

# Supporting Multimedia Services in the Future Network with QoS-routing

Leandro Alexandre<sup>(1)</sup>, Augusto Neto<sup>(2)</sup>, Eduardo Cerqueira<sup>(3)</sup>, Sérgio Figueiredo<sup>(4)</sup>,  
Rui Aguiar<sup>(4)</sup>

<sup>(1)</sup> Federal Institute of Goiás, Vale das Goiabeiras  
754000-000 Inhumas-GO, Brazil

<sup>(2)</sup> Federal University of Ceara (UFC), Department of Teleinformatics Engineering (DETI),  
Group of Computer Networks, Software Engineering and Systems (GREAt),  
74001-970 Goiânia-GO, Brazil

<sup>(3)</sup> Federal University of Pará (UFPA), EngComp/PPGEE/PPGCC, Group of Study in  
Network Computers and Multimedia Communications (GERCOM),  
66075-110 Belém-PA, Brazil

<sup>(4)</sup> Telecommunication's Institute, University of Aveiro, Campus Universitário de Santiago  
3810-193 Aveiro, Portugal

leandroalexandre@ifg.inhumas.br, augusto.deti@ufc.br,  
cerqueira.ufpa.br, sfigueiredo@av.it.pt, ruilaa@ua.pt

**Abstract** The increasing demand for real-time multimedia applications for groups of users, together with the need for assuring high quality support for end-to-end content distribution is motivating the scientific community and industry to develop novel control, management and optimization mechanisms with Quality of Service (QoS) and Quality of Experience (QoE) support. In this context, this paper introduces Q-OSys (QoS-routing with Systematic Access), a distributed QoS-routing approach for enhancing future networks with autonomous mechanisms orchestrating admission control, per-class overprovisioning, IP Multicast and load-balancing to efficiently support multi-user multimedia sessions. Simulation experiments were carried to show the efficiency and impact of Q-OSys on network resources (bandwidth utilization and packet delay). Q-OSys is also evaluated from a user point-of-view, by measuring well-known objective and subjective QoE metrics, namely Peak Signal to Noise Ratio (PSNR), Structural Similarity (SSM) Video Quality Metric (VQM) and Mean Opinion Score (MOS).

**Keywords:** Future Internet, QoS-routing, multicast, QoS

## 1 Introduction

The growing interest for multi-user sessions having a strict demand for resources (e.g. real-time multimedia, personalized and immersive services – 3D, among others) has fuelled the emergence of social added-value services such as healthcare, location based services, environmental monitoring, seismic activity or energy management. These sessions are characterized by their high sensitivity to delay (and its variation)

and packet loss, and are supported by complex structures. Internet's current method of Best-effort cannot efficiently support the requirements of these sessions. Thus, future IP networks expect autonomous and robust mechanisms to cope with key issues, such as intermittency, scalability and reliability, having the difficult challenge of providing ubiquitous access (everywhere, by anyone, anytime) with guaranteed quality in heterogeneous environments.

Controlling and managing such features are a tremendous challenge. In order to overcome the aforementioned challenges, the scientific community has proposed solutions for supporting multi-user sessions with guaranteed end-to-end Quality of Service (QoS). Management strategies for such objectives require self-organized features inside the network, since a centralized control system in packet-based networks would require stringent control mechanisms. In this distributed scenario, routing support oriented by QoS parameters (a.k.a. QoS-routing) [8] is very important for efficient network supply in order to establish sessions with appropriate Quality of Experience (QoE) over the time, involving low values of propagation delay, delay variation, jitter and packet loss. However, literature shows existing solutions' inefficiency for QoS-driven routing support, due to its high cost and limited scalability – an unfortunate issue, since such a solution would inherently embed in the network multi-session support, and would reposition the central network management system as a policy controller.

Moreover, for live entertainment, the limited IP multicast coverage area, a per-flow control architecture and non-QoS orientation, undermines efficient multi-user sessions. The implementation of load balancing for such scenarios is promising, by optimizing network processing through systematic access to the system's bandwidth. This reduces the network node's resources (processing, memory and energy) consumption due to the increase of the link's residual bandwidth, since new paths are chosen independently of depletion of others (in contrast to the state of the art). We haven't found in the literature a single solution embedding the mechanisms described above. In general, multiple tools are used together to provide such features, which greatly increase the system's complexity and maintenance cost, making it virtually unmanageable.

