A Proposal of Real-Time Publish-Subscribe Scheme Compatible with 802.11e Wireless Networks

Raimundo Viégas Jr
Federal University of Pará
Faculty of Computer Science
Belém - Brazil
Email:rviegas@ufpa.br

Paulo Portugal
University of Porto
FEUP, DEEC
Porto - Portugal
Email:pportugal@fe.up.pt

Luiz Affonso Guedes
Federal University of Rio Grande do Norte
Department of Computer Engineering
Natal - Brazil
Email:affonso@dca.ufrn.br

Francisco Vasques
University of Porto
UISPA, FEUP
Porto - Portugal
Email:vasques@fe.up.pt

***Abstract*—In this paper, we propose a real-time publish-subscribe communication scheme for IEEE 802.11e networks, named Group Sequential Communication (GSC). The GSC intends to improve the HCCA mechanism by adopting a simple approach based on the Virtual Token Passing (VTP) protocol which uses a real-time group concept, where the real-time members of the group are granted a high-priority and sequential access to the communication medium. Their main achievements are the reduction of the Polling, individual ACKs and NULL frames overheads between the controller and the polled stations. In order to improve the reliability of the GSC scheme was developed a fault-tolerant mechanism based on a block acknowledge strategy to avoid missing real-time message deadlines.**

I. Introduction

The use of Wireless Local Area Network (WLAN) is rapidly growing in industrial automation. Several factors contribute to it, such as a flexible operation, easiness of reconfiguration, mobility and reduced installation and maintenance costs. The use of these networks have been identified as an attractive option for factory automation, distributed control systems, automotive systems and networked embedded systems [1]. Usually in WLANs the generic traffic is fault-tolerant, but it does not impose significant time restrictions. However, traffic in industrial applications has real-time requirements with a low error-tolerance [2]. Therefore, it is necessary that WLANs provide solutions to cope with these requirements.

A. IEEE 802.11 Standard

The IEEE 802.11 protocol was standardized in 1999 by the Institute of Electrical and Electronics Engineers (IEEE), and was latter reaffirmed on 2003 [3]. The basic service set (BSS) is the building block of an IEEE 802.11 WLAN, which actually provides two types of configurations: independent BSS (IBSS) and infrastructure. The IBSS is the most basic type for an IEEE 802.11 WLAN, which may be composed of, at least, two stations. This operation mode is often referred to as Ad-Hoc. The infrastructure mode includes one or more Access Points (AP) that convey the communication among wireless stations.

The IEEE 802.11 MAC layer was designed to implement two distinct mechanisms [4] [5] [6]. The first one, the Distributed Coordination Function (DCF), is the basic mechanism of IEEE 802.11. It is based on a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) approach, which is random, non-centralized coordinated medium access technique that avoids collisions using a backoff-algorithm. In addition, it uses Interframe Spaces (IFS) to assign different medium access priorities to stations, called: Short Interframe Space (SIFS), Point Coordination Function Interframe Space (PIFS), DCF Interframe Space (DIFS) and Extended Interframe Space (EIFS). The DCF function offers an asynchronous data service, which is applicable in Basic Service Set and Independent Basic Service Set. The later, the Point Coordination Function (PCF), is a polling protocol that allows access to a free contention environment able to coexist with the DCF mechanism. It was proposed to support real-time traffic. PCF uses an access point (Point Coordinator) to control the access to the medium. The medium is reserved for a time interval called Contention Free Period (CFP), where a list of stations is polled. However, their complexity has limited their popularity. The large majority of WLAN cards does not have the PCF mechanism implemented.

B. IEEE 802.11e Amendment

The need for real-time communication for IEEE 802.11 networks lead to the establishment of the IEEE 802.11e task group on July 1999. On December 2005, the Task Group E published the IEEE 802.11e amendment [7]. This amendment intended to provide differentiated levels of Quality of Service (QoS) to applications, including the support of voice and video over WLANs. It incorporates an additional function called Hybrid Coordination Function (HCF), which is only used in QoS network configurations. The HCF provides two mechanisms with QoS requirements: the Enhanced Distributed Channel Access (EDCA), which delivers traffic based on differentiating user priorities (UPs), and the HCF Controlled Channel Access (HCCA), which allows the reservation of Transmission Opportunities with the Hybrid Coordinator.

