An approach to enhance the QoS support to real-time traffic on IEEE 802.11e networks

Salvatore Vittorio, Lucia Lo Bello
RETISNET Lab
Department of Computer Engineering and Telecommunications
University of Catania
Catania, ITALY
lucia.lobello@unict.it

Abstract-**This paper proposes an approach to overcome some limitations of the 802.11e protocol highlighted in recent literature and improve the QoS support provided to real-time industrial traffic. The proposed solution does not change the IEEE 802.11e protocol, but introduces a technique to reduce the number of collisions and therefore to use the channel more efficiently for real-time traffic, especially when the traffic load is high and approaches saturation conditions.**

The proposed mechanism, called a Contention Window Adapter, dynamically changes the contention window size of the different Access Categories defined by the IEEE 802.11e protocol according to the wokload conditions of the wireless network. The paper describes the rationale behind the CWA mechanism, the algorithm itself and discusses the performance obtained through simulations run using ns-2.

1. Introduction and motivation

The final version of the 802.11e standard, released by the IEEE Task Group E in 2005 [1], introduces two new access mechanisms, i.e., the Enhanced Distributed Channel Access (EDCA) and the Hybrid Coordination Function (HCF) Controlled Access (HCCA). These mechanisms correspond to the ones already present in the IEEE 802.11a/b/g standards, but provide differentiated levels of Quality of Service (QoS) to the supported applications. The EDCA mode extends the IEEE 802.11 Distributed Coordination Function (DCF) [2] by differentiating traffic into four Access Categories (ACs), mapped into the priorities defined by the 802.1D standard [3] as follows:

- AC_BK (background category) for priorities 1 and 2;
- AC_BE (best-effort category) for priorities 0 and 3;
- AC_VI (video category) for priorities 4 and 5;
- AC_VO (voice category) for priorities 6 and 7.

AC_VO is the highest priority category, while AC_BK is the lowest. Each frame arriving at the MAC layer with a defined priority will be mapped into one of the ACs. A dedicated queue for each AC exists, and different service levels are provided to each queue, based on the Arbitration Inter-Frame Space (AIFS), the Contention Window size (CW) and the Transmission Opportunities (TXOP) time interval. Each AC has its own backoff procedure. According to the values of the minimum Contention Window size (*CWmin*) and the maximum Contention Window size (*CWmax*) which, according to the standard are statically set, each AC has a different probability of accessing the channel. Such a probability is higher for the highest priority AC_VO, which features the lowest *CWmin, CWmax* values. The differentiation mechanism provided by the standard offers advantages in terms of delay, jitter and throughput for the AC_VO class: However, recent literature outlined some limitation of the 802.11e protocol when different kinds of traffic are supported on the same channel and the total offered workload is high.

Recent work [4] showed through simulations that the default parameter values of the EDCA mode are not able to guarantee industrial communication timing requirements, when the AC_VO class is used to support real-time traffic in shared medium environments, where other types of traffic are present. The paper concludes stating that new communication approaches must be devised in order to adopt IEEE 802.11e networks on the factory floor. The work in [5] showed that, even in the presence of traffic from the highest priority class only (i.e. AC_VO), according to the amount of real-time traffic and the size of packets in this class, the real-time performance can rapidly and significantly deteriorate with growing workloads. This is due to the *CWmin* and *Cwmax* settings provided by the standard for the AC_VO class, which determine a narrow range of backoff values for the packets in this class. In [5] it was shown that it is beneficial to adapt *CWmin* and *Cwmax* for the AC_VO class to allow for a larger spectrum of backoff values, thus reducing the number of collisions inside the AC_VO class. The approach was evaluated in an industrial scenario where periodic traffic was exchanged and the CWA was applied to the Contention Window size of the AC_VO class only. Simulation results in [5] showed that CWA outperforms the 802.11e standard both as far as throughput and deadline miss ratio are concerned. The reason for this is that it reduces the number of collisions in the AC_VO class.

