*Elektrotehniški vestnik 76(3): 133-138, 2009*
*Electrotechnical Review: Ljubljana, Slovenija*

# Impact of hybrid queuing disciplines on the VoIP traffic delay

**Saša Klampfer, Jože Mohorko, Žarko Čučej**

*Faculty of Electrical Engineering and Computer Science, 2000 Maribor, Slovenia*
*E-pošta: sasa.klampfer@uni-mb.si*

**Abstract.** The paper presents different combinations of queuing methods and their impact on the VoIP traffic delay within the network. In our experiment we combined three of the most useful queuing methods where we put together priority queuing (PQ) with class-based weighted fair queuing (CBWFQ), custom queuing (CQ) with CBWFQ and weighted fair queuing (WFQ) with CBWFQ. Each hybrid method is compared with the basic method. For example, the PQ-CBWFQ method is compared with the priority queuing method, and so on. All the hybrid queuing disciplines are tested using an OPNET Modeler simulation tool. Our intention is to show the reader, how queuing combinations affect VoIP traffic quality, especially, with regard to Ethernet delay and jitter. Using results of our research-based simulations, we prove usefulness of our combination concept for Ethernet delay decreasing. Ethernet delay was rapidly decreased by using the WFQ-CBWFQ queuing combination. Though such combination (WFQ-CBWFQ) most strongly affects the jitter delay, and it is still within the set limits. Using the proposed queuing combination, it is possible to minimize the Ethernet delay for IP-based time-sensitive applications, including VoIP.

**Keywords:** Hybrid methods, queuing disciplines, VoIP, delay, wired network

# Vpliv hibridnih načinov uvrščanja na zakasnitev prometa VoIP

**Povzetek.** V članku predstavljamo različne kombinacije mehanizmov uvrščanja prometa IP v čakalne vrste ter vpliv kombiniranih hibridnih metod na zakasnitve prometa VoIP znotraj ožičene strukture omrežja. Predstavljeni so kombinirani načini uvrščanja, kjer smo združili prioritetno uvrščanje (PQ) z utežnim pravičnim uvrščanjem na podlagi razredov (CBWFQ), navadno uvrščanje (CQ) s CBWFQ ter utežno pravično uvrščanje (WFQ) z načinom CBWFQ. Za vsako hibridno metodo smo izvedli primerjavo z osnovnim načinom uvrščanja, na primer: kombiniran način uvrščanja PQ-CBWFQ smo primerjali s PQ in kasneje še z vsemi drugimi uporabljenimi mehanizmi uvrščanja v čakalne vrste. Vse režime, ki jih opisujemo v članku, smo preizkusili v simulacijskem orodju OPNET Modeler.

**Ključne besede:** hibridne metode, načini uvrščanja, VoIP, zakasnitev, ožičeno omrežje

## 1 Introduction

The demand for quality internet service (QoS) in wired networks has grown rapidly over recent years especially within the networks of larger companies and organizations. QoS refers to the ability of a network (wired and wireless [19]) to provide better service to selected network traffic over various underlying technologies including Frame Relay, asynchronous transfer mode (ATM), ethernet and 802.1 networks, SONET and IP-routed networks. In particular, QoS features provide better and more predictable network service by supporting dedicated bandwidth, improving loss characteristics, avoiding and managing network congestion, shaping network traffic, and setting traffic priorities across the network. Avoiding and managing network congestion and shaping network traffic are provided by using different queuing disciplines, which are a basic part of QoS assurance. All well-known queuing disciplines have both advantages and disadvantages. Our main goal in these simulations is oriented towards improving the networks performances in terms of VoIP end-to-end delay, Ethernet delay, and jitter. This also represents the basic idea for creating research procedures in this area. We use only standard queuing methods as used in the network equipment (routers), well-known under the CQ, PQ, WFQ, and CBWFQ abbreviations. In order to prove how hybrid combinations of specific methods affect the VoIP traffic delay within the network, we created simulations where hybrid queuing disciplines are compared with basic queuing schemes (PQ-CBWFQ with PQ), etc. Our goal was oriented towards exploring how each of the queuing combinations affect the VoIP delay within the network.

