Adaptive Weighted Round Robin (AWRR) Scheduling for Optimization of the Wireless Medium Virtualisation

Gorka Hernando, Susana Pérez, and José María Cabero

TECNALIA-Telecom, Technology Center, Zamudio, 48170, Spain
Tel.: +34 94 6002266, Fax: +34 94 6002299
{ghernando,sperez,jmcabero}@robotiker.es
http://www.robotiker.es/

Abstract. Network virtualisation has been recently presented as a mean to overcome the saturation of the current Internet by sharing the same infrastructure by different network operators. This work considers virtualisation of the wireless medium, a way to share common network physical resources by different Virtual Operators based on a Time Division Multiple Access technique. We propose an Adaptive Weighted Round Robin scheduler as a means to optimize the assignment of time slots to each virtual operator and improve the performance of the system. End-to-end delay and packet loss are the metrics used in this paper to show the potential and limitations of wireless virtualisation as a way to increase the network usage. This research is presented as a simulation-based study developed over the widely used NS2 network simulator. Different scenarios and network topologies are considered in order to assess the benefits of using the proposed scheduler.

Keywords: Virtualisation, Adaptive TDMA Scheduler, Optimization, Weighted Round Robin.

1 Introduction

Many new necessities and requirements have been requested by the final users during the last years, some of them with conflicting goals and policies, making the current Internet network paradigm almost impossible to achieve. Therefore, network virtualisation has been recently presented as a key concept for the construction of the Future Internet because of its potential to allow multiple network architectures coexist [1][2]. Virtualisation is based on sharing a common physical substrate by several providers running their own services in a fully isolated and protected way. Many different research lines are boosted nowadays regarding network virtualisation: innovative network architectures [3], efficient embedding algorithms for the creation of virtual networks [4][5], techniques for the virtualisation of the physical resources [6][7], among others.

The wireless medium is a common physical resource that does not purely belong to a specific node but the whole network itself. In general, virtualisation techniques for the wireless medium look for the shared use of a wireless transmission medium in a coordinated way [8], i.e. each virtual operator (VO) has at its disposal the wireless medium during specific times (Time Division Multiple Access (TDMA)), frequencies

P. Chatzimisios et al. (Eds.): MobiLight 2010, LNICST 45, pp. 673–684, 2010.

(Frequency Division Multiple Access (FDMA)), codes (Code Division Multiple Access (CDMA)) and so on.

Our research focuses on the TDMA virtualisation of the wireless medium, based on a Round Robin (RR) scheduling algorithm [9]. In this approach, the scheduler divides the usage of the common resource into equal time slots (TS), each one assigned to a particular VO. The transmission frame is formed by all the individual pieces, and it is conformed by the repeated sequence of consecutive TSs, such that the time to finish a complete round (called 'quantum' in the following) will be the summation of the individual TSs. Weighted RR (WRR) is a special case of RR scheduling where the VOs are characterized by their weights, $W(VO_i)$. The $W(VO_i)$ value defines the percentage of quantum that VO_i will get. So the TS assigned to each VO is obtained as the product of the quantum by its $W(VO_i)$. Some particular situations could provoke undesired effects (non optimal performance) in this kind of systems. For instance, VO_A stops sending data after a long-lived transmission. Even if VO_A is not using its TS, this cannot be used by any other VO in place, although a certain VO_B could be suffering from congestion problems or so. Thereby, classical RR and WRR techniques are not covering an optimal sharing among VOs, since the duration of the TSs is somehow fixed and cannot be adapted to the network dynamics that might occur.

In this work an Adaptive WRR (AWRR) scheduler is presented. Our efforts aim at optimizing the use of the quantum time, improving the performance of the shared medium. While certain VOs are whether not transmitting or sending traffic at low speed (they do not exhaust their own TS), the AWRR algorithm readjusts the weights and tries to give some extra time (which is made free by the VO who does not consume its whole assignment) to other VOs. Thus, the objective of our work is to give some favour in terms of Quality of Service (QoS) to the VOs with higher transmission constraints (peaks of traffic, demanding applications, …), but without penalizing the others.

This paper is organized as follows: the outline of the TDMA scheduler used as basis is presented in Section 2, whereas Section 3 is focused on the analysis of the proposed adaptive WRR algorithm. Section 4 discusses the scenarios followed by the main simulation results and, finally, concluding remarks are drawn in Section 5.