Here, the approach in [5] is extended to address a general industrial scenario, where Voice, Video and

Background traffic generated by Workstations (WS) is exchanged on the same channel on which field nodes send small-sized periodic RT control traffic with tight deadlines. Our aim is to improve the performance in terms of RT throughput, delay, number of collisions experienced by industrial RT traffic (consisting of small sized periodic packets exchanged between sensors, PLCs, actuators) when generic workstations generate other kind of traffic (multimedia, background) on the same channel. Here we map the industrial RT traffic on the AC_VO class provided by the 802.11e standard.

The targeted extension is not trivial. First, it has to be considered that, when different ACs are to be supported, if the CW_{max} of the highest priority class is increased, the CW_{min} of the lower priority ACs has to be accordingly set, so as to enforce the different QoS support offered to the each AC according to the standard. The CWA has therefore to dynamically tune the $\{CW_{min}, Cw_{max}\}$ range of the different ACs defined in the IEEE 802.11e standard [1] in order to reduce the potential interference (in terms of collisions) that real-time traffic, here mapped into the highest priority class AC_VO, could suffer from other lower-priority ACs. Second, if there is a station which transmits only traffic with a priority other than the highest (AC_VO), the approach in [5] cannot react on the basis of the collisions affecting that type of traffic, as the adaptation mechanism in [5] is defined for transmissions in the AC_VO class only. The CWA extension proposed in this paper addresses both the above mentioned points.

Here we underline that the proposed mechanism does not change the IEEE 802.11e protocol, but introduces a technique to reduce the contention overhead and therefore to use the channel more efficiently for real-time flows, especially when the traffic load approaches saturation conditions.

The paper is organized as follows. Sect. 2 outlines the 802.11e standard, while Sect.3 addresses related work. Sect. 4 describes the CWA mechanism here proposed, while Sect. 5 discusses performance obtained through an extensive set of simulations run in different scenarios under the ns-2 tool [6]. Finally, Sect. 6 gives our conclusions.

2. Overview of IEEE 802.11e

The EDCA mode of the IEEE 802.11e protocol, in order to manage the different Access Categories, implements in each node a dedicated transmit queue and an independent backoff entity for each AC. Each queue works as an independent DCF station and uses its own parameter set, which includes the Arbitration Inter-Frame Space (AIFS), the minimum Contention Window size (*CWmin*), the maximum Contention Window size (*CWmax*), and the Transmission Opportunity limit (TXOPlimit).

Similar to a 802.11 DCF node, each AC starts a backoff timer after sensing an idle channel for a duration equal to an AIFS length. However, while in DCF all nodes have the same opportunity to access the channel, in EDCA the AIFS depends on the AC. As a result, the duration of an idle medium before initiating a transmission is shorter for the higher priority ACs, which thus have higher probabilities of accessing the channel than the lower ACs.

The backoff value is selected as a random number in [0, CW], with CW set as *CWmin* at the beginning of a backoff procedure and increased up to CWmax whenever collisions occur, according to formula (2.1):

$$CW_{new}[AC] = ((CW_{current}[AC] + 1)*2) - 1 \quad (2.1)$$

In case of successful transmission, the CW value of the AC queue is reset to *CWmin*. As *CWmin* and *CWmax* determine the size of the CW, the smaller *Cwmin* and *CWmax* are, the greater the chances for a node gaining access to the medium are.

Finally, TXOPlimit is the time duration an EDCA function may transmit after winning access to the medium.

According to the IEEE standard [1], the above mentioned parameters are set as in Table 1. Note that the highest priority class, AC_VO has the narrowest [Cwmin, Cwmax] range.

Table 1: ACs and relevant parameters

ACCESS CATEGORIES

Access Category	AC_VO	AC_VI	AC_BE	AC_BK
AIFS	2	2	3	7
CW_{min}	7	15	31	31
CW_{max}	15	31	1023	1023
TXOPlimit	0.003008	0.006016	0	0

3. Related work

Among the works which recently addressed the impact on the performance of the IEEE 802.11e protocol of changing the various parameters of EDCA, Xiao [7], extending the Bianchi [8] model, implemented EDCA by means of 3-dimensional Markov chains and analyzed network behaviour for CWs of various sizes. Kong [9] also used 3-dimensional Markov chains to characterize the procedures of the various ACs with variations in both the CWs and the AIFS. Both papers have shown the effectiveness of changing CW depending on the network load.