The most commonly used queuing disciplines and their combinations are described in the third and fourth sections. The fifth section provides a description of a G.729

*Received 2 September 2008*
*Accepted 25 January 2009*

voice encoder, and the sixth section presents the OPNET Modeler simulation tool which is used for meeting our expectations. The seventh section gives a description of a wired simulated network structure the eighth section presents the simulation results, and the ninth section gives a short conclusion and directions for further research in this area. Our main idea is connected with simulations in the OPNET Modeler because such simulations can be useful for potential internet providers. They can find proper settings for their own networks. Those obtained from simulations can be used on their real network equipment. Queuing combinations are the basic part of providing QoS in networks. Reading this paper, internet providers can identify a proper queuing mechanism which reduces VoIP traffic delay.

## 2 Problem definition

The main problem is of how to improve time-sensitive application qualities. The basic element for quality assurance is the QoS system, which also covers the queuing area. As each queue represents a buffer where traffic is temporarily stored, it affects traffic delay. Time-sensitive application traffic must travel undisturbed through the network and its elements (routers, switches, etc.). When traffic goes through a network element it also goes through queuing mechanisms. For such traffic, PQ and CBWFQ mechanisms are especially appropriate. The question is whether there is a queuing combination which would give better results for time-sensitive applications compared to both above-mentioned methods? This is also, where the main focus of our researches. Each queuing scheme has both good and bad properties where, using different queuing combinations and combining good properties, we try to reach a level appropriate for time-sensitive applications, such as the VoIP application used in our case. By combining them, we tend to reduce the Ethernet traffic delay for VoIP application.

## 3 Queuing disciplines

*Custom queuing (CQ):* The custom queuing task is linked to bandwidth sharing between active applications within a network, respectively between the organizations. In such circumstances the bandwidth must be proportionally divided between applications and end-users, in the sense that congestion cannot appear. The CQ mechanism [2, 10] provides a satisfactory bandwidth at a congested point and, at the same time, protects specific traffic with constant bandwidth amounts, whilst meanwhile the remaining bandwidth CQ is left for other active data traffic within the network. Traffic is managed by assigning weight amounts, respectively to places in the queue for each class of packet. CQ is composed of 17 queues numbered from 0 to 16 [10]. In this case we have 17 waiting queues – classes or more which are mutually equivalent. Each waiting queue class has an annotation which tells us how much bandwidth should be provided for transferring data onto the output link. Then, the algorithm arranges messages into one of seventeen available queues, where the queue with index 0 is intended for storing system messages, such as keepalive messages and many other signal messages. The queue-empty procedure executes the principle of the weighted size, obtained from the priorities, which means that a message with a higher priority has a greater weighted comparison than a message with a lower priority amount. The router manages the waiting queues from index one to index 16 in a round-robin manner, where each empty cycle represents a byte counter. It carries-out orders in the sense that no application can reach more than the predefined level of bandwidth, in case of large traffic being presented in the network. CQ is statically configured as for many other mechanisms, such as Priority Queuing-PQ (described below) etc. This is the reason why CQ is incapable of automatically accommodating to any dynamic changes in the network [2, 3, and 4].

*Priority queuing (PQ):* Priority queuing [2] provides undisturbed high-priority traffic to travel through the network with smaller delays in comparison to the low-priority traffic. PQ uses four different queues with different priority levels. The queue with the lowest degree is marked with ‘low’, then follows ‘usual’, ‘middle’, and ‘high’. Incoming packets are arranged into one of the four queues regarding the applied priority level in the Type of Service (ToS) [11] byte which is part of the IP packet header. Unclassified packets are assigned into the queue with a ‘usual’ annotation. The main property of the high priority queues is the absolute precedence trail between the transfer processes in comparison with low-priority queues. PQ is statically configured [1, 2].

*Weighted fair queuing (WFQ):* Weighted fair queuing [12] is the best solution in situations where we want to provide a constant response time and keep delays within a specified range, without assigning an excessive bandwidth. WFQ is an algorithm/mechanism which introduces bit-wise fairness and allows for each queue to be treated fairly. The fairness is provided by a mechanism which counts bytes (Fig. 1).

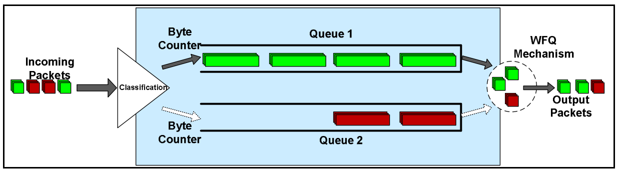


Figure 1. Weighted fair queuing discipline (WFQ).
Slika1. Utežno pravično uvrščanje.

