Multipath Routing Management using Neural Networks-Based Traffic Prediction

Melinda Barabas, Georgeta Boanea, Virgil Dobrota
Communications Department
Technical University of Cluj-Napoca
28 Memorandumului Street, 400114 Cluj-Napoca, Romania
Email: {Melinda.Barabas, Georgeta.Boanea, Virgil.Dobrota}@com.utcluj.ro

***Abstract*—Embedding forecasting algorithms into routing management systems can play an important role in guaranteeing QoS in IP networks. In this paper, we propose an intelligent routing framework, consisting of a situation aware multipath routing algorithm and a routing management system involving neural networks-based predictors with multi-task learning. The solution is characterized by QoS-awareness, load balancing and self-management. The main goal is to offer a proof-of-concept by practical implementation of predictive QoS-aware multipath routing in a real test environment. The proposed solution is compared with the OSPF and ECMP routing protocols in case of congested network links. The experimental results show that traffic prediction enables proactive routing management and improves the global performance of the network through congestion control and avoidance.**

Keywords–cross-layer QoS; multipath routing; neural networks; self-management; traffic prediction

I. Introduction

Ensuring Quality of Service (QoS) over the Internet is difficult, especially in the case of real-time multimedia applications where the retransmission of packets is not a viable option. The occurrence of congestion can severely degrade the quality of transmissions due to packet losses, increased delay and jitter [1]. Embedding forecasting algorithms into routing management systems can play an important role in guaranteeing QoS in IP networks. Traffic prediction enables proactive network management which improves the global performance of the network through congestion control and prevention.

Initially, it was believed that adaptive routing protocols, such as OSPF (Open Shortest Path First) [2], can react to congestion. Unfortunately, congested links often remain undetected because of the way OSPF assesses link connectivity. 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 problem. If the congestion is severe and no Hello messages are received from a neighbor, it is automatically considered *down* because OSPF makes no distinction between hardware failures and congestion. Thus, the involved router will not be further used and all the traffic will be rerouted to a different link which in turn can also become congested. The solution adopted by OSPF does not resolve the underlying problem, that of transmitting too much traffic on a single link.

In the present Internet, congestion control mechanisms rely on queue management algorithms (dropping packets randomly or based on their priority) or TCP (Transmission Control Protocol) congestion avoidance (reducing the sending rate). From the end-user perspective, these solutions are not optimal because they mean lost packets or a reduced bitrate, both affecting the quality of transmission, especially the QoE (Quality of Experience) of multimedia content.

The main motivation for this paper is to resolve the above mentioned limitations of legacy routing protocols and congestion control mechanisms by applying the multipath routing paradigm. We focus on the problems caused by congested network links and our goal is to improve the overall network performance by load balancing and prediction of network traffic. We envision an intelligent routing framework, consisting of the SAMP (Situation Aware Multipath) routing algorithm [3] and a routing management system.

The routing strategy presented herein is characterized by self-management and QoS-awareness, achieved via monitoring link resources through cross-layering techniques. QoS-aware routing means that not only shortest paths, but traffic-aware shortest paths are computed for optimal network performance. The traffic predictors integrated into the routing management system enable proactive decision-making, as opposed to reacting to past events. Employing a prediction-based approach helps to match network resources to the traffic demand [4]. Thanks to the early warning, a prediction-based approach will be faster, in terms of congestion detection and elimination, than reactive methods which detect congestion only after it significantly influenced the operation of the network, as demonstrated in [5].

The rest of this paper is organized as follows. Section II briefly presents previous work regarding prediction used in combination with routing systems. In Section III, neural network traffic predictors with multi-task learning approaches are described. Section IV presents the multipath routing framework. In Section V, the practical testbed is described, followed by the experimental results in Section VI. Section VII concludes the paper and discusses future work.

II. Related Work

In the literature, several works address the topic of network parameter prediction techniques integrated into single-path [6],

 ISBN: 978-1-61208-174-8

[7], [8] or multipath routing solutions [5], [9], [10].

The authors of [6] propose the PBS (Prediction Based Routing) heuristic mechanism that predicts the availability of links/routes and selects routes without taking into account network state information. In [7], a neural networks-based queuing delay prediction mechanism is integrated with a MANET proactive routing protocol OLSR (Optimized Link State Routing), increasing the packet delivery ratio and reducing the end-to-end delay. Masip-Bruin *et al.* [8] designed a routing technique based on CBR (Constraint-Based Routing) that combines the strength of prediction with an innovative link-state cost. CBR is applied in circuit-switched networks and it reduces the impact of routing inaccuracy on the blocking probability.

A data forwarding algorithm over multipath is described in [5]. The proposed solution is based on linear prediction and particle swarm optimization and it improves the QoS of real-time applications. Li *et al.* [9] proposed a Multipath Routing Algorithm based on Traffic Prediction (MRATP) to be used in Wireless Mesh Networks (WMN) in order to guarantee end-to-end QoS. A method for multipath selection based on prediction in wireless networks is introduced in [10] where neural networks are used to infer the types of the links and the paths are chosen based on predicted incremental throughput.