2 TDMA Scheduler Description

A TDMA scheduler scheme similar to the one we propose here was previously validated and evaluated as the running process for a single wireless node in Matlab [10]. The objective of this mathematical implementation was to validate the WRR model as compared to the M/M/S/K systems. In this paper we assess the benefits of an AWRR scheduler in the interconnection of several nodes in different network topologies.

When the sharing is planned in a time slotted way, the situation that we are interested in corresponds to a generic node receiving traffic coming from different VOs. The traffic management inside each node consists in storing the arriving traffic packets in specific VO queues until the TDMA scheduler sends them to the wireless interface to be transmitted. The selection of these packets is done depending on the scheduling technique and the policies applied to the different types of traffic.

The basis of the scheduling algorithm proposed is depicted in Fig 1. From left to right, the whole system starts running when the arrival of traffic packets at the VO queues is produced. The system consists of the following main modules:

– *Traffic Generator (TG):* it creates the network traffic based on models [11][12] that generate arrival instants and duration intervals for all the packets of each type such as: VoIP, IPTV, HTTP, FTP, etc. The starting time of every traffic associated to a particular VO is random inside the quantum time.

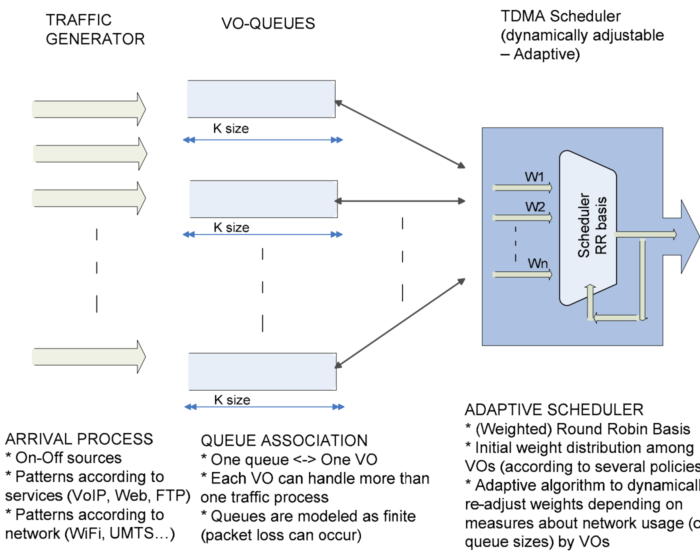

Fig. 1. AWRR scheduler scheme

– *VO Queues*: each node consists of a number of queues equal to the number of VOs running on it and one additional queue dedicated to network management issues, i.e. ARP, Routing, etc. The key concept here is the univocal correspondence between queues and VOs. All these VO queues receive the incoming packets generated by the TG and store them until they are requested by the TDMA Scheduler. The VO queues considered are finite First Input First Output (FIFO) arrays. The limitation in the size is important because it makes packet losses possible and realistic.

  The sending of management packets is prioritized over data packets, so whenever there is a packet in this queue it will be sent as soon as possible with pre-emptive priority. Thus, the management queue does not need an own time slot and it is not considered within the AWRR algorithm.

– *TDMA Scheduler:* all the VO queues are managed by the TDMA scheduler that is in charge of allowing a specific VO queue to serve the packets to be transmitted. The TDMA Scheduler follows an AWRR discipline, i.e., it initially assigns a $W(VO_i)$ per VO, being these values equal (RR) or different (WRR). The number of served packets depends on their size and the TS duration. In order to optimize the end-to-end delay and reduce the ratio of lost packets because of queue overflows, the scheduler readapts the weights depending on the queue occupation of each VO. Specific features of the AWRR algorithm will be explained in section 3.

3 The Adaptive Weighted Round Robin Algorithm

Trying to optimize the response of the TDMA scheduler, the proposed AWRR algorithm recalculates the weights assigned to each VO depending on the average number of packets enqueued during a certain time interval. If a certain VO stores in its queue a large number of packets during the time of analysis it will be granted with a larger $W(VO_i)$, so that the average end-to-end delay of this traffic can be reduced, improving the performance of the network. It must be noticed that this extra $W(VO_i)$ is obtained from idle intervals of other VOs, so the quantum time remains unchanged. That way, if a VO does not transmit anything for a long time, its $W(VO_i)$ would be decreased to 0 milliseconds. However, there is a restriction that the reduction of a $W(VO_i)$ can only be performed if this new value is greater than (at least equal to) the minimum weight, $MW(VO_i)$, arranged by contract. This policy tries to reflect that a certain VO may have a certain minimum QoS pre-arranged (minimum guaranteed TS) in the contract specification for its operation of the virtualised resource.