Mechanisms for CW tuning are presented in [10] and [11]. The approach in [10], called AEDCF, does not implement a mechanism to vary the range of CWs, but calculates an ideal CW on the basis of the network load estimated according to the number of collisions experienced by the transmitted frames. Once the ideal CW is known, the current CW (*CWcurrent*) for the next frame is set by taking whichever is the lower between the minimum CW (*CWmin*) of the AC the frame belongs to and the ideal CW. The approach is shown outperforming EDCF, the IEEE 802.11e pre-standard distributed medium access mechanism. However, it uses the Persistence Factor, a parameter that was present in an earlier version of the IEEE 802.11e standard, but not in the final one.

The AEDCA approach proposed in [11] estimates network congestion by using the value of the current CW (i.e. that of the last frame sent). The distance between the current CW and *CWmin* is compared to the maximum distance between *CWmax* and *CWmin* for the relevant AC, deriving a parameter that is utilized to calculate the new CW for the next frame to be transmitted. The

AEDCA approach, like the AEDCF one, does not provide for changing the values of *CWmin* and *CWmax*, but simply chooses the best one in that range.

Instead of setting the CW to an optimal given value in the [CWmin, Cwmax] range defined by the standard, which proved to be inappropriate in many network load conditions, the CWA mechanism proposed in [5] adjusts the range of the current CW (i.e. the values of *CWmin* and *CWmax*) of the AC_VO class on the basis of information on the newtork workload collected during a time interval. Here we extend this approach, by varying the values of *CWmin* and *CWmax* in a cascaded way for each class. This allows to have, for each class, a CW adapted to the current network status, not limited by the bounds defined by the standard. In addition, here we evaluate the CWA performance in two different scenarios and under different workloads.

4. The Contention Window Adapter (CWA)

The CW_{min} and CW_{max} values for each AC in EDCA are static [1]. Under low workload conditions, small CW values for backoff are a convenient choice. We recall that the backoff value is selected as a random number in [0, CW], where CW is set as *CWmin* at the beginning of the backoff procedure and increases whenever collisions occur up to Cwmax, according to formula (2.1). However, when the network load increases, the probability of collisions increases too, thus enlarging the contention window size could be beneficial to reduce collisions. Unfortunately, EDCA does not implement any mechanism to dynamically change the contention window size of the different ACs according to the workload on the wireless network. The solution here proposed, the Contention Window Adapter, is a mechanism which tries to adapt the CW size of each AC to the network load.

In order to clearly explain the rationale behind CWA, let us consider the results shown in Table 2. They refer to tests run under ns-2 [6] with a 11 Mbps network (DSSS) made up of 20 stations, each generating AC_VO packets of 160 Bytes with a period of 20 ms, giving an overall workload of 1280Kbps. With such small packets and high transmission rate, the AC_VO class obtains poor performance when the setting defined by the IEEE 802.11e standard, i.e. CW_{min}[AC_VO]=7 and CW_{max}[AC_VO]=15 are used. In the different sets of simulations reported in Table 2 the CW_{min} and CW_{max} were statically set at the beginning of each experiment.

Results showed that RT performance of EDCA quickly and significantly degrade with growing workloads (in these conditions the AC_VO class is highly congested [12]) even in the presence of traffic from the highest priority class only (i.e. AC_VO). The reason is the high number of collisions, as the CWmin and CWmax settings provided by the standard provide too narrow a range of backoff values for the packets in the AC_VO class.

