Supporting Triple-Play Communications with TDuCSMA and First Experiments

Andrea Vesco
Istituto Superiore Mario Boella
Torino, Italy
Email: vesco@ismb.it

Riccardo M. Scopigno
Istituto Superiore Mario Boella
Torino, Italy
Email: scopigno@ismb.it

Enrico Masala
Politecnico di Torino
Torino, Italy
Email: enrico.masala@polito.it

***Abstract*—This work addresses the implications of using the Time-Division Unbalanced Carrier Sense Multiple Access (TDuCSMA) coordination function to support triple-play services. Firstly, the theoretical background of TDuCSMA is reported, presenting its advantages and discussing its full compliance with the IEEE 802.11 standard. Secondly, a prototype of TDuCSMA is discussed in details. Then, a set of experiments with the prototype implementation of TDuCSMA is presented, showing for the first time the advantages of TDuCSMA in a realistic setting with audio, video and elastic data applications. Experimental results show the superiority of TDuCSMA over the legacy 802.11 Medium Access Control (MAC) in terms of both channel utilization and Quality of Experience (QoE) as measured at the application level.**

I. Introduction

The IEEE 802.11 [1] certainly standardizes the most used technology to implement local wireless networking systems. One of its greatest advantages is its flexibility in providing network services without the need to deploy a wired infrastructure, making the technology suitable for a wide range of scenarios. However, as in any wireless technology, the need to share the wireless medium imposes some limits, in particular the need to share the wireless bandwidth among several users.

In the first version of the IEEE 802.11 standard a distributed coordination function implementing the CSMA/CA channel access scheme was introduced. In CSMA/CA the channel access decision process is distributed among all nodes, *i.e.*, the right time to access the channel is determined by each node autonomously relying on the principle of random access. Although the CSMA/CA is distributed, easy to implement and scalable, it suffers of fundamental limitations in ensuring the necessary quality-of-service (QoS) to all applications, especially in case of many users, due to collisions and sub-optimal decisions. Moreover, network usage inefficiencies of a single application may negatively affect the performance of the whole network. The IEEE 802.11e amendment was introduced to support the increasing demand for QoS. It introduces the Hybrid Coordination Function (HCF) that defines two channel access mechanisms, namely, Enhanced Distributed Channel Access (EDCA) for differentiated QoS provisioning and the HCF Controlled Channel Access (HCCA) for parametrized QoS.

A novel coordination function called Time-Division Unbalanced Carrier Sense Multiple Access (TDuCSMA) has been recently proposed [2]. It merges the advantages of both EDCA and HCCA without inheriting their disadvantages, *e.g.*, low channel utilization with EDCA, the need for a centralized controller and changes to the standardized MAC with HCCA. The TDuCSMA relies on synchronization among nodes and time-driven switching of contention parameters inside nodes to provide a viable solution for bandwidth and traffic management, while exploiting all the available bandwidth. The TDuCSMA operating principles have been extensively investigated by simulations and analytical models both in single [2] and multi-hop [3] scenarios. Moreover, TDuCSMA has also been shown to be fully compliant with the IEEE 802.11 standard [4]. In the same work an architecture was proposed to enable the coexistence of TDuCSMA and EDCA entities on the same node. Finally, the TDuCSMA has been prototyped and assessed in [5] with preliminary experiments comprising CBR long lasting data flows.

The main contribution of this work is to investigate, for the first time in a real setting with real experiments, the performance of triple-play (*i.e.*, audio, video and data) communications over TDuCSMA, showing that TDuCSMA is well suited to support heterogeneous applications, as in home networking scenarios, while maximizing multimedia performance, ensuring high network utilization and retaining full compatibility with the IEEE 802.11 standard.

The paper is organized as follows. Section II describes the operating principles of TDuCSMA and discusses performance and limitations from a theoretical point of view. Then, Section III presents how TDuCSMA can be used to efficiently support triple-play services. The experimental setup and results are presented in Section IV and V respectively, in terms of both network and application level metrics. Section VI draws the conclusions.

II. Time-Division Unbalanced Carrier Sense Multiple Access

A. Operating Principles

In networks supporting TDuCSMA all nodes are synchronized with a common time reference (CTR) whose structure is depicted in Fig. 1. The CTR is a periodical time structure: the time-frame (TF) is the entity representing the time unit and a given number of TFs are grouped in a time-cycle (TC) structure. The time-cycle length T_c, measured in TFs, provides the periodicity of the CTR structure. Both the TF duration (T_f) and the TC duration (T_c) are configurable system parameters.