The adaption process can be triggered periodically every time-window interval, i.e. every 5 completed quantum times, or due to a certain amount of packets served, i.e. every 250 packets transmitted. The implications of choosing one method or the other will be assessed in the results section. Since the adaption process implies a certain processing delay in each node, it is very important to reach a balance between the number of (periodical) adaptations and the processing time involved.

Each queue associated to a particular VO is characterized by its size, $QS(VO_i)_n$, which represents the number of packets stored in memory at a certain instant T_n. This size remains unchanged until a new event (arrival or departure of a packet) happens in T_{n+k}. The average queue occupation of this particular queue, $QO(VO_i)$, is calculated by the expression:

$$QO(VO_i) = \frac{\sum_{t=0}^{t=n} QS(o_i)_n \cdot (T_{n+k} - T_n)}{T_{n+k}}$$

It might seem desirable that the AWRR algorithm readapts weights as quickly as possible to peak rate variations of the traffic generation. Nevertheless, we cannot completely obviate historical patterns because otherwise those VOs that stop sending data for a short time would be instantly penalized. In order to allow the TDMA scheduler to tune the AWRR algorithm depending on the network requirements, two parameters α and β are defined, where α is associated to short-lived variations and β is associated to long-lived variations. We define the LastMean, $LM(VO_i)$, as the $QO(VO_i)$ obtained during the last time-window calculation and the HistoricalMean, $HM(VO_i)$, as the $QO(VO_i)$ averaged from the beginning. The TotalMean $TM(VO_i)$ is calculated by:

$$TM(VO_i) = \alpha \cdot LM(VO_i) + \beta \cdot HM(VO_i)\,; \qquad \{\, \alpha + \beta = 1 \,\}$$

The $TM(VO_i)$ is used by the AWRR algorithm to optimize the response of the TDMA scheduler. The mean TM, $\overline{TM}$, is calculated as:

$$\overline{TM} = \frac{\sum_{i=0}^{N} TM(VO_i)}{N}$$

The adjustment of new $W(VO_i)$ is proportional to the difference between each $TM(VO_i)$ and the $\overline{TM}$. That is, for those VOs with $TM(VO_i) > \overline{TM}$ their weight will be increased and viceversa for the opposite case. See lines 1 to 10 of algorithm 1. As we explained, the new $W(VO_i)$ of all VOs must always be maintained above (at least equal to) their predefined $MW(VO_i)$. If the AWRR needs to take away more units than permitted from VO_1 (it has already reached its minimum weight), the algorithm will get the proportional amount of units from the rest of VOs. See lines 13 to 25 of algorithm 1.

In this algorithm 1 we summarize all the details involved in the AWRR Scheduler. It covers one of the iterations to readapt VO weights.

Algorithm 1. AWRR Scheduler

```
1: for all VO_I do
2:   /* Calculate the delta value to sum/rest to each W(VO_i) */
3:   Delta = ( (TM(VO_i) – mean(TM) ) / QS(VO_i) ) · W(VO_i)
4:   if W(VO_i) + Delta >= MW(VO_i) then
5:     W(VO_i) = W(VO_i) + Delta
6:   else
7:     Rest = Rest + ( Delta – (W(VO_i) – MW(VO_i))
8:     W(VO_i) = MW(VO_i)
9:   end if
10: end for
11: if Rest = 0 then
12:   finish
13: else
14:   while Rest > 0 do
15:     /* Calculate the delta value to be subtracted from each W(VO_i) */
16:     Delta = Rest / NumVO (Not Minimum weight)
17:     if W(VO_i) – Delta >= MW(VO_i) then
18:       W(VO_i) = W(VO_i) – Delta
19:       Rest = Rest – Delta
20:     else
21:       W(VO_i) = MW(VO_i)
22:       Rest = Rest – (W(VO_i) – MW(VO_i))
23:     end if
24:   end while
25: end if
26: finish
```

4 Scenarios and Simulation Results

In order to quantify the aforementioned aspects we have particularized the wireless medium virtualisation problem for several scenarios.