In the literature, a predictive approach is taken into consideration either for single-path routing approaches or for multipath routing over wireless networks. Based on this observation, we chose to integrate a network parameter prediction algorithm into a multipath routing solution over wired networks. In this way, the routing metrics will depend on predicted traffic conditions. Thereby, we intend to identify congestion in the network faster than through simple monitoring. This is achieved by predicting the available transfer rate on unidirectional network links, as opposed to other solutions which predict: *a)* the rate of packet losses [5], *b)* the delay in routing queues [7], *c)* the type of wireless links and the incremental throughput [10] or *d)* the bitrate of video flows [11], etc. Reaction to congestion is manifested by rerouting traffic, unlike alternatives such as: *a)* reduced video bitrate [1], *b)* advanced allocation of transfer rate for future transmissions [4], [8], [11], *c)* controlled dropping of packets [12], etc.

III. Neural Networks-based Prediction

The prediction of network traffic is possible because it presents a strong correlation between chronologically ordered values. The most widely used traffic forecasting methods involve Neural Networks (NN) [13], [14], [15], etc. NNs are employed for modeling and predicting traffic because of their strong self-learning and self-adaptive capabilities through which they are able to learn complex patterns. NNs are characterized by nonlinear mapping and generalization ability, robustness, fault tolerance, parallel processing, etc.

A NN consists of several layers of interconnected nodes (neurons): *a)* an input layer, *b)* one or more hidden layers and *c)* an output layer. The most popular NN architecture is feed-forward in which the information travels through the network in the forward direction: from the input layer towards the output layer. The NN model represents a nonparametric and adaptive modeling approach, the architecture and the parameters being determined solely by the observed data.

Using a NN as a predictor involves two phases: *a)* the training phase and *b)* the prediction phase. In the training phase, the training set is presented at the input layer and the parameters of the NN are dynamically adjusted to achieve the desired output. The prediction phase represents the testing of the NN. A new input (not included in the training set) is presented to the NN and the output, which represents the predicted value, is calculated.

Usually, NN predictors have a single output node and they focus on a single main task, i.e., predicting x_{t+1} based on $\{x_1, x_2, \ldots, x_t\}$. Thereby, the predictor neglects information hidden in other tasks (e.g. the relationship between the historical data and x_{t+2}, although both tasks belong to the same dataset). The Multi-Task Learning (MTL) paradigm is introduced to improve the generalization performance of NNs. A main task is trained simultaneously with extra tasks, sharing the hidden layer of the NN, as shown in Figure 1. By learning multiple tasks simultaneously, the NN can achieve better prediction accuracy. For time series forecasting through the MTL concept, usually, two extra tasks are chosen, namely the prediction of x_t and x_{t+2}, which are closely related to the main task x_{t+1}, as in [16] and [17].

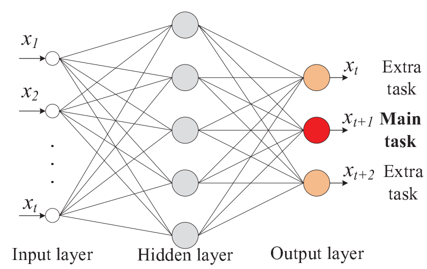

Fig. 1. NN predictor with multi-task learning

In our experiments, we selected a NN with only one hidden layer because more layers would make the network more time- and resource-consuming, but added complexity would not be justified by the improvement of the prediction accuracy.

IV. Multipath Routing Framework

The main idea of the proposed multipath routing framework is to separate the monitoring from the routing process itself: the link monitoring and the communication between neighboring nodes is realized by the routing management system, while the multipath routing algorithm deals with the routing decisions and the packet forwarding. Thus, the information regarding the state of the network becomes reusable.

A. Multipath Routing Algorithm

The multipath routing algorithm used in this paper is called SAMP (Situation Aware Multipath). Practical implementation of SAMP and simulation results are presented in detail in our previous work [18]. To ensure efficient and high quality transmissions, SAMP relies on the information provided by the routing management system regarding the network status. The key features of this solution are load balancing and congestion avoidance by fast rerouting.

1) Load balancing

To overcome the problem of inefficient link resource utilization, load balancing is employed. In order to divide the traffic among multiple routes, a split granularity at flow level is used, avoiding the problem of out-of-order packet arrivals. A flow is identified by the triplet: source IP address, destination IP address and destination port. Because SAMP takes into account the physical state of the network, the flows will be routed along paths that ensure the application's requirements in terms of transfer rate and delay.

With the purpose of providing scalability and decreasing the complexity of the solution, the network is divided into multiple routing domains, each consisting of two types of routers:

- *AR (Adaptive Router)*: located inside the domain and performing situation aware routing (reacting in case of congestion);
- *AMR (Adaptive Multipath Router)*: located at the edge of the domain. Besides the situation aware routing features, it also achieves load balancing for the traffic coming from outside the domain.

