

# Evaluation of Deficit Round Robin Queue Discipline for Real-time Traffic Management in an RTP/RTCP Environment

Iffat Ahmed\*, Leonardo Badia\*, and Khalid Hussain<sup>§</sup>

\* IMT Institute for Advanced Studies, Piazza S. Ponziano 6, 55100 Lucca, Italy

§ Faculty of Computing, Riphah International University, Islamabad, Pakistan

email: {*iffat.ahmed,leonardo.badia*}@imtlucca.it, *khalidusmani\_65@yahoo.com*

**Abstract**—Multimedia real-time traffic is deemed to be dominant in future communication systems. One of the reference applications to support real-time traffic is the Real-time Transport Protocol (RTP), which can be used to transmit multimedia contents on real-time basis. At the same time, Real-time Transmission Control Protocol (RTCP) is used for receiving feedback and getting information about the network. This paper proposes and evaluates a traffic management implementation in such an RTP/RTCP environment for congestion control. Deficit Round Robin queue discipline is used as the traffic management strategy instead of Random Early Detection and DropTail queue disciplines. A simulation campaign was performed to analyze the effects of implemented traffic strategies in RTP/RTCP environment and compare it with previous solutions. The obtained results highlight a significant difference in terms of jitter delay and packet losses and improvement the bandwidth utilization for real-time flows. Thus, we are able to provide quantitative evidence of the importance of the queue discipline to efficiently manage multimedia content.

**Index Terms**—Multimedia traffic, real time protocol, real-time transmission control protocol, congestion control, queueing, deficit round robin, random early detection.

## I. INTRODUCTION

IT IS expected that next generation communication system will deliver an ever increasing amount of multimedia traffic. Therefore, the capability to transmit this traffic, complying with real-time quality of service (QoS) constraints, is commonly regarded as one of the major upcoming research challenges for the next years.

The present paper investigates transport layer solutions for real-time delivery, especially focusing on the choice of the queue discipline for multimedia flows. In this sense, it is important not only to achieve high efficiency of the queueing policy, but also to be able to correctly manage different kinds of traffic. In fact, multimedia traffic comprises several applications with different characteristics in terms of required QoS. Moreover, it is expected that multimedia traffic will coexist with other best effort data traffic in the same network operations.

Technological solutions to achieve real-time delivery over the internet include in particular the Realtime Transmission Protocol (RTP) and the Real-Time Control Protocol (RTCP) [1]. In [2], RTP/RTCP environment was introduced and implemented within the well-known ns-2 simulator [3].

In this paper, we proceed along the lines of [2], that is, the main functionality of RTP is modeled as involving the identification of payload, the generation of RTP packets,

and finally the introduction of RTP packets time stamps and sequence numbers. RTCP is used for inquiring the network status and getting feedback. The major advantage of RTCP is that it does not interact with RTP, but it can be used as a network management entity. Thus, RTP and RTCP can operate jointly as direct and feedback loop.

The data exchanged by the nodes through this mechanism enter a buffer queue at each intermediate receiver. One basic cause of delay in the transmission of multimedia traffic is actually the queueing delay at these buffers. Thus, when multimedia or real-time traffic is concerned, it is important to select the correct type of queueing policy in order to provide the users with the required QoS. To this end, different choices are possible. Previous existing work utilizes very simple queue disciplines, such as a basic DropTail policy [4].

When real time flows are considered, it is extremely important to monitor the traffic and performing dynamic resource allocation, as argued by [5]. In that paper, the authors investigate an RTP/RTCP environment and use the queueing delay parameter to tune the congestion control mechanism. However, they do not focus on a specific queue discipline; rather their approach tries to determine whether to increase the packet train by observing the current queueing delay. We take instead a similar approach but we focus on the queue discipline choice. Moreover, we take also the delay jitter into account. In fact, as described by [6], high values of the jitter are caused by the network congestion and/or inadequate queue disciplines. Therefore, selecting the proper queue discipline can play a tremendous role in the efficiency of real-time traffic over congested network.

