighbour when it identifies a new connection or a new network node. In order to distribute topology information through the network, a controlled flooding approach is used.

```
<?xml version="1.0" encoding="UTF-8"?>
<topology>
  <con>
    <id1>R1</id1>
    <id2>R2</id2>
    <ip1>172.16.1.11</ip1>
    <net1>172.16.1.0</net1>
    <interf1>eth0</interf1>
    <mp1>0</mp1>
    <ip2>172.16.1.10</ip2>
    <net2>172.16.1.0</net2>
    <interf2>eth2</interf2>
    <mp2>0</mp2>
  </con>
  <con>
    <id1>R1</id1>
    <id2>R3</id2>
    <ip1>172.16.2.11</ip1>
    <net1>172.16.2.0</net1>
    <interf1>eth1</interf1>
    <mp1>0</mp1>
    <ip2>172.16.2.10</ip2>
    <net2>172.16.2.0</net2>
    <interf2>eth0</interf2>
    <mp2>0</mp2>
  </con>
</topology>
```

Fig.5. XML message: *topology* – used for topology discovery

4) *Determining the initial network state*: LMEs will send an XML message with initial link status information to each other, shown in Fig. 6.



```
<?xml version="1.0" encoding="utf-8"?>
<clq_info>
  <entry id="D">
    <src>
      <mac>00:13:3B:04:01:FD</mac>
      <ip>172.16.8.10</ip>
    </src>
    <dst>
      <mac>00:13:3B:04:01:E0</mac>
      <ip>172.16.8.11</ip>
    </dst>
    <par>
      <name>ATR</name>
      <unit>bps</unit>
      <avg>82049590.228571</avg>
      <diff>362.965508</diff>
      <timeint>10</timeint>
      <timestamp>2011-07-25 14:55:46</timestamp>
    </par>
    <par>
      <name>Delay</name>
      <unit>ms</unit>
      <avg>0.166000</avg>
      <diff>0.002449</diff>
      <timeint>10</timeint>
      <timestamp>2011-07-25 14:55:46</timestamp>
    </par>
  </entry>
</clq_info>
```

Fig.6. XML message: *clq_info* containing statistical QoS information

In this way, the routers can have an initial view of the status of the network and the routing algorithm will be provided with initial values of the dynamic routing metrics.

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 other entities by sending *clq_info* messages through controlled flooding.

The routing algorithm running on each node will update its routing table according to actual traffic conditions, 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, frequent oscillations of the routing process can be avoided, 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 transferred to a different path. For this, traffic flows affected by packet losses and delays have to be identified. The network state is re-evaluated and if the congestion still persists, other streams will be also re-routed. This is possible because a multipath routing solution is used. This mechanism allows for balancing the traffic load sent between a source–destination pair on different links.

Description of the Testbed

For performance evaluation a simple network topology was chosen for reasons of practical implementation on physical machines. The testbed contains six software routers (R1, R2, R3, R4, R5 and R6), a source and a destination node, each running on Linux-based machines with Fedora operating system. On each router we started a QoS-aware multipath routing application and the LME application. In this way it is possible to monitor the status of all unidirectional links between neighbouring routers. The topology and the status information is distributed across the network by LMEs in order to be used in the adaptation of the routing process and to improve the overall performance.

To demonstrate the beneficial effects of statistical QoS-aware routing management, we chose to transmit a video stream from the source node to the destination, assuming that one link on the used path becomes overloaded. Providing good video quality is a major problem since video traffic is both massive and intolerant to packet loss or latency. For the test scenario, we assume that a Variable Bit Rate (VBR) MPEG-4 video flow is sent over RTP/UDP/IP. The 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 node.

The goal of the tests is to demonstrate the capability of the routing management system to reduce the negative effects of congested links by providing statistical QoS information regarding link status to the routing application.

The performance evaluation is based on the parameters of the received video stream. We take into consideration the following Video Quality (VQ) metrics:

- *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.
- *Magnitude of loss events*: expresses the number of packets that were dropped at each loss event (a magnitude of 0 means the packet arrived successfully at the destination).

- *Discontinuity counters:* measures discontinuity events, characterizing the frequency with which discontinuities were detected, i.e. the number of times loss was detected.

Experimental Results

The test scenario involves the introduction of congestion through background traffic between R5 and R4 one minute after the experiment starts. In order to do this, the *ethtool* was used to limit the nominal transfer rate of the link to 100 Mbps (the actually observed transfer rate at the network layer is less than this theoretical maximum). After this, background traffic is generated between R5 and R4 using the *iperf* network testing tool: 10 UDP streams were started, with the total transfer rate of 100 Mbps. As a result, ATR drops below the required rate to transmit the stream and the OWD from node R5 to R4 increases.