The traffic is divided into elastic end inelastic flows [20]. The elastic flows are handled by the main routing table because they are not sensitive to delay- and throughput variations. The inelastic flows (e.g. video, VoIP, etc.) are identified and transmitted on multiple paths. This forwarding method is carried out using the VRF (Virtual Routing Forwarding) concept: depending on the path that is allocated to a flow, the corresponding routing table is used.

The AMR dictates how a flow is routed in a domain. This is possible because all the nodes have a global view of the network, possessing the necessary information concerning the behavior of all other routers in any situation. The proposed solution does not impose any restrictions regarding the number of multipath domains/nodes, but the complexity increases along with the number of domains.

2) Congestion avoidance by fast rerouting

In case of congestion on one of the links, flows transmitted along the affected link are gradually rerouted, one by one, until the congestion disappears. The new selected path for a flow will be the one that offers the highest available transfer rate and has the lowest delay. Only multimedia flows are rerouted, the rest of the traffic being considered background traffic. Because all paths in the network are precomputed, the algorithm does not dependent on the number of congested/failed links.

B. Routing Management System

The Routing Management System (RMS) is a highly distributed self-managing system, which is capable to dynamically adapt to external events, minimizing the need for human intervention. It consists of *Local Management Entities* (LME) located on every node of the network (Figure 2). LMEs located on different nodes communicate through XML messages, discovering the network topology and exchanging network status information that are stored in local databases.

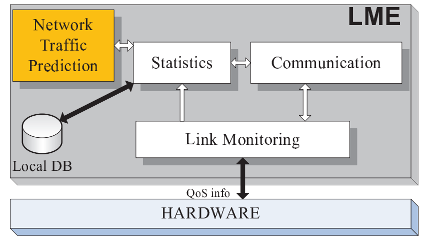

Fig. 2. Local Management Entity

LME performs real-time monitoring of the inbound links, measuring the Available Transfer Rate (ATR) and the One Way Delay (OWD) at the data link layer, as well as the missing packets at the application layer. Dropped packets can be considered an early indicator for congested links and overloaded routers. These are monitored only for multimedia streams, whose quality is the most likely to be influenced by congestion. Thereby, the system employs a cross-layer optimization strategy by making decisions at the network layer based on information derived from lower- and upper layers.

A previous version of the proposed routing management system is described in [21]. There are three main differences from the previous implementation. First of all, the RMS is a highly distributed system, as opposed to the previous solution where the congestion detection mechanism and the network status updates followed a centralized approach. Another difference lies in the monitoring of dropped packets. This enables the identification of the most affected multimedia streams which will have priority in the rerouting process. The third and most significant difference is the integration of a network traffic prediction module into LMEs. This represents a key component for the adaptive congestion control scheme. It forecasts the values of the ATR for inbound links for the next time interval (1 second). LME can detect congestion on its monitored links and it broadcasts this information through the network. It indicates to the routing algorithm when to update the routing tables. The system being highly distributed, routing decisions are not taken synchronously on every node.

V. Performance Evaluation

The practical testbed illustrated in Figure 3 is used to evaluate the performance of the proposed solution. This network offers sufficient paths between the source and destination nodes in order to employ multipath routing, but it is simple enough to allow practical implementation. The testbed consists of: *a)* six routers (`R1`, `R2`, `R3`, `R4`, `R5`, and `R6`), *b)* a source node (`S`) and *c)* two destination nodes (`D1` and `D2`). All nodes in the network are Linux-based computers with Fedora operating system. On each machine, several software applications written in C++ are running:

- multipath routing application (SAMP);
- Local Management Entity (LME);

 ISBN: 978-1-61208-174-8

- NNs based traffic predictors.

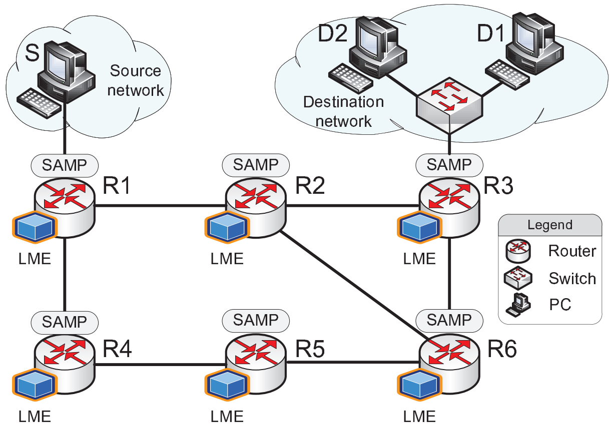

Fig. 3. Practical testbed

Providing good video quality is a major problem in congested networks since video traffic is both massive and intolerant to packet loss or latency. We demonstrate the improvements brought by the described predictive QoS-aware multipath routing framework by sending video streams from the source node `S` to the destinations `D1` and `D2` when two of the network links are affected by congestion. During the experiments, three different MPEG-4 video flows, each having an average bitrate of 1 Mbps, are sent by a VLC client over RTP/UDP: *1) Stream #1* from `S` to `D1`, *2) Stream #2* from `S` to `D1`, and *3) Stream #3* from `S` to `D2`.

