Comparing Application Performance on Distinct IP Packet Scheduling Configurations

Miguel F. de Castro[1*], **Dominique Gaïti**[2], **Abdallah M'hamed**[1], **Mauro Oliveira**[3]

[1]RST - Institut National des Télécommunications
9 rue Charles Fourier - 91011 - Evry Cedex - France

[2]LM2S - Université de Technologie de Troyes
12 rue Marie Curie - 10010 - Troyes Cedex - France

[3]LAR - Centro Federal de Educação Tecnológica do Ceará
Av. 13 de Maio, 2081 - Fortaleza/CE - Brasil

Resumo. *Convergência de serviços é um tema que tem despertado bastante interesse no mundo Internet. Apesar do protocolo IP não ter sido projetado para tal ambiente, sua popularização tem tornado inevitável seu emprego como suporte ao tráfego multimídia. Neste sentido, o IETF tem tomado providências com o intuito de prover a Internet com mecanismos ágeis, a fim de tentar responder eficientemente a esta nova realidade. Entretanto, o comportamento do trafego Internet é imprevisível e de difícil modelagem, o que pode levar a mudanças de desempenho dos mecanismos de gerência. O objetivo deste artigo é mostrar o comportamento simulado de serviços de rede quando submetidos a diferentes algoritmos de escalonamento de pacotes. Os efeitos do aumento progressivo de carga de trafego UDP sobre o desempenho da gerência também são avaliados.*

Abstract. *The Internet Protocol (IP) is the most serious candidate to receive network multimedia services convergence. Although IP was not primarily designed to isochronous medias, its rapid popularization propitiates its broad use for these services. Some efforts have being deployed to change this reality, such as* IntServ *and* DiffServ*. However, Internet traffic behavior is still unclear and unpredictable, and this unpredictability may compromise management mechanisms' behavior inside the network. The objective of this article is to show simulated behavior of distinct network services when submitted to different scheduling mechanisms. We also try to evaluate the effects of growing UDP traffic load ratios on network performance.*

1. Introduction

The Internet is willing to receive multimedia service convergence. There is an increasing availability in the Internet of services such as Live Radio and TV, Jukeboxes, On Demand Video Broadcasters, Video-Conferencing, Telephony, etc. These services are different from others because of their high requirements on Quality of Service (QoS) from the network in order to offer satisfactory results to users.

Although the Internet was not primarily designed to this kind of usage, its rapid popularization propitiates its broad use for these services. This incompatibility is due

*Supported by CAPES.

to the fact that Internet does not offer *a priori* definition and maintenance of Quality of Service to applications. This reality is in way of change after the deployment of new technologies like Service Integration (*IntServ*) [1] with resource reservation protocols, such as RSVP (Resource reSerVation Protocol) [2] and Service Differentiation (*DiffServ*) [3]. The best-effort philosophy over Internet has overcharged the end entities of the flow with the responsibility of QoS, flow and congestion control.

There is a strong trend to integrate various infrastructures such as fix and mobile telephony, wireless and Internet in only one infrastructure capable to offer customized support to different issues for services and its requirements. These services will follow specific billing rules, once users are willing to pay more to have a better service. This trend, based on IPv6 [4] and "IPng (IP New Generation)" [5], is named "*IP all-the-way*", and is being studied and developed by research centers, industry and standardization organisms, such as IETF, to be the basis for the next generation Internet.

There is still work to do for QoS management to new generation Internet. A lot of lessons were passed from ATM (Asynchronous Transfer Mode) [6] and other former efforts to grant QoS. Some of these lessons were kept (label switching, reservation schemes, virtual paths, ABR, etc.) while other characteristics were left behind. The lack of internal QoS mechanisms in the Internet carried the responsibility for QoS control to the edges of the connections. Thus, source and destination must negotiate to control their communications.

It is not yet clear to know how different network services behave under QoS control mechanisms. There are several propositions of such mechanisms, and each one has its advantages and disadvantages. It is not easy to choose only one reasonable mechanism that can give good response on different traffic scenarios. Nevertheless, such mechanisms are not always easy to implement or to configure.

The aim of this work is to show how simulated network services behave on different traffic management configurations. Network traffic is evaluated when being submitted to different combinations of scheduling techniques.

Our simulated environment evolves several different types of applications, including short, medium and long-term connections, with responsive or non-responsive behaviors. During simulation, traffic behavior is modified in order to evaluate the impact of such changes over network performance. Finally, we try to map traffic behavior and characterization with best-fit combinations of network management mechanisms and configurations.

The remainder of this paper is organized as follows. Section 2 outlines the Internet evolution towards the converged services. On Section 3, the efforts of IETF (Internet Engineering Task Force) to provide IP with QoS support are briefly described. Section 4 overlines the QoS management tasks on IP networks, including scheduling and queue management algorithms. On Section 5, we present the experimentations that were performed in order to try to find a relationship between management mechanisms' choice/configuration and traffic profile. Section 6 shows and comments the obtained results. Finally, in Section 7, we present our conclusions about this experience, and discusses the next steps to this work.