Therefore, in our vision for future systems, we consider an embedded routing approach that significantly removes the load of the central management system. In this paper we propose Q-OSys (QoS-routing and Systematic access), a QoS-routing approach orchestrating admission control, QoS-centric per-class resources over-provisioning, IP multicast and load-balancing for efficient support of multi-users session in future IP networks, under the policy control of a central management system. Q-OSys stands-out from the existing solutions due to its embedded autonomous architecture, where the complexity is implemented at the network's borders, in order to keep the network's core as simple as possible. Given that in our vision, the central management system becomes a simpler policy setting unit, we focused on the benefits of Q-OSys for traffic performance. These were evaluated through simulations, revealing low use of bandwidth and delay in the paths along the simulation time. Moreover, the impact of Q-OSys on the user experience was also assessed by analysis of real video sequences, supported by objective and subjective QoE metrics: Peak Signal to Noise Ratio (PSNR), Structural Similarity (SSM), Video Quality Metric (VQM) and Mean Opinion Score (MOS).

The remaining of this article is organized the following way: Section 2 presents a State of The Art study; Section 3 describes Q-OSys in more detail, and in Section 4 the evaluation of the mechanism is performed. Finally, Section 5 concludes the paper and proposes future work.

## 2 Related Work

This section seeks to analytically evaluate mechanisms under the scope of quality-oriented, load balancing and multi-user transport support. Our related work study exhibits that current routing patterns (with or without QoS support) are per-flow driven (e.g. [12]) with hop-by-hop operations. Hence, the consensus is that per-flow QoS-routing solutions are not cost-effective and have very limited scalability due to excessive signalling events [13]. A classic QoS-routing work is the Quality Of Service extensions to Open Shortest Path First (QOSPF), which extends the well-known OSPF protocol [1] with QoS utilities. For guaranteeing compliance with QoS-based restrictions of input flows, the coexistence with the Resource Reservation Protocol (RSVP [2]) is necessary for dynamic resource reservation; otherwise, QOSPF implements statistical methods based in bandwidth and delay characteristics of the link. QOSPF follows a flooding approach for keeping network's QoS state, where each node signals the network after handling a reservation. This way, QOSPF's scalability level is very limited, as the signalling increases proportionally with the number of input flows. Besides, QOSPF embeds neither multicast support nor load balancing. These problems are identified in [4]. Since we claim that per-flow QoS-driven routing approach is not suitable for multi-user sessions, and all existing proposals share similar conclusions regarding poor both cost-effectiveness and scaling capabilities, it is a waste of time analysing other relating proposals.

In terms of routing approaches aided with load balancing utility, the Multi-path-Iterative-Routing-Traffic-Optimizer (MIRTO) [3] [5] has gained some attention. MIRTO adopts a distributed routing protocol, which explores available paths' diversity for implementing load balancing. For such, a MIRTO agent, deployed at the network's border, distributes the received flows' load between multiple paths in a systematic way, allowing a better use of the whole system's bandwidth. Although it supports load balancing, MIRTO doesn't implement QoS control, neither multicast, which handicaps its deployment in future IP networks.

The Multiuser Aggregated Resource Allocation Mechanism (MARA) [7] is an over-provisioning centric proposal that distinguishes itself among others related solutions by its coupled edge-to-edge management of per-class over-reservations and SSM IP multicast aggregated trees. MARA adopts mechanisms to dynamically adjust over-reservations according to session demands, current network's QoS capabilities and in advance resource information (booked on the system boot-up). As such, MARA allows multiple sessions establishment without any signalling, being only necessary to readjust the over-reservation levels inside the domain. MARA objectives do not include routing, and to that it interfaces with intra-domain solutions (e.g., OSPF) for indicating paths. Thus, MARA may not guarantee the selection of the best

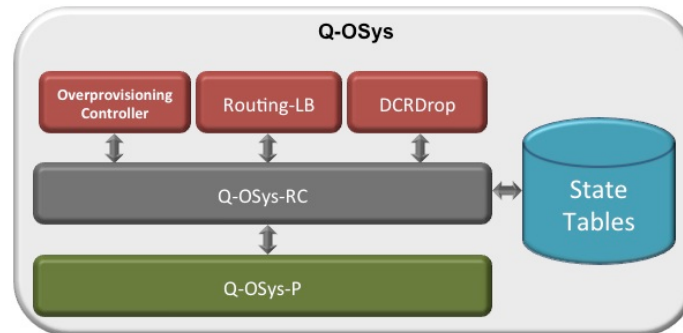
paths when the current intra-domain routing solution is not driven by QoS metrics. Besides, MARA doesn't implement load balancing.

Our related work analysis exhibits that the literature lacks of over-provisioning centric QoS-routing proposals, and current QoS-driven routing patterns are neither cost-effective nor scalable due to their per-flow basis, thus supporting the need for the Q-OSys approach.