The test scenario has a duration of 5 minutes. We generate background traffic in the network using the `iperf` network testing tool. Congestion is introduced on links `R2–R3` and `R2–R6` after 1 minute and after 2 minutes, respectively. As a result, packet losses can appear because the ATR drops below the required rate to transmit the streams and the OWD increases.

Experiments are performed in order to compare the following intra-domain routing approaches: *Case 1)* **OSPF** (Open Shortest Path First) – the most widely used routing protocol in large networks; *Case 2)* **ECMP** (Equal-Cost Multi-Path) [22] – the only multipath solution supported by current IP routers, and *Case 3)* **SAMP using NNs-based traffic prediction**.

The performance of the different routing solutions is measured in terms of their ability to reduce the negative effects of congested links. We take into consideration the following objective Video Quality (VQ) metrics of the received streams:

- *Number of lost packets*: determined by examining the sequence number in the RTP (Real-time Transport Protocol) header;
- *Magnitude of loss*: the number of packets that are dropped at each loss event, i.e., how many packets are missing between two consecutive received packets (a magnitude of 0 means the packet arrived successfully);
- *Discontinuity counter*: the frequency of detected discontinuities (i.e. packet drops);
- *Success Ratio* (SR): the number of packets received successfully divided by the total number of packets sent.

VI. Experimental Results

Case 1 (OSPF)

In order to evaluate the OSPF protocol on Linux-based machines, the Quagga Routing Software Suite [23] is used which is an advanced routing software package that provides a suite of TCP/IP based routing protocols.

For the tested network topology, OSPF determines the same path between the source and the destination nodes for all three streams, namely `R1–R2–R3`. After 1 minute, we introduce congestion between `R2` and `R3`, but OSPF does not modify the routing tables because it does not take into consideration the physical state of the links. As an effect, we observe packet losses at the destination nodes and a very poor quality of experience. At 2 minutes from the beginning of the experiment, congestion is introduced on link `R2–R6`, but this is also not detected by the OSPF protocol.

The VQ parameters of interest for the received video streams in case of OSPF routing are shown in Table I. As we can observe, each of the streams is characterized by significant losses. A total number of 32485 packets are missing at the destinations `D1` and `D2` out of 64977 packets sent by the source node `S`, i.e., 49.99% of the transmitted packets were dropped due to congestion.

Figure 4 presents the magnitude of loss events for each video stream. In the first minute of the experiment, the magnitude of loss is 0, indicating that all packets are received at the destination nodes. It can be observed that losses appear constantly after the link `R2–R3` gets congested.

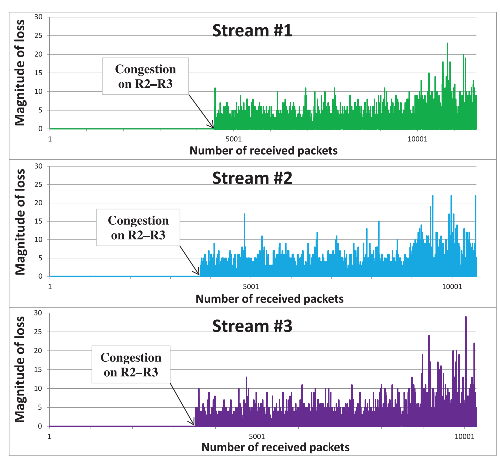

Fig. 4. Case 1 (OSPF) – Magnitude of loss

In Figure 5, the Success Ratio (SR) of the different transmissions is shown over the experiment duration. The SR corresponding to all streams starts to fall steadily after congestion is introduced on link `R2–R3`, reaching a minimum of 51.41% for Stream #1, 49.62% for Stream #2 and 48.89% for Stream #3 at the end of the experiment. The global final SR is 50.01%.

Case 2 (ECMP)

In our experiments, the ECMP routing approach is used in conjunction with the OSPF routing protocol in Quagga.

 ISBN: 978-1-61208-174-8

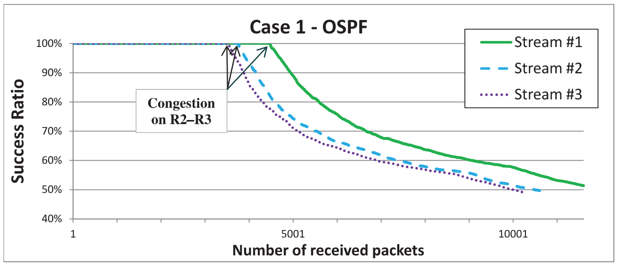

Fig. 5. Case 1 (OSPF) – Success ratio over experiment duration

This represents the only available multipath solution for Linux which allows load balancing by per-flow routing. In Linux, a flow is defined by the *source IP address* and the *destination IP address*. This means that, in our experiment, ECMP will identify only two flows which will be routed on different paths: *1)* Stream #1 and Stream #2 between `S` and `D1`, and *2)* Stream #3 sent from `S` to `D2`.