2. QoS Support in the Internet

The Internet Protocol (IP) [7] was not primarily designed to support services that have different requirements. The Internet based on IP is becoming the most accepted standard to receive service convergence. As each service may have its own requirements – or *Quality of Service* (QoS) – it is now mandatory to prepare IP to deal with this heterogeneity.

Network management role in this new generation of Internet is still more important than ever. After the first big collapse threat, the Internet was enhanced by TCP's congestion avoidance and flow control. With service convergence, the challenge is being to create management mechanisms to accommodate several service behaviors under a common infrastructure.

Moreover, as emerging streaming media applications in the Internet primarily use UDP (User Datagram Protocol) transport, it is still harder to enforce more strict control in order to avoid network congestions. These new UDP applications generate large volumes of traffic which are not always responsive to network congestion avoidance mechanisms, causing serious problems on fairness [8]. Hence, if no control is done, such unresponsive flows could lead to a new congestion collapse [9]. Some ISP networks that use ATM as layer-2 technologies can solve this problem by mapping UDP traffic to ABR (Available Bit Rate) service, but this is not a pure IP solution.

IETF research is already worried about emerging UDP applications, and a workgroup to propose a solution to this problem has been lately created. This workgroup is responsible to develop the Datagram Congestion Control Protocol (DCCP) [10], which is intended to control unreliable flow of datagrams, with acknowledgements, and reliable negotiation of options, including negotiation of a suitable congestion control mechanism. DCCP will also be compatible with Explicit Congestion Notification (ECN) [11]. These efforts must give datagram transport a *TCP-friendly* behavior, as expected by congestion control mechanisms.

3. Research in IETF

Although the Internet now runs faster and is increasing in size, its basic architecture remains unchanged since its early days. The Internet still operates as a datagram network, where each packet is delivered individually through the network. Delivery time of packets is not guaranteed, and packets may even be dropped because of congestion inside the network. This unpredictability does not mesh well with new applications such as Internet telephony or digital video conferencing, which cannot tolerate delay jitter or loss of data in transmission [12].

To overcome these problems, the Internet Engineering Task Force (IETF) has developed new technologies and standards to provide resource assurance and service differentiation in the Internet, under the umbrella term Quality of Service (QoS). IETF proposes two architectures to address QoS management over IP: *IntServ* [1] and *DiffServ* [3].

IntServ is a service model to provide fine-grained assurances to individual flows. At present, there are two services defined in the model: Guaranteed Service and Controlled Load Service. IntServ requires state information in each participating router and, if this state information is not present in every router along the given path, QoS guarantees cannot be ensured. Usually, but not necessarily, Integrated Services are associated with **R**esource re**S**er**V**ation **P**rotocol (RSVP) [2] signaling. Signaling processing times and the need for storing per flow information in each participating node are believed to lead to scalability problems, particularly in the core of the Internet [13].

DiffServ is an architecture for implementing scalable service differentiation in the Internet. This architecture achieves scalability by aggregating traffic classification state which is conveyed by means of IP-layer packet marking using the DS field [14]. Packets are classified and marked to receive particular per-hop forwarding behavior on nodes along their path. Network resources are allocated to traffic streams by service provisioning policies which govern how traffic is marked and conditioned upon entry to a differentiated services-capable network, and how that traffic is forwarded within that network. A wide variety of services can be implemented on top of these building blocks.

Both architectures proposed by IETF try to address the IP architecture's adaptation to support QoS. Each one has its qualities and limitations. IntServ, for example, has a serious drawback on scalability, but offers good guarantees of QoS to services. Although less critical than in IntServ, DiffServ has also some level scalability problems once complexity is pushed to the edges of the network. However, core routers can be simple and fast enough to give better performances inside the network, giving DiffServ the preference to be the QoS architecture to large networks.

DiffServ per-hop behaviors are implemented by a combination of management mechanisms that are made available in the router. These mechanisms constitute the basic elements upon which an IP network with service-differentiation capabilities may be built. The choice of these mechanisms as well as their tuning are essential to issue good performance [15]. Such mechanisms include scheduling algorithms and queue management schemes. Some of these mechanisms will be explained on next sections.

4. Internet QoS Management

QoS Management on an IP router consists basically of two tasks [9]: *queue management* and *scheduling*. Queue Management deals with the length of packet queues by dropping packets when necessary or appropriate, while scheduling algorithms determine which packets to forward next. Some queue management algorithms and scheduling schemes are briefly described in the following subsections.

QoS Management solutions may employ different queue management and scheduling algorithms in order to support service requirements. Operation may be based on single or multiple queues, with or without differentiated treatment to network services.

