

BULETINUL INSTITUTULUI POLITEHNIC DIN IAȘI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iași
Tomul LVII (LXI), Fasc. 1, 2011
Secția
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

PRIORITIZED WEIGTHED FAIR QUEUING

BY

LUMINIȚA SCRIPCARIU* and FELIX DIACONU

“Gheorghe Asachi” Technical University of Iași, România,
Faculty of Electronics, Telecommunications
and Information Technology

Received, June 24, 2010

Accepted for publication: November 9, 2010

Abstract. Nowadays, real-time applications are used intensively in the Internet. Delay, bandwidth and packet loss rate are the most important transmission parameters which ensure or degrade the quality of service over a path in a communication network. Weighted Fair Queuing (WFQ) is used to allocate bandwidth to applications proportionally to some weights associated with them. This protocol has best results for 100% efficiency of channel bandwidth usage and zero delay but it does not guarantee the minimum bandwidth required by some multimedia applications. We propose to adapt this protocol to real-time transmissions over the Internet using priorities attached to the differentiated service classes. The new protocol is intended for one-hop transmission, in order to ensure the requested bandwidth for critical applications with higher priorities than those non-critical network services (file download, e-mail, etc). The modified protocol named Prioritized Weighted Fair Queuing (PWFQ) is presented below and some scenarios are analysed.

Key words: Internet; bandwidth; priority; Weighted Fair Queuing; Prioritized Weighted Fair Queuing.

* Corresponding author: *email:* lscripca@etti.tuiasi.ro

1. Weighted Fair Queuing Protocol

Weighted Fair Queuing (WFQ) is a scheduling algorithm used in packet switched networks in order to adaptively share the bandwidth depending on the occupancy of a path in the network (Sayenko *et al.*, 2003). It is used efficiently in ATM (Asynchronous Transfer Mode) networks where the dimension of the data unit, called *cell*, has fixed length of about 53 bytes, no matter what type of information it carries away (data, voice, audio or video).

Each transmission request is instantly solved by re-dimensioning the allocated bandwidth to the current applications.

Each application has a weight depending on the flow type and the price admitted for transmission. Each flow type is associated with a specific weight sometimes depending on the transmission price. Let us denote by w_n the weight of the n^{th} application.

For a time-interval, the channel bandwidth is fair shared between the active applications based on their weights. For the n^{th} application, it will be allocated a percentage of the total bandwidth equal to

$$p_n = \frac{w_n}{\sum_{i \text{ for active applications}} w_i} \cdot 100, [\%]. \quad (1)$$

For example, we consider three user applications, *A*, *B* and *C*, with the corresponding weights 2, 3 and 1. We consider a particular scenario of transmission when the users start to transmit consequently and stop at different moments: *A* is the first transmitting user and it stops first; *B* is the second user and *C* is the third or the last one (Fig. 1).

Fig. 1 represents the transmission flow of each user marked with the percentage of used bandwidth on each time interval.

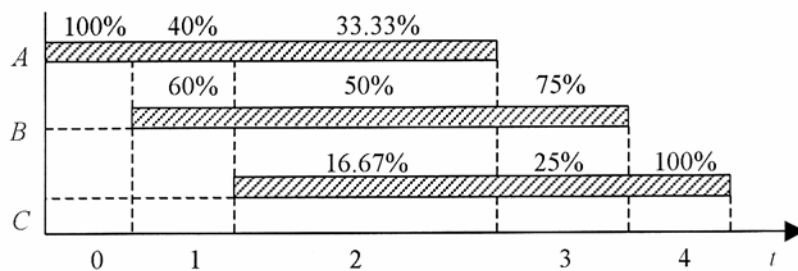


Fig. 1 – First scenario bandwidth usage.