1) HCF Controlled Channel Access (HCCA): The HCCA mechanism was proposed to improve the PCF and is intended to guarantee bounded delay requirements, based on a round robin scheme. Contrasting to the PCF scheme, the HCCA operates during both the Contention Free Period (CFP) and the Contention Period (CP), where the Hybrid Coordinator (HC) is able to capture the control of the wireless medium, by waiting a PIFS interval between transmissions of stations using the EDCA or DCF mechanism. The HC may include a CFP parameter in the Beacon frame, which informs all stations to set their Network Allocation Vectors (NAVs) for the end of the controlled phase. During the CFP, the HC controls the access to the channel by polling all stations in the polling list, even if some stations does not have packets to transmit. When the HC polls a station that has no packets to transmit, the station will transmit a Null frame. This lead to a low channel utilization with a high implementation overhead [8]. For each polled station it is granted a Transmission Opportunity (TXOP) to transmit its own traffic. The HC is also allowed to start a TXOP in the CP immediately after the channel been idle for one PIFS, that is called Controlled Access Phase (CAP). A CAP ends when the HC does not reclaim the channel after a duration of PIFS at the end of a TXOP. However, QoS stations (QSTAs) must send QoS reservation requests using a special QoS management frame, called Traffic Specification (TSPEC). The TSPEC frame contains the set of parameters that define the QoS characteristics of the network traffic [9]. This HCCA polling mechanism is illustrated in Figure 1.

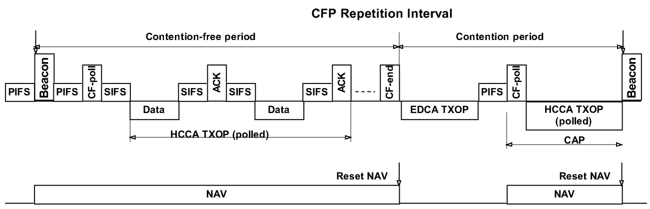

Fig. 1. HCCA Polling Mechanism.

C. Performance Improvement Mechanisms

The IEEE 802.11e also provides some other mechanisms for QoS enhancements.

- The first is the Block Acknowledgment (BlockACK). The mechanism allows the acknowledgment of a group of frames instead of a single one. It can be initialized through a setup and negotiation process between the QSTA and the QAP (QoS-enhanced AP). When the BlockACK has been established, multiple QoS data frames are transmitted in a contention free burst with a SIFS interval between adjacent frames. At the end of the block, all data frames are acknowledged by a bit pattern transmitted in a block acknowledgment frame [10]. This reduces the bandwidth overhead imposed by the conventional ACK mechanism [11].
- The second is called Direct Link Protocol (DLP). It allows two QSTAs associated with the same QAP to transmit data frames directly to each other (the packets do not need to be relayed by AP). If the link is found inactive, QSTAs are disconnected after a DLP idle time out. This mechanism saves network bandwidth [12].
- The third is a No Acknowledgment frame (No-ACK), which is used in applications that have strict delay requirements, but can tolerate packet losses. Finally, there is a Piggyback mechanism that allows data to be sent “piggybacked” on polls and ACKs frames to reduce the network overhead [11].

D. Motivation to Improvement the HCCA Mechanism

The HCCA mechanism is an improvement of the legacy IEEE 802.11 PCF. However, some preliminary research studies [13] have reported that this mechanism is not well-suited to support real-time communications, mainly due to the high polling overhead caused by stations that do not have messages to transmit.

Nevertheless, we believe that an improved HCCA mechanism has the potential to provide an interesting real-time communication service. The major challenge concerning the improvement of the original HCCA mechanism is the development of new scheduling algorithms, in order to provide adequate real-time communication services.

In this paper, the GSC scheme is proposed to reduce the HCCA network overhead by using a publish-subscribe paradigm for real-time message transmission. The GSC scheme compels the sequential transfer of messages from the members of a real-time group, without the required polling, null messages and ACK overheads. A fault-tolerant mechanism is also proposed to deal with message losses.

The remainder of this paper is organized as follows. Section II describes the state-of-the-art of real-time communication approaches based on the IEEE 802.11e standard. Section III provides a brief introduction to the use of the publish-subscribe approach in industrial communications. Section IV describes the GSC proposal. Section V presents the admission control and schedulability analysis of the GSC. Afterwards, in section VI, some conclusions are drawn.