In this paper, we propose to use, within the RTP/RTCP framework, a Deficit Round Robin (DRR) strategy. This is justified by several theoretical benefits, which we aim at validating in practice. To assess the validity of this approach, we implemented this queue policy within the ns-2 simulator and we quantify its performance in a test topology by means of a simulative campaign, evaluating several metrics of interest. In this way, we are able to verify that the proposed solution is properly able to fulfill QoS requirements of multimedia traffic.

The rest of this paper is organized as follows. In Section II, we review possible queue disciplines. Section III describes the RTP/RTCP environment, illustrates our proposed solution, based on the employment of DRR, and presents a test topology to evaluate it. In Section IV we discuss simulation results. Finally, Section V concludes the paper and reviews the overall significance of the proposed approach.

## II. REVIEW OF TRAFFIC MANAGEMENT METHODOLOGIES

Traffic management is the mechanism by which packets are received and transmitted by network routers. In particular, it requires a proper queue management to avoid congestion. Traffic control strategies can be used to maximize the network usage most favorably and/or to reserve the bandwidth for particular user or application. They can also be used to limit the total bandwidth at a known rate for particular user or application.

Traffic control may consist of some traditional elements as outlined in [7]; policing, classifying, marking, shaping, scheduling and dropping describe the mechanism by which packets can be limited, separated, altered, delayed, arranged (or rearranged) and discarded, respectively. In particular, in the following we will review these aspects under the general classification of *queue disciplines*, i.e., we will consider traffic management seen as the set of rules according to which users enter, are served, and exit the router node. The queue disciplines reviewed in the following are mostly classic examples and their detailed description can be found in many standard sources, such as [7], [8] and references therein.

FIFO (First In First Out) queueing is the most basic queue discipline. In FIFO queueing all packets are treated equally and are placed into a single queue. Then each packet is served in the same order. FIFO queueing offers the following benefits, as also presented in [7]: FIFO queueing places an extremely low computational load on the system as compared to other queue scheduling disciplines. The maximum delay is determined by the maximum depth of the queue. The behavior of a FIFO queue is very predictable, i.e., packets are not reordered. Short queue depth provides simple contention resolution for network resources. On the other hand, FIFO queueing also generally causes increased delay, increased jitter, and higher loss for real-time applications. This happens since a single FIFO queue impacts all flows equally, and the mean queueing delay for all the flows increases in case of congestion. A single FIFO queue does not allow routers to organize buffered packets. A bursty flow can consume the entire buffer space of a FIFO queue. Thus, during congestion, applications based on User Datagram Protocol (UDP) remain unaware to packet loss and continue transmitting packets at their usual rate.

Stochastic Fairness Queueing (SFQ) is another discipline, based on the more basic Fair Queueing (FQ). FQ is the base for a class of queue scheduling disciplines that are designed to provide assured fair access to network resources for each flow and prevent a bursty flow from consuming more than its fair share of output bandwidth [7]. SFQ is called stochastic because it does not really allocate a queue for the session; it utilizes a hashing algorithm which splits the traffic over a limited number of FIFO queues. Packets are then taken from the queues, which are selected in a round robin fashion. SFQ offers few advantages, as also presented by [7]: It requires fewer calculations while being almost perfectly fair. However, SFQ is required to deal with many queues, since there is one per each of the flows. Moreover, a disadvantage of SFQ is its low accuracy, since it is actually difficult to achieve high accuracy in practical implementations.

In classic Priority Queueing (PRIO) [7], packets are first classified by the system and then placed into different priority queues. Within each of the priority queues, packets are scheduled in FIFO order. Thus, the system contains an arbitrary number (set a priori) of classes of different priority. Among the pros of PRIO is worth mentioning that the traffic can be prioritized and high priority traffic gets fair access. However, when outgoing traffic rate is low and incoming traffic rate is high then the complexity of congestion avoidance is the major drawback.

Class Based Queueing (CBQ) [9], is a classful queue discipline where a hierarchy of classes is set to share the link capacity. CBQ divides user traffic into a multi-level hierarchy, e.g., based on any combination of Internet Protocol (IP) addresses, protocols and application types. To send a packet, CBQ must select first which of the classes is allowed to send. Such a decision is taken according to a Link-Sharing scheduler which adopts a Weighted Round Robin policy. Thus, the CBQ discipline includes shaping elements as well as prioritizing capabilities. However, CBQ is generally difficult to setup properly due to the presence of too many parameters.