978-1-4799-3083-8/14/$31.00 ©2014 IEEE

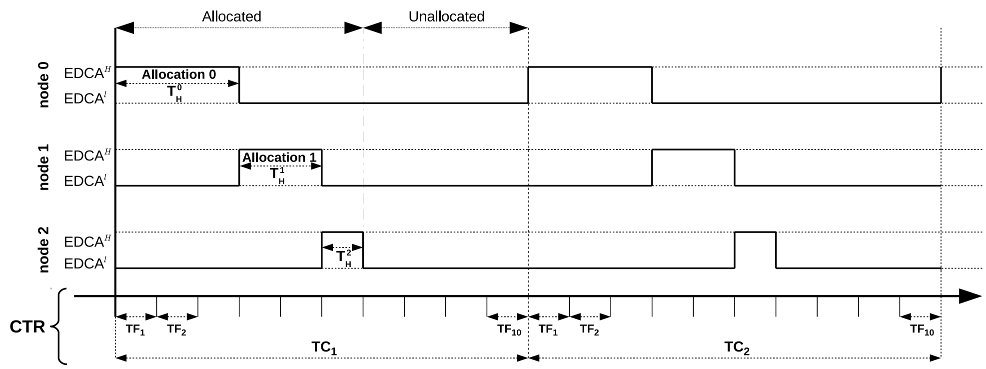

Fig. 1: Time-driven EDCA parameters switching inside three nodes; $T_H^0 = 3$, $T_H^1 = 2$, $T_H^2 = 1$ over a time cycle with $k = 10$ TFs. Some TFs can be left un-allocated to send background traffic.

The synchronization among nodes can be achieved using, for instance, a distributed leaderless solution [6], where nodes collaboratively reach a consensus on a common clock.

Note that, although a typical TDMA time structure is employed, the channel access decision process is distributed among all nodes following the CSMA/CA rules. Two sets of EDCA parameters, which include the Arbitration Inter-frame Space Number (AIFSN) and the minimum and maximum Congestion Windows (CW), are predefined and maintained by all TDuCSMA nodes accessing the network. More formally, the set is defined as $(AIFSN, CW_{min}, CW_{max})$. The two sets are referred to as *high-priority* set $EDCA^H$ and *low-priority* set $EDCA^l$. A subscript is used to distinguish the set to which the parameter belongs. The EDCA parameters are *unbalanced* in the two sets. The values are such that $AIFSN^H < AIFSN^l$ and $CW^H_{min} \leq CW^H_{max} < CW^l_{min} \leq CW^l_{max}$, thus node i, contending for channel access in accordance to $EDCA^H$, has strict priority on node j using $EDCA^l$ settings.

The key idea underlying TDuCSMA is to synchronize the contextual switching of EDCA parameters at each node such that (*i*) only one node at a time contends for channel access in accordance to $EDCA^H$ and (*ii*) all nodes maintain $EDCA^H$ for a predefined periodical time interval, referred to as T_H (measured in TFs.)

Fig. 1 shows an example of a time-driven switching of EDCA parameters inside three nodes in the same collision domain. As shown in the figure, only one node contends for channel access in accordance to $EDCA^H$ during one TF, whereas the time periods in which nodes operate in accordance to *EDCA*^{j} $\forall j = H, l$ can change over the nodes. As a result, following the TDuCSMA operating principles, a node i is very likely to gain access to the channel and maintain it for the full period T_H^i. Note that this happens only due to the CSMA/CA operations and due the values of the access parameters in $EDCA^H$ and not because of a predefined channel access as in TDMA-based solutions[1].

In principle the EDCA parameter sets are switched over time on a per-node basis, so that each node handles QoS-demanding traffic as a single aggregate. Thus, bandwidth management is performed on a per-node basis by assigning different T_H^i to nodes sharing the collision domains. However, some TFs can be left un-allocated to let nodes send background traffic in accordance to either the best-effort or the differentiated service discipline as described in [4].

