5

Influences of Classical and Hybrid Queuing Mechanisms on VoIP's QoS Properties

Sasa Klampfer[1], Amor Chowdhury[1], Joze Mohorko[2] and Zarko Cucej[2]
[1]Margento R&D d.o.o.
[2]University of Maribor, Faculty of Electrical Engineering and Computer Science
Slovenia

1. Introduction

Nowadays we can find many TCP/IP based network applications, such as: WWW, e-mail, video-conferencing, VoIP, remote accesses, telnet, p2p file sharing, etc. All mentioned applications became popular because of fast-spreading broadband internet technologies, like xDSL, DOCSIS, FTTH, etc. Some of the applications, such as VoIP (Voice over Internet Protocol) and video-conferencing, are more time-sensitive in delivery of network traffic than others, and need to be treated specially. This special treatment of the time-sensitive applications is one of the main topics of this chapter. It includes methodologies for providing a proper quality of service (QoS) for VoIP traffic within networks. Normally, their efficiency is intensively tested with simulations before implementation. In the last few years, the use of simulation tools in R&D of communication technologies has rapidly risen, mostly because of higher network complexity.

The internet is expanding on a daily basis, and the number of network infrastructure components is rapidly increasing. Routers are most commonly used to interconnect different networks. One of their tasks is to keep the proper quality of service level. The leading network equipment manufacturers, such as Cisco Systems, provide on their routers mechanisms for reliable transfer of time-sensitive applications from one network segment to another. In case of VoIP the requirement is to deliver packets in less than 150ms. This limit is set to a level where a human ear cannot recognize variations in voice quality. This is one of the main reasons why leading network equipment manufacturers implement the QoS functionality into their solutions. QoS is a very complex and comprehensive system which belongs to the area of priority congestions management. It is implemented by using different queuing mechanisms, which take care of arranging traffic into waiting queues. Time-sensitive traffic should have maximum possible priority provided. However, if a proper queuing mechanism (FIFO, CQ, WFQ, etc.) is not used, the priority loses its initial meaning. It is also a well-known fact that all elements with memory capability involve additional delays during data transfer from one network segment to another, so a proper queuing mechanism and a proper buffer length should be used, or the VoIP quality will deteriorate.

If we take a look at the router, as a basic element of network equipment, we can realise that we are dealing with application priorities on the lowest level. Such level is presented by waiting queues and queuing mechanisms, related with the input traffic connection interface.

The traffic which appears at the input connection is transferred to the queuing mechanisms and waiting queues. Which queuing mechanism from the set of available queuing mechanisms will be used depends on the network administrator's choice.

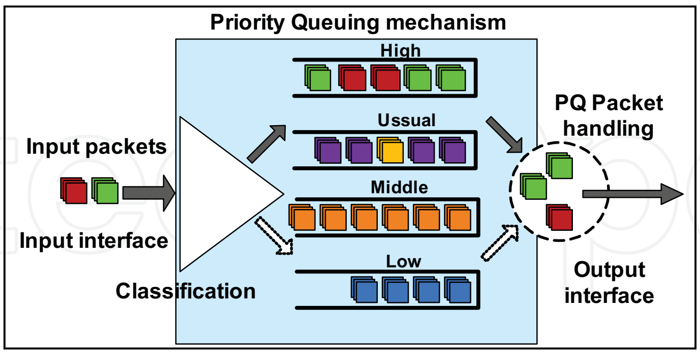


Fig. 1. Priority Queuing Mechanism

One of the QoS's most crucial components are waiting queues, where suitable queuing mechanisms take care of proper IP traffic treatment. The sophisticated queuing mechanisms also include traffic sorting and scheduling functionality. This group of regimes is called 'conscious', and includes the following queuing regimes:

- priority queuing (PQ) which sorts the packets according to their priority (see Fig. 1),
- weighted fair queuing (WFQ) which provides bandwidth fairness usage for all traffic types, and
- class-based weighted fair queuing (CBWFQ), which gives the advantage to the traffic for which the traffic class has been generated by the administrator.

First-in-first-out (FIFO) queuing and custom queuing (CQ) mechanisms belong to the old-fashioned queuing regimes, the so called 'unconscious' group. With such a group it does not matter which type of traffic appears at the input interface, but they treat the traffic as it actually is. In the FIFO case, the packet that came first in also goes first out, etc.

With individual analyses of queuing mechanism properties we get an idea of joining the advantages of two queuing mechanisms. This means that the positive properties of both mechanisms will be combined. Combining different queuing mechanisms and proving their new properties is a part of our scientific contribution. For the research we have been using a sophisticated simulation tool: OPNET Modeler. The result of our ideas and experiments are hybrid queuing mechanisms (except PQ-CBWFQ). The conclusion of our research is that the best solution of all the tested concepts is still the well-known PQ-CBWFQ method. From the set of tested hybrid methods the best results in terms of the VoIP jitter delay were obtained with our proposed WFQ-CBWFQ concept, which significantly reduces the jitter. The results of the WFQ-CBWFQ concept are according to our estimations in the VoIP jitter case even better than with the PQ-CBWFQ, but the disadvantage of the first concept reflects in a slightly higher VoIP delay in comparison to the PQ-CBWFQ.

Much similar research using simulations has been done in the area of VoIP's quality improvement; some of it is presented in the following literature: Mansour J. Karam & Fouad A. Tobagi, 2001, Velmurugan T. et al., 2009, and Fischer, M.J. et al., 2007. VoIP's quality improvement is a very popular research area, mostly focused on queuing aspects, and the problem of decreasing jitter influence as in our case. The hybrid queuing mechanism concept (except PQ-CBWFQ) is our original contribution, resulting from the research of the past three years.

2. Presentation of the quality of service and its connection to waiting queues

Here the basic terminology and facts about Quality of service and waiting queues will be explained; what QoS is, where it can be found (H. Jonathan Chao & Bin Liu, 2007), how it works, main parts of QoS (Kun I. Park, 2005), and QoS levels (M. Callea et al., 2005), how QoS handles congestions, etc. (L. L. Peterson & B. S. Davie, 2003 and Cisco Systems-Internetworking Technology Handbook, 2002). At this point, we will present the two most important areas corresponding to our research work; the so called fuzzy QoS area for distinguishing traffic, and the area which includes the mechanisms for traffic congestions management, to which the waiting queues belong.

2.1 What is QoS?

Quality of Service allows control of data transmission quality in networks, and at the same time improves the organization of data traffic flows, which go through many different network technologies. Such a group of network technologies includes ATM (asynchronous transfer mode), Ethernet and 802.1 technologies, IP based units, etc.; and even several of the abovementioned technologies can be used together.

An illustration of what can happen when excessive traffic appears during peak periods can be found in everyday life: an example of filling a bottle with a jet of water. The maximum flow of water into the bottle is limited with its narrowest part (throat). If the maximum possible amount of decantation (throughput) is exceeded, a spill occurs (loss of data). A funnel used for pouring water into a bottle, would in case of data transfer be in the waiting queues. They allow us to accelerate the flow, and at the same time prevent the loss of data. A problem remains in the worst-case scenario, where the waiting queues are overflowed, which again leads to loss of data (a too high water flow rate into the funnel would again result in water spills).

Priorities are the basic mechanisms of the QoS operating regime, which also affects the bandwidth allocation. QoS has an ability to control and influence the delays which can appear during data transmission. Higher priority data flows have granted preferential treatment and a sufficient portion of bandwidth (if the desired amount of bandwidth is available). QoS has a direct impact on the time variation of the sampling signals which are transmitted across the network. Such sampling time variation is also called jitter (T. & S. Subash IndiraGandhi, 2006). Both mentioned properties have a crucial impact on the quality of the data and information flow throughput, because such a flow must reach the destination in the strict real-time. A typical example is the interactive media market. QoS reflects their distinctive properties in the area of improving data-transfer characteristics in terms of smaller data losses for higher-priority data streams. The fact that QoS can provide priorities to one or more data streams simultaneously, and also ensure the existence of all remaining (lower-priority) data streams, is very important. Today, network equipment companies integrate QoS mechanisms into routers and switches, both representing fundamental parts of Wide Area Networks (WAN), Service Provider Networks (SPN), and finally, Local Area Networks.

Based on the abovementioned points, the following conclusion can be given: QoS is a network mechanism, which successfully controls traffic flood scenarios, generated by a wide range of advanced network applications. This is possible through the priorities allocation for each type of data stream.

2.2 How QoS works?

QoS mechanism, observed as a whole, roughly represents an intermediate supervising element placed between different networks, or between the network and workstations or servers that may be independent or grouped together in local networks. The position of the QoS system in the network is shown in Figure 2. This mechanism ensures that the applications with the highest priorities (VoIP, Skype, etc.) have priority treatment. QoS architecture consists of the following main fundamental parts: QoS identification, QoS classification, QoS congestions management mechanism, and QoS management mechanism, which handle the queue.

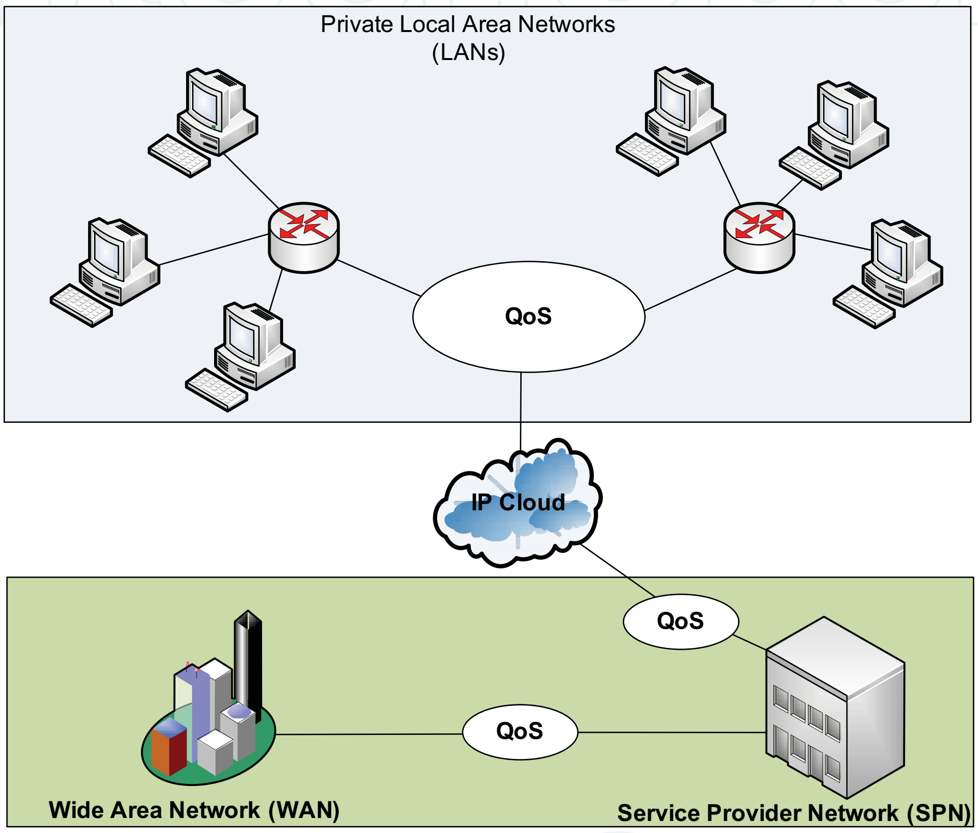


Fig. 2. QoS system's position in the network

2.2.1 QoS Identification

QoS identification is intended for data flows recognition and recognition of their priority. To ensure the priority a single data stream must first be identified and then marked (if this is needed). These two functions together partly relate to the classifying process, which will be described in detail in the next section. Identification is executed with access control lists (ACL). ACL identifies the traffic for the purpose of the waiting queue mechanisms, for example PQ - Priority Queuing or CQ - Custom Queuing. These two mechanisms are implemented into the router, and present one of its most important subparts. Their operation is based on the principle of "jump after a jump", meaning that the QoS priority settings belong only to this router and they are not transferred to neighboring routers, which form a network as a whole. Packet identification is then used within each router with QoS support. An example where classification is intended for only one router can be found with the CBWFQ (Class Based Queuing Weighted Fair) queuing mechanism. There are also techniques which are based on extended control access-list identities. This method allows considerable flexibility of priorities allocation, including the allocation for applications, users, destinations, etc. Typically, such functionality is installed close to the edge of the network or administrative domain, because only in this case each network element provides the following services on the basis of a particular QoS policy.

Network Based Application Recognition (NBAR) is a mechanism used for detailed traffic identification. For example, NBAR can identify URLs, which are located in the HTTP packet. When the packet is recognized, it can be marked with priority settings. If we look deeper into the structure of the HTTP packet, we can recognize URLs as well as the MIME type. This is a more than welcome feature of the WWW (World Wide Web)-based applications. NBAR can recognize various applications that use a variety of different ports/plugs. This functionality is performed with the procedure of checking control packets, where it finds the port through which the application will be sending the data. Such mechanism includes many useful features, which allow protocol identification and their statistical analysis at the interface entry point. The mechanism also contains a module for a linguistic description of the packet (Packet Description Language Modules - PDLM), where this functionality simplifies insertion of new protocols, which can be then identified.

2.2.2 QoS Classification

QoS classification is designed for executing priority services for a specific type of traffic. The traffic must first be pre-identified and then marked (tagged). Classification is defined by the mechanism for providing priority service, and the marking mechanism. At the point, when the packet is already identified, but it has not yet been marked, the classification mechanism decides which queuing mechanism will be used at a specific moment (for example, the principle of per-hop). Such an approach is typical in cases when the classification belongs to a particular device and is not transferred to the next router. Such a situation may arise in case of priority queuing (PQ) or custom queuing (CQ). When the packets are already marked for use in a wider network, the IP priorities can be set in the ToS field of the IP packet header. The main task of classification is identification of the data flow, allocation of priorities and marking of specific data flow packets.