We compute the percentage of used bandwidth for each active user on all the time-intervals:

a) interval 0 (only A is active)

$$p_{A_0} = \frac{w_A}{w_A} \cdot 100 = \frac{2}{2} \cdot 100 = 100\%;$$

b) interval 1 (A and B are both active)

$$p_{A_1} = \frac{w_A}{w_A + w_B} \cdot 100 = \frac{2}{5} \cdot 100 = 40\%; \quad p_{B_1} = \frac{w_B}{w_A + w_B} \cdot 100 = \frac{3}{5} \cdot 100 = 60\%;$$

c) interval 2 (all users are active)

$$p_{A_2} = \frac{w_A}{w_A + w_B + w_C} \cdot 100 = \frac{2}{6} \cdot 100 = 33.33\%;$$

$$p_{B_2} = \frac{w_B}{w_A + w_B + w_C} \cdot 100 = \frac{3}{6} \cdot 100 = 50\%;$$

$$p_{C_2} = \frac{w_C}{w_A + w_B + w_C} \cdot 100 = \frac{1}{6} \cdot 100 = 16.67\%;$$

d) interval 3 (A exits, B and C are active)

$$p_{B_3} = \frac{w_B}{w_B + w_C} \cdot 100 = \frac{3}{4} \cdot 100 = 75\%; \quad p_{C_3} = \frac{w_C}{w_B + w_C} \cdot 100 = \frac{1}{4} \cdot 100 = 25\%;$$

e) interval 4 (B exits, only C is active)

$$p_{C_4} = \frac{w_C}{w_C} \cdot 100 = \frac{1}{1} \cdot 100 = 100\%.$$

We notice that even if the corresponding weight of an application is little, when only one user utilizes the channel, then the entire bandwidth is allocated to it (100%).

In many cases, the allocated bandwidth for one user is higher than its predicted value which corresponds to the case when all users are active (for example, time-interval 2, in our scenario).

Unfortunately, real cases cannot afford to reserve so much bandwidth for all the applications. It means that real-time broadband applications will have low quality-of-service (QoS) when the network is very busy (for example, at peak-times).

2. Prioritized Weight Fair Queuing Protocol

Differentiated Services (DS) enhancements to the Internet protocol (IP) are intended to enable scalable service discrimination in the Internet. The services may include both those that can satisfy quantitative performance requirements (*e.g.*, bandwidth) and those based on relative performance (*e.g.*, “class” differentiation).

The differentiated service *Class Selector Code Point* (DSCP) contained in a field in the IP packet header of 3 bits in the fourth version and of 6 bits in the sixth version of the protocol is associated to a particular forwarding treatment, or *Per-Hop Behavior* (PHB), at each network node along a path. An DSCP may have a local meaning or can be defined in a reference document. These PHBs are useful and required in network nodes to deliver differentiated treatment of the packets (Evans *et al.*, 2007).

In the DS field from the IPv6 header there are two currently-unused bits which can be used to prioritize the traffic.

Any real-time application requests zero-delay and a specific amount of bandwidth. So, a higher priority value is given to the critical traffic which request guaranteed QoS.

A Lower-Effort Behavior (LEB) and a lower priority are chosen for sending extremely non-critical traffic. There should be an expectation that LEB packets may be delayed or dropped when other traffic is present. Use of the LEB might assist a network operator in moving certain kinds of traffic or users to off-peak times.

The modified algorithm PWFQ tries to solve instantly any real-time request for a specific bandwidth despite other applications which are non-critical and admit delays. PWFQ uses weights and priorities to classify applications. The priorities are used to control the admission and the weights are applied for bandwidth computation.

Four initial *levels of priorities* are used namely

a) Level 0 ($p = 0$) corresponds to the least priority non-critical applications (NCA) such as file downloading.

b) Level 1 ($p = 1$) is given to some multimedia applications which admits reduced delays such as audio or video-streaming (denoted ASTR, VSTR).

c) Level 2 ($p = 2$) is associated to real-time (RT) applications like voice packets (VP) and real-time video (RTV).

d) Level 3 ($p = 3$) is reserved for congestion control packets (CCP) transmitted any time the congestion notification bits are set.

Four PHBs are considered by the PWFQ namely

1. PHB-1: transmit with no delay.

2. PHB-2: wait in the queue.

3. PHB-3: interrupt the transmission.

4. PHB-4: discard the packet when the queue is loaded completely and congestion occurs.

PWFQ uses the following *notions*:

a) The initial priority of an application (INIT_PRIOR) depends on the application type.

b) The priority of a request (REQ_PRIOR) is equal to the INIT_PRIOR first time when the request is addressed to the node and then it has higher values depending how much time it waits in the queue.

c) The channel priority (CH_PRIOR) is defined as the highest value of the priorities of the ongoing flows except the congestion control packets priority.