Moreover, TDuCSMA preserves the CSMA/CA nature, *i.e.*, if a node i does not have enough traffic to send before the end of its T_H^i, any other node can gain access to the channel, thanks to CSMA/CA, and transmit. The consequence is that bandwidth reuse is easily and intrinsically implemented and bandwidth waste, one of the possible side effects of reservation, is avoided.

B. Bandwidth Reservation Model

The work in [2] showed two important consequences of the TDuCSMA operating principles:

1) only node i gains access to the channel during T_H^i, thus the congestion windows in $EDCA^H$ can be minimized to reduce back-off time between two consecutive transmissions hence bandwidth utilization is increased without affecting collision probability;
2) if node i tends to use its T_H^i with poor efficiency due to short packets, this does not affect the transmissions of the other nodes in their respective T_H periods.

Therefore, assuming $CW^H_{min} = CW^H_{max} = 1$ and neglecting the propagation delays, the theoretical bandwidth G_{id}, available for reservation, can be computed as the efficiency in channel utilization considering only the protocol overheads as follows:

$$G_{id} = \frac{R \cdot t_p}{t_p + AIFS^H + 2 \cdot t_{plcp} + t_h + SIFS + t_{ack}}, \quad (1)$$

where R is the linerate, t_p and t_h are the MAC payload and header transmission times, t_{plcp} is the transmission time of the Physical Layer Convergence Procedure (PLCP) header and preamble and t_{ack} is the acknowledgment transmission time.

In TDuCSMA bandwidth reservation is performed, on a per-node basis, by allocating one or more TFs in which channel access contention is performed using $EDCA^H$. An amount of bandwidth

$$G_i = \frac{T_H^i}{T_c} \cdot G_A, \quad (2)$$

is reserved to node i, where G_A is the available bandwidth. Nodes sending QoS-demanding traffic experience very few

[1]The transmission opportunity *TXOP* mechanism is not exploited in TDuCSMA because if a node were delayed in its channel access, *TXOP* would enforce this delay and propagate it with a disruptive effect on the underlying TDuCSMA operating principles.

collisions, basically at the boundaries of their T_H^i, *e.g.,* at the beginning of TF 1, 4 and 6 in the example depicted in Fig. 1. Due to the possibility of some collisions, the available bandwidth in practical conditions (G_A) is lower than the nominal available bandwidth (G_{id}). The work in [2], [3], [4], however, showed that G_A can be reliably estimated from G_{id} by assuming that, in practical conditions, the nominal value decreases by about 10%.

Reverting Eq. (2), it is possible to compute the number of TFs n_i that must be allocated to node i to reserve a given bandwidth G_i as follows:

$$n_i = \left\lfloor \frac{T_c}{T_f} \cdot \frac{G_i}{G_A} \right\rceil = \left\lfloor \frac{k \cdot T_f}{T_f} \cdot \frac{G_i}{G_A} \right\rceil = \left\lfloor k \cdot \frac{G_i}{G_A} \right\rceil \tag{3}$$

where the $\lfloor \cdot \rceil$ is the operator that rounds to the nearest integer.

Eq. (1) can be applied, in principle, only when packet length is constant. However, as shown in [3], the mean packet length alone is sufficient to deal with the reservations in TDuCSMA, and the detailed nature of the packet size distribution has only a secondary effect. Therefore Eq. (1) and consequently Eq. (2) and Eq. (3) can be generalized to work with variable packet lengths by estimating G_{id} as follows:

$$G_{id} = \frac{R \cdot T_P}{AIFS^H + 2 \cdot t_{plcp} + T_P + t_h + SIFS + t_{ack}}, \tag{4}$$

where T_P is the average value of the MAC payload transmission time.

III. Triple-Play Communication over TDuCSMA

This section explains how to tune TDuCSMA to provide triple-play services in the most effective way. Providing triple-play services require an adequate support for both multimedia (*e.g.*, audio and video) and data applications. Multimedia applications needs to receive data periodically with sufficiently low delay, and packet loss rate must be minimized, although few packet losses can be tolerated by some of them. Video-conferencing applications are the most demanding ones due to their low delay requirement in the order of 150 ms maximum. Other applications such as live streaming exhibit more relaxed constraints.