2.2.3 QoS congestion management mechanism

Because of the nature of audio, video and data traffic, the whole traffic amount sometimes exceeds the maximum speed of the connection. In this situation the following question can be raised: what should the router do in such situations? Will it manage and insert the packets, or better yet series of packets, into a double queue or two single queues, which will be refreshing more often? For solving such problems, a tool for managing congestions is used nowadays. Congestion management mechanism ensures that the data flows are placed into corresponding and proper waiting queues. Depending on the application type and application priorities the mechanism decides into which queue the momentary packet will be inserted. As a classic example, we can take a look at an HTTP packet. For such a packet the mechanism will provide custom queuing discipline (CQ), where the packet will be assigned into one of 16 internal queues (see section 3). In case of priority queuing such a mechanism (PQ) would insert the HTTP packet into the lowest internal queue (*low*).

2.2.4 QoS queuing management mechanism

We have to be aware that the round-robin waiting queues (single, double) do not have an infinite length, meaning that sooner or later they are full or congested. Another disadvantage is that each memory structure involves additional delays during data transfer. When the queue is full, it cannot accept any new packets, meaning that a new packet will be rejected. The reason for rejection has been already discovered: the router simply cannot avoid discarding packets when the queue is full, regardless of which priority is applied in the ToS field of the packet. From this perspective the queue management mechanism must execute two very important tasks:

- Try to ensure a place in the round-robin queue or try to prevent the queue from becoming full. With this approach a queuing management mechanism provides the necessary space for high-priority frames;
- Enable the criterion for rejecting packets. The priority level applied in the packet must be checked at the beginning, after which the mechanism decides which packet will be rejected and which not. Packets with lower priority are rejected earlier in comparison to those with a higher priority. This allows undisturbed movement of high-priority traffic flows, and if there is some additional space at the available bandwidth, other low-priority traffic flows can also pass through the network.

Both described methods are included in the *Weighted Random Early Detect* mechanism, which can be found in various sources under the acronym WRED.

2.3 QoS service levels

Service levels are related to the QoS capabilities of the system, which help ensuring the proper delivery of specific traffic through the network to its destinations. QoS service levels differ in accuracy and consistency (QoS strictness). Such levels define how much bandwidth a certain application requires, how latency and jitter influence it, and how each service level manages the packet loss characteristics. Three basic service levels are provided across the entire heterogeneous network, as shown in Figure 2:

- Best effort service has no guaranteed service. A good example for this level is FIFO queue, which has no capability to differ individual traffic types.
- Differentiated service presents the so-called »soft« QoS. With its application all traffic types are treated in a better way, which also speeds up the treatment, improves the average threshold of bandwidth and reduces the low-priority traffic data loss. This type of service includes the traffic classification mechanism and QoS queuing mechanisms such as PQ, CQ, WFQ and WRED, which are going to be explained in detail in section 3. Basically, this level of service has a statistical advantage in comparison to the above-presented best effort service, but a guaranteed service, which is the main property of the last service level, is still not applied here.
- Guaranteed service level is representative of the so-called high-level QoS. It is primarily intended to maintain the network resources for specific traffic. Such level is provided by Resource Reservation Protocol (RSVP) and CBWFQ queuing mechanism.

To conclude: which service level is more appropriate for use in a particular network depends on the following factors:

- If a user tries to solve a communication problem for a particular application, each of the above mentioned levels could solve this problem. Performance which could be achieved depends on the requirements of the user applications.

- In everyday life, situations where users want to flexibly upgrade their communication infrastructure often appear. For this purpose there must be an upgrading technology, which offers support to all listed services which are tightly connected with each other.
- The cost of the guaranteed service implementation is slightly higher compared to implementing the differentiation service.

2.4 Congestions management concerning the waiting queues

One way how the network elements can manage and handle the transport routes and eliminate congestions and bottle-necks, is by using a queuing algorithm, which sorts the traffic and then decides which priority allocation method will be in use to dispatch packets to an output connection. A typical example is the Cisco's IOS software equipment, which includes the following queuing tools/mechanisms:

- FIFO queuing, which is based on the first-in first-out principle
- Priority Queuing (PQ)
- Custom Queuing (CQ)
- Weighted Fair Queuing (WFQ)
- Class-Based Weighted Fair Queuing (CBWFQ)

Each queuing algorithm is designed to solve a specific network traffic problem, and each algorithm also has an impact on the network performance. This will be described in more detail for each of the above mentioned queuing schemes in the next section.

3. Waiting queues used in present-day routers

Queues are very important parts of a router, and there are many different waiting queues. The basic waiting queues (FIFO, double FIFO, Custom Queuing (CQ), Priority Queuing (PQ), Weighted Fair Queuing (WFQ) (Yunni Xia† et al., 2007 and Anirudha Sahoo & D. Manjunath, 2007), and Class Based WFQ (CBWFQ) (T. Subash & S. IndiraGandhi, 2006 and L. L. Peterson & B. S. Davie, 2003) will be described and presented more precisely in this section. We will also describe the so-called 'worst case scenario' which can happen to VoIP when the traffic amount is high and the simplest queuing regime (FIFO) is in use.

To understand how waiting queues work, we have to say a few words about a single queue, as the simplest representative. Single waiting queue is a data structure that behaves as an ordered list, where data is inserted at one end, and output data comes out at the other end. This method is called FIFO (first-in first-out), and is presented below.

3.1 The FIFO waiting queue

The FIFO waiting queue can be illustrated with an example of people standing in a line in front of the cash register - who came first, will be the first to pay the cashier. Elements coming into a single queue from the left side in the serial order *a, b, c, d* can be removed from the queue in the same order (first *a,* then *b, c, d*). Figure 3 shows an example of a single line filling and emptying on the basis of the first-in first-out (FIFO) principle. FIFO queues are often implemented as round-robin queues, as shown in Figure 4.

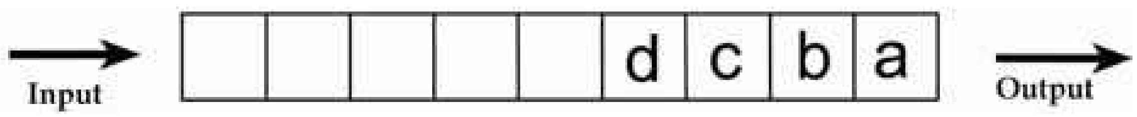


Fig. 3. The FIFO queue's filling and emptying procedure.

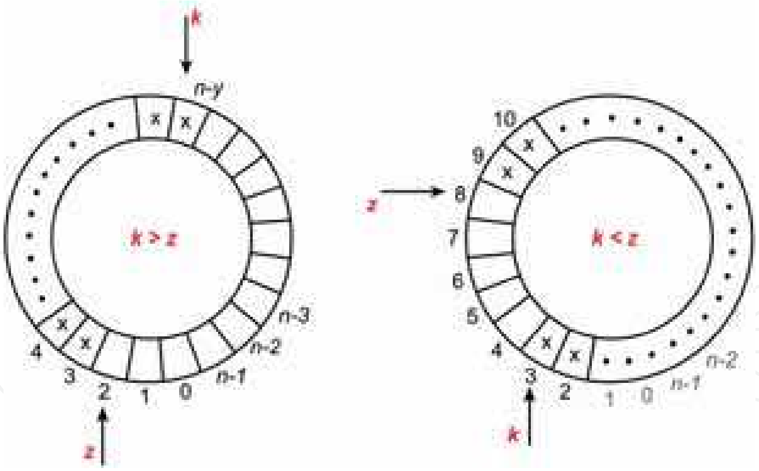


Fig. 4. The round-robin waiting queue with indexation.

Accessing items is quite limited when this method is in use. It is usable in situations where we need only the first element in the row – e.g., when printing documents. In networks, this type of a waiting queue is unsuitable for practical use, particularly with traffic flows with assigned priorities. A different and faster way than a regular FIFO is the double FIFO mechanism, where data is inserted and taken out on both sides. More about this concept is provided in the next sub-section.

3.2 The double FIFO waiting queue

The double FIFO waiting queue is a combination of two data structures (stack and single queue), which allows inserting and taking out elements on both sides. The advantage is in a faster data access, compared to a single queue or a stack. Since the circular structure operates in a round-robin mode of insertion and taking out, we are not limited with the end or the beginning of a permanently fixed structure. This is why such a concept is so flexible. Generally, we are only limited with the available size of the storage space. Operation of the double queue and its possible scenarios are illustrated in Figure 5:

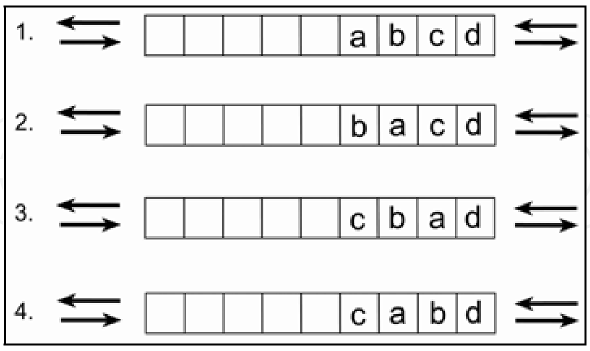


Fig. 5. The procedure of inserting the elements and the procedure of taking the elements out of the double waiting queue for the following scenarios: (1.) Element *a* is inserted from the right side, but all others are taken out on the left side in the following order; *b, c, d.* (2.) First we insert element *a* and then *b* from the left side, and then element *c* and *d* from the right side of the double waiting queue. (3.) The elements in the sequence *a, b, c* are inserted from the left side and the last element *d* from the right side. (4.) First we insert element *a* from the left side then *b* from right side, then again *c* from the left side, and finally *b* from the right side.

Operations executed upon the waiting queues must satisfy the conditions, which describe the behavior of the queue and the data in it. Operations should allow us to insert an element at the end of the queue, remove the element from the beginning, check which element is located at the beginning of the queue, and check if the queue is currently empty.

3.3 The CQ waiting queue

The primary purpose of custom queuing (CQ) is proportional sharing of the available network bandwidth among applications or organizations to avoid congestions in the network. CQ reserves the guaranteed bandwidth amount at a possible congestion point in the form of a constant ratio of bandwidth assurance, while the rest of the available bandwidth is left for other network traffic. Traffic management is performed by the allocating procedure according to the free space in the queue for each class of packets. It then starts the serving queue process in a circular manner, as shown in Figure 6. Furthermore, in each of the internal queue classes (up to 17), the amount of bandwidth, necessary for individual packets' transmission at the output connection is always calculated.

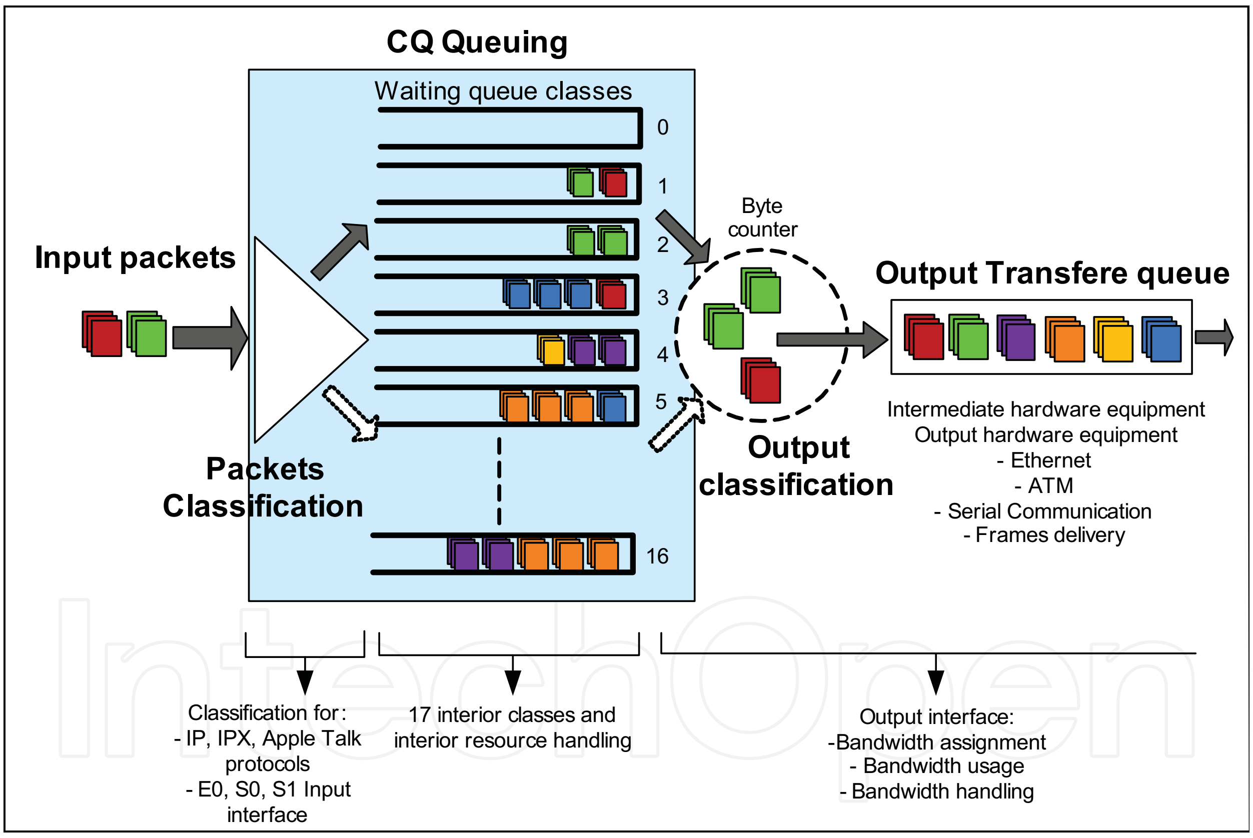


Fig. 6. Custom queuing (CQ) serves 17 internal queues in a circular manner.