Round-Robin (RR) [16] and its derivative Weighted Round-Robin (WRR) serve the packet flows in a round-robin fashion. Each flow served by any of the algorithms has a number of bits that are transmitted in each round-robin. When the scheduler is RR the number of bits is equal for all the flows and, in the case of the WRR scheduler, the flows can receive different amount of service within each pass. The differentiation allows WRR to atribute priorities to the served flows [17]. Cisco implemented a variant of WRR scheduler for IP routers, named Modified WRR (MWRR) [18], which is also object of comparison in this work.

A modification of the WRR algorithm called Deficit WRR (DWRR) [19] enables the served flows to save the service they do not receive in a round-robin as a result of the packet size variability. That is, if a packet from a flow being currently served is so long that its transmission would exceed the service quantum in this round-robin, the resulting amount of service undelivered to the flow is saved until the next round-robin and is added to the service quantum.

The Modified Deficit Round-Robin (MDRR) scheduling algorithm is based on the Deficit Round-Robin (DRR) [20] mechanism which implements a number of queues that are served in a round-robin fashion. For DRR each queue has assigned to it a configurable value called a *service quantum*. A service quantum provides a measure of how much traffic should be handled from the queue in each round. Packets from that queue are serviced until their cumulative length (byte count) exceeds the service quantum. A deficit counter, which is a memory mechanism designed to enhance fairness and packet size independence, is used as a credit mechanism. The deficit counter value is added to the service quantum to determine the measure of service available for each queue during each round [21].

MDRR extends the DRR mechanisms by including for each set of class-of-service queues a low-latency, high-priority (LLHP) queue designed to handle special traffic (*e.g.* voice) different from the other queues. Except for the LLHP queue, MDRR services all queues in round-robin fashion. RED or WRED can be configured for each of the MDRR queues, specifying a discrete RED/WRED profile for each.

The WFQ (Weighted Fair Queuing) supports the fair distribution of bandwidth for variable-length packets by approximating a generalized processor sharing (GPS) system. While GPS is a theoretical scheduler that cannot be implemented, its behavior is similar to a weighted bit-by-bit round-robin scheduling discipline. In a weighted bit-by-bit round-robin scheduling discipline, the individual bits from packets at the head of each queue are transmitted in a WRR manner. This approach supports the fair allocation of bandwidth, because it considers packet length. As a result, at any moment, each queue receives its configured share of output port bandwidth. Although transmitting packets from different queues one bit at a time can be supported by a TDM network, it cannot be supported by a statistically multiplexed network. However, if one can imagine the placement of a packet reassemble at the far end of the link, the order in which each packet would eventually be fully assembled is determined by the order in which the last bit of each packet is transmitted. This is referred to as the packet's finish time.

Custom Queuing (CQ) is a solution proposed by Cisco as a method of guaranteeing bandwidth for various protocols or incoming interfaces. This is done by assigning protocols or interfaces to one of 16 possible queues. These queues are then handled in a round-robin fashion. One can define how much is transmitted from each queue at a time so that some queues can transfer more than other queues meaning that they will be able to have a greater share of the bandwidth than other queues. There is one queue that cannot be changed, called *queue 0* (equivalent to LLHP queue in WFQ). This queue will be emptied before all others. It handles system packets such as *keepalives*.

Choosing a fair set of mechanisms to implement a per-hop behavior in a service-differentiated network is not an easy task. The diversity of mechanisms available turn the number of possible combinations too big and a bad choice can lead to a poor performance to very constrained QoS applications. Furthermore, even when a good choice is made, finding a good configuration to these mechanisms is still a challenge.

5. Some Experiments

The objective of this article is to observe management mechanisms' behaviors under different network traffic scenarios. We try to make a link between management mechanism choices and some traffic profiles in which performance is the best-fit.

We choose to use OPNET simulation tool to run our experiments. This choice is justified by the simplicity and the flexibility of the tool to simulate the different scenarios we planned, and because of the quality and accuracy of the statistics offered by this tool.

An enterprise network may describe our environment of interest, where several clients (400 terminals) ask for services from a number of servers located on the other side of a wide-area serial link.

The topology of our experiments is composed by two QoS-enabled IP routers, interconnected by a 4 Mbps link, with propagation delay of 60 ms. In one side, some servers are made available. We assume that servers' performances are good enough to avoid processing bottlenecks. In the other side, we have several PC's enabled to use any of the services offered by the enterprise servers. Our experimental enterprise network is illustrated in Figure 1.