TDuCSMA offer bandwidth and traffic management capabilities well suited to the triple-play scenario. An appropriate reservation must be performed at TDuCSMA node, in accordance with the model described in Section II-B, on the basis of the characteristic of multimedia flows. However, note that the exact value of the reservation is not so critical because TDuCSMA is, by its nature, adaptive since it intrinsically allows bandwidth reuse. Hence any remaining part of a video flow that does not fit into the reservation can exploit the bandwidth unused by the other flows, with obvious benefits in terms of QoE. This is a great advantage with respect to traditional TDMA-based solutions where reuse must be implemented with specific functions and indeed it represents a cost in terms of complexity.

The data applications can be efficiently supported by accommodating their elastic traffic flow within the remaining bandwidth left un-reserved.

IV. Experimental Settings

A. WiFi Node

The legacy 802.11 node is made of a router board running OpenWRT [7], a Linux distribution for embedded devices. The router board is equipped with one miniPCI wireless card based on the Atheros chipset. The wireless Atheros drivers are open-source and implemented by the *ath5k kernel module* within OpenWRT.

B. TDuCSMA Node

The prototype of a TDuCSMA node [5] is built on top of the 802.11 node as an extension of the wireless Atheros drivers implemented by the *ath5k kernel module*. This module gives the kernel the capability to easily load and unload the miniPCI wireless card module and to control the inner functioning of the chip. It does not implement the IEEE 802.11 MAC layer, since this is implemented both in hardware and in the *mac80211 kernel module* included with OpenWRT.

The TDuCSMA implementation simply consists of two files added to the *ath5k kernel module*. The header file contains the definition of the tdu_csma structure, which defines the contention parameters in the $EDCA^H$ and $EDCA^l$ sets, the CTR structure configuration and a representation, in the form of an array, of the TF allocations within a time cycle. Each element in the array is a flag, equal to 1 if the corresponding TF is allocated to the node and 0 otherwise. Moreover, there is a timer to schedule the beginning of the next TF. This timer is implemented by using the Linux 2.6.x kernel built-in high resolution timer (*hrtimer*) that provides greater precision compared to other legacy solutions. The source code file contains the implementation of three functions that are called at specific instants during the *ath5k kernel module* execution to implement the TDuCSMA operating principles:

1) tducsma_setup() is called when the *ath5k kernel module* is loaded into the kernel and initializes the tdu_csma structure with default values defined in the header file;
2) tducsma_localization() is periodically called to localize the node within the CTR structure depicted in Fig. 1. It determines the current TF from the system time τ that is synchronized with the other TDuCSMA nodes as follows:
$$\mathrm{TF} = \mathrm{mod}\left(\left\lfloor \frac{\tau(t)}{T_f} \right\rfloor, TC \right) \tag{5}$$
and then schedule the beginning of the next TF after a period of time
$$T_{\mathrm{comp}} = \left\lceil \frac{\tau(t)}{T_f} \right\rceil - \tau(t) \tag{6}$$
to compensate for possible drift in TF beating among nodes;
3) tducsma_callback() is called at the beginning of each TF and schedules itself after a period equal to T_f to implement the periodic TF beating. It writes the contention parameters in the Atheros chipset in accordance with the TF allocation array at each execution. It uses the parameters in $EDCA^H$ during the TF allocated to the node and $EDCA^l$ otherwise.

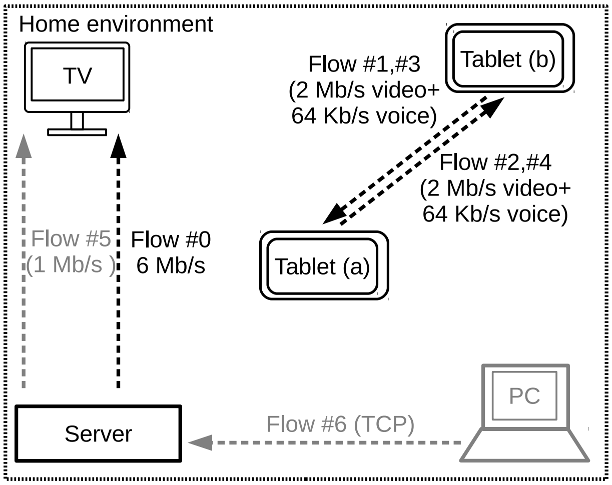

Fig. 2: Experimental setup showing the location of the nodes and the communication flows. Those in grey are active only in some scenarios.