Custom queuing algorithm places the packets in one of the seventeen internal waiting queues, where the queue with the index 0 is reserved for system messages, such as the so-called "keep-alive" messages and various warning messages. Queues discharging procedure is executed according to packets' weights. This means that the message with a higher priority has a smaller "weight" compared to the message with a lower priority (larger weight). The routers this way manage queues from 1 to 16 in a circular mode. Such functionality ensures an order, where no application (or group of applications) can take up more than a predetermined level of the overall bandwidth capacity, even in situations, where the link is over 90% full. CQ queuing mechanism is statically configured.

3.4 The PQ waiting queue

Priority queuing mechanism provides a smooth transition of important traffic (packets), through the network, using management at all intermediate points. PQ works by giving priority to the most important traffic. Priority queuing can be flexible regarding the allocation of different traffic parameters such as: the network protocols (IP, IPX, AppleTalk, etc.), input interfaces, the size of packets, source/destination addresses, and so on. In the PQ case, each packet (according to the entered priority in the ToS field), is classified into one of the four queues that are distinguished by different levels (priorities). The lowest level is marked with a label "low", and then the levels go up in the following subsequent order: "normal", "medium" and "high". Packets are individually sorted into appropriate queues according to the declared priority. Packets which are not classified or have not yet been classified (see the section on data flows classification) through the above described classification mechanism, automatically fall into the "normal" waiting queue as shown in Figure 7. During the data transmission the algorithm first handles the high-priority queues and then the low-priority queues.

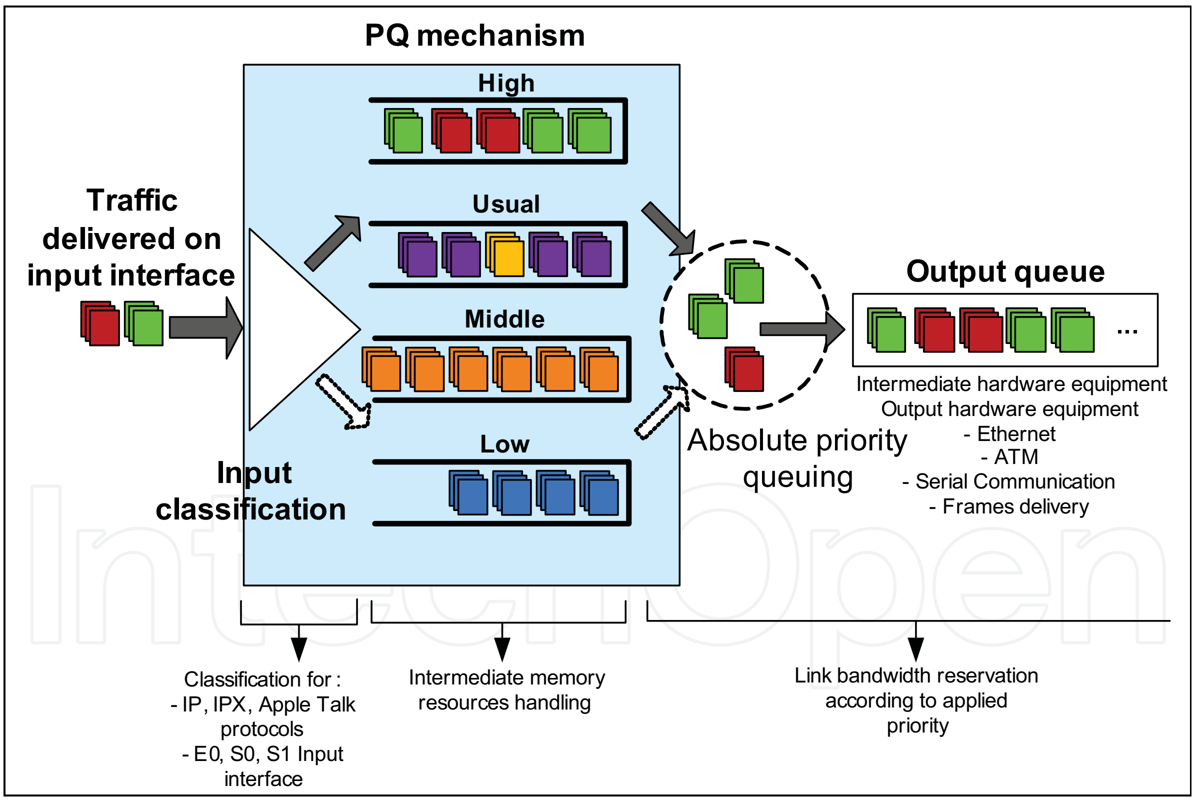

PQ mechanism
High
Usual
Middle
Low
Traffic delivered on input interface
Input classification
Absolute priority queuing
Output queue
...
Intermediate hardware equipment
Output hardware equipment
- Ethernet
- ATM
- Serial Communication
- Frames delivery
Classification for :
- IP, IPX, Apple Talk protocols
- E0, S0, S1 Input interface
Intermediate memory resources handling
Link bandwidth reservation according to applied priority

Fig. 7. Different priority classes into which the packets are inserted according to their priority.

PQ is particularly useful in situations, where the most important traffic must be treated and transmitted over different network types (WAN, LAN, etc.) first. PQ currently uses static configuration, and because of this it is not to able automatically adjust to the changing requirements in the network.

3.5 The WFQ waiting queue

In situations where it is desirable to provide a constant response time for more demanding users or applications without adding an excessive bandwidth, the ideal solution is the weighted fair queuing (WFQ) mechanism. This is an algorithm that provides bit-wise fairness, which allows each queue to be served fairly, where fairness is guaranteed by the number of bytes.

For example, let's take a closer look at two waiting queues below. The first queue has at the specific moment 100 inserted packets, while the other queue contains at the same moment 50 packets. In this situation the WFQ algorithm takes two packets from the second queue for each packet taken from the first queue. With this procedure both queues will be empty at the same time. WFQ ensures that no one queue suffers a lack of bandwidth. This way the low-level traffic can smoothly travel through the network, which represents a compromise for the majority of traffic. This increases the service efficiency, since an equal number of low-level and high-level packets are transmitted. The described operation is illustrated in Figure 8.

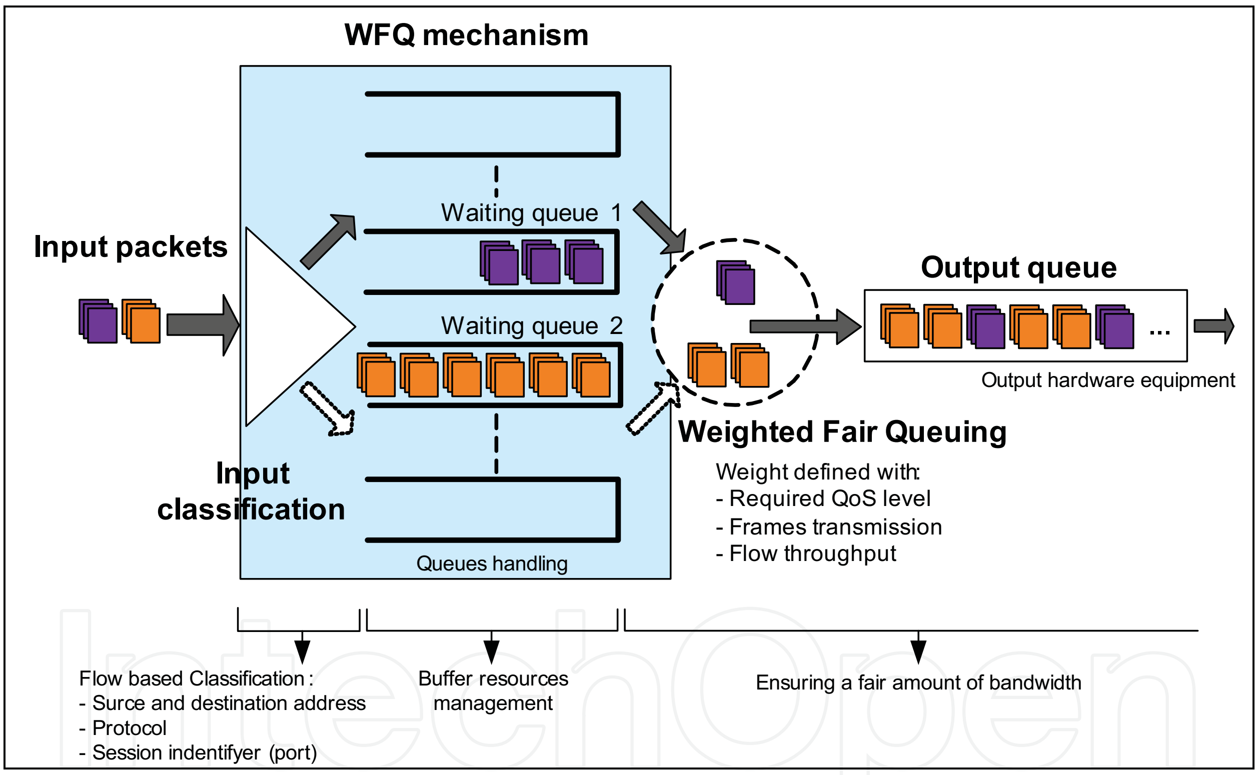


Fig. 8. The operating principle of the WFQ

The weighted fair queuing minimizes the configuration costs. Such mechanism can also automatically adapt to the changing temporary network traffic conditions. The fairness concept has been in practice very well established as the default mode on the majority of communication interfaces. The weighted amount is in the WFQ case calculated from IP priority bits, which provides a better performance for all queues. For IP priorities the values from 0 to 5 are in use (settings 6 and 7 are reserved), and the WFQ algorithm calculates how many additional services must be provided for every queue.

This method can use any available bandwidth for traffic transmission. Such operating principle is essentially different from the concept based on strict time-division multiplexing

(TDM), which simply increases the bandwidth and leaves it unused if the traffic is not present. WFQ can operate in association with the IP priority settings, as well as with the resource reservation protocol (RSVP).

WFQ algorithm also has the ability of addressing the problem of variable round-trip delay. This clearly improves the algorithms, such as SNA, LLC (logical link control) and transmission control protocol (TCP), as well removes congestions and speeds up slow connections. Results are much more predictable over the entire route, while the response time for each active flow can be reduced even for a multiple factor, as shown in simulation results in Figure 9. Time diagrams in the figure show round-trip delays for traffic without WFQ (left graph) and for the same traffic with WFQ (right graph) in milliseconds. The impact of the WFQ algorithm is more than evident.

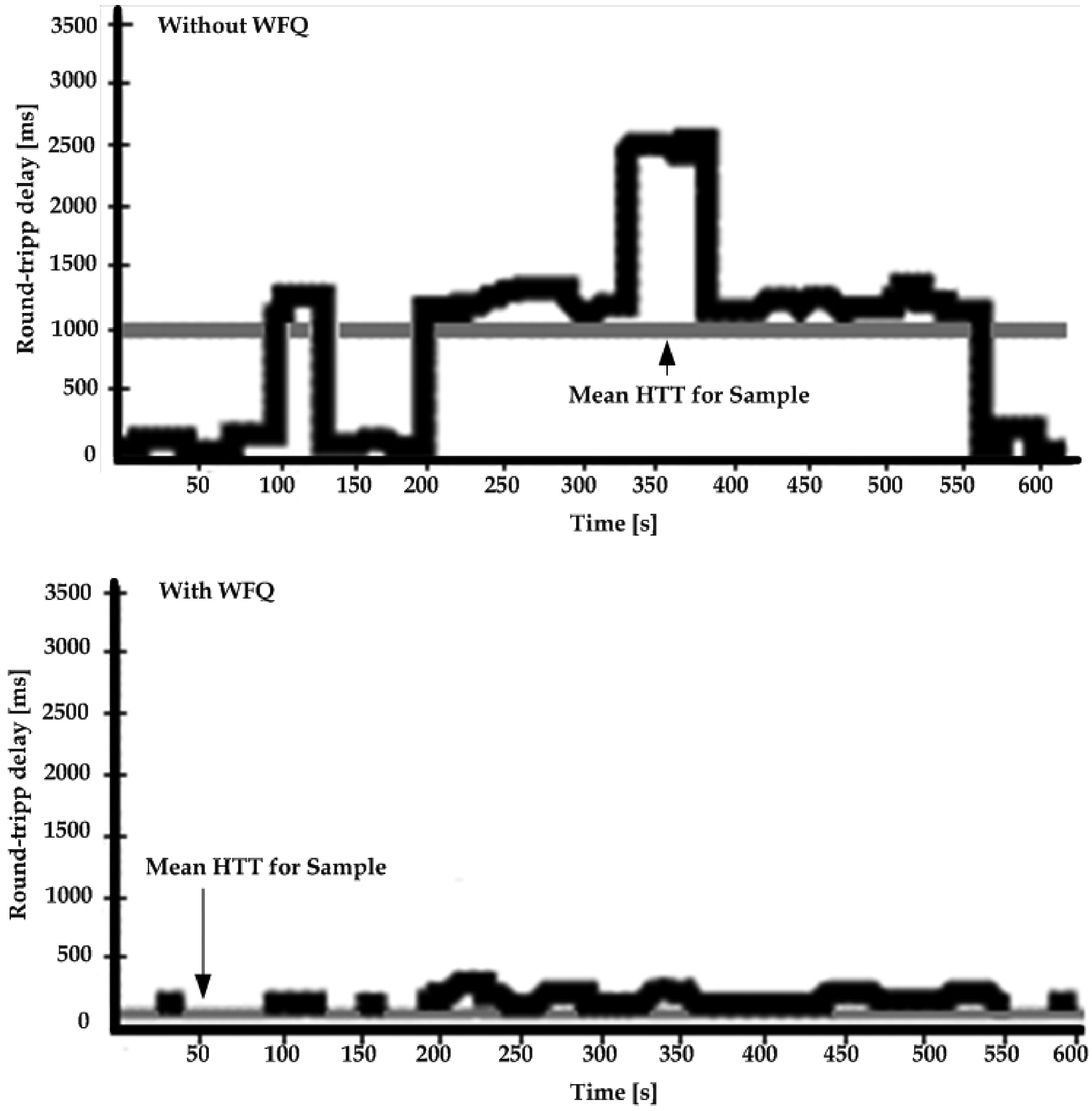


Fig. 9. Round-trip delay of transmitted, frames without WFQ (up) and with WFQ (down), using a WAN 128kbps connection.