The results in Table 2 reveal that, although all the packets which are delivered arrive on time (i.e. they meet the 20 ms deadline assigned to them), the throughput is significantly higher for settings such as CW_{min}[AC_VO]=15, CW_{max}[AC_VO]=31 onwards than with the default settings CW_{min}[AC_VO]=7 and CW_{max}[AC_VO]=15 provided by the IEEE 802.11e protocol.

CWmin	CWmax	Aver.delay (ms)	Throughput (%)
7	15	17	62
15	31	16	89
31	63	7	99

Table 2: Effects of varying CWmin, CWmax on AC_VO performance

In addition, Table 2 shows that, with wider contention windows, the delay experienced by RT packets is reduced. This is due to the smaller number of collisions experienced by RT packets thanks to the broader range of backoff values.

From the various tests run it also emerged that a good way to reduce the collision probability is to vary CW_{min} and CW_{max} by doubling both of them. These results were also confirmed by other tests run with a greater number of stations and different workload conditions.

Similar considerations can be made as far as the other ACs are concerned. In this paper we focus on industrial environments more general than the one addressed in [5], where the only traffic is RT, periodic and mapped into the AC_VO class. Here instead we address a scenario where a number of Workstations (WSs) transmitting different types of traffic (Voice, Video and Background) share the same channel with RT nodes transmitting small size periodic process control frames with tight deadlines. The aim of CWA here is to improve the performance of RT traffic. The parameter used in the CWA to assess the network load is the ratio between the number of collisions affecting the highest priority packets (*coll*(AC_VO)) and the total number of packets sent (*pkts_sent*(AC_VO)) in that AC during a given observation interval Δt:

$$ratio = \frac{coll(AC\_VO)}{pkts\_sent(AC\_VO)} \qquad (4.1)$$

This parameter gives the average number of collisions per packet, and is an index of the level of congestion on the network, with particular reference to the AC_VO class. The number of collisions can be easily obtained. Here we run ns-2 simulations, but also real measurements using network boards equipped with open source drivers are possible. In order to minimize the bias against transient collisions, an Exponentially Weighted Moving Average (EWMA) estimator was used. In a generic *i* interval, the value of r^i_{avg}, to be used by the algorithm, is updated in the following way:

$$r^i_{avg} = (\lambda - 1) \cdot r^i_c + \lambda \cdot r^{i-1}_{avg} \qquad (4.2)$$

The CWA, when adapting the *CWmin* and *CWmax* of the various ACs, takes this parameter into account, in a two-step procedure. First, the CWmin and CWmax of the highest priority class, i.e., AC_VO, are set. Second, to enforce the different QoS support offered to each AC according to the 802.11e standard, suitable changes to the CWmin and CWmax values for the other ACs are made. Let us examine the first step.

- If ratio is below a given minimum threshold α, then the network is underloaded, so it is safe reducing (i.e. halving) CWmin and CWmax for AC_VO to speed up the backoff procedure;
- If ratio is in between two values α and β, which are heuristically set depending on the requirements of the supported RT application, then the current CW_{min} and CW_{max} values for AC_VO are maintained;
- If ratio is higher than β, but still below a maximum threshold γ, then the network load is high and thus, to reduce the probability of collisions, CWA increases CW_{min} and CW_{max} for AC_VO doubling their current values;
- If ratio exceed the maximum threshold, then the network is heavily congested, so in order to drastically react, CWA doubles 2 times both CW_{min} and CW_{max} for AC_VO. This is to male the system more reactive to sudden traffic peaks.

In the second step, CWA arranges the CWmin and CWmax values for the other ACs in order to maintain the differentiation between them, and thus between the various types of traffic. CWA, therefore, whenever the CWmin and CWmax values for AC_VO increase, enforces a cascaded increase in the CWmin and CWmax values for the other ACs, i.e. AC_VI, AC_BE and AC_BK.