As the simplest example, we can observe two queues of the same length. The first one contains 100 packets and the second only 50. In this case the WFQ will work in the following manner; firstly it will take two packets from the first queue, then one packet from the second queue and keep repeating this until fairness is achieved.

Such an algorithm creates service fairness for each participating queue. Low-level priority traffic can also travel undisturbed through the network, which is a good compromise for each traffic flow within the network. WFQ is capable of configuring cost minimization, because it can automatically adjust to any dynamic network traffic changes. Because of its good qualities, it is used on the majority of serial interfaces configured for E1 operation at speeds of 2048 Mbps. Weighting is, also defined in this case, using the IP priority amount in the ToS field [11] of the IP packet header. For IP priority settings, the range from 0 (best effort) to 6 (IP quality speech) is in use, whilst the setting 7 is reserved. The algorithm then uses data from the ToS field to calculate how many additional services need to be provided for each individual queue. The calculated results define the bandwidth amount needed for a specific case. WFQ must reserve only the calculated amount of the bandwidth, the other part is left to other applications. Such a principle is opposite to the time-division multiplexing (TDM) mechanism [13], which permanently reserves all available bandwidths [4, 5]. WFQ reserves only as much bandwidth as a specific application needs.

*Class-based weighted fair queuing (CBWFQ):* CBWFQ represents the newest scheduling mechanism intended for handling congestions while providing greater flexibility [5]. It is usable in situations where we want to provide a proper amount of the bandwidth to a specific application (in our case VoIP application). In these cases, the network administrator must provide classes with defined bandwidth amounts, where one of the classes is, for example, intended for a video-conferencing application, another for VoIP application, and so on. Instead of waiting-queue assurance for each individual traffic flow, CBWFQ determines different traffic flows. A minimal bandwidth is assured for each of such classes. One case where the majority of the lower-priority multiple-traffic flow can override the highest-priority traffic flow is video transmission which needs half of the T1-connection bandwidth [7]. A sufficient link bandwidth would be assured using the WFQ mechanism, but only when two traffic data flows are present. In a case where more than two traffic flows appear, the video session suffers the regarding bandwidth, because the WFQ mechanism works on the fairness principle. For example, if nine additional traffic flows make demands of the same bandwidth, the video session will get only 1/10th of the whole bandwidth, and this is insufficient when using a WFQ mechanism. Even if we put an IP priority level of 5 into the ToS field [11] of the IP packet header, the circumstances would not change. In this case, the video conference would only get 6/15 of the bandwidth, and this is not enough because the mechanism must provide half of all the available bandwidth on the T1 link. This can be provided by using the CBWFQ mechanism. The network administrator just defines, for example, the video-conferencing class and installs a video session into that class. The same principle can be used for all other applications which need specific amounts of the bandwidth. Such classes are served by a flow-based WFQ algorithm which allocates the remaining bandwidth to other active applications within the network [5].

## 4 Hybrid queuing disciplines

*Combination of PQ-CBWFQ queuing discipline:* Priority queuing/Class-based Weighted Fair Queuing (PQ-CBWFQ) is a queuing scheme that adds strict priority queuing to CBWFQ. Strict-priority queuing allows for delay-sensitive data, such as voice, to be sent first, before packets in other queues are de-queued. This scheme gives delay-sensitive data preferential treatment over other traffic. PQ-CBWFQ belongs to the queuing method group that offers smaller delays. Such queuing allows, for specific flow classes, defined using IP priorities, to be served within a strict priority queue (Fig. 2).

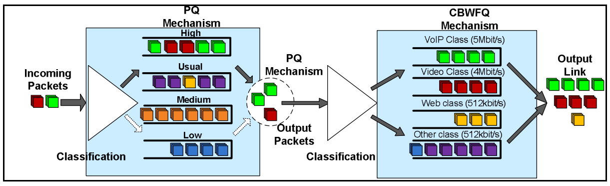


Figure 2. PQ-CBWFQ combination.
Slika 2. Hibridna metoda uvrščanja PQ-CBWFQ

The highest-priority class is served before all other priority classes, compared to other ordinary queuing schemes [5].
Such a combination can be found in Cisco's routers. All other combinations presented below (CQ-CBWFQ and WFQ-CBWFQ) are not implemented in real routers, and just reflect our ideas, tested in the OPNET simulation tool.