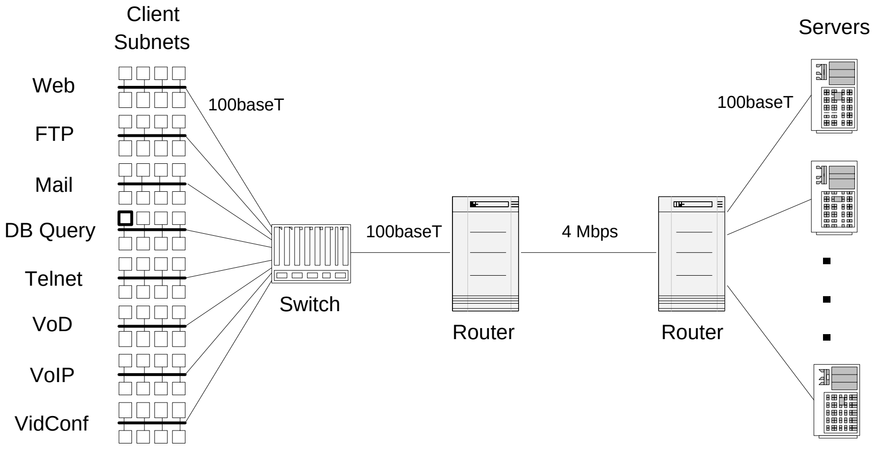


Figure 1: Simulated Topology.

Clients have the possibility to use one or more of the following applications:

Web	Simple World-Wide Web browsing.
FTP	Medium-sized file downloading.
Mail	Transfer of Simple Mail Transfer Protocol (SMTP) messages.
DB	Server-centered database queries.
Telnet	Remote terminal emulation.
VoD	Server-centered streaming media to medium-quality video on-demand.
IPTel	IP-based telephony over the Internet with medium quality.

Applications are configured as follows. The *Web* application is characterized by a sequence of HTTP 1.1 connections, where each connection is one page load. The mean time between page loads is 15 seconds (varying exponentially). Each object contained in one page has between 500 and 20000 bytes of length. Each page has an average number of 5 objects (exponential distribution).

The *FTP* application has always download operations (GET). Each client downloads objects with 1 MByte length (exponentially distributed), with a mean time between downloads of 10 seconds (exponentially distributed).

The *Mail* application is configured as follows. Each client sends and receives messages. Message size is average 10 Kbytes (varying exponentially). The mean message interarrival time is 40 s (exponential), and the mean message interdepart time is 20 s (exponentially variable).

The *DB* application is characterized by client making small queries to database servers, and receiving results of mean size 32 kbytes. The mean time between queries is 12 s (varying exponentially).

Telnet application has clients that submit a heavy load of commands, with a mean interarrival time of 30 seconds (normal distribution with variance of 5 s). The average length of each command sent to the server is 25 bytes (normal distribution with 25 bytes of variance), and the traffic returned to the client follows a normal distribution of mean outcome 60 bytes and 144 bytes of variance, varying following normal distribution.

The *Video on Demand* (VoD) service is characterized by servers that provide UDP streaming media to a medium-quality video of average 10 fps (frames per second). Each frame's average size is 2.5 Kbytes, exponentially distributed. Each video stream generates 200 Kbps traffic from the server to the client. The traffic is characterized by almost no variation along time. As UDP is employed, this service has no congestion control mechanism.

IPTel service is a two-way UDP non-responsive flow. It emulates a GSM-quality telephony conversation over IP, without silence detection. Each connection endpoint produces a 35 Kbps constant flow.

All configurations for applications followed OPNET simulator suggestions on default values. Multimedia applications employ UDP as transport protocol. The service adopted in our experiments does not include parallel flow control and congestion avoidance mechanisms to UDP flows, what make them *Non-Responsive* flows. TCP implementation is based on New Reno, with some enhancements, like Window Scaling, Selective ACK (SACK), ECN capability, and Nagle's algorithm.

Our experimentation is based on a differentiated service environment, and compares the following scheduling algorithms: Custom Queuing (CQ), DWRR (Differential Weighed Round-Robin), MWRR (Modified Weighted Round-Robin), MDRR (Modified Differential Round-Robin) and WFQ (Weighted Fair Queuing). We have chosen a configuration that would have the same effects on priorities of services in all tested scheduling algorithms. These configurations define the weights to each service as well as the queue management scheme to be applied to each service queue. Configurations[1] used in this ex-

[1]To RED configuration, we defined $Min_{th} = 0.8 * Max_{th}$. The parameter *Weight* is used in WFQ,

periment are described on Table 1. During the simulation time, traffic pattern is changed in order to test management mechanisms' performances. These changes are made by varying the contingent of users (clients) to each one of the offered services.

Parameter	*Web*	*FTP*	*DB*	*Mail*	*Telnet*	*VoD*	*IPTel*
Weight	10.99	8.54	30.52	8.54	12.21	10.99	30.52
Byte Count	9000	7000	25000	7000	10000	9000	25000
Max Queue Size	100	200	150	250	100	75	50
RED Parameters	RED+ECN	FIFO	RED+ECN	RED+ECN	RED	FIFO	FIFO
Queue Category	Default	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	LLHP

Table 1: Scheduling Algorithms' Configurations.