As soon as the *ath5k kernel module* is loaded into the kernel, the tducsma_setup() function is called. Then the *ath5k kernel module* performs a reset operation of the wireless card and the function tducsma_localization() is called for the first time in order to acquire the synchronization and to start the periodic TF beating. Then, the function tducsma_callback() is called at the beginning of each TF to write the contention parameters into the registers of the Atheros chipset in accordance with the TF allocations and to schedule the beginning of the next TF.

Note that the TDuCSMA implementation only requires minor changes to the *ath5k kernel module* to enable the switching of contention parameters in accordance with the CTR structure and TF allocations. Conversely, the basic CSMA/CA operations implemented by the *mac80211 kernel module* are not altered at all and the same is, of course, for the underlying hardware. This fact itself confirms that TDuCSMA is IEEE 802.11 standard compliant, as claimed in [2], [3], [4].

The TDuCSMA relies on the CTR structure to coordinate EDCA parameters switching inside nodes. The synchronization procedure involves two steps: nodes refer to a local clock — the system time τ — that is kept synchronized with the other nodes, then they exploit this common time reference in order to deploy the CTR structure and to localize themselves correctly within the CTR. However, the synchronization in TDuCSMA is not so critical [6], because it needs to synchronize nodes at TF-level — typically the TF has 1ms duration — and not at bit-level, which would be much more complex. Moreover the TF boundaries are not so strict as in a traditional TDMA-based solution, in which packets must be completely transmitted within the time slot.

Since the accuracy required for the synchronization is not so critical the system clock τ of each node is maintained synchronized with the others using the Network Time Protocol (NTP) [8] by relying on a single, out of band connection to a remote NTP server[2].

While NTP fulfills the first step of the synchronization

[2]The testbed leverages the NTP free service provided by Istituto Nazionale di Ricerca Metrologica (INRIM) available at address *ntp1.inrim.it*

TABLE I: Characteristics of the flows and composition of each scenario.

Flow ID	Application type	Video sequence name and resolution	Avg. bit-rate [kb/s]	Mean pkt length	Scenario A	Scenario B	Scenario C
0	HD video	*flatirons* (1280×720)	6678	1399	yes	yes	yes
1	videoconference	*spectrum* (640×480)	2001	1353	yes	yes	yes
2	videoconference	*spectrum* (640×480)	2001	1353	yes	yes	yes
3	voice	-	64	160	yes	yes	yes
4	voice	-	64	160	yes	yes	yes
5	videosurveillance	*coastguard* (352×288)	1001	1239	no	yes	yes
6	data (TCP)	-	variable	1460	no	no	yes
Total mean offered load [kb/s] (Note: TCP goodput in Table III)					10808	11809	11809 +TCP

procedure, each node implements the second by leveraging the tducsma_localization() and tducsma_callback() functions. Note that each node schedules the beginning of the next TF by tducsma_callback() in accordance with its local clock. This clock is prone to drift, hence the TF beating progressively gets un-synchronized among nodes. Therefore, each node has to resynchronize the TF beat after a given number of time cycles. This is achieved by calling again the tducsma_localization() function from tducsma_callback() when needed.

All TDuCSMA related parameters are stored, as variables, within the tdu_csma structure in the *ath5k kernel module*. For convenience, the virtual file system *sysfs* is exploited to enable their configuration by simply writing the desired values into specific files mapped to the variable themselves. In this way it is also possible to change TF allocations, hence bandwidth reservation, dynamically at run-time.

C. Testbed

The testbed in Fig. 2 is made of five static wireless nodes operating in ad-hoc mode, using the IEEE 802.11a physical layer, on channel 40 at 5.200 GHz. The transmit power is set to 10 dBm and the line-rate is R=18 Mb/s with auto-fall-back disabled. Each node can either operate as a WiFi node, contending for channel access in accordance to CSMA/CA, or as a TDuCSMA node.

The TDuCSMA parameters are set as follows:

$$AIFS^i = SIFS + AIFSN^i \cdot slotTime \; \forall i = H, l$$

where $AIFSN^H = 2$ and $AIFSN^l = 7$, $CW^H_{min} = CW^H_{max} = 1$, $CW^l_{min} = 31$, $CW^l_{max} = 1023$, $TC = 25$ TFs, $T_f = 1$ ms.