A major limitation of ECMP is that it only uses paths having equal costs. Initially, in our network topology the costs of all links were by default 10. Because there exist no multiple paths between the source and destination networks with the same cost, to be able to use ECMP, the cost of link `R2`–`R3` is set to 20. Thereby, ECMP identifies two paths with the same cost (40): *1)* `R1`–`R2`–`R3`: used for the first flow (Stream #1 and #2) and *2)* `R1`–`R2`–`R6`–`R3`: used for the second flow (Stream #3).

The parameters of the received video streams in case of ECMP routing are shown in Table I. As we can observe, the percentage of lost packets for Streams #1 and #2 is more pronounced, than for Stream #3. This can be explained by the fact that they are routed on different paths: the first two are affected by congestion for a period of 4 minutes, while the third only experiences congestion in the last 3 minutes. Out of the total number of 64977 packets sent by `S`, only 44860 reached the destination nodes, i.e., 30.96% were dropped.

Figure 6 illustrates the magnitude of loss for the video streams. In the case of Stream #1 and #2, losses appear constantly after the first minute, while packets of Stream #3 are dropped only after 2 minutes, leading to a lower frequency and average magnitude of loss events.

Figure 7 shows the SR of the different transmissions over the experiment duration. The success ratio corresponding to Stream #1 and Stream #2 starts to fall steadily after congestion is introduced on link `R2`–`R3`, reaching at the end of the experiment a minimum value of 68.86% and 65.26%, respectively. The SR corresponding to Stream #3 decreases after link `R2`–`R6` is also congested, its final value being 73.08%. The global final SR is 69.04%.

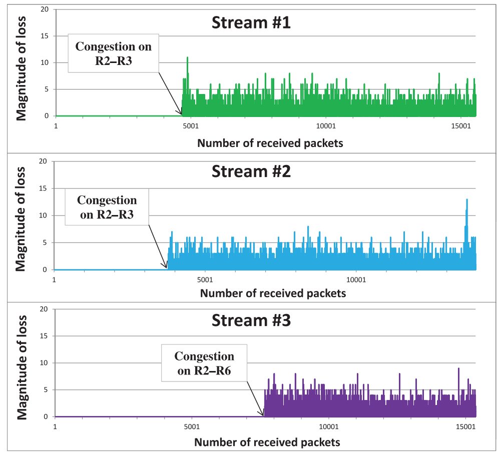

Fig. 6. Case 2 (ECMP) – Magnitude of loss

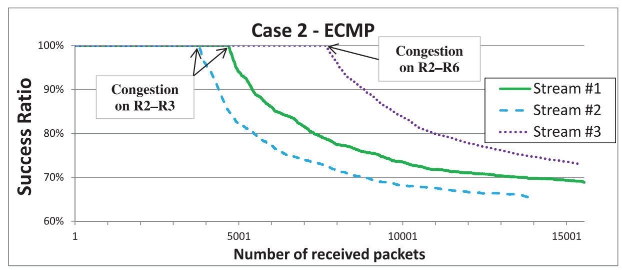

Fig. 7. Case 2 (ECMP) – Success ratio over experiment duration

Case 3 (SAMP with prediction)

The NN predictors integrated into the LMEs are implemented using *Flood*, an open source NNs C++ library [24]. A different NN is utilized to predict the ATR on every inbound link monitored by a LME. In order to reduce the overall complexity, the NNs have a small topology: 4–5–3, i.e., 4 input nodes, 5 hidden neurons and 3 output neurons. As a training algorithm, the Quasi-Newton method is used. The training lasted for 100 epochs and the learning rate was set to 0.01. The NNs are trained *offline* (i.e. before starting the

TABLE I
PARAMETERS OF THE RECEIVED VIDEO STREAMS

	Case 1 – OSPF			Case 2 – ECMP			Case 3 – SAMP with prediction		
	Stream #1	**Stream #2**	**Stream#3**	**Stream #1**	**Stream #2**	**Stream#3**	**Stream #1**	**Stream #2**	**Stream#3**
Sent packets	22567	21376	21034	22567	21376	21034	22567	21376	21034
Received packets	11602	10606	10284	15539	13950	15371	22356	21238	21034
Lost packets	10965	10770	10750	7028	7426	5663	211	138	0
% of lost packets	48.59%	50.38%	51.11%	31.14%	34.74%	26.92%	**0.94%**	**0.65%**	**0%**
Avg. magnitude of loss	0.945	1.015	1.045	0.452	0.532	0.369	0.009	0.006	0
Max. magnitude of loss	23	22	29	11	13	9	7	6	0
Discontinuity counter	4395	4386	4290	4283	4453	3355	139	81	0

experiments) with a special dataset of length 200, to detect congestions.