The evaluation of the proposed scheduler has been developed within the NS2 environment [13], a C++ and TCL based discrete event simulator widely used in academic research. An implemented version is available at [14]. NS2 implements full protocol stacks and is able to simulate wired and wireless topologies using different transmission technologies. Typically, transport layer protocols, routing protocols, interface queues, and also link layer mechanisms can be configured. Moreover, propagation times between nodes are considered by the simulator. In NS2 network physical activities are translated to events that are enqueued and processed in the order of their scheduled occurrences.

4.1 Ad-Hoc Network Topology

The first scenario proposed to assess the benefits of the AWRR Scheduler is an Ad-Hoc network where a set of 4 nodes are located in a straight line topology (bus), so that transmissions consist of several hops among edges. The wireless technology used in this scenario is the standard 802.11. We consider that VOs can only send traffic during their TS, but can receive packets any time during the quantum.

Traffic generation is characterized by the inter-arrival time, λ, defined as a constant bit rate. That way, the analysis of the queue evolution is done just by the adaption of the weights by the AWRR scheduler and independently of the variations on the inter-arrival time by processes such as Poisson. All the packets transmitted have a constant size of 100 bytes.

Table 1 and 2 show the set of transmissions defined in the simulation and the evolution of the VOs in the communication network. We assume that the TSs and the active operators are perfectly synchronized in all nodes at the beginning of the simulation.

Table 1. Set of transmission flows

VO	Src Node	Dst Node	Start time	Finish Time	λ (ms)
1	0	3	20	1000	50
1	3	0	100	600	100
2	2	0	200	400	25
2	2	0	600	800	20
3	3	2	100	400	80
4	1	3	400	800	60
5	3	0	610	900	40

Table 2. Virtual Operator scheduled events

VO	Action	Time
3	Disappear	450
5	Appear	600

First, the benefits of the proposed tuning parameters to smooth the effects of the appearance of new flows and VOs are assessed. We evaluate the first flow of VO_1 (i.e. 3 hops) because it is the longest connection and the scheduled events of flows and VOs occur inside its duration time. Figure depicts the calculated average delay time (normalized per time unit) in several possible combinations: only the $HM(VO_i)$ is considered ($\alpha = 0$, $\beta = 1$); only the $LM(VO_i)$ is considered ($\alpha = 1$, $\beta = 0$); and the intermediate case ($\alpha = 0.5$, $\beta = 0.5$). The AWRR algorithm has been executed every 10 quantums, i.e. every 4 seconds.

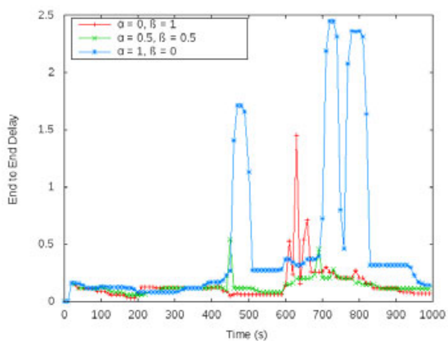

Fig. 2. End to end average delay for the comparison of the tuning parameters

As we can observe, the intermediate case (marked in the graph with *) is the one showing best performance, since extreme cases (marked in the graph with X and |) present more difficulties to adapt to changes and lead to congestion situations. If we only consider the $LM(VO_i)$ when a new flow belonging to a new transmitting VO appears (its $W(VO_i)$ was at $MW(VO_i)$), the queue utilization value will be suddenly very high and it will be granted by the adaption algorithm with a very large $W(VO_i)$, provoking congestion to the other VOs in the following re-adaptations. On the contrary, if we only consider the $HM(VO_i)$, the system will spend a lot of time to provide a large $W(VO_i)$ to new flows causing congestion and many packets losses.

Once the benefits of the tuning parameters have been exposed, the performance improvement for the intermediate case ($\alpha = 0.5$, $\beta = 0.5$) will be assessed. We can clearly see two different situations in the graph depicted in figure 3. Until second 600 the depicted flow is the one with the bigger bit rate so the AWRR scheduler reduces the delay compared to the classical RR case (marked in the graph with *). With the appearance of the flows associated to VO_2 and VO_4, which are characterized by a lower inter-arrival time, we can observe how the delay of VO_1 increases because the AWRR scheduler reduces $W(VO_1)$ and increases $W(VO_2)$ and $W(VO_4)$.