The legacy WiFi access parameters are set as follows:

$$DIFS = SIFS + 7 \cdot slotTime$$

and the congestion window varies between $CW_{min} = 15$ and $CW_{max} = 1023$.

Several multimedia applications are included in the experiments, ranging from simple videosurveillance to more demanding audio and videoconferencing and high-definition (HD) video streaming. Uncompressed video sequences (available at *www.cdvl.org*) have been encoded at 30 frames per second using the H.264/AVC test model software JM18 [9]. Three different scenarios, with an increasing number of applications up to the triple-play case, are evaluated as reported in Table I. Note that audio applications generate CBR flows while both video and data applications generate VBR flows with variable

TABLE II: Average packet delay (ms) experienced by the various applications, for the three experimental scenarios, for TDuCSMA (no brackets) and WiFi (in brackets).

Scenario	HD video	videosurveillance	videoconference #1	videoconference #2	voice #3	voice #4
A	25 (43)	-	10 (6)	11 (7)	7 (3)	8 (5)
B	27 (42)	23 (31)	10 (8)	13 (9)	8 (4)	10 (6)
C	48 (132)	40 (126)	32 (11)	32 (104)	30 (8)	30 (100)

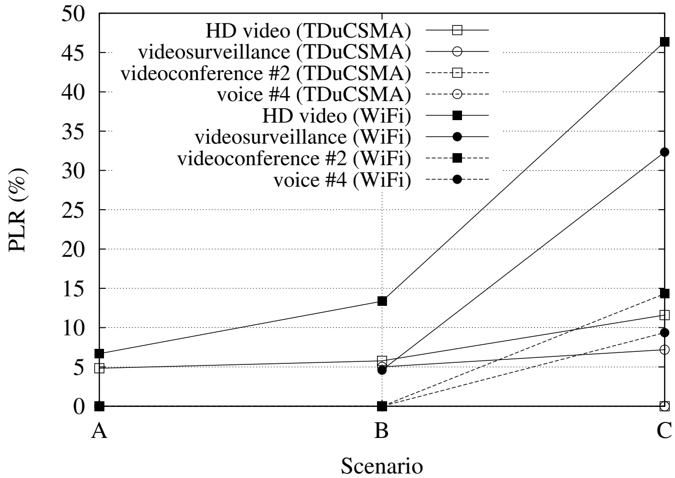

Fig. 3: PLR in the three experimental scenarios. Videoconference #1 and voice #3 are not shown since they presents PLR = 0% for both TDuCSMA and WiFi.

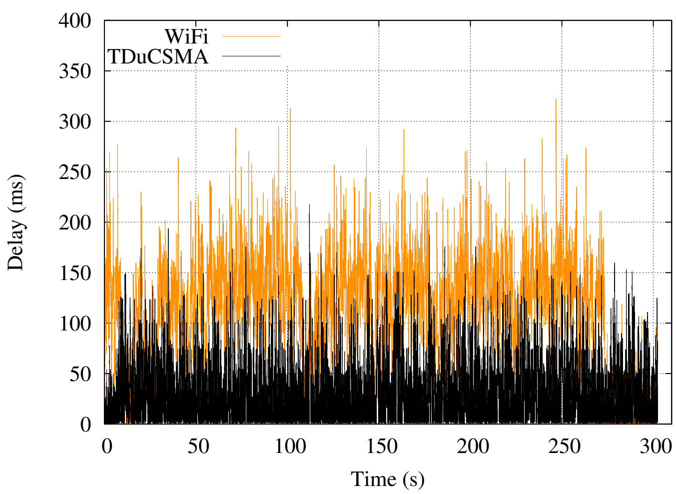

Fig. 4: Average packet delay experienced by the videoconference #2 application, in Scenario C, as function of time.

packet length. This is the most general traffic condition under which a new coordination function can be assessed.

A bandwidth share, equal to the mean bit-rate of the generated flow, is reserved to multimedia applications, while the remaining bandwidth is left un-reserved for contention among multimedia and data applications.

V. RESULTS

The real-time multimedia applications use the IP/UDP/RTP protocol stack. Therefore, it is easy to detect the loss of packets by means of the RTP sequence number. Using the information about packet losses the performance at the application level are computed in terms of a synthetic video quality index, the Peak Signal to Noise Ratio (PSNR), which is a widely used quality measure in the multimedia communications research community.