3.6 The CBWFQ waiting queue

The Class-Based Weighted Fair queuing (CBWFQ) is a modern tool for managing congestions, and it provides a better flexibility in allocating a minimum bandwidth amount on the fair-queuing basis as well as on the basis of administrator-defined classes. Instead of providing a queue for every data stream, classes defined by the network administrator are used. If the traffic flow corresponds to an admin-defined class, it is immediately placed into such class, where it already has a reserved link bandwidth. If the traffic flow does not correspond to any of the admin-defined classes, it can use only the remaining link bandwidth, which is not reserved for any other class. For each defined class a minimum required bandwidth is guaranteed.

At this point, we could provide a concrete example, where the CBWFQ mechanism is very useful to avoid situations where more low-priority flows could overflow the high-priority data stream. A typical example is video-stream transmission, which requires almost half of the available bandwidth on the T1 connection. A sufficient amount of bandwidth could also be provided with the WFQ mechanism, but only in cases where only two data streams are present. In cases where more than two traffic flows are present at the same time, the video session will get less bandwidth (in the WFQ case), as the WFQ mechanism works on the principle of fairness. If, for example, 10 streams at the same time require T1 link bandwidth, the video session stream will get only one tenth (1/10) of the whole T1 link bandwidth, which is unacceptable for a video session. Even if IP priority 5 is set for the video session, the situation would not change significantly. The queuing mechanism must reserve at least half of the T1 link bandwidth for the video session. This can be ensured with the CBWFQ queuing mechanism. The network administrator determines the class, and places a video meeting into such class. This indicates that the router must provide 768 kbps service for such class, which is exactly half of the total bandwidth of the T1 connection. The needed bandwidth is thus allocated to the video. The remaining bandwidth is used for other (unclassified) data streams. These classes are serviced through the use of stream-based WFQ algorithm, which allocates the remaining bandwidth to other applications (in our case the remaining half of the T1 connection's bandwidth).

It should be noted that low latency queues (LLQ) can be marked so that the actual priority queue is differentiated. Such feature is known as the PQ-CBWFQ, which is a priority class-based weighted fair queuing. Low latency queuing allows a specific class to be served as a strict priority queue. Traffic in such classes will be serviced before all other traffic placed in other classes, and at the same time the necessary amount of bandwidth will be guaranteed. All traffic that is above the level of bandwidth reservation is simply discarded.

With the CBWFQ a minimum amount of bandwidth can be reserved for a given class. If there is some free bandwidth available, it can be used by such class. Similarly, when a class does not use all the guaranteed bandwidth, it can be used by other applications.

4. Hybrid waiting queues

Because different queuing mechanisms have different advantages, our idea was to combine different queuing mechanisms and join their positive (but also negative) properties into new hybrid queuing methods. The aim of hybrid methods is to concentrate the most possible positive properties of individual methods. Many different hybrid queuing methods are possible. This section provides descriptions of hybrid queuing disciplines for our proposed combinations CQ-CBWFQ and WFQ-CBWFQ (Sasa Klampfer et al., 2009 and Sasa Klampfer et al. 2007) as well as for the known PQ-CBWFQ introduced by Cisco Systems. Each of these methods was evaluated with simulations, as described in section 6.

The negative side of hybrid methods is duplication of the memory of the mechanism which forms a queue. It is a well-known fact that every memory element and its size involve certain latency or delay for traffic which goes through these interfaces. The higher the number of these interfaces, i.e. waiting places, the bigger are the delays, which is not a desirable feature for time-sensitive applications (VoIP, video conference, etc.). This is why we have to make a compromise between the number, size and length of the intermediate buffers to avoid excessive data spillage (or data loss) when a buffer is too small, and to also avoid scenarios where the buffers are too big, and are increasing the delay. This aspect and the so-called jitter effect will be presented in detail in Section 5.

4.1 The CQ-CBWFQ hybrid waiting queue

This hybrid method combines the properties of the custom queuing (CQ) and the CBWFQ mechanisms (Figure 10). In the first phase the custom queuing allocates the available bandwidth among all active network applications so that congestion cannot appear. This is the main reason why we combined these two queuing schemes. In first phase we try to avoid congestions with custom queuing. In the CQ step traffic is managed by assigning weighted amounts, and is arranged into 16 queues. Once the packets are sent to the output CQ interface they arrive to the CBWFQ input interface. 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 with fixed amounts of bandwidth. All the advantages of the CBWFQ are retained. With this method we reduce the delays within the network, which is not the case with the ordinary CQ scheme.

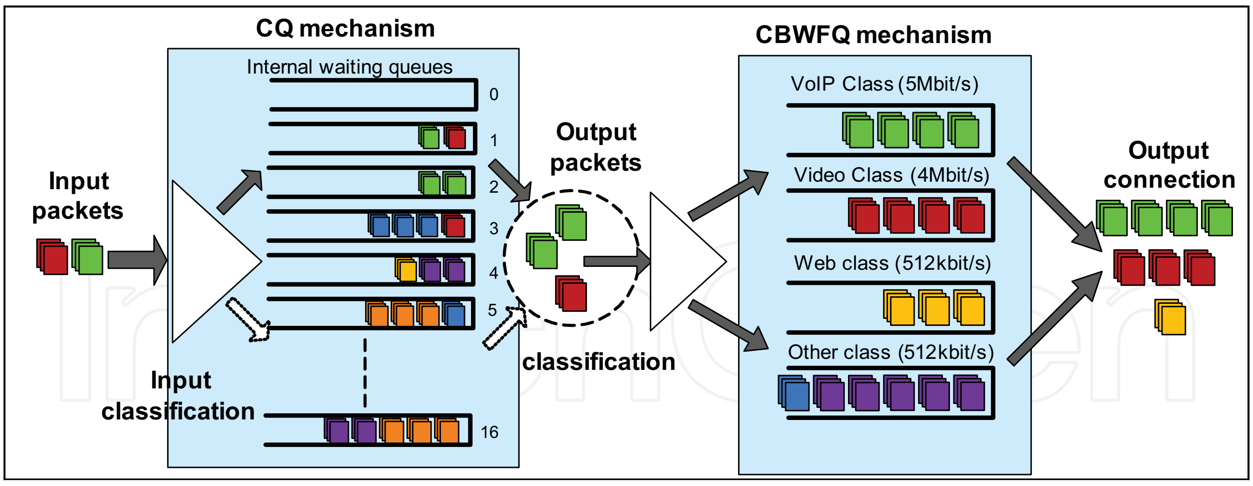


Fig. 10. The hybrid queuing mechanism consisting of the CQ and the CBWFQ regimes.

4.2 The PQ-CBWFQ hybrid waiting queue

This mechanism consists of two previously mentioned queuing mechanisms; the priority queuing mechanism and the admin class-defined queuing mechanism (CBWFQ). Since the properties of both mechanisms that construct the PQ-CBWFQ method have been already mentioned in previous sections, we should now take a look at the hybrid mode concept shown in Figure 11.

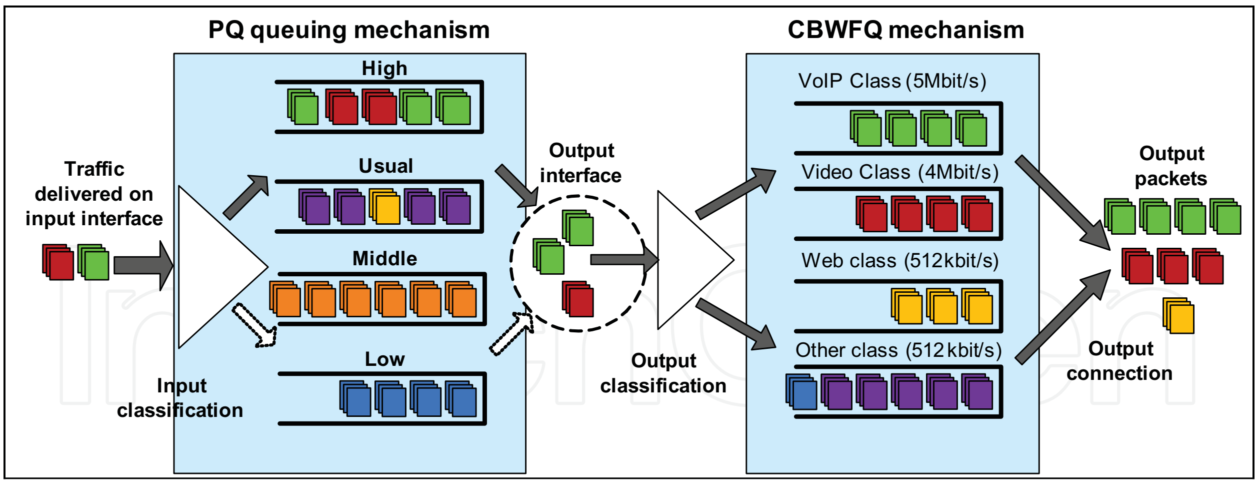


Fig. 11. The hybrid queuing mechanism consisting of the PQ and the CBWFQ regimes.

In the first step the traffic is arranged into waiting queues according to the priorities set in individual packets' ToS fields. According to the ToS priorities the packets are arranged by their importance to four different internal priority queues. In the second step, the output interface algorithm first serves the highest-priority data stream (packets that are in the queue with the highest importance) and then all other lower ranking queues. Once the packets appear at the outgoing interface of the priority queuing mechanism, they are again scheduled into admin-defined classes of CBWFQ mechanism. Such defined classes already have the needed bandwidth pre-reserved as set by the network administrator. This way the packets at the CBWFQ mechanism output interface do not need to fight for bandwidth, as it is guaranteed in advance. This accelerates the transfer of high-priority flows, and such flows become independent of all other lower-priority flows.

We can take the VoIP traffic as an example. VoIP traffic will already have provided a sufficient pre-allocated bandwidth, meaning that its pre-reserved bandwidth cannot be used by any other application or other traffic flow, whose classification does not fulfill the terms of class reservation. This way the output connection can transmit even the lower-priority traffic parallel to the high-priority traffic, but only in quantities and at rates established by the remaining bandwidth. This hybrid method (PQ-CBWFQ) is representative of low latency queues (LLQ) (S. Büchel, 2004). Low latency queuing mechanism allows a class that is served as a strict priority queue. Traffic in such class will be served before all other traffic in the remaining classes. Bandwidth-amount reservation is also guaranteed in this case. All traffic which is above the level of bandwidth reservation is simply discarded. Furthermore, the same hypotheses regarding the compromise between the choice of the size and the number of intermediate buffers mentioned in the above case are valid also with this method.

4.3 The WFQ-CBWFQ hybrid waiting queue

This mechanism consists of the weighted fair queuing (WFQ) and the class-based weighted fair queuing (CBWFQ). With this method we can show how important the first step is to ensure fairness for all applications where the internal WFQ queues are emptied by the principle of fairness (see section about the WFQ). At this step, we ensure undisturbed flow throughput for all active applications that appear at the WFQ mechanism's outgoing interface. In the next step the CBWFQ classification takes care of proper packets' assignment into admin-defined classes. This way every application at the first stage gets a fair treatment, and in the second phase high-priority applications get its own classes with the pre-reserved bandwidth. The rest of the bandwidth is left for all other active applications. The fairness and fluidity movement apply for all active applications (Figure 12).

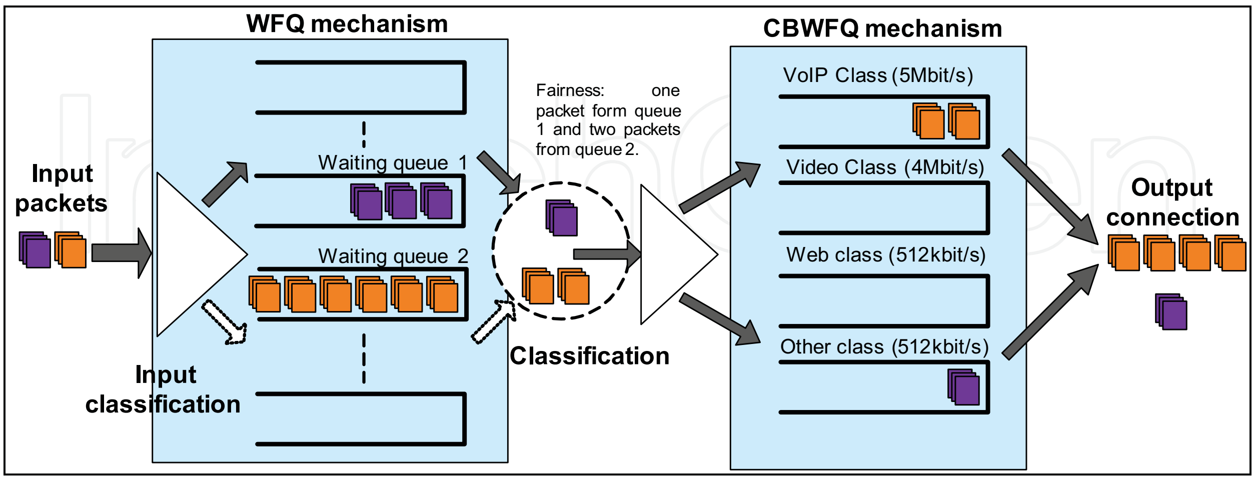


Fig. 12. The hybrid queuing mechanism consisting of the WFQ and the CBWFQ regimes.