Focusing now on the adaption methods (i.e. packets sent and quantum rounds), after analyzing a set of simulations with different values for each one, the best results are displayed in figure 3. We can conclude that the one triggered by the quantum rounds (marked in the graph with |) behaves much better. At the beginning of the simulation only VO_1 is sending data, so the packet triggered method (marked in the graph with X) needs a long time to send the number of packets established as the threshold, and the adaption is performed very slowly, becoming inefficient. Moreover, in the second half of the simulation when a lot of packets are being sent, the adaption is performed too fast, which makes the process a bit unstable. We can observe in the graph two peaks for the adaption case triggered by the quantum rounds: they are caused by the congestion derived by variations in the transmission flows and the fixed time interval between adaptions.

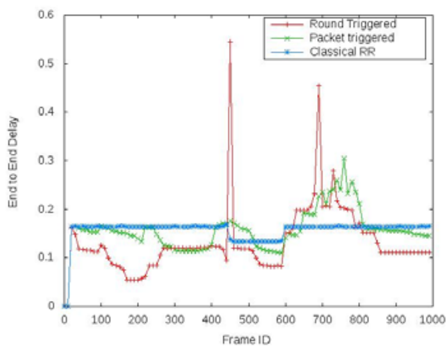

Fig. 3. End to end average delay for the comparison of the adaption methods

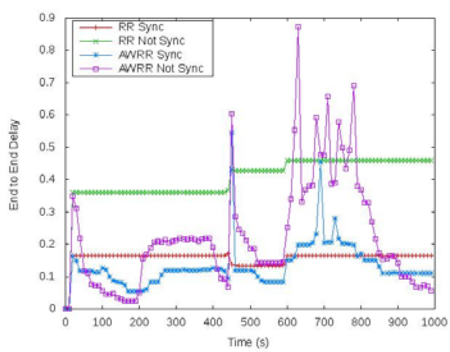

Fig. 4. End to end average delay for the synchronized and non-synchronized cases

Up to now we have assumed synchronization among all nodes. However, sometimes this situation can not be assumed. In figure 4 we show the average delay calculated for the same topology previously analyzed, but comparing the synchronized and non-synchronized transmission. As we could expect, the synchronized case behaves much better since packets can be forwarded directly from the source to the destination in the same time slot without intermediate stops. We can also observe that the AWRR scheduler (marked in the graph with *) has a better response compared to the classical RR scheduler (marked in the graph with).

4.2 Infrastructure Network Topology

The second scenario we have assessed consists in sending several video streams, each one belonging to a particular VO in an infrastructure based topology. The wireless technology chosen for sending the video frames has been the mobile extension of the IEEE 802.16 standard (802.16e, Mobile WiMAX). This technology provides a better solution in terms of bandwidth and coverage area than other wireless schemes such as 802.11. Since 802.16e is not implemented in the official NS2 release, the widely used NIST provided simulation patch [15] has been used.

Several video transmissions have been carried out by using the Evalvid tool [16] as TG, integrated within the NS2 network simulator. Evalvid generates the input source file from the information of video frames to be sent in our scenario in NS2, and also provides the tools to assess the quality of the video streams at the reception node.

The proposed network architecture is depicted in figure 5. We can observe the existence of a central video server in the picture, which sends and routes the video frames to the base stations. Since the requirements of each base station could be different depending on how many VOs are sending video streams at a time, the AWRR scheduler is applied individually in each base station.

Our research has been focused on the transmission of streaming video or Video on Demand (VoD). In order to reduce the jitter experimented by video frames [17], this type of video transmission assumes the existence of a buffer of some seconds at