### 3 Q-OSys Description

In terms of QoS, an over-provisioning approach orchestrated with admission control and dynamic resource provisioning is deployed in Q-OSys, allowing the setup of multiple group-based sessions (multi-user) without per-flow signalling. In practice, several of the typical management functions on a per-flow aware network are taken by the routing concept we developed. Q-OSys keeps a state table with QoS information regarding best paths, providing QoS-routing. Thus, routing decisions can be implemented autonomously and more efficiently by means of a global view of the network and its updated QoS capacities, unlike solutions that only keep next-hop information using inefficient metrics, and updated using constant flooding operations (e.g. OSPF). Figure 1 depicts Q-OSys proposed architecture, with the following blocks:

- Q-OSys Protocol (Q-OSys-P): Provides the system with support for inter-Q-OSys agents communication;
- Q-OSys Resource Controller (Q-OSys-RC): interacts with elements of the Internet model for handling QoS state (over-reservation by class) and connectivity (aggregated multicast trees);
- Over-provisioning Controller: Provisions QoS features and connectivity, and controls their adjustments according to the session's demand;
- Routing with Load Balance (LB-Routing): makes decisions about the quality-driven routing and load balancing according to demand for new sessions or due to re-routing;
- Detection of Congestion or Router Drop (DCRDrop): detects congestion based in the queues of the routers through Random Early Detection (RED);
- State table: contains information regarding network state, including routing tables, metrics for QoS per class of service and per tree, active sessions, among others.



**Fig1.** Q-OSys Architecture

Basically, internal interfaces are implemented to enable inter-communication between modules of Q-OSys and local standards, and external interfaces to expose Q-OSys facility to mechanisms and standards outside the system. In this text we neglect the policy functions, which can be centralized in a single central entity.

### 3.1. Q-OSys Features

This section describes the features supported by Q-OSys as well as the operations implemented by the mechanism that fulfil them.

#### 3.1.1. QoS Resources and Connectivity Provisioning

The QoS resources and connectivity provisioning mechanism aims to allocate per-class over-reservations and aggregated multicast trees in advance. Therefore, at network bootstrap, the Q-OSys mechanism located at the Ingress Router (IR) starts by flooding the whole network through RESERVE messages, with flag I activated, that way realizing per-class over-reservation, taking into account an initialization factor set by the network operator (e.g.  $\frac{1}{2}$  or  $\frac{1}{4}$  of the capacity of the local link). In order to avoid redundant operations, each node should check some issues before configuring the resources; for example, they should guarantee per-class over provisioning is initialized only once.

Thereafter, each router that received the RESERVE message adds in its RSVPATH field its local IP address. In addition to information about the path, the QoS parameters are also analysed, being replaced when there is updated information (bandwidth, delay, jitter and loss). Finally, upon reaching the egress Router (ER), a new RESPONSE message is composed, containing the OK flag enabled. Subsequently, the ER sends this message to the corresponding IR, which contains all information from the communication path.

Based on the received unicast paths, the IR starts the composition of the multicast paths. Since a large number of trees can be created, Q-OSys filters and retains only the best trees according to some criteria. For example, trees that have the IR in the

centre of the network are discarded so as to maintain downstream flows only (from IR to ER). After filtering these trees, Q-OSys define IP multicast addresses to a set of them and sends the message RESERVE (M) towards the network, responsible for creating multicast trees in network routers by means of PIM-SSM [9].

### 3.1.2. QoS-routing

As a support for QoS-routing, the routing tables are built at initialization of the system, i.e., during the flooding process using the RESERVE(I) message (alternatively, this initialization may be made by the central policy management system). In first place, the downstream routing tables are built. Each sent message contains the router's source IP address, and based on that, the routers at the core map this IP address to the network interface at which the message was received. Besides, other parameters are added to the routing table and to the message, such as each of the interface's QoS metrics (which allows calculating the cost of the message transmission to each of the local interfaces) and the Autonomous System (AS) ID from which the message was sent. At last, when arriving to the ER, a new message is built, a RESPONSE (OK), which permits the composition of the upstream routing tables. The same principle is applied, with the IP address being set to the network interface from which the message was received. The Algorithm 1 corresponds to the pseudo code developed, in which the session will be mapped.

The QoS-routing algorithm of Q-OSys was developed according to the classical theorem of "minimum cost flow", which is based on the method of Ford-Fulkerson and Dijkstra [6]. It tries to group different streams of the same session on the same path in order to save resources and provide better use of the bandwidth of the path used. However, this is not always possible and for this reason, the algorithm was designed to distribute data flows between two or more paths for the same session.