Until there is no congestion in the network, as illustrated in Figure 8, the proposed multipath solution sends each stream on a different path: *1)* Stream #1 on R1–R2–R3; *2)* Stream #2 on R1–R2–R6–R3, and *3)* Stream #3 on R1–R4–R5–R6–R3.

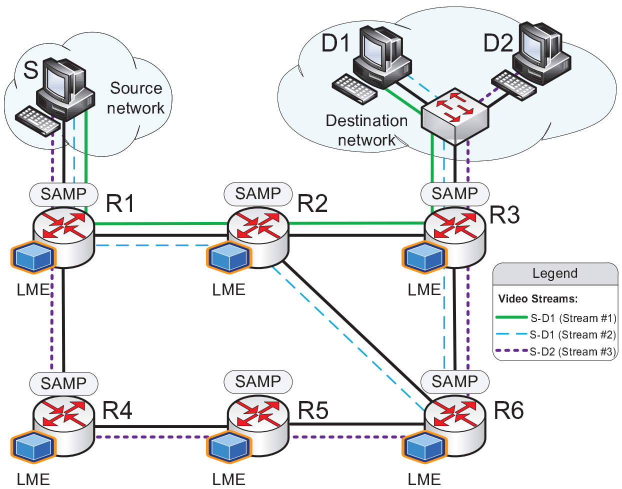

Fig. 8. Case 3 (SAMP) – No congestion

After 1 minute, we introduce congestion on link R2–R3 by starting several UDP streams between the two nodes. Thereby, the ATR on the affected link will decrease. By examining values of the ATR, the local management entity located on R3 will predict the appearance of congestion 1 second before it would be detected through simple monitoring. LME will trigger an alarm, indicating to the routing algorithm to recalculate the routes. The new best path followed by the affected Stream #1 is: R1–R2–R6–R3, as shown in Figure 9. The selection is based on the current state of the network links, choosing the path with the highest ATR and the lowest OWD.

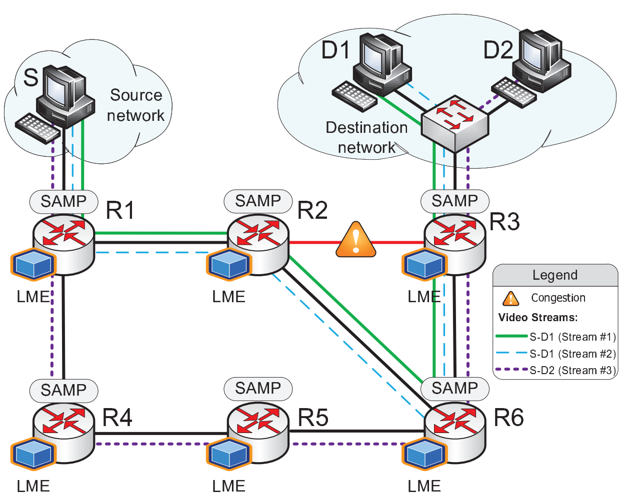

Fig. 9. Case 3 (SAMP) – Congestion on link R2--R3

The quality of the streaming is affected just for a very short period of time, mainly as a result of router reconfigurations. These are not carried out synchronously due to the distributed nature of the routing management system. At this moment, only 121 packets corresponding to Stream #1 are lost. Note that packets are also considered lost when they arrive out-of-order, because rearranging them at the destination is not feasible in case of video transmissions.

After an additional minute, congestion is introduced between nodes R2 and R6. LME on R6 detects lost packets and predicts the congestion. As a result, the multipath routing application will reroute the affected streams to the path used by Stream #3, namely R1–R4–R5–R6–R3, as presented in Figure 10. During this situation, 90 packets corresponding to Stream #1 and 138 packets from Stream #2 will be considered lost at the destination. The percentage of lost packets at the end of the experiments is 0.54% of the total number of packets sent.

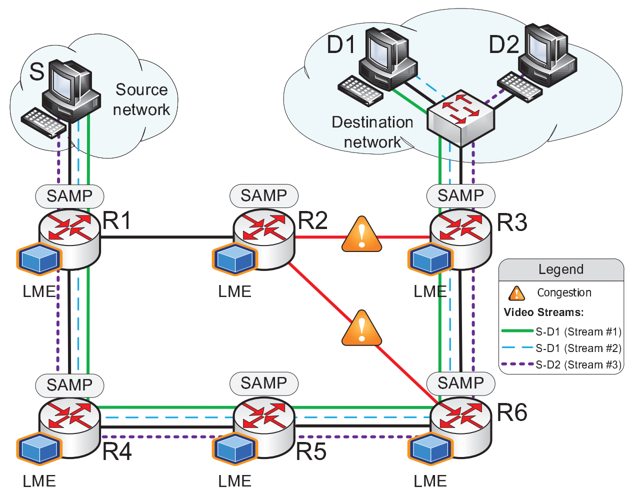

Fig. 10. Case 3 (SAMP) – Congestion on link R2--R3 and R2--R6

The VQ metrics of the received video streams when using the proposed predictive multipath routing framework are shown in Table I. Figure 11 illustrates the magnitude of loss events for the received videos. In case of Stream #1 there are two short time-intervals and for Stream #2 there is one short period in which losses occur. These correspond to the appearance of congestion and the rerouting.