In these experiments, we want to test the behavior of management mechanisms under different charges of UDP non-responsive traffic. The number of clients that uses each service at the same time is configured, and router behavior – in terms of network overall performance and application-specific measures – is evaluated. UDP traffic loads on scenarios represent 9.63, 19.3, 28.9, 38.5, and 48.1 % of output link. These situations' configurations are shown on Table 2.

App	*Sit1*	*Sit2*	*Sit3*	*Sit4*	*Sit5*
Web	35	35	35	35	35
FTP	15	15	15	15	15
Mail	25	25	25	25	25
DB	30	30	30	30	30
Telnet	20	20	20	20	20
VoD	2	4	6	8	10
IPTel	1	2	3	4	5

Table 2: Clients' Configuration.

In general, all these situations show a high percentage of network utilization. Network services in these experiments present sometimes performances that are behind of average. Although these measures may not correspond to real-life, it is important to evaluate network behavior on critical situations in order to better model network management functions.

6. Results and Discussion

Each of the five traffic situations (scenarios described on Table 2) was simulated and produced 15-minute logs. Each scenario simulation was repeated to test each one of the five scheduling algorithms (CQ, DWRR, MDRR, MWRR, and WFQ). Due to space constraints, we will show plots for only three of the five simulated situations: Sit1, Sit3 and Sit5.

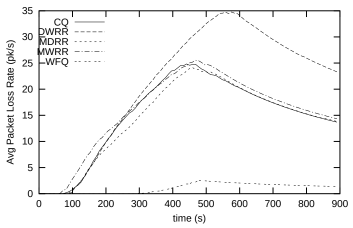
DWRR, MWRR, and MDRR, while *Byte count* is used in CQ. LLHP is only applied to WFQ and CQ.

Packet Loss

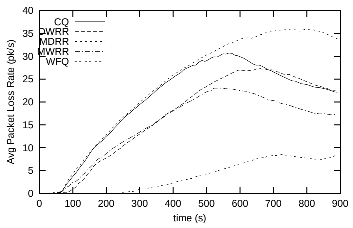
Results in Figure 2 shows that WFQ obtains the smallest overall packet loss rate. As UDP traffic load increases, WFQ maintains the lower loss rate. On the other hand, UDP traffic load influenced DWRR, increasing packet loss rate. CQ shows a relatively low loss rate when UDP load is low, but its performance decreases on higher UDP load.

Web service

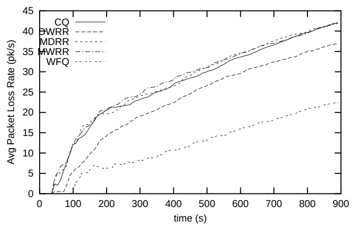
For Web service (Figure 2), Round-Robin-based algorithms had the smaller page load times. The algorithm that has presented the worst performance on Web in all situations is CQ. In all situations, MDRR and WFQ presented close performances, just as MWRR and MDRR.



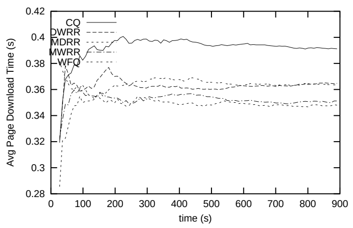
(a) Pkt Loss Sit 1



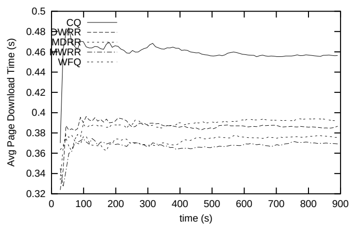
(b) Pkt Loss Sit 3



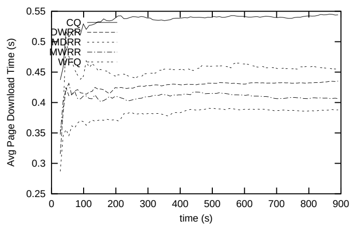
(c) Pkt Loss Sit 5



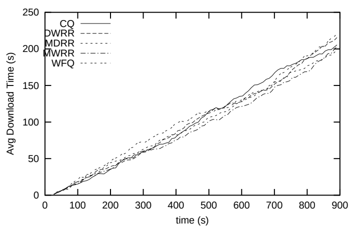
(d) Web Sit 1



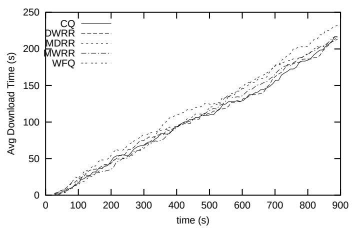
(e) Web Sit 3



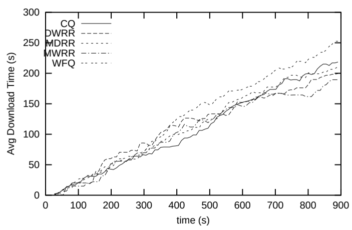
(f) Web Sit 5



(g) FTP Sit 1



(h) FTP Sit 3



(i) FTP Sit 5

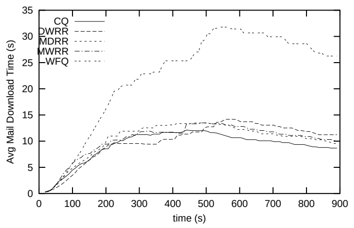
Figure 2: Packet Loss, Web and FTP Performances.