PWFQ works based on some *rules* for a one-hop scenario *i.e.*

1. When two transmission requests are received simultaneously in the network node, the priority request is solved first.

2. When multiple requests are processed at a moment, those having priorities higher than the current channel priority are admitted and the other are delayed and put on the queue in order to be processed next time.

3. Any time an application is delayed, the priority of its request is incremented by 1.

4. All the requests included in the queue are reconsidered periodically or each time an application ends or another request occurs.

5. An ongoing application keeps its priority unchanged, equal to its initial priority.

6. When a priority application is admitted for transmission and its requested bandwidth cannot be satisfied keeping active all the other ongoing applications, the bandwidth computation is made with the least priority flows having progressively reduced weights (half value, only three times) till the desired bandwidth is obtained for the critical application and after that some less priority applications are interrupted and moved in the queue with their initial priority.

7. When congestion is imminent (90% used bandwidth percentage), an CCP request is made and it virtually reserves 25% of the channel bandwidth in order to free bandwidth and solve the congestion.

8. When an imminent congestion has to be solved, different low-priority applications are interrupted and no other requests are admitted till the critical situation is solved, except the real-time applications.

9. When the node tries to solve congestion and only RT applications are active, for RT applications the minimum requested bandwidth is allocated unconditionally. If the total bandwidth is unavailable, then progressively some broadband RT applications are stopped, till a convenient situation. All interrupted applications are moved to the queue with their initial priorities.

10. When congestion is solved and all the ongoing applications have enough bandwidth, then the node return to WFQ till another critical situation occurs.

R e m a r k s

1. When an application with low initial priority is delayed, its request

priority is increased every time it is delayed, so after 1, 2 or 3 time intervals it will have the highest possible priority and it will surely be admitted for transmission.

2. PWFQ does not guarantee the bandwidth for low priority applications.

3. PWFQ admits unconditionally RT application requests, with zero delay and guaranteed bandwidth.

Unfortunately if a Denial-of-Service (DoS) attack is started, it is possible that the hop becomes unable to solve even RT requests if the total amount of requested bandwidth exceeds the channel bandwidth.

The *risk of congestion* is considered not null when more than 75% of the total bandwidth is used. In this case, the bandwidth for the low-priority applications is reconsidered with reduced weights.

When 90% of the bandwidth is used, then an *imminent congestion* is notified and a congestion control protocol is applied. In respect to this, PWFQ uses Congestion Control Packets (CCPs) to solve a critical situation.

3. Scenarios Analysis

We analyse an One-Hop transmission scenario, with the following features:

a) Maximum number of applications: 10xVP, 2xRTV, 2xVSTR, 4xASTR and 10xDP channels (N_x denotes the number of channels)

b) The wireless bandwidth: 11 Mbps

c) The minimum required bandwidth for each application type: 64 kbps for 1xVP (0.58%), 4 Mbps for 1xRTV (36.36%), 1 Mbps for 1xVSTR (9%), 128 kbps for 1xASTR (1.1%) and 2.75 Mbps as a virtual bandwidth for CCP.

d) The corresponding default weights: 1 for DP, 1 for VP, 37.5 for RTV, 10 for VSTR, 1.5 for ASTR and 25 for CCP.

The weights can be changed when the minimum bandwidth for a type of application is not obtained. For example, the weight of DP will be progressively reduced to 0.5, 0.25, 0.1, 0.05 and 0.025. After that, if it is necessary the weight of some narrowband applications, such as voice or ASTR, will be also reduced to 0.8, 0.7 or 0.6 from its initial value but the resulting allocated bandwidth should not be lower than its minimum requested bandwidth.

These applications request to send consequently, according to a scenario described in Fig. 2. On each time interval the number of packets of same kind is specified.

At the time moment t_0 , two voice channels, one RT video channel, one audio streaming channel and four data transfer channels are active.

At the next moment, t_1 , the network node receives other transmission requests: two for voice applications, one for an ASTR application and three from some data channels. There are also the other ongoing applications started at t_0 .

At t_2 , the network node receives more requests: two for voice applications, one for a VSTR application and two for ASTR transmissions.

At t_3 , the network node receives others requests and it becomes busy (peak-time case): two for voice applications, one for an RTV application and two for data transmissions.