Figure 3 shows the packet loss rate (PLR) for the three experimental scenarios described in Table I. These results only take into account network losses and not the ones due to excessive delays. The data clearly show that, as the offered load increases, the PLR increases when the WiFi nodes are employed, while it remains at much lower levels with TDuCSMA nodes. The voice flows do not experience noticeable losses with TDuCSMA whereas in Scenario C voice #4 with WiFi presents about 10% PLR which makes the flow unusable. These results experimentally confirm that TDuCSMA is able to support more heterogeneous communications with better QoS. Also the result about average packet delay in Table II confirms this claim. Note, in particular, the increase of the delay for the videoconference, both for video (#2) and voice (#4).

To achieve good quality videoconferencing the end-to-end delay should be lower than about 150 ms [10], a constraint which will not be completely satisfied by videoconference #2 in Scenario C with WiFi nodes. In fact, as shown in Fig. 4, the delay with WiFi is higher and more variable than with TDuCSMA.

In order to better understand the implications of the playout buffer on the overall performance at the application level, Fig. 5 shows the video quality, in terms of PSNR, as a function of the imposed playout buffer. While all curves are almost flat with TDuCSMA nodes, when a playout buffer equal to 150 ms is employed the videoconference application suffers due to the fact that WiFi nodes cannot keep the delay low. Also other flows exhibit the same behavior, however for them it is less important since their delay constraint is more relaxed. Videoconference #1 is not included to improve figure clarity since it presents no losses in all scenarios.

However, the optimal playout buffer size depends on the application requirements. While for the case of videoconferencing 150 ms is the upper bound to allow good interaction between the participants, for streaming applications the situation is more relaxed. In this work a 500 ms playout buffer is employed for both the HD video and the videosurveillance application. With these settings, the overall performance at the application level, in terms of PSNR, is shown in Fig. 6 for the three experimental scenarios. It is clear that the TDuCSMA is able to provide good QoE to all applications while the WiFi strongly suffers in heavy load conditions such as Scenario C.

To further confirm and quantify the advantages of TDuCSMA over WiFi, Fig. 7 shows the PSNR as a function of time for the HD video flow in the heavy load Scenario C. Since standard test video sequences are usually short (about

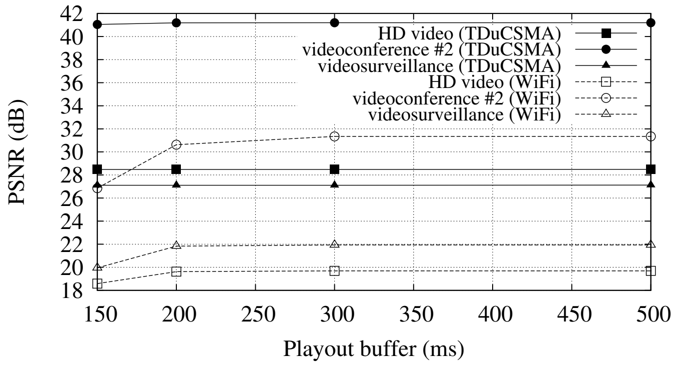

Fig. 5: PSNR as a function of the playout buffer for the video applications, in Scenario C.

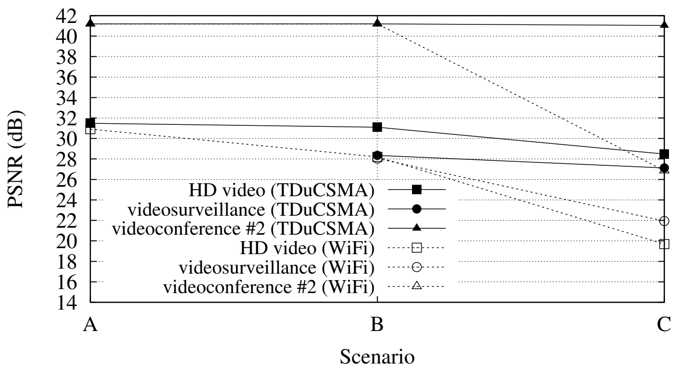

Fig. 6: PSNR of the video applications in the three experimental scenarios.