Here the upper bound for CW_{max} in the AC_VO class has been set to 63, because higher values, although reducing the probability of collisions, would excessively penalize the performance of this class, introducing too long backoff times. Fig.1 shows the above described procedure, written in pseudo-code.

```
procedure Contention Window Adapter
    if (n_pkt_sent(AC_VO) != 0) then
        r_avg := EWMA(n_coll[AC_VO]/n_pkt_sent[AC_VO])
        if (r_avg ≤ α) then
            call procedure Decrease()
        else if ((α < r_avg) && (r_avg ≤ β)) then do nothing
        else if ((β < r_avg) && (r_avg ≤ γ)) then
            call procedure Increase(1)
        else if (r_avg > γ) then
            call procedure Increase(2)
        end if
    end if
end procedure
```

Fig. 1 The CWA algorithm

The Increase procedure is used to increment CW_{min} and CW_{max} for the various ACs. It requires as parameter an integer value (equal to 1 or 2), which indicates the amount of increment to be done on the CW_{min} and CW_{max} values of all the ACs, according to the ratio value. Here we recall that here an increment means doubling the current values. Table 3 shows the possible combinations of CWmin and CWmax for the various ACs which may occur.

The Decrease procedure is run whenever ratio is lower than the α threshold. Here a decrement means that the current CW_{min} and CW_{max} values are halved. For example, in Table 3, if the current combination at time t is the one displayed on the third row, and at time *t+ Δt* ratio is less than α, the Decrease procedure will be invoked by CWA and the CW_{min} and CW_{max} values of the various delle ACs will be set as in the second row.

	CW min (VO)	*CW max (VO)*	*CW min (VI)*	*CW max (VI)*	*CW min (BE)*	*CW max (BE)*	*CW min (BK)*	*CW max (BK)*
1	7	15	15	31	31	1023	31	1023
2	15	31	31	63	63	1023	63	1023
3	31	63	63	127	127	1023	127	1023
4	31	63	127	255	255	1023	255	1023
5	31	63	255	511	511	1023	511	1023

Table 3: CW_{min} and CW_{max} combinations for CWA

There is a potential problem when using CWA in the presence of stations that do not have traffic in the AC_VO class. If a station tries to transmit only traffic with a priority other than the highest (AC_VO), the CWA cannot react on the basis of the collisions affecting that type of traffic, as ratio is defined for the AC_VO class.
To overcome this problem, the CWA shall consider the AC_VI ratio, expressed as (4.3)

$$ratio = \frac{coll(AC\_VI)}{pkts\_sent(AC\_VI)} \quad (4.3)$$

The CWA behaviour will be the same as before, with the only difference that, to be sure that CW_{max} for AC_VO in those stations which do transmit RT traffic in that class is no greater than CW_{min} for AC_VI in the stations which do not transmit traffic in the AC_VO class, the minimum possible value for CW_{min} for AC_VI will be set to 15 or 63, according to the situation.

To obtain the right value, the station will use a parameter, here called RT_NAV, which is obtained by summing the Duration Field of the AC_VO packets sent during a given observation interval (her chosen equal to 300 ms). The AC_VO packets are identified by the Flow Identification field. All the stations are able to read the header of any packets in transit on the network as, according to the 802.11e standard, the Duration Field is used to calculate the Network Allocation Vector (NAV). The NAV and its associated timer are used to regulate the access to the medium for the various stations avoiding that a transmission trial would interfere with an on-going transmission. The CWA can therefore exploit these features of the standard to enable a station, which does not have traffic in the AC_VO class to transmit, to assess whether there are RT stations sending AC_VO traffic on the shared channel. If, after an observation interval, the station has a non-null RT_NAV value, CWA sets the CWmin value for the AC_VI traffic in that station to 63, otherwise the value 15 will be chosen.

In the following Section, CWA performance obtained in different scenarios will be presented and discussed. In all the addressed scenarios, CWA is run on Workstations nodes (WSs), i.e., stations transmitting interference traffic (Voice, Video, Background), while RT stations, which transmit only periodic small-sized process control frames in the AC_VO class, will use the static CW_{min}(AC_VO) and CW_{max}(AC_VO) values defined by the standard [1], i.e. 7 and 15.