*Combination of CQ-CBWFQ queuing discipline:* Custom queuing shares the available bandwidth between active network applications in the sense that congestion cannot appear. This is the main reason why we combined these two queuing schemes. In CQ, traffic is managed by assigning weighted amounts and is arranged into 17 queues. A CBWFQ packet-classification mechanism, attached behind the custom queuing mechanism arranges traffic into traffic classes defined by a class-based weighted fair queuing algorithm. Such classes are then ensured by fixed amounts of the bandwidth. All the advantages of the CBWFQ are retained. Using this method, we reduce delays within the network unlike is the case with the ordinary CQ scheme.

*Combination of WFQ-CBWFQ queuing discipline:* WFQ is suitable for operating with IP priority settings, such as the resource reservation protocol (RSVP) [15], which is also capable of managing round-trip delay problems. Such queuing clearly improves algorithms such as SNA (Systems Network Architecture) - Cisco SNA (CSNA) which is an application that provides support for SNA protocols to the IBM mainframe. Using a

Cisco 7000, 7200, or a 7500 Series router with a Channel Interface Processor (CIP) or Channel Port Adapter (CPA) and Cisco SNA (CSNA) support enabled, we can connect two mainframes (either locally or remotely), connect a mainframe to a physical unit (PU) 2.0 or 2.1 device, or connect a mainframe to a front-end processor (FEP) in another Virtual Telecommunications Access Method (VTAM) domain [16], logical link control (LCC) [14], and transmission control protocol (TCP). WFQ-CBWFQ is, at the same time, capable of accelerating slow features and removing congestion within the network (merged good properties of both queuing schemes). Results become more predictable over the whole routing path, meanwhile Ethernet delays can be greatly decreased (see the simulation results section) compared to other queuing disciplines (CQ, PQ, WFQ). The WFQ and CBWFQ queuing combination can represent the best solution (merged good properties from both methods) for reducing the Ethernet delay. This assumption can be proved within the seventh section, where delays within the network are greatly reduced compared to all other queuing schemes.

## 5 Used voice encoder

The G.729 speech coder is an 8 kbps Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP) speech compression algorithm approved by ITU-T. G.729. It offers high quality, robust speech performance at the cost of higher complexity. It requires 10 ms input frames and generates frames of 80 bits in length. Within the G.729 coder the processing signals are 10 ms frames with an additional 5 ms look-ahead. The total delay introduced by the algorithm is 15ms. Since G.729 is based on the Code-Excited Linear Prediction (CELP) model, each produced 80 bit frame contains linear prediction coefficients, excitation code book indices, and gains parameters that are used by the decoder in order to reproduce speech. The input/output of this algorithm is a 16 bit / 8 kbps linear PCM audio stream that is converted from/to a compressed data stream [7].

*VoIP and UDP short introduction:* most data travelling over the Internet uses the Transmission Control Protocol (TCP) [2] for the transport layer because it guarantees data delivery and integrity. VoIP [18] does not need the kind of delivery guarantee which TCP provides, so most VoIP transmissions use the faster User Datagram Protocol (UDP) [2] as the transport layer. VoIP uses UDP as the transport layer. The UDP protocol only provides a direct method of sending and receiving data over an IP network, and offers very few error recovery services. UDP has no mechanisms in place to notify the application of any loss in transmission whilst delivering packets of data; it also sends data unordered with no guarantees of the data being presented in the receiving application.

## 6 Experiment description

The experimental part was tested within the OPNET Modeler simulation tool [6, 17] where a network structure was modeled, which is a copy of a private company network. As the company that realized it had VoIP quality problems, a study and a model were made for such network (links, equipment, applications, etc.). Different queuing mechanisms and the proposed hybrid concepts were tested within a simulated network (Fig. 3).
Fig. 3 represents a fictitious network composed for proving advantages and disadvantages of the proposed hybrid queuing methods. Our main goal in these simulations is oriented towards improving the network performances with regard to the VoIP end-to-end delay, Ethernet delay, and jitter. We try to use different queuing schemes and to find the most appropriate one for VoIP traffic (smallest delay).