When congestion occurs, the simplest approach to take is to accept packets until the queueing buffer is full. After that, any excess packet is simply dropped. In the following, this policy will be referred to as DropTail. To improve this approach, one can use Random Early Detection (RED). This queue discipline uses one queue and one dropping probability, which is used to determine whether to process the packets further or drop/discard them. This avoids penalizing burst of packets, which suffer a higher risk of being discarded with DropTail. One can also discard packets proactively, i.e., before the queue gets full. Such an approach, which generally behaves better against congestion, is called Active Queue Management. A possible extension of RED in this sense is known as RIO (RED In/Out) [10]. In this discipline, packets are marked either as IN or OUT packets. If the packet arrival rate exceeds a predetermined target rate, packets are marked as OUT. Otherwise, packets are marked as IN. IN packets are prioritized over OUT packets; therefore, OUT packets can be dropped in congestion states [11]. This approach can be further extended by considering the queue length as a key parameter for the level of congestion control. However, this requires to set a proper target value for the queue length, a task which is often simply left to the network operators, according to the type of the networks and the specific needs of the system. Finally, RED can be extended as Generic Random Early Drop (GRED) [7] where multiple drop priorities are set. It is worth noting that for multimedia application, RED or any of its extension may suffer from a major drawback, that is, they are not particularly careful in regard to real-time flows. In fact, multimedia content may be very sensitive to arbitrary packet dropping, which may result in a severe decrease of the QoS perceived by the users.

The Token Bucket Filter (TBF) [8] is a control mechanism based on the presence of “tokens” in a (virtual) bucket that determines when traffic can be transmitted. Tokens work as credit units for packet exchange. According to the presence of an adequate number of tokens in the bucket, the packets are shaped, marked, scheduled and/or discarded. A TBF queue

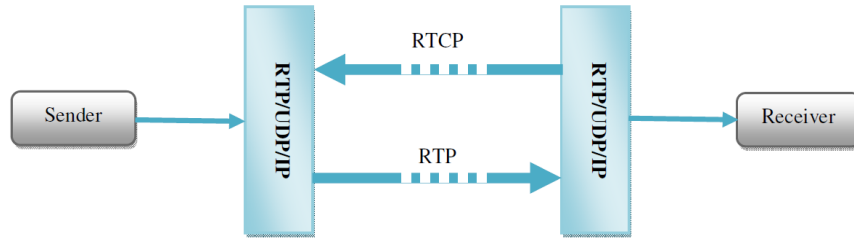


Fig. 1. The RTP/RTCP workflow

discipline is easy to implement, it is a practical method for rate limiting and allows fixed rate of flows. It is useful in managing network bandwidth and is generally precise, network and processor friendly. It can also be expanded to a hierarchical version to achieve a CBQ-like queue discipline, which translates into multiple buckets to be filled with tokens. Somewhat similar to TBF, the Deficit Round Robin (DRR) is an extension of Weighted Round Robin (WRR). DRR uses, instead of token and buckets, a deficit counter which can increase or decrease and is employed as a credit mechanism to determine whether the head packet can be withdrawn from a queue or not. The result is also a fair and efficient queue discipline [12]. In spite of its advantages, DRR is still an easily implementable queue discipline and has  $O(1)$  running time per packet. Moreover, it can handle various sizes of packets, which makes it suitable for heterogeneous multimedia and other realtime applications.

### III. THE RTP/RTCP ENVIRONMENT AND THE PROPOSED SOLUTION

RTP was developed by the Audio-Video Transport Working Group. It uses regimented packet format for multimedia contents that is, audio and video [1]. It was designed for multicast applications. RTP provides the following services: Identification of payload, time stamping, sequence numbering, and delivery notification. It provides end-to-end network transport functions, but is not responsible for guaranteed QoS. Therefore, RTP and RTCP are most frequently used in a joint manner, because RTP is used to transport multimedia data and RTCP is employed for monitoring the network QoS [13]. A scheme depicting this interaction is represented in Fig. 1.