Two experiments were carried out in order to compare the behaviour of the following routing approaches in case of congestion:

- Case *1:* OSPF (Open Shortest Path First) routing protocol;
- Case *2:* Statistical QoS-aware distributed routing management system.

If there is no congested link in the network, the same path is used in both cases for routing the packets (source–R1–R5–R4–destination) and no loss events occur during the streaming

Case 1:

To assess the OSPF protocol on Linux machines the Quagga Routing Software Suite is utilized [11], enabling on each router the *ospfd* and *zebra* daemons after being set up properly. Fig. 7 shows the routing of the video stream when using the OSPF protocol.

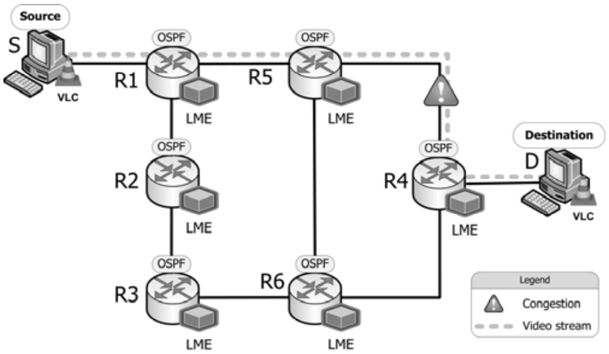


Fig.7. Case 1 – Testbed

After congestion is introduced on link R5–R4, OSPF does not modify the path followed by the video flow since it does not take into consideration the state of the links and the current traffic conditions in the network. Because at least 1 out of 4 Hello messages sent by R5 is received by the OSPF module on node R4, the existing problems on the link are not identified. As an effect, packet losses can be observed at the destination node, the Quality of Experience being very poor. In this case 6461 packets were missing at the destination out of 21980 transmitted packets, i.e. 29.39% of the transmitted packets were dropped during the experiment.

In Fig. 8 the magnitude of loss events is presented. It can be seen that losses occur continuously after the introduction of congestion.

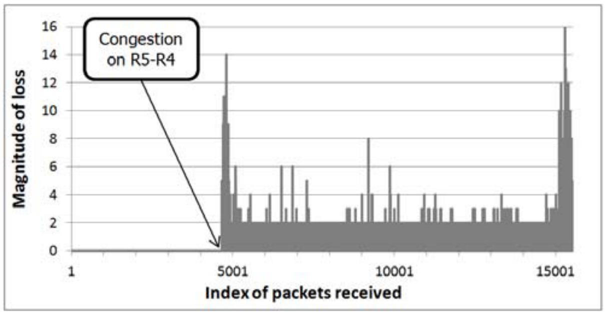


Fig.8.Case 1 – Magnitude of loss events

Case 2:

In this case, the congestion is detected by the local management entity located on router R4 when the SMA calculated for the ATR parameter drops below 5 Mbps (corresponding to 5% of the maximum nominal rate 100 Mbps of the link). The QoS-aware routing solution is notified by the routing management application about the problem. It will determine a new route from source to destination, avoiding the congested area based on the updated routing metrics. In this scenario, presented in Fig. 9, 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.

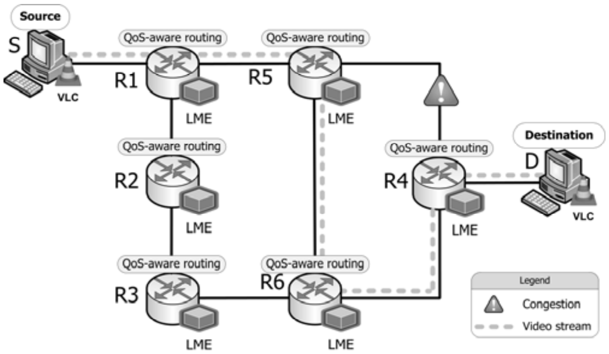


Fig.9. Case 2 – Testbed

Fig. 10 illustrates the magnitude of loss events in Case 2. It can be observed that losses appear just for a short period of time, until the stream is rerouted. From the end-user's perspective this means that the video image freezes for a few seconds, after which no further disruptions are observed.