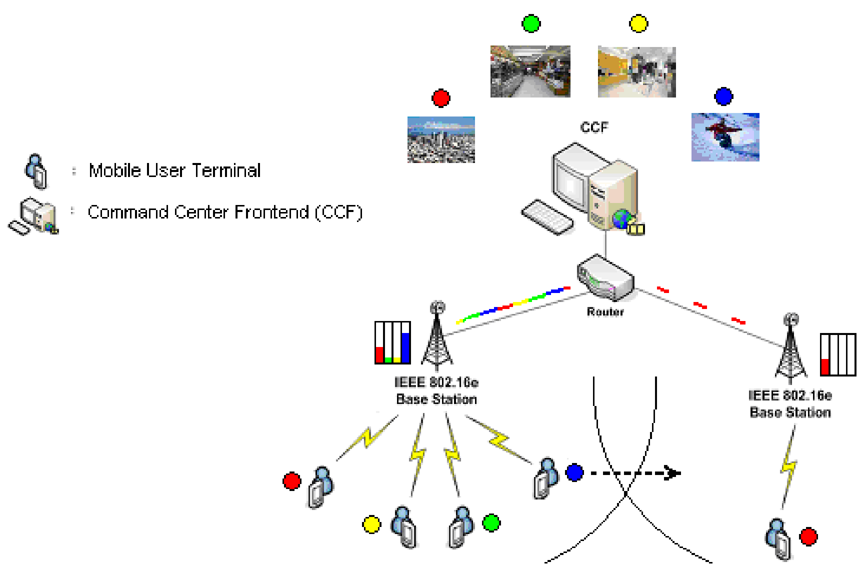

Fig. 5. Network scheme for video transmission

the reception node. In this case the main constraint to overcome is to maintain data losses below the 5 percent recommended for such applications [17].

Four video samples in raw yuv format with a duration longer than three minutes and CIF resolution (352×288 pixels) are considered in our analysis. All the video files where compressed using the H.264/AVC [18] video codec for the transmission. The selection of the video samples is done by covering different levels of dynamism within the frames as we can observe in table 3. When we send a video with higher grade of movement and we do not want to lose quality, we need to increase the number of frames per second or reduce the Group of Pictures (GoP) so that the distance between two intra-frames (I frames) is shorter (increase the percentage of I frames in the coded video file). This fact implies the need of a higher bandwidth, and thus, much more packets to be sent.

Table 3. Transmitted video files

Video File	Level of Dynamism	Bit Rate
SecurityCamera1.yuv	Low	400 kbps
SecurityCamera2.yuv	Low	400 kbps
City.yuv	Medium	800 kbps
Ski.yuv	High	1200 kbps

The first case of analysis consist in the transmission of the four video files, each one belonging to a particular VO, with a 400 milliseconds quantum time and a queue size of 50 packets. These resources are enough for the transmission of videos with low grade of dynamism, but not for those with a higher level, what causes a queue overflow and some frame losses. The adaption is performed every 4 seconds.

Figure 6 – left shows the overall percentage of lost frames per video file depending on the TDMA scheduler used in one base station. As we can observe, the percentage of losses for the AWRR scheduler is much lower and therefore the quality of the videos will be better preserved. In this case the videos with greater grade of dynamism are granted with a larger weight, reducing the amount of lost packets without penalizing the rest of transmissions. Of course, if we would send three or less videos instead of four without reducing the number of VOs in the system, the percentage of losses for the AWRR Scheduler would be even lower.

Usually the scheduler will have to deal with dynamic scenarios where users move towards the radio coverage of another base station while they are receiving the video stream (handover). A user is able to request a new video or stop the one he is watching whenever he wants, so the video transmissions start and stop unexpectedly. Since we cannot plan when a client is going to require a handover or restart a transmission, the base stations cannot remove any of the VO queues permanently even if no video files are being transmitted by that operator. Figure 6 – right shows the percentage of lost packets for a dynamic scenario where users require handovers and transmissions begin randomly within the simulation time. Once again the AWRR results in a better performance. In the graph we can observe an increase in the percentage of lost frames using the AWRR scheduler. This occurs because when a VO starts a new flow after a while without transmitting anything, its $W(VO_i)$ will be at its minimum, $MW(VO_i)$. Many frames will get lost until the scheduler readapts the weights.

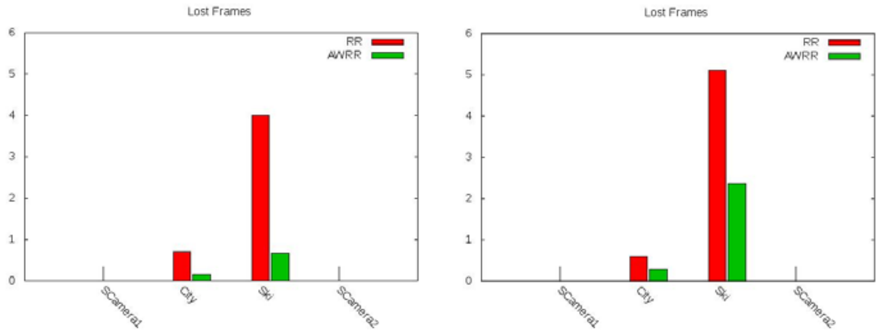

Fig. 6. Percentage of lost frames

5 Conclusions

The proliferation of new operators willing to offer services over physical substrates of others infrastructure providers makes network virtualization an important research field for the construction of the Future Internet. Virtualization of the wireless medium pursues the maximization of usage of a specific wireless physical resource by several VOs running services through the same wireless interface. This work has presented an AWRR algorithm that improves the average performance of a TDMA scheduler.