II. Review of Related Works

Currently there are multiple innovative approaches being developed to provide real-time behavior in wireless-supported applications. In this section, we briefly describe some of the most relevant approaches that have been proposed to support real-time traffic upon the IEEE 801.11 Wireless Networks. Although this standard supports both Ad-hoc and Infrastructure modes, but only solutions based on infrastructure mode are presented here.

Lo, Lee and Chen [14] designed a multipolling mechanism called Contention Period Multipoll (CP-Multipoll), which incorporates the DCF access scheme into the polling scheme. It uses different backoff time values for the multiple message streams in the polling group, where each station executes the backoff procedure after receiving the CP-Multipoll frame. The contending order of these stations is the same as the ascending order of the assigned backoff time values. The station in the polling list initializes its transmission immediately after receiving the CP-Multipoll frame. This action avoids the interference from other stations performing the backoff procedures in the DCF mode. Moreover, in order to avoid the repeated collisions between stations that are operating on the same channel in the overlapping space, the values assigned in the CP-Multipoll among neighboring BSSs must be different.

Son, Lee, Yoo and Park [9], proposes an effective and simple polling scheme to reduce the number of polling times for a station with no packet to transmit. In this scheme, when the polled station transmits a data holding packet, the station can get polls more frequently by reducing the polling period. If the polled station is changed to the maximum polling period that is decided by data characteristics. For example, the simulation result shows that the throughput increases by 35.8% when fifteen stations out of thirty stations in the polling list have packets to transmit in normal data transmission conditions, there is not more advantage than original HCCA keeping the same HCCA problems.

Lim at al [8] describes the enhancements of the draft IEEE 802.11e standard for supporting QoS in WLANs. They have described a MAC architecture that when implemented at a station, allows it to support the features of the standard. An algorithm for ensuring fair access of a granted TXOP to different streams at a station by combining them into a single FIFO queue, thereby allowing for minimal MAC processing when a poll is received, is described. They also introduce a novel scheduler that when used in conjunction with the HCCA mechanism optimizes the delivery of QoS sensitive traffic. The paper has also been demonstrated that the described scheduler algorithm can simultaneously meet the QoS specifications of different types of traffic streams. At the same time the delay performance metrics were found to be insensitive to background traffic. Based on your simulation results, they conclude that the scheduled HCCA mechanism is suitable for both voice and high quality AV delivery over WLANs.

All these solutions presented are designed to optimize the Wireless Network scheme, but not avoid the low performance to transmission for small size data packets in the presence of deadlines with high reliability and do not save bandwidth when there are not data packets to be transmitted by real-time stations in the industrial environment. For this situation we propose the Group Sequential Communication (GSC) for dealing with this cases.

III. Publish-Subscribe in the Industry

Nowadays, there are typically two types of paradigms associated to industrial communications: master-slave and publish-subscribe. The first paradigm uses a unicast mechanism, where the master always starts the communication to a specific slave and this slave just answers to the master question. The second one, uses a multicast mechanism, where the publisher component always starts the communication publishing their messages upon the network. In such a case, messages are not addressed to any specific host. Instead, each message has an identifier and one or more subscribers components that consume it. For example, two of the most popular fieldbus networks, Profibus and Foundation Fieldbus, use master-slave and publisher-subscriber approaches, respectively [2].

The master-slave paradigm is the simplest communication mechanism, where the sender just needs to know the address of the receiver to send its messages. However, this approach is completely centralized and it does not offers multicast/broadcast data exchanges. Due to this aspect, master-slave mechanisms have scalability problems. Conversely, a publisher-subscriber approach is more decentralized and flexible than master-slave. Beyond that, its communication mechanisms are intrinsically multicast, which saves network bandwidth. Thus, the publish-subscribe is a more adequate solution to distributed real-time industrial applications [2], where small packets must be periodically transferred between sensors, controllers and actuators according to strict transfer deadlines.

IV. Group Sequential Communication (GSC) Proposal

The GSC scheme works similarly to the HCCA mechanism, where the Hybrid Coordinator (HC) captures the control of the wireless medium by waiting a PIFS interval between station transmissions using the EDCA or DCF procedures. As in the HCCA mechanism, it also includes a Contention Free Period (CFP) parameter in the Beacon frame to set all NAVs to the end of the controlled phase. Moreover, the GSC uses a real-time group concept, where the members of this group have the highest priority to the medium access.