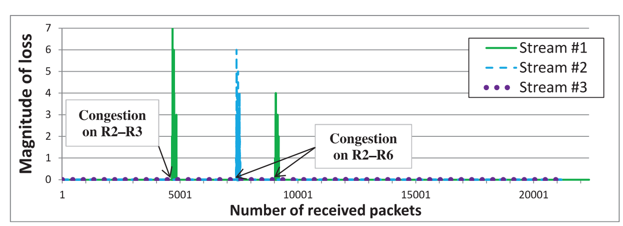

Fig. 11. Case 3 (SAMP) – Magnitude of loss

In Figure 12, the SR over the experiment duration can be observed. The SR for Stream #1 presents two local minimums: *1)* 97.56% when R2–R3 is congested and *2)* 97.76% when R2–R6 is congested; but after that, it recovers, increasing to the final value of 99.06%. For Stream #2 the SR drops for a short time to 98.21% when R2–R6 is congested, but it increases by the end of the experiment to 99.35%. The SR for Stream #3 has a constant value of 100% because it is not affected by congestion. The value of the global final success ratio is 99.46%.

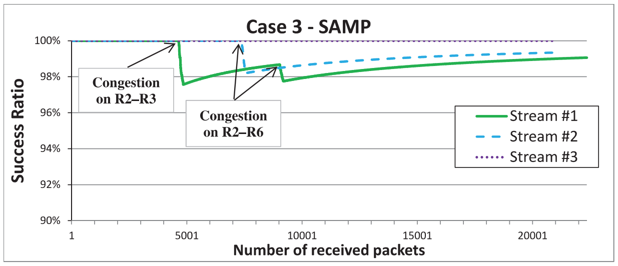

Fig. 12. Case 3 (SAMP) – Success ratio over experiment duration

Over the duration of the experiment, the prediction accuracy of the ATR is very high, in terms of NMSE (Normalized Mean Square Error) and E (Efficiency coefficient). Note that for a perfect prediction: NMSE $= 0$ and $E = 100\%$. In case of link R2–R3, we obtain NMSE $= 0.00091$ and $E = 99.91\%$, while for link R2–R6 we get NMSE $= 0.0011$ and $E = 99.89\%$.

In order to evaluate the beneficial effect of traffic forecasting, we performed the same test without predictors. In this case, the congestions is detected at a later moment, leading to a larger percentage of lost packets: 3.51%, 1.69%, and 0% for Stream #1, #2, and #3, respectively. In conclusion, if no prediction is used, the total loss (1.78%) is more than three times higher then in the case of embedding NNs-based predictors into the multipath routing framework (Table II).

TABLE II
PERCENTAGE OF LOST PACKETS

	OSPF	ECMP	SAMP no prediction	SAMP with prediction
Lost packets [%]	49.99	30.96	1.78	**0.54**

VII. CONCLUSION AND FUTURE WORK

This paper presented a multipath routing framework able to improve the global performance of the network, in case of congestion, by applying a predictive congestion control scheme. The goal was to offer a proof-of-concept by practical implementation in a real test environment. In our test scenario, the total lost percentage was: *1)* 49.99% with OSPF, *2)* 30.96% when employing ECMP, and *3)* $\mathbf{0.54\%}$ when implementing our proposed solution. This approach significantly improves the link utilization and reduces the loss rate. We cannot demonstrate it at the moment, but we foresee that similar results would be obtained in a larger network topology. As future work, we intend to verify the results through simulation.

REFERENCES

[1] Y. Li, Z. Li, M. Chiang, and A. R. Calderbank, ”Content-Aware Distortion-Fair Video Streaming in Congested Networks”, IEEE Trans. on Multimedia, Vol. 11, No. 6, pp. 1182–1191, 2009.

[2] J. Moy, RFC 2328 ”OSPF Version 2”, Internet Engineering Task Force, 1998.

[3] G. Boanea, M. Barabas, A. B. Rus, and V. Dobrota, ”Design Principles and Practical Implementation of a Situation Aware Multipath Routing Algorithm”, Intern. Conf. on Software, Telecommunications and Computer Networks 2010, pp. 321–325. Split–Bol, Croatia, 2010.

[4] Z. Fan, ”Bandwidth Allocation for MPEG-4 Traffic in IEEE 802.11e Wireless Networks Based on Traffic Prediction”, Future Generation Communication and Networking, pp. 191–196. Jeju-Island, Korea, 2007.