---

**Algorithm 1: Choose the Way**

---

```

Result: Chosen Paths
Declaration of Variables:
BW ← Bandwidth requested;
cos ← CoS session;
ipDestination ← IP address of the destination host;
bwResidual ← Residual bandwidth;
eRouters[] ← List of egress routers;
treeMaxBWRes[] ← Trees with higher bandwidth residual;
treeLowerCost[] ← Trees with lower cost;
begin
  eRouters[] ← Filter and select the paths that have Egress Routers reaching the
  destination host. For this, run some algorithm EGP;
  treeMaxBWRes[] ← Filter and select the trees that have higher Residual BW;
  treeLowerCost[] ← Filter and select the trees that have lower cost;
  while (BW > 0) do
    treeMaxBWRes ← Select from treeMaxBWRes [], the tree that has the
    highest residual BW;
    treeLowerCost ← Select from treeLowerCost [], the tree that has the lowest
    cost;
    chosen path[] ← chosen path;
    BW ← bwResidual - BW;
  end

```

---

### 3.1.3. Detection of Congestion and Router Drop for Resilience

The Q-OSys has an autonomous module in charge to provide robustness for the system, which is done through managing link drop and router failure events in the network for adding resilience, called Detection of Congestion and Router Drop (DCRDrop). This mechanism tries to avoid packet loss in situations of network overload [11] or routers failure. In order to monitor the queues of the routers in the network, a detection mechanism based on RED queues [10] was developed. The RED mechanism is designed to work together with the TCP protocol. However, as most of the considered sessions require the use of UDP protocol for data transmission (e.g. video and audio streaming), RED was adapted so that the transport protocol did not influence the effectiveness of our mechanism. Another adjustment made refers to the packet dropping procedure. Q-OSys mitigates quality degradation by deploying admission control, and in cases of network congestion, it performs load balancing to increase users' satisfaction level and allow more effectively leveraging network resources. *Algorithm 2* shows the pseudocode corresponding to the congestion detection mechanism of the RED queue.

---

**Algorithm 2: Congestion Detection**

---

**Result:** Packet queuing and congestion detection

**Declaration of Variables:**

avg  $\leftarrow$  Average Queue Size RED;

Minth  $\leftarrow$  Lower limit of the RED queue;

Maxth  $\leftarrow$  Upper limit of the RED queue;

count  $\leftarrow$  Number of times the package should have been marked;

**begin**

    Calculate avg;

**if** (avg < Minth) **then**

        | Queues the packet

**else**

**if** (avg > Minth) and (avg < Maxth) **then**

            Calculates the probability  $P$  the package is discarded;

**if** (The package has arrived probability that  $P$  to be discarded) **then**

                | Notify agent Q-OSys in the eminence of network congestion;

                | Queues the packet;

**else**

                | Queues the packet;

**else**

**if** (avg > Maxth) **then**

                | Notify agent Q-OSys in the eminence of network congestion;

                | Queues the packet;

**end**

---

In case of router failure, Q-OSys performs fault tolerance procedures, avoiding the unexpected termination of active sessions. For this, it uses mechanisms for monitoring active routers, and in case of failure in any of these, a load-balancing algorithm is invoked to re-route all active sessions of the affected path.

### 3.3.4. Load Balancing

The load-balancing algorithm aims to provide systematic access to the bandwidth of the system, maintaining the maximum possible residual bandwidth for network processing optimization. The main idea is to connect the incoming flows to the multiple available paths. The main impact of this strategy is congestion, because multiple streams will be re-routed. For this, the DCRDrop is constantly monitoring the core routers. In case of congestion or routers failure, the DCRDrop sends an alert message to the IR, which triggers the load-balancing algorithm. *Algorithm 3* presents the proposed pseudocode for this process.

The first action of the load balancing algorithm is to verify the source of the problem which occurred on the network: *i*) congestion of RED queues of the routers or *ii*) router failure. In case congestion is detected (i.e., when router queues reach the limit of *maxth*) the algorithm will initially calculate the rate of utilization of each network tree and thereafter, the overall use rate. This will allow the algorithm to check the amount of sessions that should be re-routed. An example would be as such: the global resources utilization of the network is 50% and the utilization rate of the degraded tree is 80%. In this case, the re-routing of 30% of the flows from the degraded tree is required, so that burden is distributed among the other trees according to their current capabilities. Another action taken into account by the algorithm is selecting sessions for re-routing. The load-balancing algorithm always picks the most recent sessions, as the oldest sessions are usually close to completion.

For router's fault detection, the load-balancing algorithm will perform the redirection of all active sessions for that data path. For each of these sessions, the path selection algorithm is triggered, always taking into account the initial requirements of the session, the cost of the path (that must always be the minimum), and QoS, which should be ensured, to the user.