The GSC scheme operates during the CFP period. It does not use a polling scheme to grant TXOP to each of the real-time stations previously registered in the group. Instead, it sequentially grants the medium access rights to each of the real-time group stations. Once a real-time group is triggered, a Virtual Token Pass (VTP) procedure is started among the real-time group members (hereafter also referred as GSC stations). This procedure serializes the message transfers from the real-time group members, without the need to exchange any extra polling messages between the HC and the real-time group stations.

The VTP procedure uses a set of local counters that are incremented at specific slot boundaries. Such local counters implement a distributed variable that defines the station that has granted the medium access. Whenever a GSC station does not have a frame to be transmitted, it transfers the medium access rights to the next station in the group, without wasting bandwidth with extra message exchanges. The definition of Transmission Opportunities (TXOP) must allow the transmission of just one message per GSC station in each Service Interval (SI), and in the end of real-time transmission process it is verified if all messages were successfully transmitted using a Real-Time Block Acknowledgment packet.

A. Scheme Description

The GSC procedure considers a Sequence Group (SG) with np members where circulates a virtual token. The membership is represented as $L = \{GI_1, GI_2, ..., GI_{np}\}$, where GI_i is used as station identification (ID), np is the maximum number of admitted stations in real-time group and alpha (α) is a percentage of CFP.

In the beginning of the CFP repetition interval, called Service Interval (SI), the HC sends a Beacon frame (based on the information of the TSPEC management frames) to set all NAVs to the end of the controlled phase and to set the value of the distributed variable local Sequence Counter (SC) to 1. Specifically, all members of SG group maintain a local SC that is an image of the distributed variable SC. Whenever a GSC station has a packet ready to be transferred and its SC image equals its ID ($SC = GI_i$) during the reserved TXOP interval, it means that the GSC station has authorization to transfer its real-time messages immediately after the medium being idle during at least a SIFS interval ($aSIFSTime$).

Conversely, if the GSC station authorized to transmit and does not have any message to transmit, the SC value will be incremented after $aSlotTime$ (ST) in all the GSC stations. It allows the next GSC station to transfer its messages successively until the end of all GSC stations transmissions.

The publish-subscribe paradigm is implemented by using broadcast frames to transmit GSC messages. The GSC scheme answers to all real-time stations with just one single packet called $RTBlockAck$ containing the information of which frames have been received correctly. When a message is transmitted all stations receive it, and those which are interested consume it. This results in a strong reduction in the polling overhead. However, in a noisy channel it is possible that messages become corrupted. Since the GSC uses a non-confirmed approach this implies that a message loss becomes undetectable. Since the next transmission of this message will occur only in the next GSC round, this is equivalent to delaying its transmission. This is an undesirable situation since it could lead to deadline missings which degrades system's performance. This would lead to a message delay until the next GSC round. However, such delay in undesirable in industrial communications. So, it is necessary a mechanism that guarantees that a message can be retransmitted within a minimum delay.

The GSC implements this mechanism by means of a new transmission round named *Second Chance*. At the end of all GSC transmissions, the HC (which had listened all messages transmitted during the CFP), sends a $RTBlockAck$ packet. This packet is based in the Block Acknowledgment packet defined in the IEEE 802.11e standard, and it is transmitted using a broadcast frame. When a GSC station receives this packet, it analyzes its content and finds if their transmission was correct or not. Afterwards, a new GSC round is initiated. However, in this case only the stations that received a negative confirmation will retransmit their messages.

The second chance algorithm works therefore as fault-tolerant mechanism that minimizes the chances of missing or delaying real-time messages. After the second chance round, the HC sends a CF-End frame (message to finish a remaining CFP) resetting all NAVs, thus allowing the initialization of the CP, where standard DCF/EDCA stations can start contending together for the medium transmission with subscribe of new GSC stations.

At the beginning of a new SI interval, another Beacon frame will be sent to set all the parameter values of the GSC group. The GSC scheme includes procedures in the CP to add and remove real-time stations to and from the group.

These procedures must ensure the two following features:

Synchronization: Each GSC station must agree with the values of *SC*. That is, at whatever instant of time, all stations know the address of the token holder station in the *SG* group.

Subscription: To all GSC stations must be assigned an unique Node Address (*NA*), which its range is between 1 and *np*.

Unless these two features are satisfied, mutual access to the medium by the real-time stations cannot be ensured. Besides these properties, an assumption specifying that the logical SG group has already been initialized with np stations. The algorithms 1 and 2 show the details of how the GSC scheme works.