VoIP traffic flows were fully meshed between all the groups containing VoIP users (‘3VoIP’, ‘2VoIP’, ‘5VoIP’, ‘VoIP’ and ‘Misc’). The network structure consists of servers, such as Web Server, FTP server, etc., which are connected over a 10BaseT connection through a 16 port switch on IP Cloud, as shown at the top of Fig. 3. Four local-area segments (LANs) are connected to the routers where different kinds of users (VoIP, Web users) are placed. Each Cisco router is connected by a 16 port switch, where it is then connected to an IP cloud. Users use different application clients such as VoIP, Web, and FTP.

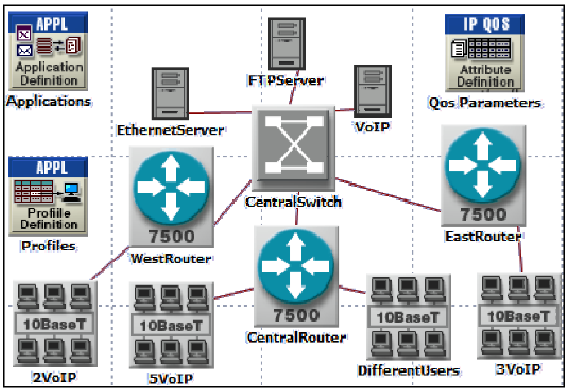

Figure 3. Network simulation structure.
Slika 3. Struktura simuliranega omrežja.

Web and FTP applications are applied only for creating low-priority traffic flows. Such applications are defined using the ‘Applications’ node shown at the top-right side of Fig. 3. Using the ‘Profiles’ node (besides the applications node), client profiles are defined, or the task what the User Equipment (UE) should do, or which UE application could be used. In IP, QoS node (QoS parameters) defines the traffic policy for the network. Within each router there are traffic classes configured for the CBWFQ queuing method where specific traffic flows (VoIP flow, Video session flow, Web browsing flow, etc.) are placed. Each router has three traffic classes, where the first-one, with 9Mbit/s, belongs to VoIP traffic, whereas the second and the third-one belong to low-priority FTP and HTTP traffic flows defined by 512 kbit/s bandwidths for each class.

Table1. Number of users and defined classes for individual active application within the network

| Application | Users | Class | Bandwidth |
|---|---|---|---|
| VoIP | 10 | 1 | 9 Mbit/s |
| FTP | 490 | 2 | 512 kbit/s |
| Web | 10 | 3 | 512 kbit/s |

## 7 Simulation results

Simulation resuts were collected after each successive simulation run for both the ordinary and hybrid queuing methods described above.

The obtained results even in a graphical form clearly show the impact of each combination on the Ethernet delay and Jitter. The impact is obvious in cases where each queuing combination is compared with a basic queuing scheme (PQ with PQ-CBWFQ).

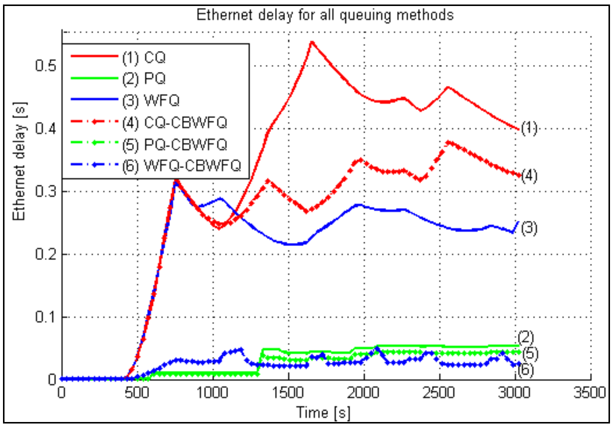


Figure 4. Ethernet delay for ordinary queuing methods.
Slika 4. Ethernet zakasnitev običajnih metod uvrščanja.

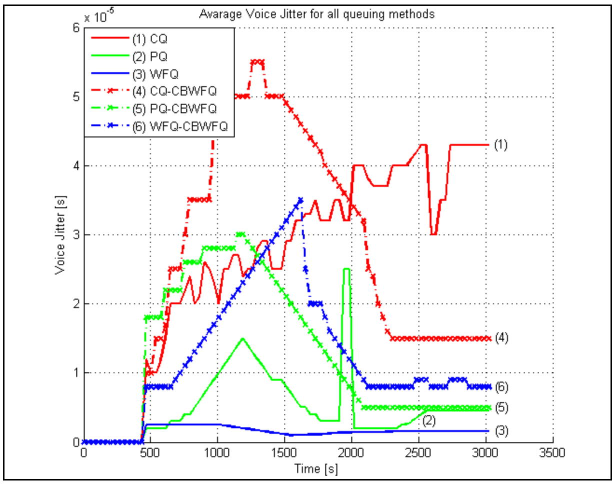


Figure 5. Ethernet delay for hybrid queuing methods.
Slika 5. Ethernet zakasnitev hibridnih metod uvrščanja.