---

**Algorithm 3: Load Balancing**

---

**Result:** Re-routing of flows

**Declaration of Variables:**

treeResources[]  $\leftarrow$  Resources used by each tree, in percentage;

networkResources  $\leftarrow$  Average total resources used by all the trees of the network, in percentage;

treeHigherRatesResourcesUtilization  $\leftarrow$  Tree with higher rates of resource utilization;

selectingTreesDegraded[]  $\leftarrow$  Trees that are degraded by the router;

amountTreeDegraded  $\leftarrow$  Number of trees damaged;

selectingTreesUndegraded[]  $\leftarrow$  Trees that are not degraded by the router;

**begin**

    Calculate the utilization of each tree:

    treeResources[]  $\leftarrow \sum_{i=1}^n \text{FunctionForCostCalculation} / n$ ;

    \*n: Number of routers in the tree.

    To calculate the use of the global resources of the net:

    networkResources  $\leftarrow \sum_{i=1}^n \text{treeResources}[] / n$ ;

    \*n: Number of trees.

    Selection of Trees:

    selectingTreesDegraded[]  $\leftarrow$  Filter and select the trees that are degraded by the router;

    amountTreeDegraded  $\leftarrow$  number of trees damaged;

    selectingTreesUndegraded[]  $\leftarrow$  Filter and select the trees that are not degraded by the router;



```

if (Any network degradation) then
  for ( $i \leftarrow 0$  to  $\text{amountTreeDegraded} - 1$ ) do
    if (Rate of resources used by the tree is > Average total resources used by the network) then
       $\text{treeHigherRatesResourcesUtilization} \leftarrow$  Rate of resources used by the tree.
    Provide load balancing:
    What is the number of streams to be routed?
     $\text{amountFlowsRouted} \leftarrow \text{treeHigherRatesResourcesUtilization} - \text{networkResources}$ ;
    What flows are routed?
    Flows are routed latest.
    After this, run the algorithm to choose the way: its restriction is related to the fact that he should choose only those trees that are not degraded by the router.
  else
    if (It will have fall of Link or Router) then
      Route all flows passing through the link or router that crashed.
      After this, run the algorithm to choose the way: its restriction is related to the fact that he should choose only those trees that are not degraded by the router.
  end
end

```

## 4 Q-OSys Evaluation

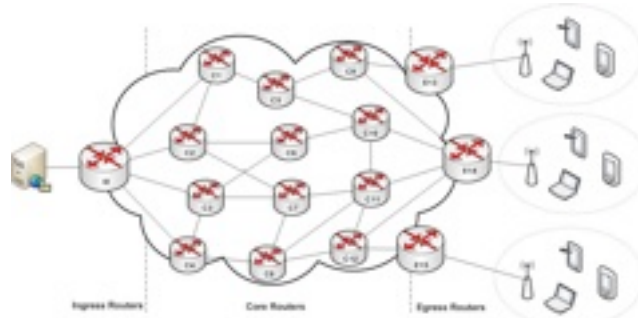
The tests and evaluations performed with Q-OSys were developed with the goal of determining the efficiency of the mechanism when compared to other solutions. For this, the functionality of the simulator NS-2<sup>1</sup> (Network Simulator) was extended, and measurements regarding the network (QoS) and the users (QoE) status were collected. The policy flexibility and complexity of Q-OSys were not evaluated, since it would require a very different evaluation environment.

The selected simulation model used a topology based on real networks with 16 nodes, bandwidth of 10Mbps and varying link delay, jitter and losses. For the differentiation of classes, a structure based on DiffServ was used, as well as Weighted Fair Queuing (WFQ) scheduling, Token Bucket policing and Random Early Detection (RED) queue management algorithms. The system has a network entry point, the Ingress Router (IR), and three exit routers, called Egress Routers (ER). The mechanism of choosing the paths and load balancing is located in the IR, and the user requests are routed from the ERs to the IR. We consider 4 distinct service classes: *i*) Expedited Forward (EF), *ii*) Assured Forward 1 (AF1), *iii*) Assured Forward 2 (AF2) and *iv*) Best-effort (BE), and 250 sessions for each of those classes, summing up to a total of 1000 flows. The streams' rates can have a constant data transmission of 128 Kbps or 256Kbps for each one. In order to make the scenario more realistic, the sessions were initialized at varying time intervals, by using a random time generator. The total duration of simulations is of 60 seconds. Finally, to demonstrate the impact generated by Q-OSys on the user experience, both objective and subjective measurements were made using QoE metrics in a real video. The tool Evalvid<sup>2</sup> was

<sup>1</sup> NS-2, The NS-2 Home Page, Web site: <http://www.isi.edu/nsnam/ns/>

<sup>2</sup> Klaue, J., Rathke, B., and Wolisz, A. "Evalvid - a video quality evaluation tool-set", 2008.