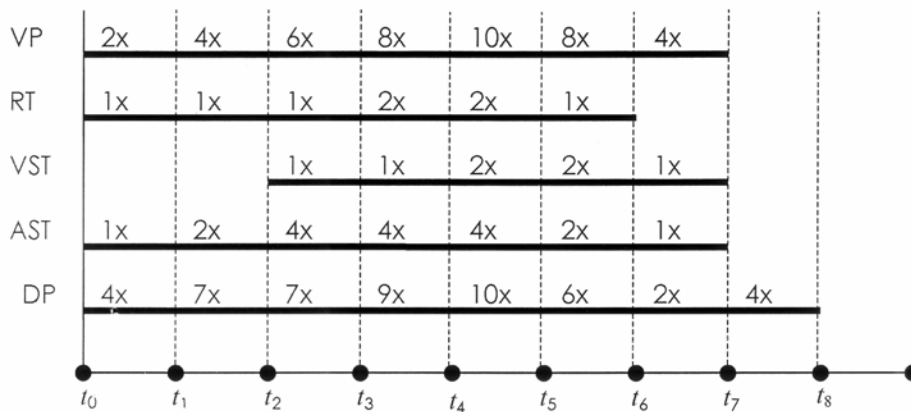


Fig. 2 – One-Hop transmission request scenario.

At t_4 , the network node receives more requests and it is fully occupied: two for voice applications, one for a VSTR application and one for a data channel; it is the worst case.

At t_5 , some relaxation is observed; some applications end to transmit: two voice applications, one RTV, two ASTR and four DP applications.

At t_6 , more applications end to transmit: four voice applications, one RTV, one VSTR, one ASTR and another four DP applications.

At t_7 , all multimedia applications exit the node and only four DP applications use the entire bandwidth of the channel.

3.1. WFQ Scenario

WFQ solves any transmission request unconditioned, with zero delay and unguaranteed bandwidth but the channel efficiency of the used bandwidth is maximum.

Table 1 presents the allocated bandwidth percentages for different types of flows applying WFQ on the scenario presented above for all the eight time intervals considered in this scenario (0 to 7).

On the last row, the percentage of the *Total Used Bandwidth* (TUB) is computed considering the used bandwidth and not the allocated bandwidth for multimedia applications.

In the time intervals no. 3 and no. 4 we can see that some multimedia applications have not enough transmission bandwidth (cells filled with grey). This is reflected in QoS degradation. For the real-time video application, some discontinuities of transmitted frames occur and a perceptual discomfort is produced. The same thing is observed for the video-streaming.

Table 1
WFQ Scenario

Traffic type	0	1	2	3*	4*	5	6	7
VP	2x 2.22%	4x 2%	6x 1.36%	8x 0.92%	10x 0.83%	8x 1.34%	4x 5.71%	–
RTV	1x 83.33%	1x 75%	1x 51.0%	2x 34.72%	2x 30.99%	1x 50.3%	–	–
VSTR	–	–	1x 13.6%	1x 9.26%	2x 8.26%	2x 13.4%	1x 57.0%	–
ASTR	1x 3.33%	2x 3%	4x 2.04%	4x 1.38%	4x 1.24%	2x 2.01%	1x 8.57%	–
DP	5x 12.5 %	7x 2%	7x 1.36%	9x 0.92%	10x 0.83%	6x 1.34%	2x 5.71%	4x25%
TUB	46.37%	54.9%	62.9%	96%	97%	69.4%	23.9%	100%

* Critical intervals.

The other multimedia applications (voice and audio streaming) works fine, the minimum required bandwidth being allocated to them in all time intervals.

Data packets are transmitted with variable data rates, minimum on the critical intervals no. 3 and no. 4, and very good for the last interval when no multimedia application is active.

We conclude that WFQ does not works well for broadband and real time applications at the peak-times. WFQ does not guarantee the minimum bandwidth in heavy traffic.

The total used bandwidth has very high values in the critical time intervals, when congestion is imminent. In this case it is useful, for WFQ, to consider a reduced shared bandwidth of about 75% or 80% when it computes the allocated bandwidth for each application. This procedure makes worth the transmission conditions for the multimedia and real-time applications but avoid congestion.

3.2. PWFQ Scenario without Congestion Control

The same scenario is reconsidered with PWFQ for the critical time-intervals when WFQ faults occur. PWFQ is applied only when WFQ does not ensure the minimum bandwidth for all the flows. The scenario flows differently for the next time intervals, because PWFQ delays some transmission requests with low priorities.