5. Performance Evaluation

Here two different scenarios have been considered to compare the performance of the EDCA with and without CWA. Simulations were run using the Network Simulator version 2.28, with the patch [13][14]. The thresholds used in the CWA procedure were $\alpha = 0.2$, $\beta = 0.6$, $\gamma = 2$, while in (4.2) λ=0.8. As said before, two different station types are present: RT and WSs. All the RT stations transmit traffic to the same Base Station, while WS exchange data between them and with an Access Point. RT traffic is periodic, of CBR type, with a period=20ms, bit rate=18kbps and packet size= 45 byte. In industrial environments, RT stations could be sensors which transmit field variables to a PLC, while WS are generic stations which use the same wireless channel to transmit consumer traffic (Video, Audio, http, ftp etc.).
The performance parameters measured are throughput and delay for all the types of traffic, and the number of collisions for RT traffic. The Physical and MAC layer parameters in ns-2 were set as in Table 4:

PARAMETER	*VALUE*
MAC	802.11e
Physical Layer	802.11b
SIFS	10us
SlotTime	*20us*
PreambleLength	144 bits
PLCPHeaderLength	48 bits
PLCPDataRate	1Mbps
DataRate	11Mbps
BasicRate	1Mbps
ShortRetryLimit	7
LongRetryLimit	4
cfb	Disabled
ifqLen	50
Routing	DSDV

Table 4: Simulation parameters for scenario 1

5.1 Scenario 1

In this scenario, on the same wireless network there are RT stations sending in the AC_VO class periodic traffic with tight deadlines to the BS, and WSs exchanging large packets generated with high data rates in the AC_VO class too. The WSs are inside the same geographic area of the RT stations.

Simulations were run with a growing number of WSs, in the range [2, 10]. Each WS transmits packets of 1000 Byte at a 1Mbps. This kind of traffic will be henceforward called "greedy". The aim of this scenario is to highlight the RT performance in the presence of such WSs sending greedy interference traffic with the same priority of RT ones, i.e., in the AC_VO class. As said before, here CWA is run on the WSs, while the RT stations use the static setting foreseen in the EDCA protocol. Fig. 2 compares the throughput obtained with and without CWA. The CWA parameters The comparison reveals that there is a significant difference in throughput with and without CWA. For example, with 10 WSs, the throughput ratio is about 1/3, i.e., while standard EDCA is able to transmit about one frame out of three, CWA succeeds in transmitting almost all the RT frames. Even the throughput of greedy traffic improve, thanks to the CWA mechanism, which reduces the number of collision within the AC_VO class.

Fig. 3 depicts traffic delay. As shown in the figure, when CWA is run, the RT delay is quite lower than the one experienced when standard EDCA is used.

Finally, Fig.4 shows the number of collisions experienced by a RT station with and without CWA. The results prove that adapting the contention window size according to the network workload significantly reduces the number of collisions for RT traffic. This means that both the timing performance of RT traffic and the overall bandwidth exploitation improve.

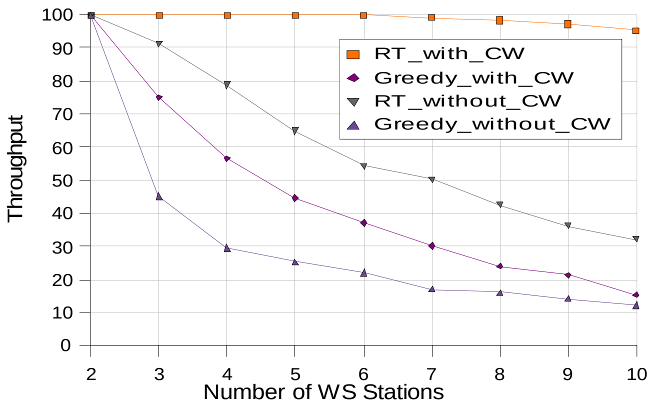

Fig.2: Throughput comparison (percentage)