Algorithm 1. HC - GSC Main Algorithm

```
Input: - SC
       - SI
       - Alpha
       - NP
       - GIi

cfp = SI * alpha;
cp = SI - cfp;

function main() {
  start wait_timer;
  while (wait_timer <= PIFS) {
    if (medium idle)
      restart wait_timer;
  }
  stop wait_timer;

  send beacon frame; // CFP Start and Set NAV
  create RTBlockAck[NP] = gsc(NP,cfp);

  counter = 0;
  for i=1 to np {
    if RTBlockAck[i] = 0
      counter = counter + 1;
  }
  Wait SIFS;
  broadcast_blockack(RTBlockAck);
  gsc(counter,counter*(transmission time + SIFS));

  Wait SIFS;
  CF_End; // CFP End and Reset NAV

  start wait_timer; // CP Start
  while (wait_timer <= cp - PIFS) {
    listen medium; // join/leave GSC and EDCA/DCF
        Stations;
  }
  stop wait_timer; // CP End
}
function gsc(nstas,duration) {
  create RTBlockAck[nstas];
  sc = 1;
  start cfp_timer;
  while (cfp_timer <= duration) {
    if (new event receveid) {
      if (successfull) {
        RTBlockAck[sc] = 1;
        sc = sc + 1;
      } else {
        RTBlockAck[sc] = 0;
        sc = sc + 1;
      }
    } else {
      Wait aSlotTime;
      RTBlockAck[sc] = 1;
      sc = sc + 1;
    }
  }
  return (RTBlockAck);
}
```

Algorithm 2. GSC Station Algorithm

```
Input: − Beacon
       − RTBlockAck
       − CF_End
       − np

GIi = join network()

while (true) {
  Wait Beacon frame;
  success = false;
  TempBlockAck = cfploop(GIi,np,false);

  function cfploop(idx,npx,second_chance) {
    sc = 0;
    while (true) {
      Wait SIFS;
      sc = sc + 1;
      if (idx = sc) {
        Transmit RT Data Packet;
      } else if (sc = npx + 1) {
        if (second_chance = false) {
          Receive RTBlockAck;
          if RTBlockAck[idx] = 1 {
            success = true;
          }
        }
        Break;
      } else if (medium busy) {
        Wait RT Transmission End;
      } else{
        Wait aSlotTime;
      }
    }
    return (RTBlockAck);
  }

  if (success = true) {
    Wait CF_End;
  } else {
    counter = 0;
    for i=1 to length(TempRTBlockAck) {
      if TempRTBlockAck[i] = 0 {
        counter = counter + 1;
        if GIi = i {
          new_GIi = counter;
        }
      }
    }
    cfploop(new_GIi,counter,true);
    Wait CF_End;
  }

  join or leave medium; // CP
  Wait PIFS;
}
```

Figure 2 shows an example of the GSC scheme. The HC transmits a Beacon frame that starts the CFP. The GI_1 station will start a real-time message transmission ($SC = 1$) just after the medium being idle during a SIFS interval. Considering that all stations received the Beacon frame and station GI_1 started its transmission successfully, it will be allowed to control the medium for a TXOP interval at most. At the end of TXOP, after a $aSIFSTime$, all GSC stations increment their local SC value passing the virtual token ($SC = 2$) to the next station (GI_2)[1] that initiates its transmission and the message is corrupted by noise. However, the HC takes account for this situation, and will signalize a retransmission in the BlockAck procedure. After another $aSIFSTime$, the GI_3 station should starts its transmission. However, considering that GI_3 station does not have any message to be transferred, the SC counter is incremented after a $aSlotTime$ (ST), and GI_4 station is allowed to start its transmission ($SC = 4$). At the end of the GI_4 station's TXOP, and after a $aSIFSTime$, the HC sends a $RTBlockAck$ packet for all real-time stations. Real-time stations that did not have a successfully transmission, will have a new chance to transmit their packet (Second Chance Algorithm). After the end of the new transmission cycle plus a $aSIFSTime$, the GSC sends the CF-end frame resetting all NAVs. This allows the initialization of the CP, where standard DCF/EDCA stations can start contending for the medium and the GSC stations will be able to subscribe for a new SI.

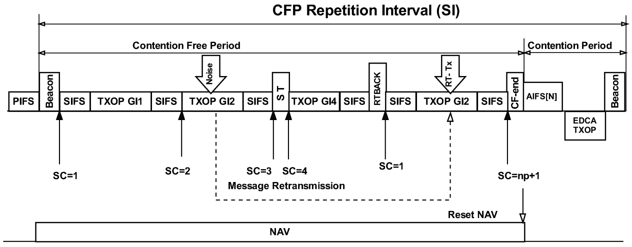

Fig. 2. GSC Repetition Interval.