FTP service

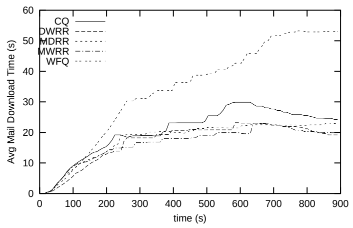
Mean FTP transfer time was not influenced by UDP load, as shown in Figure 2. However, we can see that a bigger discrepancy among all algorithms is observed when UDP load is higher. WFQ presented the worst download time in almost all situations, while Round-Robin algorithms have presented the best performances.

Mail service

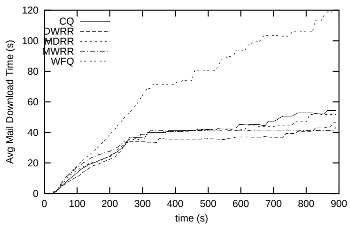
On Mail service (Figure 3), the worst message load time was obtained by adopting WFQ. Initially, CQ has presented the best performance, but as UDP traffic load increased, Round-Robin algorithms came up with better performances.



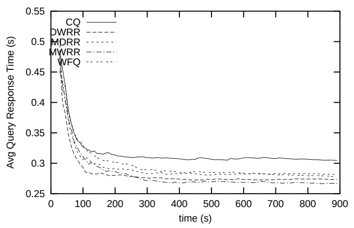
(a) Mail Sit 1



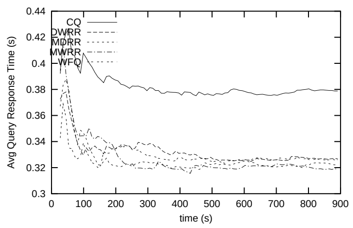
(b) Mail Sit 3



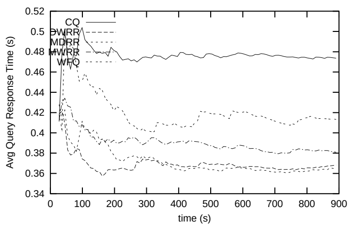
(c) Mail Sit 5



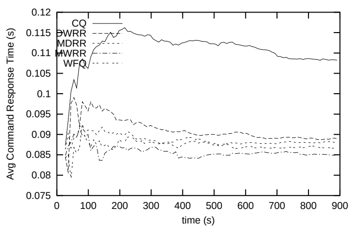
(d) DB Sit 1



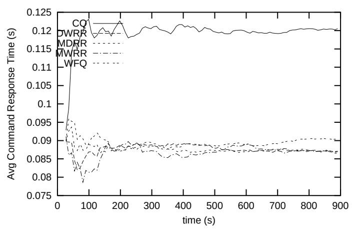
(e) DB Sit 3



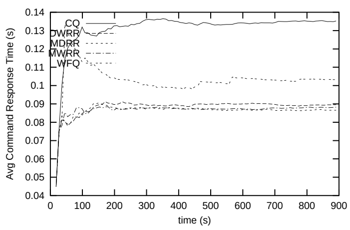
(f) DB Sit 5



(g) Telnet Sit 1



(h) Telnet Sit 3



(i) Telnet Sit 5

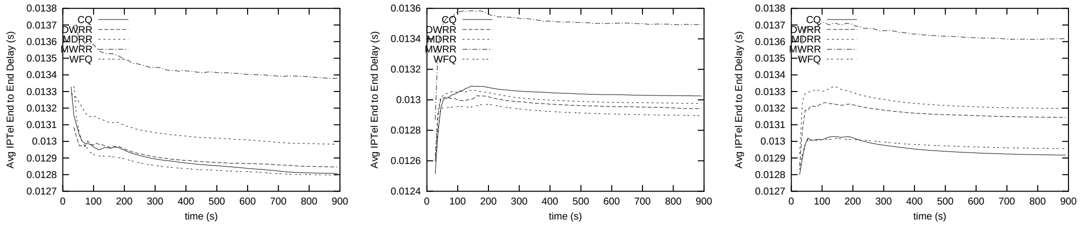
Figure 3: Mail, DB and Telnet Performances.