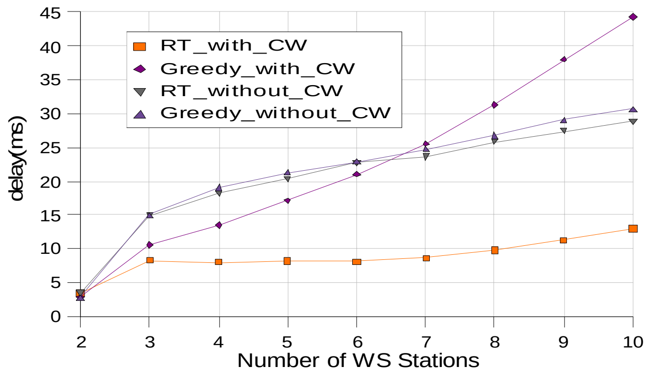

Fig.3: Delay comparison

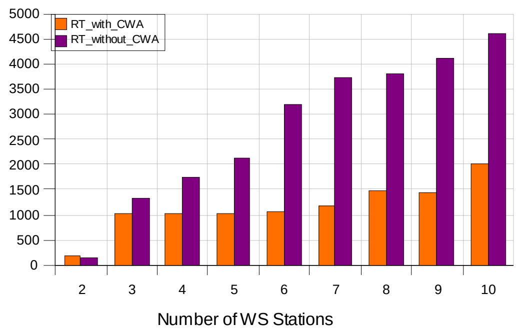

Fig.4: Number of collisions for RT traffic

5.2 Scenario 2

In this scenario, an Open Communication Environment (OCE) similar to the one in [4] is considered. Here, RT stations share the same channel with 5 WSs generating multimedia (Voice and Video) and Background traffic (BK). The aim of this scenario is to evaluate the RT performance in the presence of these interefering WSs. The RT traffic is the same used in scenario 1. For the set of WSs, the offered load, here indicated as GST, ranges from 10% to 100% of the 802.11b PHY data rate (11 Mbps). Each WS generates λ packets per second for the different types of traffic, with the same rate, but different packet size. In order to impose the requested GST overall network load, λ is obtained as in formula (5.1)

$$\lambda = \frac{G_{st}}{(PK_{VO} + PK_{VI} + PK_{BK})} \quad (frame/s)$$

where PK_{VO}, PK_{VI} and PK_{BK} represent the packet size (bits) transmitted in each AC by the WSs (Table 5).

Parameter	*RT Stations*	*Workstations*		
Traffic	RT	VO	VI	BK
CW	CWmin[VO]=7 CWmax[VO]=15	CWA	CWA	CWA
AIFSN	2	2	3	7
Packet size (byte)	45	160	1280	1600

Table 5: Traffic parameters in Scenario 2

Figs. 5-6 show the performance of RT stations, while Figs. 7 depicts the throughput obtained for WSs. For RT traffic, the benefits of CWA, in terms of throughput (fig.5) and delay (Fig.6), are evident.

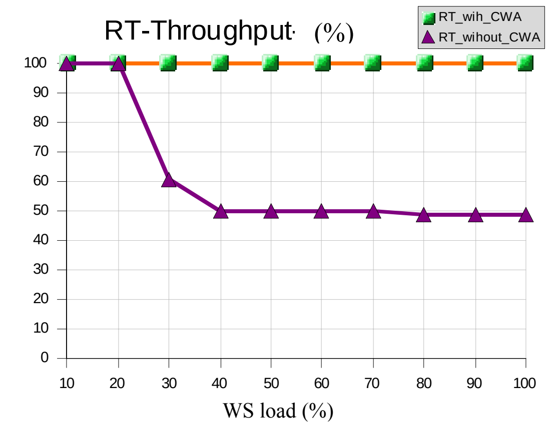

Figure 5: RT Throughput comparison in Scenario 2

There is an improvement even for traffic generated by WSs, both in throughput (Fig.7) and delay values (not shown for reasons of space).

6. Conclusions

CWA proved to be successful in reducing the number of collisions while maintaining traffic differentiation between the different ACs in both scenarios investigated in this paper. Further work will deal with implementation of CWA on COTS network boards.