In Fig. 4 can we see the Ethernet delay for all queuing schemes, where (1) presents the Ethernet delay for custom queuing (CQ), (2) priority queuing (PQ) Ethernet delay, (3) Ethernet delay for weighted fair queuing (WFQ) method, (4) CQ-CBWFQ combination, (5) PQ-CBWFQ, and (6) Ethernet delay for WFQ-CBWFQ hybrid queuing method. Fig. 6 shows Voice jitter for all queuing schemes where (1) presents custom queuing (CQ) jitter delay, (2) priority queuing (PQ), (3) weighted fair queuing (WFQ), (4) CQ-CBWFQ, (5) PQ-CBWFQ, and (6) voice jitter delay for WFQ-CBWFQ hybrid queuing method.

## 8 Discussion

Comparing a specific hybrid queuing method (Fig. 5) with a specific ordinary queuing method (example: (1)-CQ with (4)-CQ-CBWFQ) we can see Ethernet delay reduction in the CQ-CBWFQ case. The same is true for a comparison between (2)-PQ and (5)-PQ-CBWFQ, and also between (3)-WFQ and (6)-WFQ-CBWFQ queuing methods. WFQ-CBWFQ is obviously the best combination for reducing the average Ethernet delay within the network. Besides its advantages, the disadvantages of the method is the jitter issue. The best results are obtained with the PQ-CBWFQ queuing discipline (Fig. 6). Results of none of other combinations satisfy our expectations for VoIP and other time-sensitive internet applications.

PQ-CBWFQ, which is usually known as LLQ (low latency queue provides a strict priority queue for voice traffic and a weighted fair queue for any other traffic class. As we see in Fig. 5, the PQ-CBWFQ combination works fine for strict-priority traffic flows such as, for example, VoIP (tested in our case), video conferencing, video on demand, etc. High-priority traffic has, in the case of PQ-CBWFQ, the smallest delay, which is comparable with the WFQ queuing scheme.

Fig. 6 presents voice jitter for all queuing schemes (hybrid and ordinary). Jitter is a variation or dislocation in the pulses of a digital transmission. The usual causes include connection timeouts, connection time lags, data traffic congestion, and interference. Jitter is an undesirable manner when we transfer high-priority data, such as real-time traffic, VoIP traffic, etc.

To understand jitter [8, 9], we must remember that, for example, an audio file (either audio, video, pictures or text) consists of proper sizes which are then sent to the output link. The data is split-up into manageable 'packets' with headers and footers that indicate the correct order of the data packets when they travel to its destination, where they are then organized into proper sequences. When a jitter occurs, some data packets may be lost in transit or the code for data packet assembly in the receiving machine may be wiped out. But this is important only for UDP packets, which VoIP certainly is. This isn't true for TCP packets because, in the case of loss, the TCP packet system requires retransmission. In our case, UDP transmission protocol was used. Thus when jitters occur, voice packets can be corrupted. Jitter is especially important when real-time applications are used. If the buffer length is too short, the incoming packet won't be delivered to its destination on time, so it cannot be played at the right time. Due to its undesirable consequences, jitter is an important issue in the design of all communications links. In this case the WFQ scheme has the smoothest and also the lowest jitter value. Speaking generally, the CQ-CBWFQ and WFQ-CBWFQ queuing schemes are the worst case. The latter gives the best results in the Ethernet delay case. But such jitter can negatively affect the VoIP speech quality. As we expected, the PQ-CBWFQ also reaches small jitter values, which is desirable for VoIP and other real-time or near real-time applications. In any other queuing scheme, jitter values are larger, and also acceptable in the VoIP case where the maximal value reaches only 40us. The critical jitter border is 150ms. Each delay in voice application which is larger than 150ms can be detected by the human ear. Voice packets must arrive at their destinations within 120ms, which is near the real-time frame defined as 100ms ± ΔT, where ΔT is equal to 20ms. Jitter is measured between audio frames, and this is the main reason why it is acceptable. The situation would be different if such jitter appeared between individual audio samples at 8 kHz sampling rate (Ts = 125us), but we focus only on jitter between audio frames. The reason for bad results in the jitter case for hybrid methods can be found within the buffer area. To minimize the adverse impact of jitter in media file downloads, the 'buffer' is usually employed. The buffer serves as the storage area in the system where incoming packets for digital audio or video are arranged before they are played back - the computer is given the time needed to ensure that the incoming data packets are complete before they can be played.