WFQ is suitable for operating with IP priority settings, such as Resource Reservation Protocol (RSVP) (A. Kos & S. Tomazic, 2005), which is also capable of managing round-trip delay problems. This is the main reason, why we have combined the WFQ with the CBWFQ mechanism. 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), to a physical unit (PU) 2.0 or 2.1, or connect a mainframe to a front-end processor (FEP) in another Virtual Telecommunications Access Method (VTAM) domain (http://www.cisco.com/en/US/tech/tk331/tk332/tk126/tsd_technology_support_subprotocol_home.html), logical link control (LCC) (http://www.erg.abdn.ac.uk/users/gorry/course/lan-pages/llc.html), or transmission control protocol (TCP). WFQ-CBWFQ is at the same time capable of accelerating slow features and removing congestions in the network (merged positive properties of both queuing schemes). Results become more predictable over the whole routing path, while Ethernet delays can be greatly decreased (see the section with the simulation results) compared to other queuing disciplines (CQ, PQ, WFQ). The WFQ and the CBWFQ queuing combination can represent the best solution (merged positive properties of both methods) for reducing the Ethernet delay. This assumption is confirmed in Section 6, where the simulation results show the delays in the network are most reduced with the WFQ-CBWFQ combination.

5. Reducing the VoIP jitter by decreasing the router’s buffer lengths

Our intention in this section is to show the reader how queuing combinations affect VoIP traffic quality, especially in terms of Ethernet delay and jitter (Mansour J. Karam & Fouad A. Tobagi, 2001). With the results of our simulation-based research (Section 6) we can prove the usefulness of our hybrid concept for decreasing Ethernet delay. Ethernet delay was rapidly decreased when the hybrid queuing combinations WFQ-CBWFQ and PQ-CBWFQ were used. Though the jitter is highly increased with such combinations the delay is still within the useful limits. By using the proposed queuing combination it is possible to minimize the Ethernet delay for IP-based time-sensitive applications, including VoIP (Cole R. Rosenbluth, 2000, and Frank Ohrtman, 2004). Because we combine different queuing disciplines, the waiting queues are also combined. This means that buffer lengths have been in many cases doubled or at least extended. This is why we have included this section about buffer length (L. Zheng & D. Xu, 2001, M. Kao, 2005, and TIPHON 22TD047, 2001) influence on the VoIP jitter. Jitter can also cause some VoIP packets falling out, because of which the quality of the conversation over VoIP can be significantly reduced. The use of such method results in an increased VoIP round trip delay as well as higher packet delay variation (jitter), which could with an improper buffer length exceed the limits of acceptability. The method is thus not acceptable in the VoIP case where limited delays are required. For this reason we also present the possibilities of reducing such delays by using proper buffer lengths. One approach involves reducing the buffer length, which however also must not be too short so that other packets do not fall out. The proper solution for such a case will be presented in this section.

5.1 Introduction to jitter

Jitter is defined as a variation in delays between the audio packets (VoIP), which can occur due to congestions in the network. On the transmitter side, the packets are sent in a continuous stream where they are equally time-gapped between each other. The jitter is caused by network congestions, improper selection of the queuing mechanism or improper network element's (router) configuration. Such a scenario is shown in Figure 13.

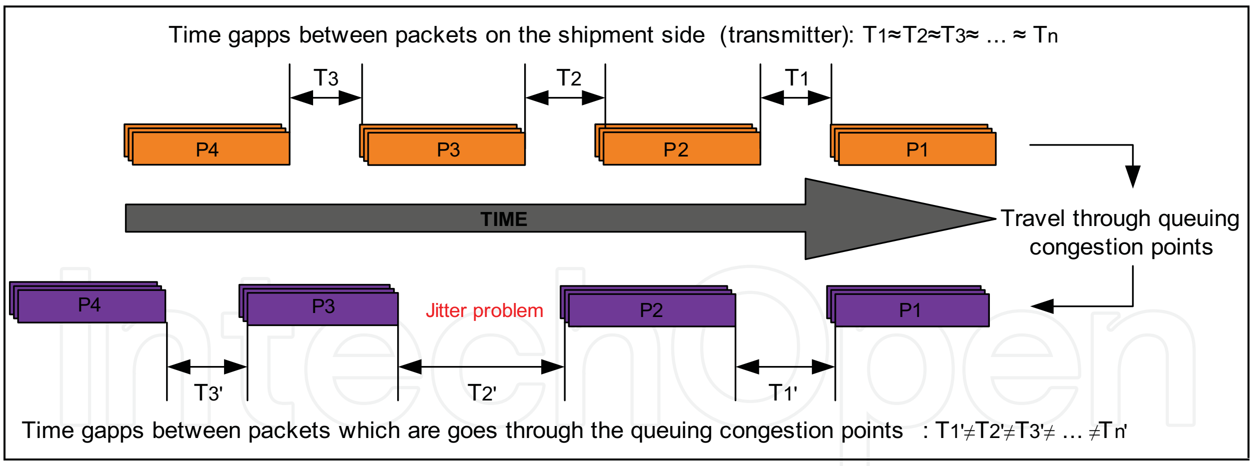

Time gapps between packets on the shipment side (transmitter): T1≈T2≈T3≈ … ≈ Tn
T3
T2
T1
P4
P3
P2
P1
TIME
Travel through queuing congestion points
P4
P3
Jitter problem
P2
P1
T3'
T2'
T1'
Time gapps between packets which are goes through the queuing congestion points : T1'≠T2'≠T3'≠ … ≠Tn'

Fig. 13. Schematic description of the jitter

When a router accepts an audio data stream via real-time protocol (RTP-Real Time Protocol) which is provided for VoIP, it must ensure jitter compensation. To eliminate jitter effect, the so-called playout delay buffer must be used. Such an output memory is sometimes also called *de-jitter buffer* (Fig. 14). The packets are then sent in a regular sequence to the digital signal processor, which reproduces the sound.

In cases when the jitter is so big that the packets sent to the "de-jitter buffer" fall out of the allowed area, the packets are simply discarded. Rejecting packets leads to poor conversation quality. For smaller packet loss effect (e.g., individual packets) the DSP interpolation algorithm is provided; it replaces the lost packets with an alternative content, which is usual neighboring a successfully accepted packet (Cisco Systems, Jitter data sheet, 2003).

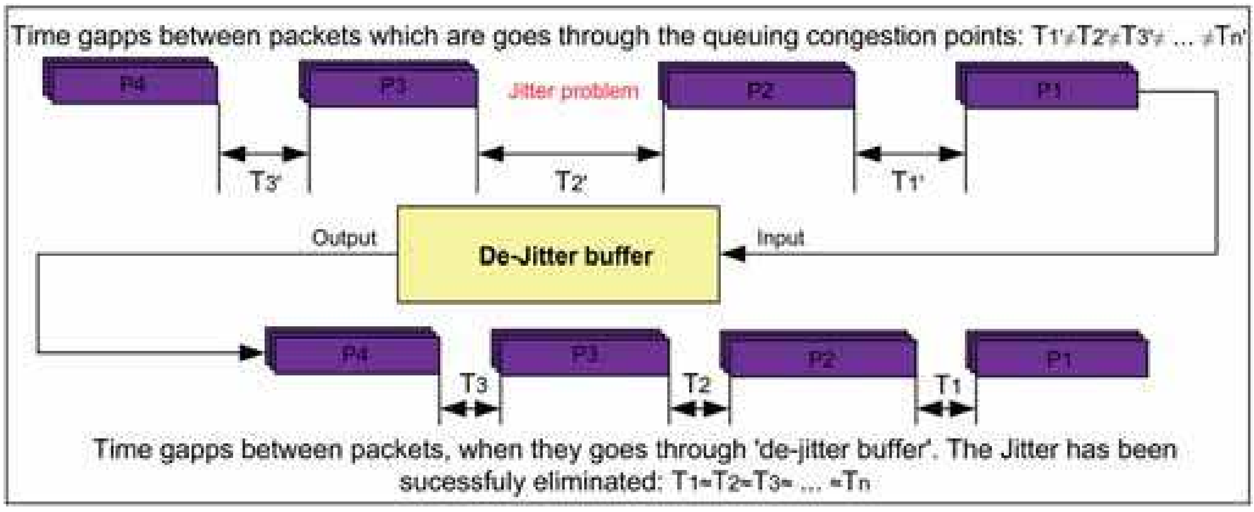

Time gapps between packets which are goes through the queuing congestion points: T1'≠T2'≠T3'≠ ... ≠Tn'
P4
P3
Jitter problem
P2
P1
T3'
T2'
T1'
Output
De-Jitter buffer
Input
P4
P3
P2
P1
T3
T2
T1
Time gapps between packets, when they goes through 'de-jitter buffer'. The Jitter has been sucessfuly eliminated: T1=T2=T3= ... =Tn

Fig. 14. De-jitter buffer's application

5.2 Encapsulation and its connection with jitter

Generally speaking the simplest way to detect jitter is at the interfaces of the existing routing equipment, because we there have full access. The success of eliminating this phenomenon is largely related to the packets and connection's encapsulation type. ATM networks are well known for being relatively robust regarding jitter occurrences. The main reason for this can be found in its asynchronous data transfer mode implementation. If the ATM is configured correctly, jitter practically does not occur. When we deal with point-to-point (P2P) applications, which use Point to Point Protocol (PPP) encapsulation, the jitter is always present in the form of serialization delay. Such delays can be controlled in a simple and easy way using fragmentation and interleaving in the PPP connection mechanism.

Recently, two queuing mechanism types came to be used within IP networks which are able to reduce the jitter in the VoIP session case. Both belong to the so-called low latency queuing mechanisms:

- IP RTP Priority Queuing
- PQ-CBWFQ (LLQ)
- PQ-WFQ (LLQ)

6. Examples of simulation for testing queuing mechanisms

In this section, we will first present the OPNET Modeler tool (OPNET Modeler Technical Documentation, 2005 and S. Klampfer – Diploma Thesis, 2007), used in our simulation experiments. The simulations include VoIP applications where we have used the G.729a voice encoder scheme. This is an 8 kbps Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP) speech compression algorithm approved by ITU-T. G.729 (G.729 Datasheet, 2005).

In our experiments we prepare simulation examples with typical network topologies (Morgan Kaufmann, 2006 and Tadeusz Wysocki et al., 2005), network architecture, routers, connection types, used application types within the network, statistical distributions of the simulated application traffic, and number of users. These parameters influence traffic congestions according to their intensity and used application types (ITU-T Y.1541 Network Performance Objectives for IP Based Services, 2001 and ITU-T SG12 D74 IP Phones and Gateways: Factors impacting speech quality, 2002). In these simulation experiments, we have included three of the most useful queuing methods, where we combined 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.

Using the simulation results we will present the method with the best results on the simulation test bed regarding VoIP jitter and VoIP delay. We are going to make a comparison between the well known PQ-CBWFQ and the best of the proposed hybrid solutions (WFQ-CBWFQ). Our main goal is to determine the factor, which will tell us how much more suitable is a specific queuing method for VoIP traffic class, during whole routing procedure from end-to-end.

6.1 A short description of the OPNET modeler simulation tool

OPNET Modeler represents (OPNET Modeler Technical Documentation, 2005) one of the most useful simulation tools in the area of communications industry. The tool enables designing and studying telecommunication infrastructures, individual devices, protocols, applications, etc. It is based on object-oriented modeling. Individual modules included in specific libraries represents models of real building blocks used in real communication infrastructure. The created simulation models thus present a good approach compared to equivalent real networks. Support for modeling of all types of communication networks, included in advanced technologies such as Wi-Fi, UMTS, GSM, Fast Ethernet, etc. is also available. The tool allows modeling of PSTN, ISDN, xDSL, as well as optical networks. The user interface is based on a series of hierarchical graphic interfaces, which enable editing at each stage, as well illustrate the structure of protocols, devices and networks. The tool also supports animations, which can provide a better understanding of the simulation results and events appearing in the simulated networks. The user can also observe individual packets traveling during simulation execution (slow motion support). OPNET Modeler offers a rich existing model library of standard equipment and protocols, including the possibility of modeling new or upgrading existing ones, which can be done by using code level in C/C++ C/C++ programming languages.

6.2 Example 1: hybrid queuing mechanisms

Let's assume a company has VoIP quality problems in their communication infrastructure, and they call communication experts to solve these problems. The experts use the OPNET Modeler simulation tool to model a network structure of a private company's network on the level of links, equipment, applications, etc. Their goal is to find the optimal setup for communication equipment which would solve the problems.

Different queuing mechanisms and the proposed hybrid concepts were tested within a simulated network shown in Figure 15 for proving the advantages and the disadvantages of the proposed hybrid queuing methods. The main goal of these simulations is improving the network performance in terms of the VoIP end-to-end delay, Ethernet delay and jitter. Different queuing schemes are used in order to find the most appropriate one for the VoIP application's traffic.

VoIP traffic flows were set up randomly among all 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 through a 10BaseT connection and through a 16 port switch on the IP Cloud, as shown at the top of Figure 15. 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 with a 16 port switch, where it is then connected to an IP Cloud. Users use different application clients such as VoIP, Web, and FTP.