DB service

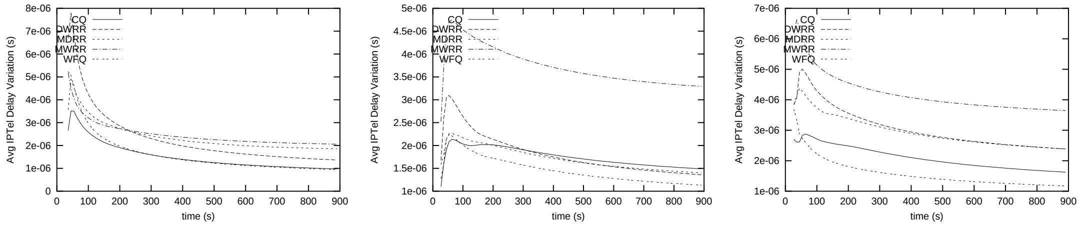
Initially, on light UDP load, WFQ presented to DB service a medium performance, close to the other two algorithms. However, as UDP load increases, WFQ presents a poorer performance. CQ presented the worst performance for all situations. It was also observed that as UDP load increases, discrepancy of performance between MWRR, which has the best performance in all situations, and the other algorithms lightly increases. Once more, Round-Robin-based algorithms had better performance with higher UDP load, as shown in Figure 3.

Telnet service

A similar behavior as observed on DB service was also obtained in Telnet service. However, discrepancy between biggest response time of CQ and the other algorithms was smaller. Just as in DB service, Round-Robin algorithms had the best performances. At first, with light UDP load, WFQ presented an acceptable response time to this service relatively to the others. With increasing UDP load, WFQ response time increases. The best response time is, then, obtained by MDRR. Measures can be seen on Figure 3.



(a) IPTel Delay Sit 1 (b) IPTel Delay Sit 3 (c) IPTel Delay Sit 5



(d) IPTel Jitter Sit 1 (e) IPTel Jitter Sit 3 (f) IPTel Jitter Sit 5

Figure 4: IPTel End-to-End Delay and Jitter.

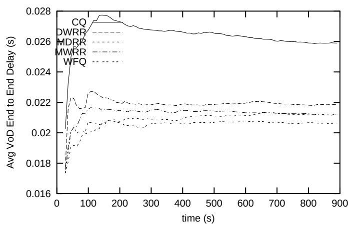
IPTel service

Measures for IP end-to-end delay and jitter are presented in Figure 4, In general, the end-to-end delay on IPTel service, as expected, has presented very low levels. It was not possible toobserved any behavior that would characterize influence of UDP load on any algorithm's performance. MWRR presented the worst performance for all situations, while WFQ presented the best end-to-end delay. The usage of low-latency, high-priority (LLHP) queue by WFQ and CQ obtained light performance gains when compared to Round-Robin algorithms, which do not offer LLHP definition for queues. On Jitter measures, practically the same relative behavior was observed.

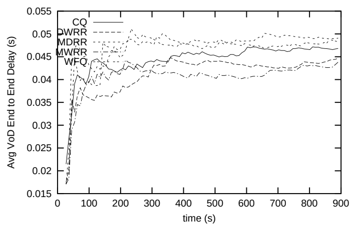
VoD service

On VoD service (Figure 5), it was observed that WFQ obtained lower end-to-end delay and CQ obtained the worst performance when UDP load was light. As UDP load increases, WFQ looses performance and CQ presents near best values. It was also observed that WFQs loss of performance is more accentuated as UDP load increases.

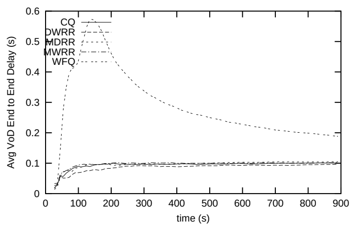
The relationship between algorithms on VoD service was reasonably different from IPTel service. This is because VoD is an UDP service just as IPTel, but it was not configured as LLHP on algorithms that support this type of queue. Hence, overall trend that presents best performance of Round-Robin algorithms on higher UDP loads may be reasonably verified. On Jitter measures, almost the same relative behavior can be observed.



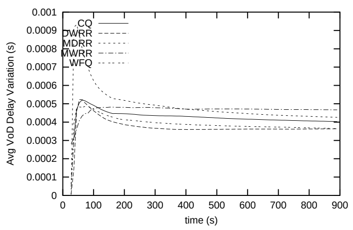
(a) VoD ETE Delay Sit 1



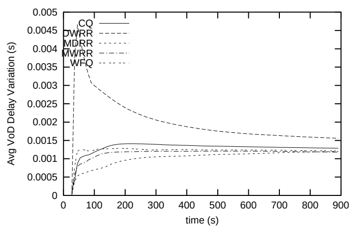
(b) VoD ETE Delay Sit 3



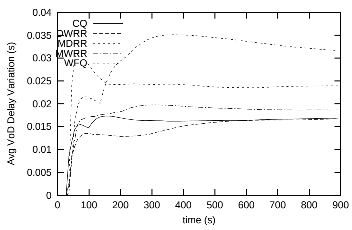
(c) VoD ETE Delay Sit 5



(d) VoD Jitter Sit 1



(e) VoD Jitter Sit 3



(f) VoD Jitter Sit 5

Figure 5: VoD End-to-End Delay and Jitter.