Let us exemplify PWFQ for the two critical time intervals no. 3 and no.

4, *without congestion control*.

At t_3 , because WFQ does not ensure the minimum bandwidth for some applications, PWFQ is applied. The network node has to solve concurrent requests with different priorities: two for voice applications ($p = 2$), one for a RTV application ($p = 2$) and two for data packets transmission ($p = 0$). The channel priority at t_3 is equal to 2, so only the voice and RTV requests are admitted and the others (DP) are delayed and their priority request is incremented by 1. A queue is created and it includes at this moment two requests (2xDP) with the REQ_PRIOR = 1. The bandwidth for each active application is computed based on their weights and to ensure the minimum bandwidth for all the RT applications, the data transmission weight is reduced to 0.5. QoS of all ongoing multimedia applications is ensured (s. Table 2).

Table 2
Allocated Bandwidth Percentage in the Critical Time Intervals Using PWFQ without Congestion Control

Traffic type	Minimum requested bandwidth	Weight	Priority	3	4	5	6	7
VP	64 kbps	1/0.75/0.6	2	8x 1.0%	10x 0.97%	8x 0.585%	4x 4.88%	–
RTV	4 Mbps	37.5	2	2x 37.5%	2x 36.58%	2x 36.55%	–	–
VSTR	1 Mbps	10	1	1x 9.95%	1x 9.70%	2x 9.75%	1x 48.78%	1x 66.67%
ASTR	128 kbps	1.5/1.2	1	4x 1.5%	4x 1.46%	2x 1.17%	1x 7.32%	–
DP	any value	1/0.5/0.25 /0.1/0.05	0	7x 0.5%	7x 0.49%	8x 0.049%	5x 4.88%	5x 6.67%
<i>Total Used Bandwidth (TUB)</i>				94.41%	95.48%	98.12%	36.9%	42.44%

At t_4 , the network node has to solve more requests: two for voice applications ($p = 2$), one for a VSTR application ($p = 1$), one for a data channel ($p = 0$) and two requests for DP from the queue ($p = 1$). The channel priority is still equal to 2 so only the voice requests are admitted and all the others are moved to the queue with incremented priorities: 1xVSTR ($p = 2$), 2xDP ($p = 2$), 1xDP ($p = 1$). The weight of DP is still reduced to 0.5 to ensure the minimum bandwidth for all the multimedia applications.

At t_5 , some applications end to transmit (2xVP, 2xASTR, 1xDP). The node has to solve the delayed requests from the queue. The channel priority is still equal to 2. From the queue those requests with the highest priority are admitted (1xVSTR, 2xDP). The other DP request ($p = 1$) still remains in the

queue with an increased priority ($p = 2$). The DP weights go down to 0.05, the voice weight is reduced to 0.6 and for ASTR the weight goes down to 1.2 in order to ensure the minimum bandwidth for RTV. All multimedia applications are transmitted with the minimum requested bandwidth.

At t_6 , more applications end to transmit and only one transmission DP request is in the queue ($p = 2$). The channel priority is still high but the DP request was delayed two times and its priority is high enough to be admitted. All the applications have enough bandwidth to transmit at high QoS.

Analysing Table 2 it shows that PWFQ offers the requested bandwidth for all the multimedia applications so it is a guarantee for QoS.

3.3. PWFQ Scenario with Congestion Control

It is obvious that for some time intervals (no. 3, 4 and 5), the node works with a high risk of congestion, in fact an imminent congestion is announced by the TUB in the time-interval no.3. It is necessary, at t_4 , t_5 moments, to generate *Congestion Control Packets (CCP)* to reduce the used bandwidth to avoid congestion. CCP has the highest priority ($p = 3$), it reserves 25% virtual bandwidth and it last one time-interval.

Table 3 presents a modified solution for the same transmission scenario with a congestion control mechanism applied.