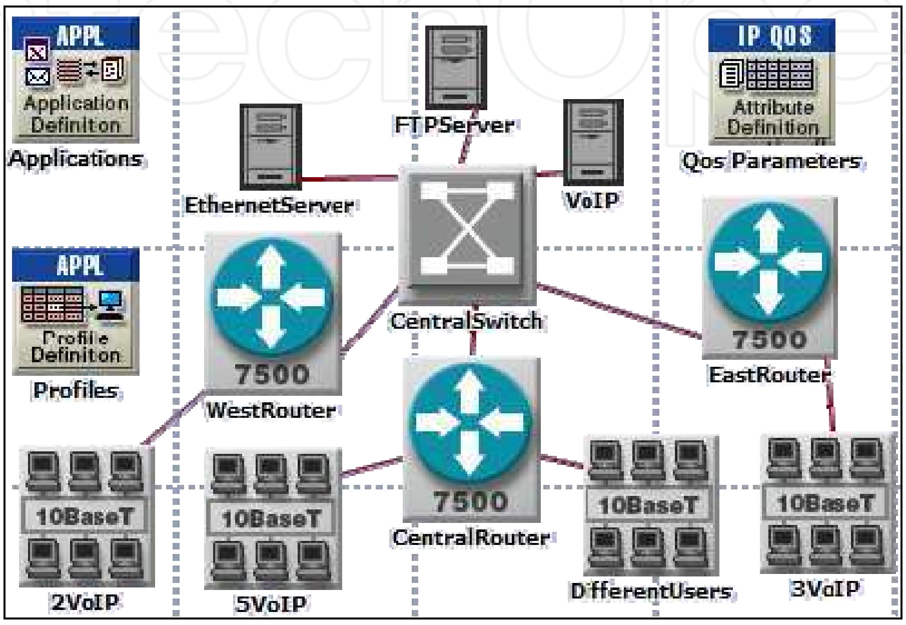


Fig. 15. The network simulation structure

Web and FTP applications are applied only for creating low-priority traffic flows. Such applications are defined with the 'Applications' node shown at the top-right side of Figure 15. Using the 'Profiles' node (besides the applications node) the client profiles are defined, as well as the User Equipment's (UE's) tasks and which UE application can be used. In IP the 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, shown in Table 1, where the first-one, with 9Mbit/s, belongs to the VoIP traffic, and the second and the third-one belong to low-priority FTP and HTTP traffic flows with 512 kbit/s bandwidths defined for each class.

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

Table 1. The application's traffic classes

6.2.1 Example 1: simulation results

Simulation results were collected after each successive simulation run, both for ordinary and hybrid queuing methods described above. The obtained results even in a graphical form clearly show the impact of each combination on Ethernet delay and jitter. The impact is obvious when each queuing combination is compared to a basic queuing scheme (PQ with PQ-CBWFQ).

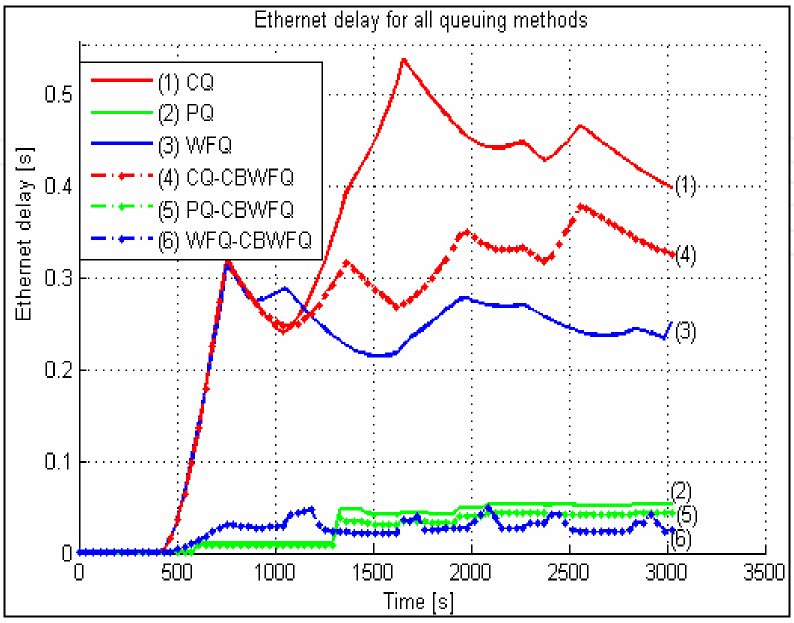


Fig. 16. The Ethernet delay for all queuing methods, where (1) presents Ethernet delay for custom queuing (CQ), (2) priority queuing (PQ) Ethernet delay, (3) Ethernet delay for weighted fair queuing (WFQ) method, (4) CQ-CBWFQ combination, (5) PQ-CBWFQ, and (6) Ethernet delay for WFQ-CBWFQ hybrid queuing method.

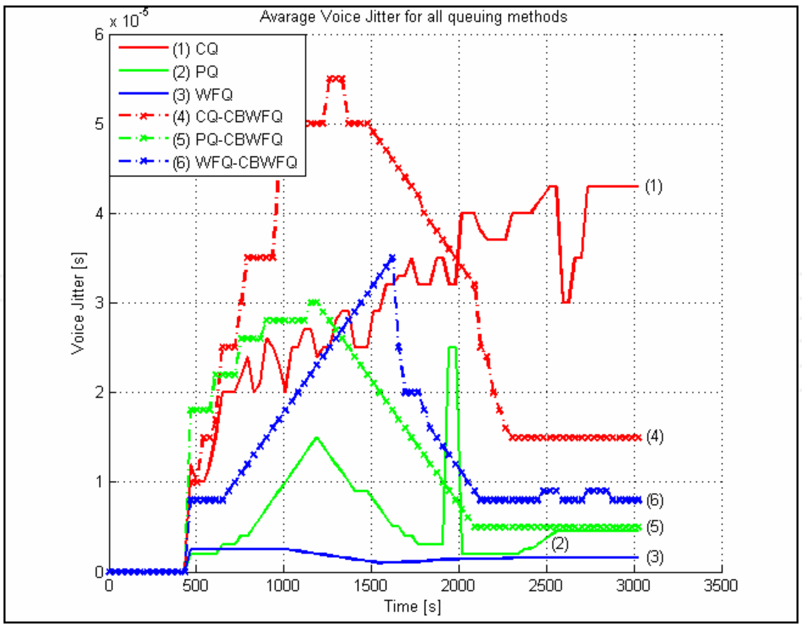


Fig. 17. The voice jitter for all queuing schemes, where (1) presents the custom queuing (CQ) jitter delay, (2) the priority queuing (PQ), (3) the weighted fair queuing (WFQ), (4) the CQ-CBWFQ, (5) the PQ-CBWFQ, and (6) voice jitter delay for the WFQ-CBWFQ hybrid queuing method.

Comparing a specific hybrid queuing method (Figure 16) with a specific ordinary queuing method (example: (1)-CQ with (4)-CQ-CBWFQ) we can see the Ethernet delay reduction in the CQ-CBWFQ case. The same is true for a comparison between (2)-PQ and (5)-PQ-CBWFQ, as well as between (3)-WFQ and (6)-WFQ-CBWFQ. WFQ-CBWFQ is obviously the best combination for reducing the average Ethernet delay within the network. Besides its advantages, the disadvantage of the method is the jitter issue. The best results are obtained with the PQ-CBWFQ queuing discipline (Fig. 16). Results of none of the 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 Figure 16, the PQ-CBWFQ combination works fine for the 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.

Figure 17 presents voice jitter for all queuing schemes (hybrid and ordinary). In this case, the WFQ scheme is the smoothest and has the lowest jitter value. Speaking generally, the CQ-CBWFQ and WFQ-CBWFQ queuing schemes are the worst possibilities. The latter gives the best results in the Ethernet delay case. However, such jitter can negatively affect the VoIP speech quality. As we have expected; the PQ-CBWFQ also reaches low jitter values, which is desirable for VoIP and other real-time or near real-time applications. In any other queuing scheme, jitter values are higher but still acceptable in the VoIP case where the maximum value reaches only 40ms. The critical jitter limit is 150ms. Any delay in voice application 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. 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 in 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.

6.3 Example 2: PQ and CQ mechanisms compared to PQ-CBWFQ

The test network consists of remote servers, VoIP and Web clients (spread across specific geographic areas), switches, routers, etc. With the “IP Cloud” element we describe some properties of the entire wide area network, such as delay, packet loss, etc. The whole network structure (see Figure 18), public network, individual users, etc. is connected through an IP cloud to remote servers in the WAN network.

Four external LANs (LAN1, LAN2, LAN3 and LAN4), where each of them contains of 50 VoIP users, establish connections to the VoIP users at the other end of the WAN network using a 10 Mbit/s wired broadband connection. In each of the local area networks there are also World Wide Web (WWW) users, which exploit a part of the available bandwidth. These users can affect the VoIP traffic delay, but only in the cases, when inappropriate QoS and queuing mechanisms are used. A fast connection allows exchange of large amounts of data between units, and at the same time ensures small time delays, which is crucial for the VoIP sessions.

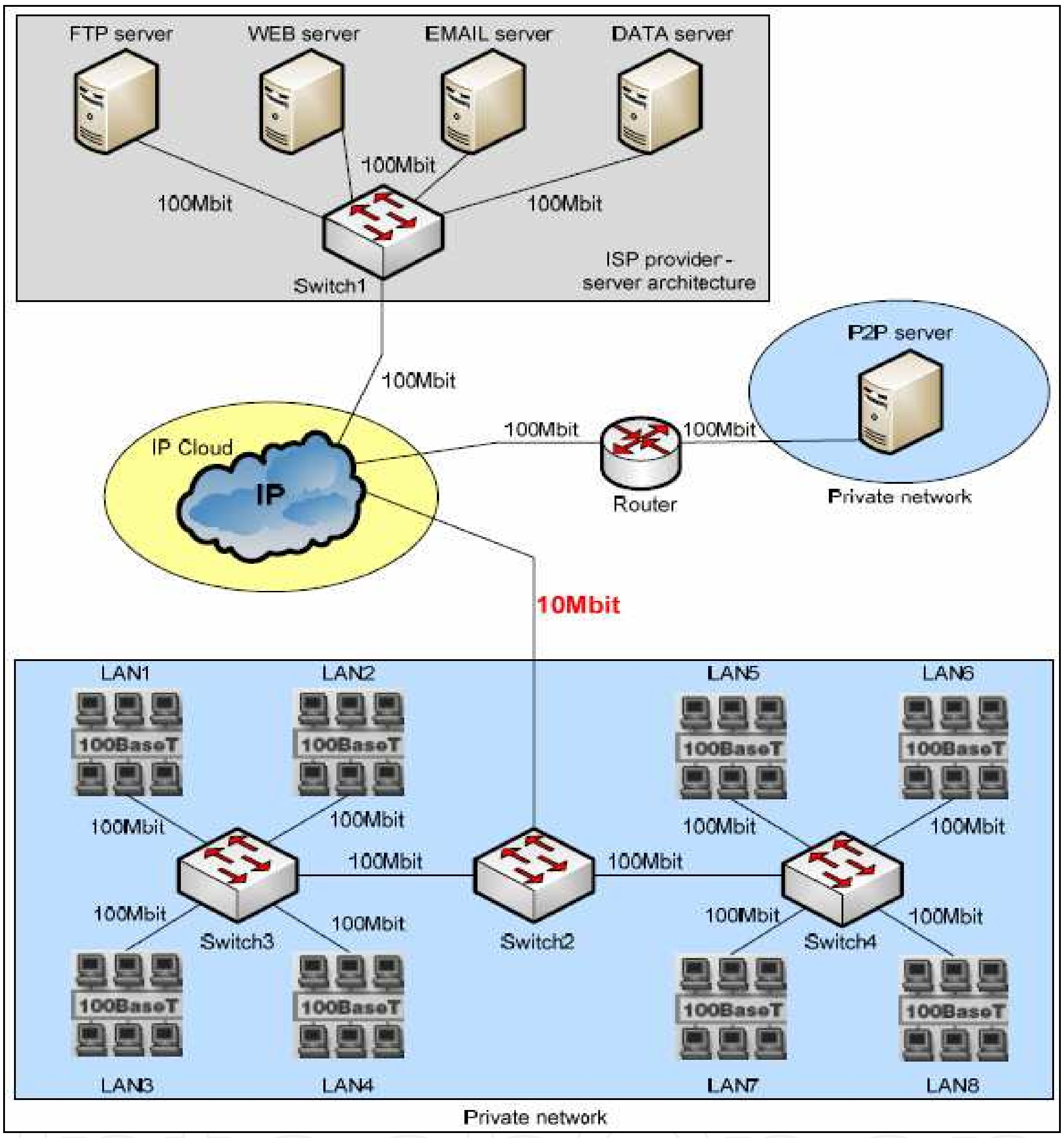


Fig. 18. Simulation structure of the wide area network

The wide area network (WAN) simulation structure is shown in Figure 18. All active applications are designed in the OPNET Modeler simulation tool in the form of three different scenarios. The first scenario consists of the CQ queuing method, second only of the PQ queuing method; while the third scenario consists of the PQ-CBWFQ queuing regime, which belongs to the low latency queuing group. Through a comparison of all mentioned scenarios we obtain the following results.

6.3.1 Example 2: simulation results

During network simulations where we used different queuing methods for IP traffic we have measured the traffic delays corresponding to each queuing method. Results are presented in Figure 19. Curves (1), (2) and (3) illustrate the average VoIP traffic delay for the used CQ, PQ and PQ-CBWFQ queuing mechanisms.

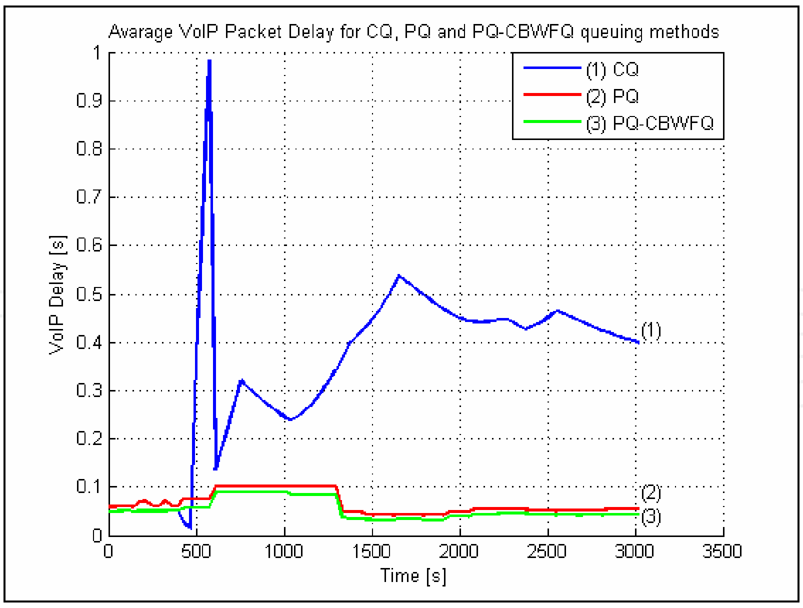


Fig. 19. The average VoIP traffic delay for the used CQ, PQ and PQ-CBWFQ queuing mechanisms.