For multimedia sources, adaptive transmission rate algorithm is introduced in [14] using a TCP friendly rationale. The algorithm considers the maximum transmission rate, minimum transmission rate and the granularity. In the adaptive algorithm, the sender changes its transmission rate according to the adaptive algorithm schema. The authors of [14] have simulated TCP-friendly and constrained TCP-friendly flow control and their results proved that the constrained TCP-friendly version reaches a higher degree of fairness than the plain TCP-friendly one. This combination of RTP for real-time flows and RTCP to monitor the QoS of the network was also used in [15], where new feedback control mechanism for video transmissions are presented. By means of simulation, this contribution compares the packet losses of UDP flow against the UDP\_RTP flow. Results proved that UDP\_RTP can improve the video transmission.

RTCP was also used in [16] to get the network information and tune the system accordingly for real-time traffic. But the emphasis of this paper is on multimedia traffic management in ad hoc networks. Authors are also using RTCP for getting end-to-end feedback information about the packet loss performance as well as delay jitter. RTP/RTCP protocol suite is well known for getting feedback from the receiver and it is also utilized by [17] to exemplify the real-time traffic flow for unicast and multicast environment. In the present analysis, we are also focusing on multicast scenario, particularly for real-time traffic. The authors of [17] present a vast survey of multimedia synchronization and the main technique is exemplified by the existing RTP/RTCP suite. Congestion Control is a challenging issue particularly for real-time traffic, this argument is also supported by [18], where however it is proposed to use RED queueing discipline for congestion control or for dropping the packet at the time of congestion. However, this may violate the need for real-time traffic to receive fair bandwidth allocation and especially the requirement for lossless delivery of information.

Therefore, we also take a RTP/RTCP environment as the starting point of our evaluation. However, as will be argued in the next sections, we stress the importance of an efficient queueing discipline at the buffers. For this reason, in the following we review and discuss this point. Further, we comparatively evaluate different choices in this respect.

The simplest queue discipline, called DropTail, follows a very basic policy, i.e., it treats all the packets equally in a single queue, and each packet is served in the same order as received; that is, this is simply a FIFO approach which adopts the dropping rule that all the packets exceeding the buffer capability are discarded when the buffer is full. As explained in Section II, this approach involves very low computational complexity and easily predictable behavior [4]. However, it also causes significant drawbacks, i.e., increased delay, jitter and packet losses for real-time applications. Further, entire bursts of packets might be discarded and a significant queueing delay may increase the overall network congestion.

The buffer management scheme used in previous RTP/RTCP framework, such as [2], is simply to employ Droptail at the sender and receivers ends, whereas RED queue discipline is used at the router ends. The RED queueing policy uses priority levels to drop packets; however, since the RED queue is applied at the link between the two routers, therefore it treats all the flows identically. Actually, RTP flows should be given some priority over non real-time flows. Therefore, considering the jitter and packet losses as QoS metrics, jit-

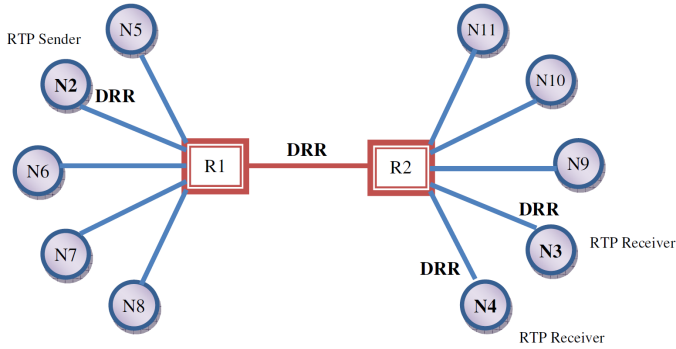


Fig. 2. Evaluation network topology

ter was introduced due to the employment of Droptail and packet loss was increased due to the packet dropping by the RED queue. Alternatively, traffic management strategy can be applied to this scenario for real-time flows, so that real-time can be given better QoS over non real-time traffic. This will result in enhancing the QoS in terms of increased share of bandwidth and decreased jitter and packet losses for real-time flows. In the proposed traffic management strategy, Deficit Round Robin (DRR) is used as a queue discipline for the source and destinations of real-time flows as well as for the link between the routers.