The benefits of the proposed algorithm have been assessed for two scenarios where the constraints and the requirements of the VOs were quite different. Results in both cases have been pretty promising since the performance and the QoS parameters monitored have been improved with respect to other static scheduling techniques.

As future work to be outlined, the adaption could be tackled from different points of analysis, i.e. the utilization of the channel and the average waiting time, so that a more exhaustive comparison could be carried out. Another future line is to enhance the adaption both introducing more complexity in the algorithm and the tuning parameters or trying to envisage future needs of the VOs according to new generation services and applications.

Acknowledgements

This work has been co-funded by the European Project 4WARD (FP7 Project reference: 216041), and the Future Internet Project, a Research Project from the Basque Government inside the Etortek Program.

References

1. Peterson, L., Shenker, S., Turner, J.: Overcoming the Internet impasse through Virtualization. In: ACM HotNets III (2004)
2. Feamster, N., Gao, L., Rexford, J.: How to lease the Internet in your spare time. Georgia Tech. Technical Report GT-CSS-06-10 (August 2006)
3. Carapinha, J., Jiménez, J.: Network Virtualization – a View from the Bottom. In: ACM SIGCOMM Workshop on Virtualized Infastructure Systems and Architectures (VISA), Barcelona, Spain (August 2009)
4. Hernando, G., Pérez, S., Cabero, J.M.: Mobility-Aware Distributed Embedding (MADE) of virtual networks. In: Future Network & Mobile Summit 2010, Florence, Italy (June 2010)
5. Lu, J., Turner, J.: Efficient Mapping of Virtual Networks onto a shared substrate. In: IEEE International Conference on Network Protocols (2006)
6. Sachs., J., Baucke, S.: Virtual radio: A framework for configurable radio networks. In: International Wireless Internet Conference (WICON), Maui, USA (2008)
7. Egi, N., Greenhalgh, A., Handley, M., Hoerdt, M., Huici, F., Mathy, L.: Towards High Performance Virtual Routers on Commodity Hardware. In: ACM CoNEXT, Madrid, Spain (December 2008)
8. GENI: Global Environment for Network Innovations: Technical document on wireless virtualization (September 15, 2006)
9. Round-Robin Scheduling, `http://en.wikipedia.org/wiki/Round-robin_scheduling`
10. Pérez, S., Cabero, J.M., Miguel, E.: Virtualization of the Wireless Medium: a Simulation-Based Study. In: 69th Vehicular Technology Conference: VTC 2009-Spring, Barcelona, Spain (April 2009)
11. Anastasi, G., De Stefano, E., Lenzini, L.: QoS provided by the IEEE 802.11 wireless LAN to advanced data applications: a simulation analysis. Wireless Networks 6(2), 99–108 (2000), ISSN:1022-0038
12. Chuah, C.N., Katz, R.H.: Characterizing packet audio streams from internet multimedia applications. In: ICC 2002 - IEEE International Conference and Communications (April 2002)
13. Network Simulator 2, NS2, official web page, `http://www.isi.edu/nsnam/ns/`
14. AWRR patch for NS2, `http://www.tecnalia.es/telecom-products.php`
15. NIST Wimax and MIH Module for NS2, `http://w3.antd.nist.gov/seamlessandsecure/`
16. Klaue, J., Rathke, B., Wolisz, A.: EvalVid - A Framework for Video Transmission and Quality Evaluation. In: 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation
17. Szigeti, T., Hattingh, C.: Quality of serving design overview, `http://www.ciscopress.com/articles/article.asp?p=357102&seqNum=2`
18. Stockhammer, T., Hannuksela, M.M., Wiegand, T.: H.264/AVC in Wireless Environments. IEEE Transactions on Circuits and Systems for Video Technology 13(7), 657–673 (2003)