7. Conclusions and Future Work

This article presents a brief comparison of some scheduling algorithms. Performance evaluation took into account mainly measures applied to applications, instead of general network performance measures. Nevertheless, packet loss rate was also taken into account as network performance measure.

In our experiments, we have tested the influence of UDP traffic load to applications' performances when submitted to several QoS management configurations. Among all configurations tested, it was not possible to observe any algorithm that could have the best performance in all circumstances and to all applications. For instance, CQ is a good algorithm choice to IPTel application when UDP load is high, but it is the worst choice for Web service.

Furthermore, we observed that some management configurations in a given network condition were more helpful to a given application, in detriment to the others. For instance, for Remote Terminal application, WFQ is a good choice when UDP traffic is heavy, but when UDP traffic decreases, WFQ presents a better relative performance. Hence, management choice may take into account general policies that would define which application (or class of applications) would be more important. This policy may serve as a decision parameter to choose the best-fit management configuration that would issue a good performance to this (these) service(s) specially, in detriment of the others.

Results have also shown that configurations that had good performance on a given network condition not always maintain good operation when conditions change. Hence, network status is also a good parameter in order to configure a network.

An important trend observed in these experimentations is that Round-Robin-based algorithms (DWRR, MDRR and MWRR) are better adapted to situations where UDP traffic is heavier, with exception to IP Telephony service, where other mechanisms like WFQ and CQ have offered LLHP flag to this service.

Hence, a parameter like UDP traffic load, which represents a simple statistic of network usage, has crucially influenced scheduling algorithms' performances. This reinforces the importance of network condition information to a good management configuration choice.

Although only one network usage information (UDP load) has been tested on our experiments, we think that this kind of information must be taken into account to choose a good management configuration. As network conditions frequently changes, it would be possible that if management configuration could change together with traffic conditions, we could have a more effective network management, and network would become more and more self-healing.

As future work, we intend to exploit more parameters that could influence on management performance, like higher loads of specific applications, and we also need to test application performance over a more complex topology. We intend to add new management configurations to our comparison.

References

[1] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: An overview." IETF RFC 1633, Jun 1994.

[2] R. Braden, L. Zhang, S. Berton, S. Herzog, and S. Jasmin, "Resource ReServation Protocol (RSVP) - Version 1: An Overview." IETF RFC 2205, Sep 1997.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services." IETF RFC 2475, Dec 1998.

[4] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification." IETF RFC 2460, Dec 1998.

[5] S. Bradner and A. Mankin, "The recommendation for the IP next generation protocol." IETF RFC 1752, Jan 1995.

[6] The ATM Forum, af-uni-0010.002, *ATM User-Network Interface Specifi cation Version 3.1*, 1994.

[7] DARPA, "Internet protocol." IETF RFC 791, Sept 1981.

[8] D. P. Hong, C. Albuquerque, C. Oliveira, and T. Suda, "Evaluating the impact of emerging streaming media applications on TCP/IP performance," *IEEE Communications Magazine*, pp. 76–82, Apr 2001.

[9] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, "Recommendations on queue management and congestion avoidance in the Internet." IETF RFC 2309, Apr 1998.

[10] E. Kohler, M. Handley, S. Floyd, and J. Padhye, "Datagram congestion control protocol (DCCP)." IETF Internet Draft draft-kohler-dcp-04.txt, Jun 2002.

[11] K. Ramakrishnan, S. Floyd, and D. Black, "The addition of explicit congestion notification (ECN) to IP." IETF RFC 3168, Sep 2001.

[12] Z. Wang, *Internet QoS: Architectures and Mechanisms for Quality of Service*. Morgan Kaufmann Publishers, 2001.

[13] A. Markin, F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, and L. Zhang, "Resource ReSerVation Protocol (RSVP) version 1 applicability statement: Some guidelines on deployment." IETF RFC 2208, Sep 1997.

[14] K. Nichols, S. Blake, and F. B. D. Black, "Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers." IETF RFC 2474, Dec 1998.

[15] O. Medina and L. Toutain, "State of the art in DiffServ," tech. rep., ITEA - Information Technology for European Advancement, Feb 2001.

[16] J. Nagle, "On packet switches with infinite storage," *IEEE Transactions on Communications*, vol. 1, Apr 1987.

[17] S. Belenki, "Traffic management in QoS networks: Overview and suggested improvements," tech. rep., Dept. of Computer Engineering, Chalmers University of Technology, 2000.

[18] S. Vegesna, *IP Quality of Service*. Cisco Press, 2001.

[19] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round-robin," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 375–85, Jun 1996.

[20] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round robin," in *Proc. of SIGCOMM '95*, (Cambridge, USA), pp. 231–42, 1995.

[21] E. Aarstad, "Differentiated services - network configuration and management (DISCMAN)," tech. rep., Eurescom 0149-1006, Jun 2001.