The network topology is described in Fig. 2. In the above figure, N2 is the node 2, which the sender of RTP flows, R1 and R2 are the edge routers and N3 and N4 are the receiver nodes. Instead of simple DropTail strategy, we have implemented a DRR queue discipline; in this manner, the jitter delay and the packet loss rate are significantly decreased, whereas the bandwidth utilization for real-time flows is improved. Results have been proven through simulation, described in the following section.

#### IV. PERFORMANCE EVALUATION

We utilized the ns-2 environment [3] to perform a simulation campaign. Focusing on a topology as previously described, we implemented the DRR policy for its use at the queueing buffers. The RTP sender and receiver use the DRR queue, similarly, DRR is also used instead of RED queueing system, but for all the other nodes Droptail is used. The queue length is set to 50 packets for each of the queue, whereas for the DRR queue between the routers has the capacity of 100 packets. Further, RTP-RTCP multicast environment is arranged in such a way that node 2, node 3 and node 4 join the multicast group, where node 2 is the transmitter of RTP traffic whereas node-3, node-4 are the RTP traffic receiver.

We simulated 100 seconds of transmissions and we evaluated the resulting system performance in terms of three different metrics: bandwidth allocation, jitter, and smooth loss. We compare our results with those reported in [2], where DropTail and RED traffic management strategy was adopted; thus, the curves referring to this approach are labeled “without traffic management.” We will show instead that in our proposed solution, buffer management plays very important role

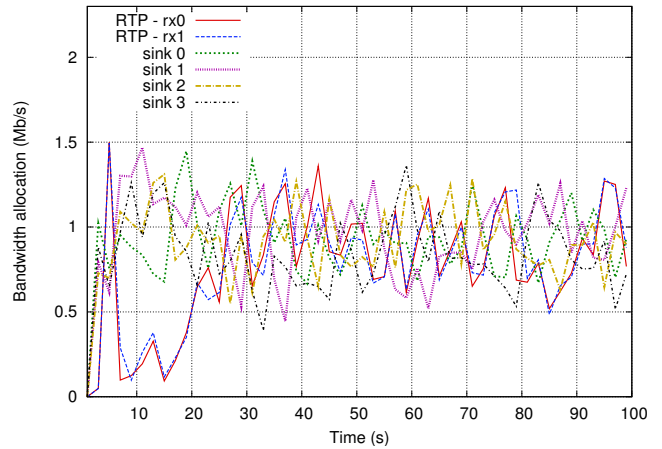


Fig. 3. Bandwidth allocated to the flows without traffic management

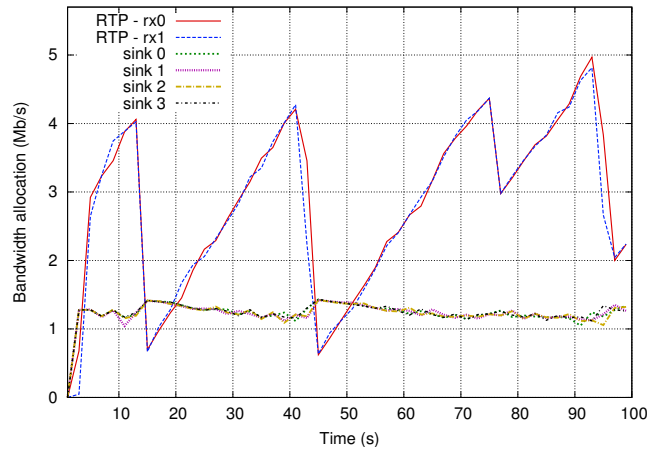


Fig. 4. Bandwidth consumptions of the flows with traffic management

in terms of bandwidth allocation and other traffic management aspects; for this reason, we refer to our solution as “with traffic management” in the graphs. The following subsections detail the analysis for each of the investigated metrics.

##### A. Bandwidth Allocation

Figs. 3 and 4 compare the bandwidth allocation of the flows, for the two cases without (previous solution) and with traffic management (proposed solution), respectively. In Fig. 3, it is observed that all the flows get almost equal (and not very high) amount of bandwidth. No preference is given to real-time flows. Finally, the bandwidth allocations of all flows fluctuate considerably, and in a very variable manner from flow to flow.