[5] L. Cai, J. Wang, C. Wang, and L. Han, ”A Novel Forwarding Algorithm over Multipath Network”, Intern. Conf. on Computer Design and Applications, pp. V5-353–V5-357. Qinhuangdao, China, 2010.

[6] E. Marin-Tordera, X. Masip-Bruin, S. Sanchez-Lopez, J. Domingo-Pascual, and A. Orda, ”The Prediction Approach in QoS Routing”, IEEE Intern. Conf. on Communications, pp. 1020–1025. Istanbul, Turkey, 2006.

[7] Z. Guo, S. Sheikh, C. Al-Najjar, H. Kim, and B. Malakooti, ”Mobile ad hoc network proactive routing with delay prediction using neural network”, Wireless Networks, Vol. 16, No. 6, pp. 247–262, 2009.

[8] X. Masip-Bruin, E. Marin-Tordera, M. Yannuzzi, R. Serral-Gracia, and S. Sanchez-Lopez, ”Reducing the Effects of Routing Inaccuracy by Means of Prediction and an Innovative Link-State Cost”, IEEE Communications Letters, Vol. 14, No. 5, pp. 492–494, 2010.

[9] Z. Li, R. Wang, and J. Bi, ”A Multipath Routing Algorithm Based on Traffic Prediction in Wireless Mesh Networks”, Intern. Conf. on Natural Computation, pp. 115–119. Tianjin, China, 2009.

[10] Suyang Ju and J.B. Evans, ”Intelligent Multi-Path Selection Based on Parameters Prediction”, ICC Workshops, pp. 529–534. Beijing, China, 2008.

[11] Y. Liang, ”Real-Time VBR Video Traffic Prediction for Dynamic Bandwidth Allocation”, IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 34, No. 1, pp. 32–47, 2004.

[12] Y.-R. Chuang, C.-S. Hsu, and J.-W. Chen, ”Implementation of a Smart Traffic Prediction and Flow Control Mechanism for Video Streaming”, Intern. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, pp. 240–243. Darmstadt, Germany, 2010.

[13] P. Cortez, M. Rio, M. Rocha, and P. Sousa, ”Internet Traffic Forecasting using Neural Networks”, Intern. Joint Conf. on Neural Networks, pp. 2635–2642. Vancouver, Canada, 2006.

[14] D.-C. Park, ”Prediction of MPEG Traffic Data Using a Bilinear Recurrent Neural Network with Adaptive Training”, Intern. Conf. on Computer Engineering and Technology, pp. 53–57. Singapore, 2009.

[15] V. B. Dharmadhikari and J. D. Gavade, ”An NN Approach for MPEG Video Traffic Prediction”, Intern. Conf. on Software Technology and Engineering, pp. V1-57–V1-61. San Juan, USA, 2010.

[16] S. Sun, ”Traffic Flow Forecasting Based on Multitask Ensemble Learning”, ACM/SIGEVO Summit on Genetic and Evolutionary Computation, pp. 961–964. Shanghai, China, 2009.

[17] J. Rodrigues, A. Nogueira, and P. Salvador, ”Improving the Traffic Prediction Capability of Neural Networks Using Sliding Window and Multi-task Learning Mechanisms”, Intern. Conf. on Evolving Internet, pp. 1–8. Valencia, Spain, 2010.

[18] G. Boanea, M. Barabas, A. B. Rus, V. Dobrota, and J. Domingo-Pascual, ”Performance Evaluation of a Situation Aware Multipath Routing Solution”, RoEduNet Intern. Conf. Iasi, Romania, 2011.

[19] S. Kandula, D. Katabi, S. Sinha, and A. Berger, ”Dynamic load balancing without packet reordering”, ACM SIGCOMM Computer Communication Review, Vol. 37, No. 2, pp. 51–62. 2007.

[20] R. Li, L. Ying, A. Eryilmaz, and N. B. Shroff, ”A Unified Approach to Optimizing Performance in Networks Serving Heterogeneous Flows”, IEEE/ACM Trans. on Networking, Vol. 19, No. 1, pp. 223–236. 2011.

[21] M. Barabas, G. Boanea, A. B. Rus, and V. Dobrota, ”Routing Management Based on Statistical Cross-Layer QoS Information Regarding Link Status”, Intern. Conf. on Knowledge in Telecommunication Technologies and Optics, pp. 8-13. Szczyrk, Poland, 2011.

[22] D. Thaler and C. Hopps, RFC 2991 ”Multipath Issues in Unicast and Multicast Next-Hop Selection”, Internet Engineering Task Force, 2000.

[23] Quagga Routing Software Suite, http://www.quagga.net/ (Last accessed: 20.09.2011)

[24] R. Lopez, (2010). Flood: An Open Source Neural Networks C++ Library (Version 3) [software]. Retrieved from www.cimne.com/flood (Last accessed: 20.09.2011)