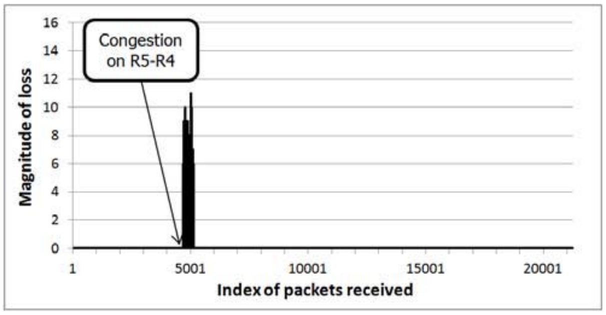


Fig.10.Case 2 – Magnitude of loss events

Table 1 depicts different parameters of the streaming performance in the two cases analyzed earlier.

Table 1. 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

In Fig. 11 the evolution of the percentage of lost packets is shown over the experiment duration. This is determined at a given moment by the ratio of the number of packets missing and the total number of packets sent by the source. In Case 1 (when using OSPF) the loss percentage is increasing steadily upon the appearance of congestion, reaching a maximum value of 29.39% at the end of the experiment. In Case 2 the loss percentage increases for a short time to 12.6%, but after that it recovers, dropping to the final value of 3.37%.

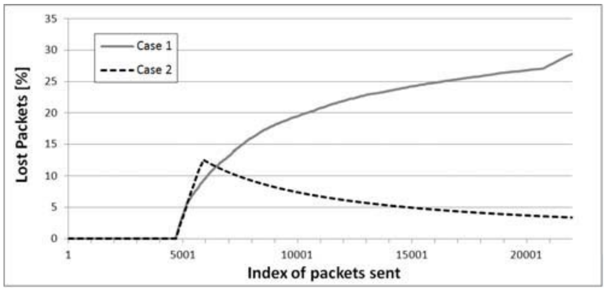


Fig.11.Percentage of lost packets over experiment duration

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 without applying a QoS assurance mechanism. 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.

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 ATR 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.

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] Medhi D., Routing Management in the PSTN and the Internet: A Historical Perspective, *Journal of Network and Systems Management*, 15 (2007), 503-523

[2] Dudkowski D., Brunner M., Nunzi G., Mingardi C., Foley C., de Leon M.P., Meirosu C., Engberg S., Architectural principles and elements of in-network management, *IEEE Int. Symp. on Integrated Network Management*, Long Island, USA (2009), 529-536

[3] Kim S.-S., Choi M.-J., Ju H.-T., Ejiri M., Hong J.W.-K., Towards Management Requirements of Future Internet, *Lecture Notes in Computer Science*, 5297 (2008), 156-166

[4] Roth R, Wolff F., Zseby T. (eds.), D-4.1 Definition of scenarios and use cases, *FP7-ICT-2007-1-216041-4WARD – "Architecture and Design for the Future Internet"*, Revision 2.0 (2009)

[5] Lipu M.S.H., Haque M.N.-U., Karim T.F., Rahman M.L., Morshed M.H., Quality of Service Model Selection Criteria and Traffic Controlling Algorithm of a Native IP Network for Telecommunication Services, *Int. Conf. on Educational and Information Technology*, Chongqing, China (2010), V3-144–V3-148

[6] Cisco Visual Networking Index: Forecast and Methodology, 2010-2015, White Paper (2011)

[7] Qazi I. A., Znati T., On the Design of Load Factor Based Congestion Control Protocols for Next-Generation Networks, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 55 (2011), No. 1, 45-60

[8] Rus A.B., Barabas M., Boanea G., Kiss Z., Polgar Z., Dobrota V., Cross-Layer QoS and Its Application in Congestion Control, *IEEE LANMAN – Workshop on Local and Metropolitan Area Networks*, Long Branch, USA (2010), 1-6

[9] Correia L.M., Abramowicz H., Johnsson M., Wuenstel K. (eds.), CLQ-Based Testbed used for Generic Path, Chapter 12 "Prototype Implementations", *Architecture and Design for the Future Internet. 4WARD Project*, 1st Edition, Springer Science + Business Media (2011), 271-276

[10] Barabas M., Routing Management in the Future Internet, *PhD Thesis* (2011), Technical University of Cluj-Napoca

[11] Quagga Routing Software Suite, http://www.quagga.net/

***Authors**: MSc. Melinda Barabas, Ph.D., MSc. Georgeta Boanea, Ph.D., MSc. Andrei Bogdan Rus, Ph.D., Prof. MSc. Virgil Dobrota, Ph.D., Technical University of Cluj-Napoca, 26-28 George Baritiu Street, Room 364, 400027 Cluj-Napoca, Romania, E-mail: Melinda.Barabas@com.utcluj.ro, Georgeta.Boanea@com.utcluj.ro, Bogdan.Rus@com.utcluj.ro, Virgil.Dobrota@com.utcluj.ro.*