Conversely, when we apply our proposed traffic management strategy, there is a great difference of bandwidth consumption between the real-time flows and the others. Fig. 4 shows that the two RTP flows get a higher amount of bandwidth than what reported in Fig. 3, and also they receive a better allocation than the other flows, which is a sign that real-time traffic is correctly provided with better QoS than best effort traffic. It is also worth observing that the oscillations are significantly reduced. Still, they are not avoided; observe, for example, that the bandwidth assigned to real-time flows drops



Fig. 5. Jitter analysis

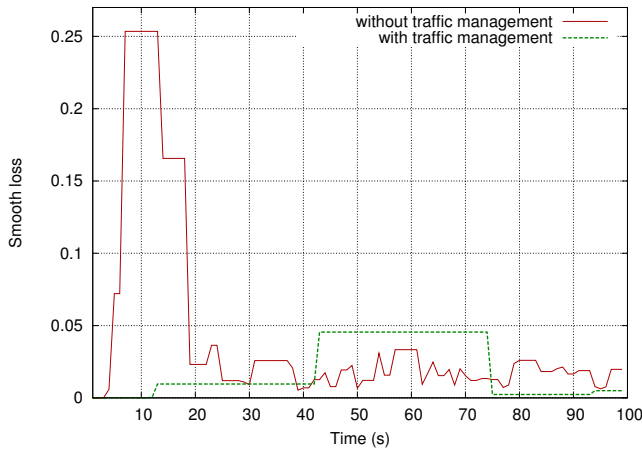


Fig. 6. Smooth loss analysis

in many points, due to congestion. Yet, the plots of Fig. 4 highlight a more regular behavior, that is, similar kinds of flow enjoy similar QoS at the same time.

### B. Jitter

Thanks to its better traffic management capability, our proposed strategy is also able to effectively decrease the jitter values as compared to the previous solution with Droptail and RED. Fig. 5 shows a comparison of the jitter values for multimedia nodes between the solution of [2] and new proposed solution with traffic management for RTP/RTCP environments. We distinguish jitter values for our proposed strategy “with traffic management” and the implementation of [2] which is referred to as “without traffic management.”

The plot emphasizes that jitter is significantly decreased when our traffic management technique is applied on RTP/RTCP environment. It is also important to notice that the jitter values decrease and become smoother as time goes by, but still the curve of the proposed queue discipline stays considerably below the one without traffic management features.

### C. Smooth Loss

Real-time traffic is very sensitive and requires more priority as compared to non real-time flows. Thus, by applying some traffic management strategies, real-time traffic can be given precedence over non real-time. Thus, when real-time traffic is prioritized, packet losses are likely to be decreased for real-time flows. This is investigated in Fig. 6, which represents the smooth losses [19] comparison between the previous RTP/RTCP solutions with our solution containing the traffic management component. The smoothing factor, i.e., the weight coefficient of the loss rate value evaluated in the previous time instant, is set to  $\alpha = 0.9$ . Again we report two plots, “with” and “without” our traffic management approach.

The considerably better behavior in terms of packet losses is expected, depending on the application in use, to reflect in an enhanced QoS for real-time flows.

## V. CONCLUSIONS AND FUTURE WORK

We considered an RTP/RTCP environment where RTP is used to transmit multimedia and realtime data and RTCP is employed in conjunction with RTP to monitor the network statistics and maintain the overall end-to-end network QoS in a feedback-based manner. In related research work, RED queue discipline, or even simpler DropTail, are used to manage multimedia traffic. However, the adoption of different queue disciplines can improve the transmission of RTP traffic. For instance, in the considered scenario, the introduction of Deficit Round Robin significantly improves the performance in terms of bandwidth allocation, jitter, and packet losses for real-time flows.

The advantages of DRR have been proven by means of simulation results comparing existing solutions with a proposed approach based on DRR in terms of all these performance metrics. The correct choice and setup of queuing policy at the network nodes has been proven to be key for meeting QoS constraint of real-time multimedia traffic.