Table 3
Allocated Bandwidth Percentage in the Critical Time Intervals Using PWFQ with Congestion Control

Traffic type	Minimum requested bandwidth	Weight	Priority	3*	4**	5**	6	7
VP	64 kbps	1/0.75/0.6	2	8x 1.0%	10x 0.58%	8x 1.06%	4x 3.12%	–
RTV	4 Mbps	37.5	2	2x 37.5%	2x3 6.36%	2x 39.89%	–	–
VSTR	1 Mbps	10	1	1x 9.95%	–	1x 10.64%	2x 31.25%	1x 45.45%
ASTR	128 kbps	1.5	1	4x 1.5%	–	–	4x 4.69%	2x 6.82%
DP	any value	1/0.5/0.25/ 0.1/0.05	0	7x 0.5%	–	2x 0.53%	2x 3.12%	9x 4.55%
CCP	25%	25	3	–	1x 19.2%	–	–	–
Total Used Bandwidth (TUB)				94.41%	80.8%	87.42%	31.24%	52.31%

* imminent congestion; ** risk of congestion.

At t_4 , a CCP is generated and it requests for at least 25% of bandwidth. In this case, the channel priority is 2 and only the CCP and the two voice applications requests are admitted. The queue includes: 1xVSTR ($p = 2$), 2xDP ($p = 2$), 1xDP ($p = 1$). The node must assure first the bandwidth requested by

CCP. To reserve the minimum bandwidth for the RT applications, the DP transmissions are interrupted and a first computation of bandwidth is made. 7xDP ($p = 0$) are included in the queue. It results insufficient bandwidth for RT applications. Secondly, the ASTR and the VSTR applications are interrupted and moved to the queue with their initial priority: 1xVSTR ($p = 1$), 4xASTR ($p = 1$). The remained ongoing applications are only RT, so the minimum requested bandwidth is allocated.

At t_5 , the risk of congestion is not null. The weights of low-priority applications are reduced ($w = 0.5$ for DP). The queue includes: 1xVSTR ($p = 2$), 2xDP ($p = 2$), 1xDP ($p = 1$), 1xVSTR ($p = 1$), 4xASTR ($p = 1$), 7xDP ($p = 0$). The number of voice channels decreases and more bandwidth is free. Because the channel priority is 2, only the priority requests ($p = 2$) are admitted. The remaining queue is: 1xDP ($p = 2$), 1xVSTR ($p = 2$), 4xASTR ($p = 2$), 7xDP ($p = 1$).

At t_6 , the risk of congestion exists. With less voice channels and no RT video channels, the congestion is solved and the queue will be reevaluated. The node admits some requests from the queue: 1xVSTR ($p = 2$), 4xASTR ($p = 2$). The DP requests remain in the queue: 7xDP ($p = 2$). The ongoing DP works with weight equal to 0.5.

At t_7 , the node has returned to WFQ. The data rate is progressively increasing. No collision is notified.

4. Conclusions

WFQ works fast, with no delays but does not guarantee the necessary bandwidth for multimedia applications. PWFQ offers the minimum bandwidth for RT traffic with some delays for the less priority packets and reduced bandwidth for the data packets. PWFQ is presented in detail and some scenarios are analysed with and without congestion control.

REFERENCES

- Evans J.W., Filsfils C., *Deploying IP and MPLS QoS for Multiservice Networks: Theory and Practice*. Morgan Kaufmann Publ., San Francisco, USA, 2007.
- Sayenko A., Hämäläinen T., Joutsensalo J., Siltanen J., *An Adaptive Approach to WFQ with the Revenue Criterion*. Proc. of the Eighth IEEE Internat. Symp. on Comp. a. Commun. ISCC '03, Remer, Antalya, Turkey, 2003, 181-193.
- * * * *Congestion Management and Queuing*. ONT Ch. 4 2009; from Cisco, <http://ciscocertifiedhell.com/2009/07/04/ont-chapter-4-conegstionmanagement-queuing/>.

ALGORITMUL PWFQ

(Rezumat)

Aplicațiile în timp real sunt întâlnite tot mai des în Internet. Întârzierea, lățimea de bandă și rata de pierdere a pachetelor sunt cei mai importanți parametri prin care se apreciază calitatea serviciului într-o rețea de comunicații. WFQ este folosit pentru alocarea proporțională a lățimii de bandă către diverse aplicații, fără întârzieri dar fără să garanteze lățimea de bandă minimă pentru unele aplicații multimedia. Se propune adaptarea acestui protocol pentru transmisii prin Internet în timp real prioritizând diferitele clase de servicii. Noul protocol, denumit PWFQ, asigură lățimea de bandă cerută de aplicațiile cu prioritate mai mare în detrimentul celor mai puțin critice. Sunt prezentate și analizate câteva scenarii de transmisie.