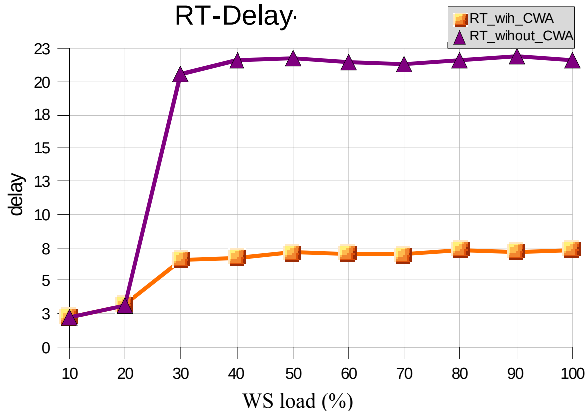

Figure 6: RT Delay comparison in Scenario 2

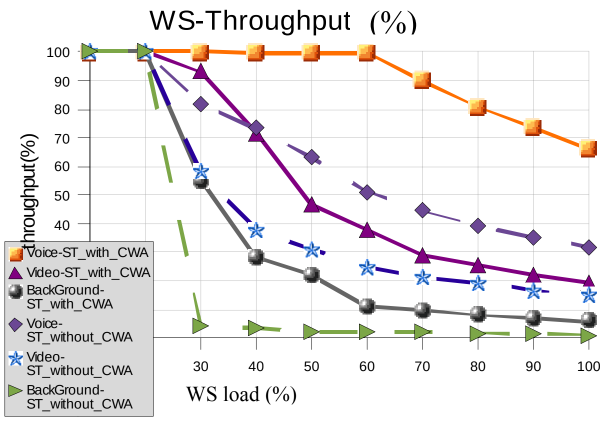

Figure 7: WS throughput comparison in Scenario 2

References

[1] "IEEE 802.11e-2005, Medium Access Control (MAC) Quality of Service Enhancements," 2005.

[2] "IEEE 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999.

[3] "IEEE 802.1D-2004, MAC Bridges," 2004.

[4]R.Moraes, P. Portugal, F. Vasques, "Simulation Analysis of the IEEE 802.11e EDCA Protocol for an Industrially-Relevant Real-Time Communication Scenario", IEEE ETFA'06, Prague, Czech Republic, Sept. 20-22, 2006.

[5] S. Vittorio, G. Kaczynski, L. Lo Bello, "Improving the real-time capabilities of IEEE 802.11e through a Contention Window Adapter", RTAS'07_WIP, pp.64-67, Bellevue, USA, Apr.2007.

[6] The Network Simulator. http://www.isi.edu/nsnam/ns

[7]Y.Xiao, "Performance analysis of IEEE 802.11e EDCF under saturation condition," *Proc. of IEEE ICC*, 2004, pp.170-174.

[8] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *JSAC*, 18, 2000, pp.535–547.

[9] Z.Kong, D.Tsang, B. Bensaou, D. Gao, "Performance Analysis of IEEE 802.11e Contention-Based Channel Access," *JSAC*, 22, 2004.

[10] L. Romdhani, Q. Ni, T. Turletti, "Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks," *Wireless Commun. and Mobile Comp.*, 2004.

[11] J. Sawaya, B. Ghaddar, S. Khawam, H. Safa, H. Artail, and Z. Dawy, "Adaptive Approach for QoS Support in IEEE 802.11e Wireless LAN," *Wireless Mobile*, 2005.

[12]J. Jun, P. Peddabachagari, M. Sichitiu, "Theoretical Maximum Throughput of IEEE 802.11 and its Applications," *IEEE NCA03*, 2003, pp. 249–256.

[13] S. Wiethoelther, M. Emmelmann, C. Hoene, "TKN EDCA model for NS-2," *TR TKN-06-003*. Telecom. Network Group, Tech. Univ. Berlin, 2003.

[14]TKN-802.11e.http://www.tkn.tu-berlin.de/research/802.11e_ns2