Possible extensions of the present work include the analysis of similar traffic management strategies in different topologies and/or with different combinations of existing queue disciplines. Moreover, it is also possible to envision the application of similar approaches to RTP/RTCP environments realized over wireless networks.

## REFERENCES

- [1] H. Schulzrinne, S. Casner, R. Fredrick, and V. Jacobson, “RTP: A transport protocol for real time applications,” *RFC 3550*, July 2003.
- [2] C. Bouras, A. Gkamas, and G. Kioumourtzis, “Extending the functionality of RTP/RTCP implementation in network simulator (NS-2) to support TCP friendly congestion control,” *Proc. of SIMUTools*, March 2008.
- [3] K. Fall and K. Varadhan, *ns Manual*.
- [4] B. Hubert and B.V. Netherlabs, “Linux HowTo,” [online] available on: [lartc.org/howto/lartc.qdisc.classless.html](http://lartc.org/howto/lartc.qdisc.classless.html), accessed on 06-07-10.
- [5] S.-S. Wang, H.-F. Hsiao, “TCP-friendly congestion control for the fair streaming of scalable video,” *Computer Communications*, to appear (published online: Feb. 2010).
- [6] S. Karapantazis, F.-N. Pavlidou, “VoIP: A comprehensive survey on a promising technology,” *Computer Networks*, vol. 53, pp. 2050–2090, 2009.
- [7] SourceForge, Inc., “Queueing Discipline HowTo,” [online] available on: [Linux.com/base/ldp/howto/Traffic-Control-HOWTO/](http://Linux.com/base/ldp/howto/Traffic-Control-HOWTO/), accessed on 20-09-09.

- [8] A. S. Tanenbaum, *Computer Networks*. 3rd Edition, Prentice-Hall, 1996.
- [9] P. Kemper, D. Müller, and A. Thümmel, "Combining response surface methodology with numerical methods for optimization of Markovian models," *IEEE Trans. on Dependable and Secure Computing*, vol. 3, pp. 259–269, Sept. 2006.
- [10] Y. Hori, T. Ikenaga, and Y. Oie, "Queue Management of RIO to achieve high throughput and low delay," *IEICE Transactions on Communications*, vol. E84-B, No. 6, pp. 1473–1479, 2001.
- [11] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transaction on Networking*, vol. 1, pp. 397–413, Aug. 1993.
- [12] M. Shreedhar and G. Varghese, "Efficient Fair Queueing using Deficit Round Robin," *IEEE/ACM Transaction on Networking*, vol. 4, no. 3, 1996.
- [13] N. Baldo, M. Kampmann, U. Horn, and F. Hartung, "RTCP feedback based transmission rate control for 3G wireless multimedia streaming," *Proc. IEEE Personal, Indoor and Mobile Radio Conference (PIMRC)*, Barcelona, Spain, September 2004.
- [14] C. Wanxiang and L. Zhenming, "The research on constrained TCP-friendly congestion control," *Proc. Intern. Conf. on Info-tech and Info-net*, Beijing, 2001.
- [15] L. Xu and S. Ai, "A New Feedback Control Strategy of Video Transmission Based on RTP," *Proceedings of IEEE Conference on Industrial Electronics and Applications*, pp. 1–4, May 2006.
- [16] C. Liu, Y. Shua, L. Zhanga, Z. Zhaoa and X.-Y. Li, "A multi-layer experimental study of multimedia and QoS communication in wireless mesh networks," *Perv. and Mob. Computing*, vol. 5, pp. 19–36, 2009.
- [17] F. Boronat, J. Lloret, M. Garcia, "Multimedia group and inter-stream synchronization techniques: a comparative study," *Information Systems*, vol. 34, pp. 108–131, 2009.
- [18] C. Bouras, A. Gkamas, G. Kioumourtzis, "Smooth multicast congestion control for adaptive multimedia transmission," *Next Generation Internet Networks*, pp. 146–152, 2008.
- [19] B. Piyatamrong and P. Piyakhun, "Smooth loss with variability smoothing factor to increase bandwidth utilization," *Proc. IEEE PACRIM Conference on Communications, Computers and Signal Processing*, vol. 2, pp. 493–494, 2001.