Based on the simulation results shown in Figure 19, we have calculated and determined the relationship factors, which describe how much is the chosen method of classification for the specific observed network traffic better than the basic method. We have used CQ as a basic reference method in comparisons. Relationships were calculated by averaging CQ delay of VoIP traffic and dividing it by averaged delays of the VoIP traffic with other methods (Table 2).

Method	**CQ**	**PQ**	**PQ-CBWFQ**
Average delay [s]	0.346488	0.064444	0.052956

Table 2. The calculated average delay for each of the individual waiting queue methods.

Methods in comparison	$\frac{CQ}{PQ}$	$\frac{CQ}{PQCBWFQ}$	$\frac{PQ}{PQCBWFQ}$
Relationship factors	5.37	6.54	1.21

Table 3. Calculated relationship factors, which describe the usefulness of an individual method in comparison to others.

From the calculated factors we can see, that the PQ and PQ-CBWFQ queuing mechanisms are most suitable for time-sensitive applications such as for example VoIP. From their comparison we can conclude that the PQ method is better for a factor 5.37 than the custom queuing method, and PQ-CBWFQ combination is for a factor 6.54 better than the basic CQ method. In simulation results this can be observed in the form of the smallest delays for a specific application. In PQ and PQ-CBWFQ cases, the VoIP delay is lower than in the CQ case, and it does not exceed the critical delay (150ms), which represents the limit where the human ear can detect it. When both sophisticated methods are compared, the PQ-CBWFQ is for a factor 1.21 better than PQ queuing regime. Simulation results show how important the right choice and configuration of the queuing mechanisms are for time-sensitive traffic.

6.4 Example 3: testing ordinary queuing mechanisms (CQ, WFQ, CBWFQ, MWRR, DWRR)

Test simulation network architecture is an imitation of a real network belonging to a private company. Our main goal is to improve the network's performances. The highest level in Figure 20 represents the network server architecture offered by the internet service provider (ISP). Servers' subnet consists of five Intel servers where each of them has its own profile, such as; web profile (web server), VoIP, E-mail, FTP and video profile. These servers are connected through a 16 port switch and through a wired link to the private company's router. Company's network consists of four LAN segments including different kinds of users. In the west wing of the company are the VoIP users who represent technical support to the company's customers. In the south wing of the company is a conference room where employees have meetings. Two places here are meant for two simultaneous sessions. In the north wing there is a small office with only 10 employees who represent the development part of the company, and they use different applications needed for their work. For example, they are searching information on the web; calling their suppliers, exchanging files over FTP, and so on. The remaining east wing includes fifty disloyal employees who are surfing the net (web) during work time, downloading files, etc. (heavy browsing).

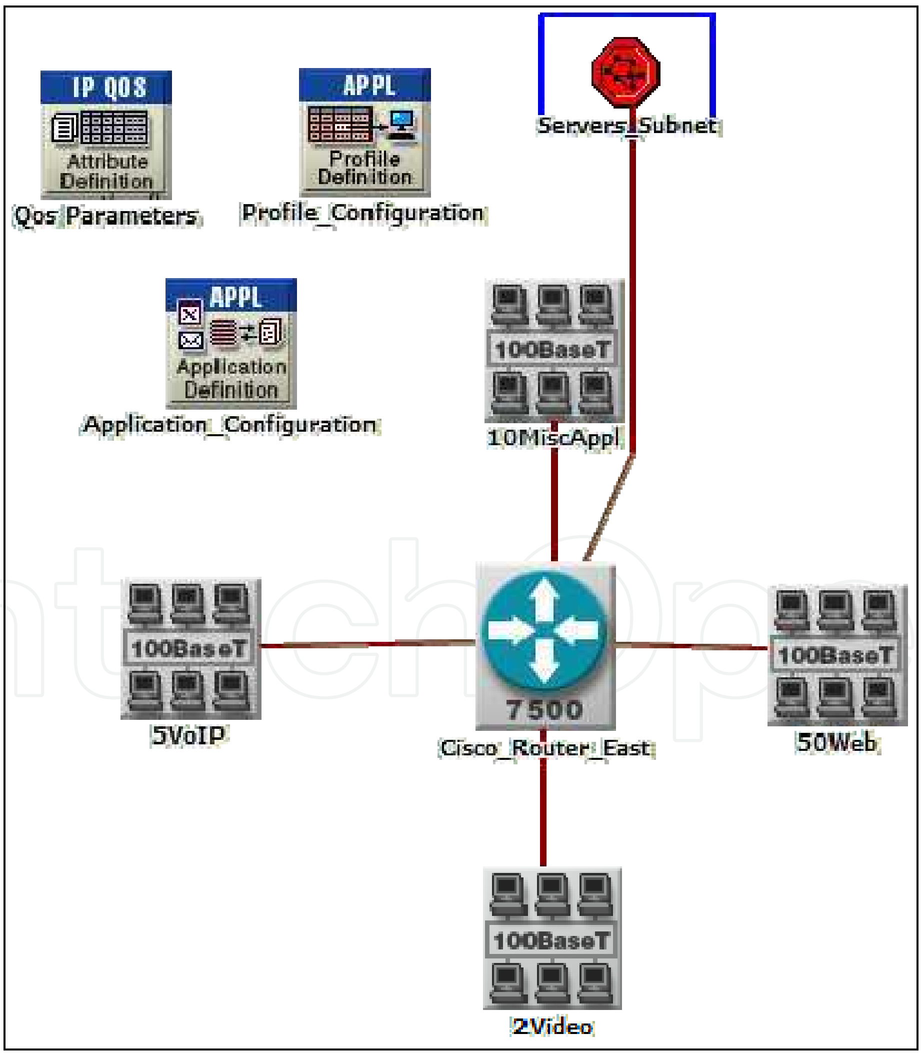


Fig. 20. A wired network architecture, which is an imitation of a real network.

Each of the company's wings is connected through a 100BaseT link to the Cisco 7500 router. This router is further connected to the ISP servers' switch through a wired (VDSL2) 10Mbit/s ISPs' link. Connections between servers and the switch are also type 100BaseT. The wired link in this case represents a bottleneck, where we have to involve a QoS system and apply different queuing disciplines.

Application	Number of users
Heavy web browsing	50 clients
FTP	4 clients
Video conferencing	7 clients
VoIP	5 clients
E-mail	1 client

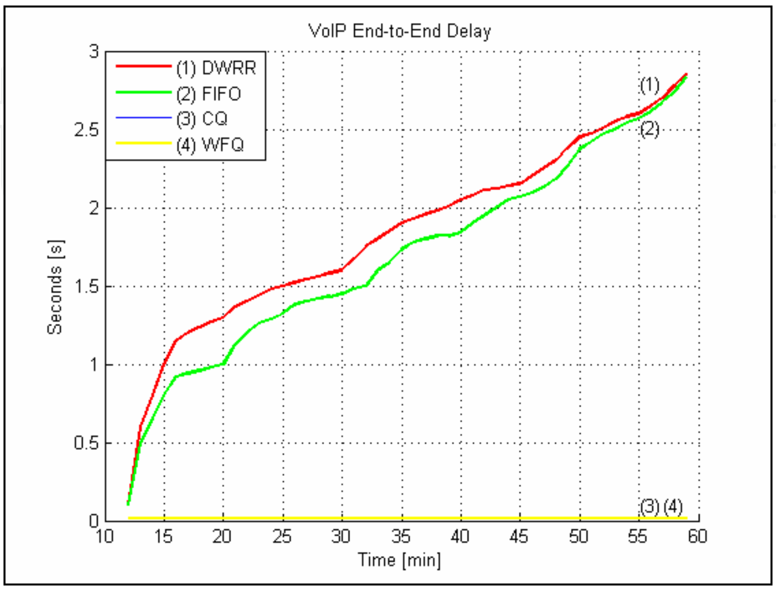
Table 4. User distribution

We have created six scenarios; in the first scenario, we have tested the custom queuing discipline, which represents the basis for comparison with the WFQ (second), the MWRR (third), the DWRR (fourth), the CBWFQ (fifth) and with the combined hybrid PQ-CBWFQ (sixth scenario) queuing disciplines. The network topology remained the same in all scenarios; the differences are only in the used queuing disciplines. Through a comparison of simulation results for different scenarios we have tried to prove how each queuing discipline serves the used network applications. The obtained results are the following.

6.4.1 Example 3: simulation results

As we have mentioned before, we have collected delay statistics from six different queuing discipline scenarios (CQ, WFQ, MWRR, DWRR, CBWFQ and PQ-CBWFQ) for two different active applications (VoIP and HTTP) in the network and with different applied priorities by the ToS field of the IP packet header. We have defined VoIP traffic flows between clients where such flows represent high-priority traffic; while HTTP traffic represents low-priority flow, based on a best effort type of service. In our scenarios, we have 82.09% users who use lower-priority HTTP traffic and only 17.91% users who use the high-priority VoIP application.

In Figure 21, we can see that only 17.91% of users take up a majority part of bandwidth, so the lower-priority HTTP traffic, which represents a majority of all traffic, must wait. This is the reason why delays rapidly increase as can be clearly seen in Figure 21. Evidently VoIP traffic has lower delays in comparison with HTTP traffic. Best results are obtained with the custom queuing method, which ensures the required bandwidth at possible congestion points and serves all traffic fairly. After CQ queuing scheme follow WFQ, DWRR, MWRR, CBWFQ and PQ-CBWFQ. WFQ, DWRR, MWRR and CBWFQ have worse results in terms of delays because of fairness queuing discipline. Similar results are obtained also in case of HTTP. If the CBWFQ scheme is in use, high-priority traffic will be ensured with a fixed amount of available bandwidth defined by the network administrator. For example, the network administrator, using CBWFQ, defines 9Mbit/s for VoIP, in which case only 1Mbit/s remains for all other applications; so the majority of low-level traffic will be affected by rapid increasing of delays, as shown in Figure 22.



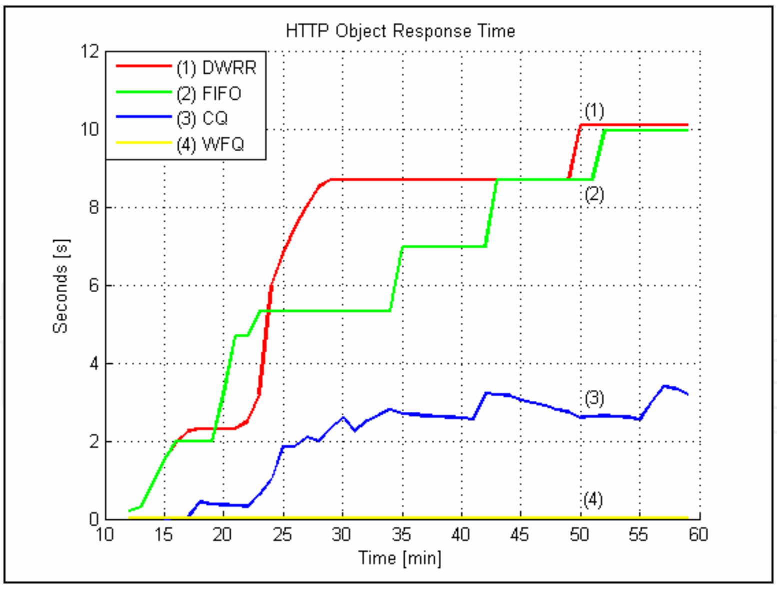


Fig. 21. VoIP End-to-End Delay (top) and HTTP Object Response Time (Bottom) when using different queuing disciplines upon VoIP and HTTP traffic.