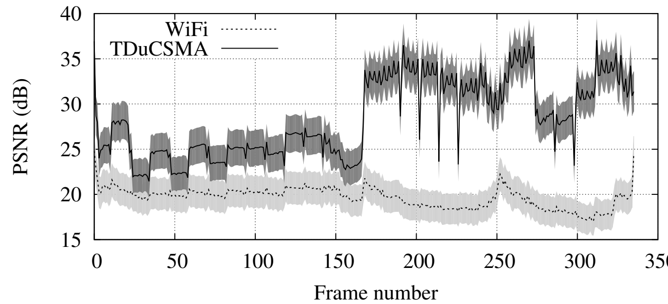

Fig. 7: PSNR of the HD video application, in Scenario C, as a function of time. The bands represent the 95% confidence interval.

10 seconds), the original video sequences have been repeated several times to achieve a duration equal to the time of the experiment, *i.e.*, about 5 minutes. This allows to average the PSNR value for each frame in each repetition, thus smoothing out occasional interferences and packet losses. The shaded area around each curve shows the 95% confidence interval. The figure shows that the average performance provided by the TDuCSMA solution is always superior to the one of the WiFi and that the confidence intervals almost always do not overlap.

Table III provides the performance, in terms of goodput, of the data application using the Transmission Control Protocol (TCP) in Scenario C. The results show that TDuCSMA can efficiently support also elastic data communications. Moreover it is worth noting that the node running data application in Scenario C is a legacy WiFi node. Thus the experiment fully demonstrates that TDuCSMA is backward compatible and that TDuCSMA and WiFi nodes can coexist.

All in all, the TDuCSMA appears to be a suitable solution to efficiently provide triple-play services while maintaining full compatibility with existing WiFi nodes.

VI. Conclusions

This work addressed the implications of using the Time-Division Unbalanced Carrier Sense Multiple Access (TDuCSMA) coordination function to support triple-play services. The TDuCSMA has been discussed from the theoretical point of view as well as using actual transmission results in a prototype setting, highlighting its advantages and, in particular, its full compliance with the IEEE 802.11 standard. The rich set of results has demonstrated the superiority of TDuCSMA over the legacy 802.11 MAC in providing efficient triple-play services while ensuring high channel utilization, a remarkable feature especially in wireless domains where bandwidth is a scarce but precious resource, and high QoE.

TABLE III: Goodput of the TCP source.

Scenario C	WiFi	TDuCSMA
Goodput [kb/s]	2200	2340

Acknowledgments

The authors would like to thank Davide Nicassio for running part of the experiments presented in this paper.

References

[1] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. ISO/IEC 8802-11, ANSI/IEEE Std 802.11, 1999.

[2] A. Vesco and R. Scopigno, "Time-division access priority in CSMA/CA," in *Proc. of IEEE Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2009)*, Tokyo, Japan, Sep. 2009, pp. 2162–2166.

[3] ——, "A distributed bandwidth management scheme for multi-hop wireless access networks," in *Proc. of Wireless Communications and Mobile Computing Conference (IWCMC 2011)*, Istanbul, Turkey, Jul. 2011, pp. 534–539.

[4] ——, "Advances on time-division unbalanced carrier sense multiple access," in *Proc. of IEEE Conference on Computer Communications and Networks (ICCCN 2011)*, Maui, Hawaii, Aug. 2011, pp. 1–6.

[5] A. Vesco, G. M. Toso, and R. Scopigno, "Prototyping of time-division unbalanced carrier sense multiple access and first experiments," in *Proc. of IEEE Global Communications Conference (GLOBECOM 2012)*, Anaheim, California, USA, Dec. 2012, pp. 5500–5505.

[6] A. Vesco, F. Abrate, and R. Scopigno, "Convergence and performance analysis of leaderless synchronization in Wi-Fi networks," in *Proc. of ACM Workshop PM2HW2N*, Miami, FL, Nov. 2011, pp. 49–58.

[7] OpenWRT Team. (2013, Sep.) OpenWRT. [Online]. Available: http://openwrt.org

[8] D.L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[9] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG. (2011, Nov.) Joint Model Number 18.0 (JM-18.0). [Online]. Available: http://iphome.hhi.de/suehring/tml/download

[10] ITU-T G.114, "One-way transmission time," *ITU-T*, May 2000.