## 9 Conclusion

While the observed/proposed queuing combinations express satisfactory results for one criterion, they also yield unsatisfactory results for another criterion, such as jitter in the case of VoIP application. The approach is a good example how we can control Ethernet delays within the network. The analyzed combinations are useless for the VoIP, but further research and analysis will show if it is possible to use such an approach on other internet applications which are not-so sensitive in terms of jitter. In future, we will also try to simulate different queue lengths, so as to reduce jitter in the VoIP case. Based on our simulation results, we can prove that PQ-CBWFQ is a low-latency queuing scheme and a proper solution for a VoIP time-sensitive application because it has the smallest average Ethernet delay compared to any other queuing schemes.

## 10 Future work

Our future work will be towards proving the theory of using the low-latency queuing discipline on other time-sensitive internet applications.

## References


[1] T. Subash, S. IndiraGandhi. Performance Analysis of Scheduling Disciplines in Optical Networks. MADRAS Institute of Technology, Anna University, 2006.

[2] L. L. Peterson, B. S. Davie. Computer Networks. *Edition 3*, San Francisco 2003.

[3] S. Bucheli. Compensation Modeling for QoS Support on a Wireless Network. *Master degree thesis*, 2004.

[4] K. M. Yap, A. Marshall, W. Yu. Providing QoS for Multimodal System Traffic Flows in Distributed Haptic Virtual Environments. Queen's University Belfast, 2005.

[5] Internetworking Technology Handbook – Quality of Service (QoS), Cisco Systems.

[6] OPNET Modeler Techical Documentation.

[7] G. 729 Data Sheet.

[8] L. Zheng, D. Xu. Characteristics of Network Delay and Delay Jitter and its Effect on Voice over IP (VoIP) Communications. ICC 2001, IEEE International Conference, 2001.

[9] M. Kao. Timing Jitter Control of an ADD/drop Optical Module in a convergent Network, Wireless and Optical Communications, 2005. 14th Annual WOCC 2005, International Conference.

[10] http://www.opalsoft.net/qos/WhyQos-2423.htm

[11] http://en.wikipedia.org/wiki/Type_of_Service

[12] http://www.opalsoft.net/qos/WhyQos-2424.htm

[13] http://en.wikipedia.org/wiki/Time-division_multiplexing

[14] http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/llc.html

[15] A. Kos in S. Tomažič. Nov način združevanja RSVP pretokov. Fakulteta za Elektrotehniko, Univerza v Ljubljani.

[16] http://www.cisco.com/en/US/tech/tk331/tk332/tk126/tsd_technology_support_sub-protocol_home.html

[17] S. Klampfer. Simulacija omrežij v Opnet Modeler-ju. Diplomska naloga univerzitetnega študijskega programa, Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru, 2007.

[18] http://www.voip4business.biz/what_is_voip.htm

[19] I. Humar, J. Bešter, M. Pogačnik, M. Meža. Extending Differentiated Services with Flow Rejection Mechanism for Wireless IP Environments. Elektrotehniški Vestnik 1-2005.



**Saša Klampfer** received his B.Sc. degree in electrical engineering from the University of Maribor, Slovenia, in 2007. Currently, he is a Ph.D. student and Junior Researcher in the Signal Processing and Remote Control Laboratory at the Faculty of Electrical Engineering and Computer Science. His research interests are in the field of telecommunication systems and technologies, and automatization.

**Jože Mohorko** received his B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the University of Maribor, Slovenia, in 1990, 1994 and 2002, respectively. He is a docent and researcher at the Faculty of Electrical Engineering and Computer Science in Maribor, University of Maribor. His research interests include image processing, modeling, simulating and evaluating telecommunication infrastructures.

**Žarko Čučej** is a full professor for automatic control and robotics, and telecommunication at the Faculty of Electrical Engineering and Computer Science of the University of Maribor. His recent research interests include signal processing and industrial data networks.