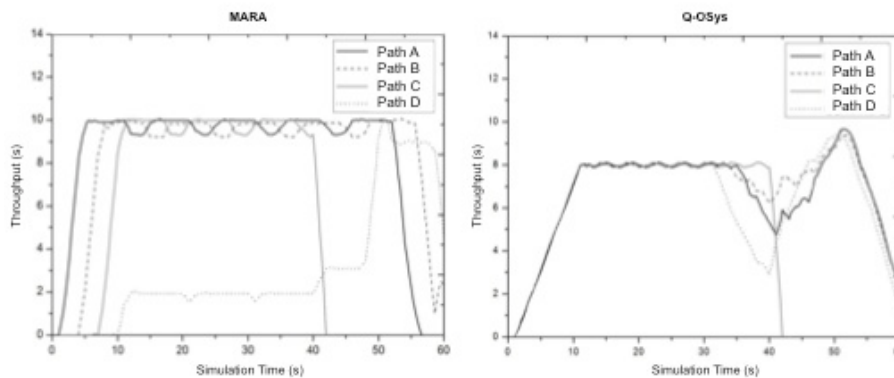
used to assess and validate the QoE evaluation. Figure 2 shows the scenario used for the simulation described above.



**Fig2.** Network topology of simulated scenario

#### 4.1. Network-based experiments

To evaluate the ability of re-routing, load balancing and support for fault tolerance, a link break was simulated between the router input I0 and core router C1, at  $t = 41$  seconds. Figure 3 shows the simulation results for the utilization rate of the links for each of the selected paths, where the results revealed that paths A, B, C and D have been selected over the entire experiment by the best path selection algorithm. MARA [7] was used as comparison against Q-OSys.

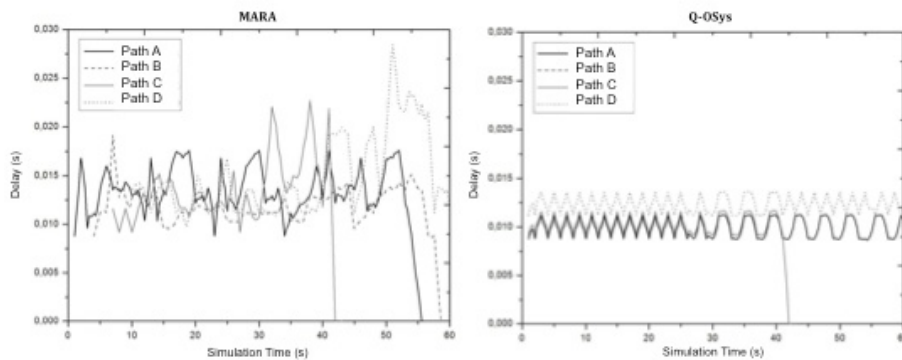


**Fig3.** Flow comparison: MARA vs. Q-OSys

From the referred figure, the sessions' distribution between the different data paths (A, B, C and D, in this case) can be depicted, for both Q-OSys and MARA mechanisms. Looking at the graphic of MARA, we realize that path B is selected only from the moment that path A runs out of bandwidth capacity. The same happens for the other paths. When the link break event occurs, it can be observed that the MARA starts redirecting all sessions to path D, despite the existence of any other way with lower cost or with higher bandwidth waste. The random session end of each of the

sessions leads to a fluctuation in the data rates of all the paths, which translates into to global instability of networks that use the MARA.

Opposed to this, the simulation results of the Q-OSys show a very different behaviour for the bandwidth and support for fault tolerance on all selected paths. During the simulation, all paths have experienced a uniform resource utilization, with an average of 8 Mbps per link, before breaking the link, reaching maximum capacity only once, at time 53 seconds, to compensate for the loss of the path C. Furthermore, the impact of load balancing with QoS-routing allows Q-OSys the uniform allocation of bandwidth on all paths. Thus, the reduction of bandwidth waste in all the paths evidences the efficiency of Q-OSys, when compared with MARA, which can be performed by analysing the delay values, presented in Figure 4.