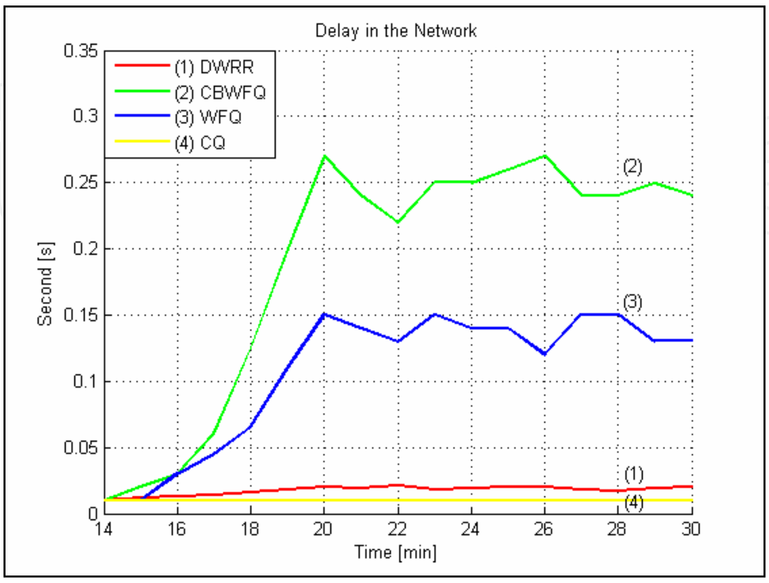


Fig. 22. Time average global delay in the network

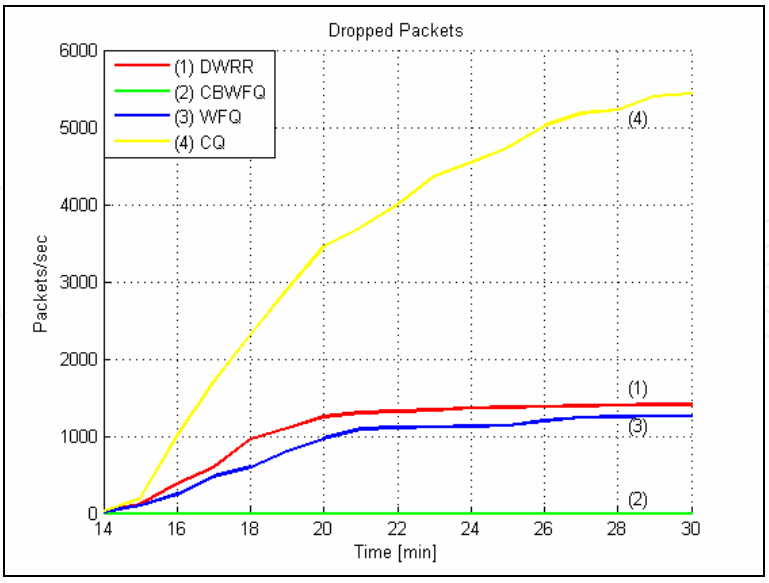


Fig. 23. Amount of VoIP dropped packets

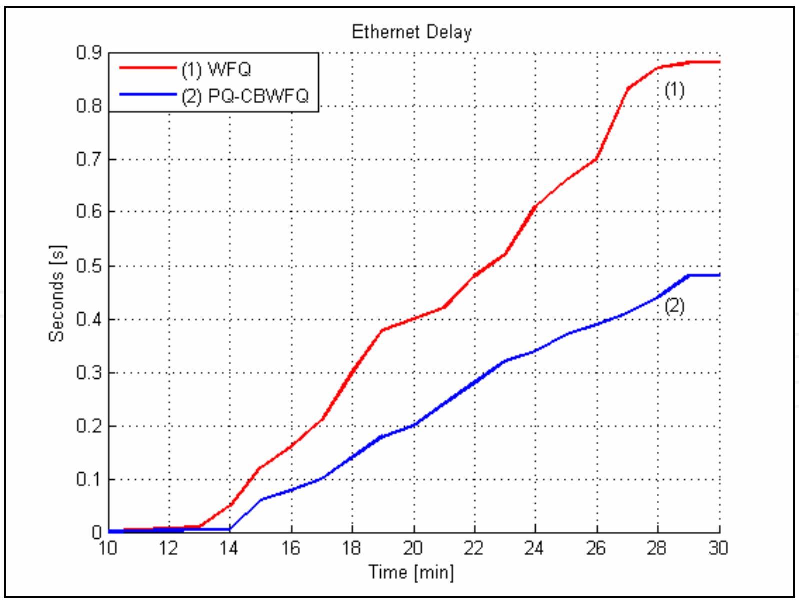


Fig. 24. Ethernet delay (in seconds) for combined PQ-CBWFQ method, compared with WFQ

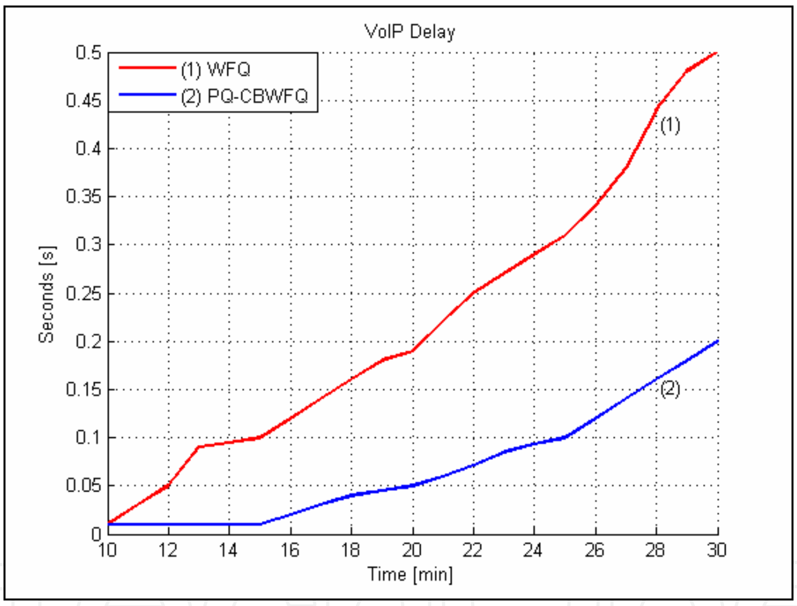


Fig. 25. VoIP delay (in seconds) for combined PQ-CBWFQ method in comparison with WFQ

Figure 23 shows the amount of VoIP dropped packets, when using different queuing schemes. As we have mentioned above, best results are in that situation obtained with CBWFQ method, which has a fixed guaranteed amount of bandwidth. WFQ, DWRR, MWRR and CQ queuing scheme follow. The situation is quite the opposite when we take delays into consideration. CBWFQ introduces the biggest delay, because a majority low level traffic must wait.

Figure 24 shows, how the combined queuing method PQ-CBWFQ improves the delays in comparison with the delays presented in Figure 22.

Using PQ-CBWFQ, the delay is smaller than with the WFQ, as we can see in Figure 24. However, the ordinary CBWFQ method involves a bigger delay than the WFQ, observed in whole Ethernet segment, as shown in Figure 22. Such combinations can perceivably improve network performance. Similar effect as shown in Figure 24 can also be seen in Figure 25 for VoIP delay.

Using the combined queuing method the delay for the VoIP traffic is also reduced in comparison with the ordinary WFQ queuing. In the VoIP application delays play an important role in the quality of perception. The smaller they are, the better voice quality can be offered.

After many simulation runs and graph analysis we can say that queuing policy discipline significantly influences the quality of service for network applications. In many cases CQ queuing discipline was the best choice; in case when we have only two traffic flows WFQ was the best choice; but when on the other hand we need to handle multiple traffic flows, the CBWFQ was the best solution. The CBWFQ method also has its disadvantages; in our case, we have defined only one class with a bandwidth amount 9Mbit/s reserved for VoIP, and the rest of the bandwidth is allocated to the majority of low-priority HTTP traffic. The majority traffic however does not have enough bandwidth and must wait, which causes delays. This is the main reason why CBWFQ has the highest average delays in the network. Regardless of that delays the VoIP delay is however constant during the simulation because of the bandwidth ensured by the defined class. Then again, if we want fairness queuing discipline, which serves all applications fairly, we should use WFQ or CQ mechanism. However, if we only want that the highest-priority traffic flows pass through the network, we should use priority queuing PQ.

Delays in CBWFQ case can be reduced using the PQ-CBWFQ hybrid queuing scheme (see examples 1 and 2). Our simulations show that we must look for solutions also in combined queuing methods. All other available combinations represent a challenge for further research in that area.

7. Conclusion

The results of the simulation examples presented in Section 6 show that when we deal with time-sensitive applications (like VoIP), we have to choose a member of the low latency queuing family. Regarding jitter and VoIP delays the PQ and PQ-CBWFQ queuing schemes are most suitable. In such cases also the voice quality is on a higher level, compared to those where ordinary queuing schemes (CQ, for example) are used. In cases where we have to make a compromise between important traffic and traffic of lower importance, the WFQ-CBWFQ hybrid method gives satisfying results. Our conclusion, according to the obtained simulation results, is to use the following queuing schemes for the following purposes:

- Time-sensitive applications (most recommended PQ-CBWFQ, CBWFQ, optionally PQ)
- Web and other low-importance applications (CQ, WFQ)
- Time-sensitive applications + low-importance applications (WFQ-CBWFQ)
- Other very-low-importance applications mutually equivalent according to the applied priority in the ToS field of the IP packet header (WFQ)

8. References

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

L. L. Peterson, B. S. Davie. Computer Networks. Edition 3, San Francisco 2003.

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

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

Internetworking Technology Handbook – Quality of Service (QoS), Cisco Systems. OPNET Modeler Techical Documentation. G. 729 Data Sheet.

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.

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.

http://en.wikipedia.org/wiki/Time-division_multiplexing

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

A. Kos in S. Tomazic. "Nov nacin zdruzevanja RSVP pretokov (The new method of merging RSVP flows)", ERK 2007, 26. - 28. september 2005, Portoroz, Slovenija, IEEE Region 8, Slovenska sekcija IEEE, 2005, zv. A, pg.. 175-178

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

S. Klampfer. "Simulacija omrežij v Opnet Modeler-ju (Network simulations using OPNET Modeler", Diploma thesis, Faculty of Electrical Engineering and Computer Science, Univesity of Maribor, 2007.

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

Sasa Klampfer, Joze Mohorko, Zarko Cucej, "Simulation of Different Router Buffer Sizes which Influences on VoIP Jitter Delay within the routed Network", Informacije MIDEM, 2011 (confirmed but not published yet)

Sasa Klampfer, Joze Mohorko, Zarko Cucej, "IP packet queuing disciplines as basic part of QOS assurance within the network", Informacije MIDEM, junij 2009, letn. 39, št. 2(130)

Cole, R. Rosenbluth J. Voice over IP Performance Monitoring, AT&T Preprint September 2000

TIPHON 22TD047 Problems with the behavior of Jitter Buffers and their influence on the end-to-end speech quality, source KPN Research, March 2001

ITU-T Y.1541 Network Performance Objectives for IP Based Services RFC1889 Real Time Control Protocol

ITU-T SG12 D74 IP Phones and Gateways: Factors impacting speech quality, France Telecom, May 2002

Sasa Klampfer, Joze Mohorko, Zarko Cucej, "Impact of hybrid queuing disciplines on the VoIP traffic delay", Electrotechnical Review 2009

Sasa Klampfer, Joze Mohorko, Zarko Cucej, "Vpliv različnih načinov uvrščanja na karakteristiko prepustnosti omrežja (Influence of different queuing methods on the common permeability network characteristic)", ERK 2007, 24. - 26. september 2007, Portorož, Slovenija, IEEE Region 8, Slovenska sekcija IEEE, 2007, zv. A, pg.. 100-103

Frank Ohrtman, "Voice over 802.11", Artech House, Boston, London, 2004

Morgan Kaufmann, "Routing, Flow and Capacity Design in Communication and Computer Networks", Warsaw University of Technology, Warsaw, Poland, 2006

Kun I. Park, “QoS in packet networks”, The mitre corporation USA, Springer 2005

Tadeusz Wysocki, Arek Dadej, Beata J. Wysocki, “Advanced wired and wireless networks”, Florida Atlantic University, Springer 2005

H. Jonathan Chao and Bin Liu, “High performance switches and routers”, John Wiley and Sons, 2007

Huan-Yun Wei, Ying-Dar Lin, “A survey and measurement – Based comparison of bandwidth management techniques, IEEE Communications Survey, 2003, Volume 5, No. 2

Mansour J. Karam, Fouad A. Tobagi, “Analysis of the Delay and Jitter of Voice Traffic Over the Internet”, IEEE InfoCom 2001

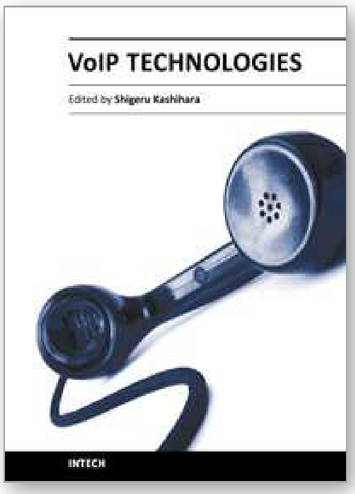
Yunni Xia†, Hanpin Wang‡, Yu Huang, Wanling Qu, “Queuing analysis and performance evaluation of workflow through WFQN”, IEEE Computer Society, First Joint IEEE/IFIP Symposium on Theoretical Aspects of Software Engineering (TASE'07)

Anirudha Sahoo and D. Manjunath, “Revisiting WFQ: Minimum Packet Lengths Tighten Delay and Fairness Bounds”, IEEE COMMUNICATIONS LETTERS, VOL. 11, NO. 4, APRIL 2007

Velmurugan, T.; Chandra, H.; Balaji, S.; , "Comparison of Queuing Disciplines for Differentiated Services Using OPNET," Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09., Vol., no., pp.744-746, 27-28 Oct. 2009

Fischer, M.J.; Bevilacqua Masi, D.M.; McGregor, P.V.; "Efficient Integrated Services in an Enterprise Network," IT Professional, vol.9, no.5, pp.28-35, Sept.-Oct. 2007
Cisco Systems, Understanding Jitter in Packet Voice Networks (Cisco IOS Platforms),
http://www.cisco.com/en/US/tech/tk652/tk698/tech_tech_notes_list.html
G.729 DataSheet, http://www.vocal.com/data_sheets/g729.pdf

M. Callea, L. Campagna, M.G. Fugini and P. Plebani. “Contracts for Defining QoS Levels In a Multichannel Adaptive Information System”, IFIP International Federation for Information Processing, 2005, Volume 158/2005, 2005



VoIP Technologies
Edited by Dr Shigeru Kashihara

ISBN 978-953-307-549-5
Hard cover, 336 pages
Publisher InTech
Published online 14, February, 2011
Published in print edition February, 2011

This book provides a collection of 15 excellent studies of Voice over IP (VoIP) technologies. While VoIP is undoubtedly a powerful and innovative communication tool for everyone, voice communication over the Internet is inherently less reliable than the public switched telephone network, because the Internet functions as a best-effort network without Quality of Service guarantee and voice data cannot be retransmitted. This book introduces research strategies that address various issues with the aim of enhancing VoIP quality. We hope that you will enjoy reading these diverse studies, and that the book will provide you with a lot of useful information about current VoIP technology research.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821