**Fig4.** Delay measurements: MARA vs. Q-OSys

The results presented in Figure 3 show that Q-OSys enabled a better experience in terms of delay on all paths - with a minimum of 9 ms and a maximum of 13 ms, and with standard deviations of 0.0012, 0.0012, 0.0024 and 0.0012 for trees A, B, C and D respectively - than that of MARA - reaching ~ 27 ms peak delay, with standard deviations of 0.0049, 0.0041, 0.0072 and 0.0065 for trees A, B, C and D, respectively. The load balancing with QoS-routing enabled the Q-OSys a uniform behaviour regarding the propagation delay, even after the link break. In addition, a small delay variations were verified, which is of paramount importance to ensure the quality of multimedia sessions. The jitter increased by 40%, which occurred after the link failure, having an average value of 0.9 ms. Therefore, the conclusions to be drawn of the network-based tests is that the supply of resources using load balancing and QoS-routing is more efficient than the MARA (and therefore other solutions).

#### 4.2. User-based experiments

In experiments based on the user, the Q-OSys was analysed using a regular Internet setup (IP standard), configured with a default QoS per class (DiffServ) and Best-Effort Routing (OSPF). Thus, the IP Standard setup did not apply admission control or resource reservation operations. On average, the numerical results showed that Q-

OSys did not present packet loss and blocked only 2.8% of video sessions. Using “IP standard” configuration, 24% of packets were lost considering all the video sessions.

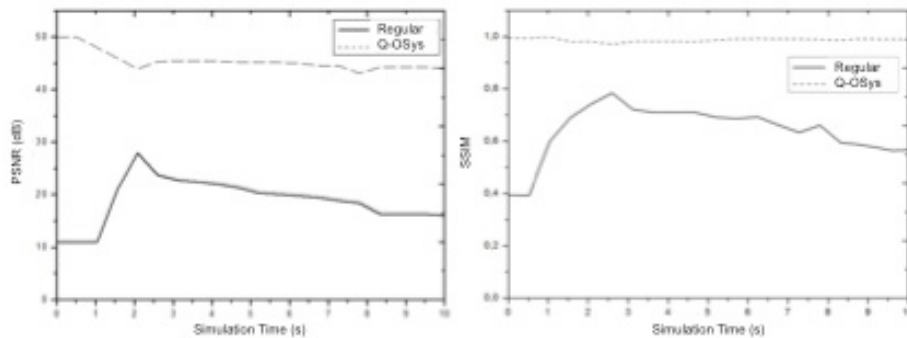
The Peak Signal to Noise Ratio (PSNR) metric is more objective and traditional metric, and performs frame-by-frame comparison between the quality of the video received by the user and the original video. For a video to be considered with good quality from the user perspective, it must have an average PSNR of at least 30 dB. This is based on the mapping of PSNR to MOS values, as shown in Table 1, where the MOS is considered the most popular subjective measure.

**Table 1.** Mapping between PSNR and MOS values

PSNR (dB)	MOS
> 37	5 (Excellent)
31 – 37	4 (Good)
25 – 31	3 (reasonable)
20 – 25	2 (Poor)
< 20	1 (Bad)

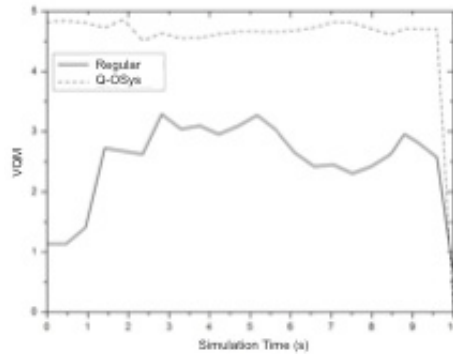
In our simulations, we use the video file “Foreman”, provided by the site Evalvid. News for the video, the average PSNR for “IP Standard” was 19 dB (with standard deviation of 4.6), as illustrated in Figure 5. Furthermore, the video is considered poor according to the user experience, as presented in Table 1. However, when the Q-OSys is used, the average PSNR passed to 45 dB (with standard deviation of 1.9), thus maintaining the excellent video quality, even in periods of congestion.

In order to make a comparison that takes into account the structure of objects and provides a better assessment than the PSNR, the metric Structural Similarity (SSIM) was obtained, which breaks the sent and received images, taking into account three HVS components: brightness, contrast and structural distortions. The SSIM index is a decimal value between 0 and 1, where 0 means there is no correlation with the original image, and 1 means it is the same image. Figure 6 shows the SSIM values for both mechanisms, where it is clear that Q-OSys enables the video content in real time to be supported with an excellent level of quality throughout the experiment (SSIM of 0.99 on average), whereas using the “IP Standard” the SSM values are around 0.63 (with a standard deviation of 0.12).



**Fig5.** PSNR vs. SSIM

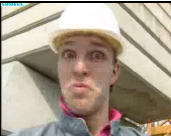





Video Quality Metric (VQM) uses as input the original and the processed videos and checks the level of multimedia quality based on the perception of the human eye and in subjective parameters, including blurring, noise, global block distortion and colour distortion. The VQM values range from 0 to 5, where 0 is the best possible score. As shown in Figure 7, on average, the VQM values are 2.57 (standard deviation of 0.73) and 4.69 (SD: 0.96) for “IP standard” and Q-OSys, respectively.



**Fig6. VQM**

In order to show the impact of Q-OSYS (compared to “IP Standard” setup) from the user’s point of view, some randomly selected video frames were captured when the system is experiencing ~ 18 % of congestion (see Table 2). The benefit of the Q-OSys is visible in the video frames.

**Table 2.** Frames of “Foreman” Video with Q-OSys and Regular configurations

Configuration	Frame No [92]	Frame No [93]	Frame No [94]
Q-OSys			
Regular			

## 5 Conclusion and Future Work

We have proposed a powerful distributed approach to routing in order to support multimedia services in the future Internet. In our management vision, a policy system configures this distributed control solution, providing scalability and low response time. The simulations for assessing Q-OSys performance have demonstrated its advantages in comparison with relevant related work (MARA experiments for network-based) as well as current Internet QoS and routing standards. In network-based experiments, Q-OSys drastically streamlined the experienced delay in all

selected data paths (14ms delay peak, while MARA reached 27ms). In addition, Q-OSys evenly balanced the workload on the paths, allowing a greater amount of residual bandwidth. In user-based experiments, Q-OSys did not show any packet loss, while using the “IP Standard” configuration, 24% of packets were lost. Moreover, when it comes to user experience, it became clear that the video content displayed in real time by Q-OSys has an excellent level of quality, proven by the QoE measurements, where the average SSIM value was 0.99, a VQM value of 4.69, and a PSNR of 45 dB. Whilst using “IP standard” configuration we obtain an average of 0.63 in SSIM, a VQM of 2.67 and a PSNR of 19 dB.

These results provide a strong basis for evaluating the Q-OSys through prototyping, for more precise conclusions. We claim that its benefits can solve some of the performance problems of current wireless mesh networks. Therefore, future work will include extending Q-OSys for IEEE 802.11s compliant networks.

## 6 Acknowledgments

CNPq supported this work. Part of this work has been sponsored by the EU project Medieval.

## 7 References

- [1] Moy, J., “OSPF Version 2”, RFC 2178, Internet Engineering Task Force, April 1998.
- [2] Brades, R. *et al* “Resource reservation protocol RSVP Version 1 Functional Specification”. RFC 2205, Internet Engineering Task Force, September 1997.
- [3] Muscariello, L. and Perino, D. “Evaluating the performance of multi-path routing and congestion control in presence of network resource management”. In Proc. of the IEEE ICUMT, San Petersburg, Russia, 2009.
- [4] Daxin, Z., Danlin, C., “Research and simulation of distributed QoS routing algorithm”, In Proc. of the 2<sup>nd</sup> IEEE IC-BNMT, 2009.
- [5] Nardelli, B., Muscariello, L. and Perino, D. “Towards real implementations of dynamic robust routing paradigms exploiting path diversity”. In Proc. of the IEEE ICUMT’09, San Petersburg, Russia, October 2009.
- [6] Dijkstra, E. W., “A note on two problems in connexion with graphs”. Journal of Numerische Mathematik, Vol. 1, Issue 1, pp. 269-271, Springer Berlin / Heidelberg, 1959.
- [7] Neto, A. *et al*. “Scalable resource provisioning for multi-user communications in next generation networks”. IEEE Globecom’08, New Orleans, LA, USA, 2008.
- [8] Crawley *et al*. “A framework for QoS-based routing in the internet”. RFC 2386, Internet Engineering Task Force, August 1998.
- [9] Bhattacharyya, S. “An Overview of Source-Specific Multicast (SSM)”. RFC 3569, Internet Engineering Task Force, July 2003.
- [10] Steiglitz, K. and Papadimitriou, C. H. “Combinatorial Optimization: Algorithms e Complexity”. General Publishing Company, Ltd. 30 Lesmill Road, Don Milss, Toronto, Ontario, January 1998.
- [11] Braden, S. *et al*. “Recommendations on Queue Management and Congestion Avoidance in the Internet”. RFC 2309, Internet Engineering Task Force, April 1998.
- [12] Huang, L., Zhang, Y., Ren, Y., “Two multi-constrained multicast QoS routing algorithms,” In Proc. of the 8<sup>th</sup> ACIS SNPD, Qingdao, China, 2007.
- [13] Ariel Orda. 2005. “QoS Routing: Challenges and Solution Approaches”. In Proc. of the QShine’05, Orlando, FL, USA, August 2005.