V. Basic Concepts of Admission Control

In this section, the admission control of the GSC scheme is presented. Specifically, schedulability analysis equations for the publish-subscribe paradigm are deduced, in which each GSC station transmits its messages with different data payload.

The proposed GSC scheme uses the TSPEC as defined in the IEEE 802.11e HCCA standard. The TSPEC is an element sent through a management frame that contains information about the characteristics and QoS expectation of a traffic stream. As mentioned before, Similarly to HCCA, the GSC defines two contention periods. The first period is used as a CFP by GSC stations that had reserved TXOPs and the second period is used as a CP for low priority stations (DCF or EDCA stations). The maximum SI specifies the maximum time interval between two consecutive service periods ($SI = CFP + CP$). The scheduling and the admission control of a new traffic stream is made by the HC, where the scheduled SI and the TXOP duration are calculated. It is worth noting that the TXOP is calculated in order that each GSC station transmits only one real-time message. Recalling that, all stations can transmit their messages again with second chance algorithm, if necessary.

Considering that the CFP is represented by a percentage of the SI interval (α) and, during a CFP, all GSC stations usually have messages to transfer. The CFP can be defined as follows:

$$CFP \leq \alpha * SI \quad (1)$$

[1]The end of a TXOP period can be detected by all stations whenever the medium is idle for a $aSIFSTime$ (ST).

The maximum CFP is equal to the product of the sum all TXOPs assigned to the np members of the group SG separated by SIFS intervals. Thus,

$$CFP = \sum_{i=1}^{np} (TXOP(i) + SIFS) + \beta \tag{2}$$

where β is the maximum time to transfer the Beacon, the Real-Time Block Acknowledgment and the CF-End frames.

Therefore, considering np admitted traffic streams, a new stream $(np + 1)$ can be admitted if it satisfies the following inequality:

$$\sum_{i=1}^{np+1} (TXOP(i) + SIFS) + \beta \leq SI - CP \tag{3}$$

From equation 1, it is also possible to obtain the following relation:

$$\alpha \geq \frac{\sum_{i=1}^{np} (TXOP(i) + SIFS) + \beta}{SI} \tag{4}$$

When considering real-time scenarios, it is of strong interest to assess a worst-case timing analysis, which shall demonstrate that the Service Interval is upper-bounded. Similarly to the HCCA mechanism, the GSC proposal also allows that standard DCF or EDCA stations contend for the medium access during the CP interval. An important assumption that must be considered to determine the worst-case scenario is that the wireless communication medium is essentially an *open communication environment* [15]. That is, any new participant can try to access the communication medium at any time (according to the MAC rules) and establish its own communication channels.

The worst-case scenario will occur when a set of stations do not receive the Beacon frame[2] and, at an instant of time immediately before the end of the CP interval, a default station can access the medium and use all the allowed TXOP time, causing a delay in the beginning of the next SI. This case is show in the Figure 3 .

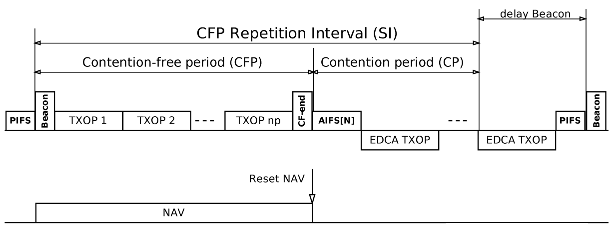

Fig. 3. Beacon Delay for GSC Mechanism.

VI. Conclusion

This paper presents a new scheme proposal compliance with HCCA called Group Sequential Communication (GSC). This mechanism does not use the HCCA polling scheme, but it uses a Virtual Token Passing (VTP) procedure. The main advantage of this approach is the low overhead, because it is not necessary to transmit the polling, null messages and individual ACK packets. Moreover, this technique guarantees that the transmission of real-time stations is prioritized over the generic stations original scheme. The GSC scheduler synchronizes the real-time stations by using an admission parameter (α). To improve the confiability of GSC in a noisy channel, a fault-tolerant mechanism based on block acknowledgement strategy is presented. Then, we believe that the GSC is a viable solution to deal with the non-determinism of the medium access in real-time wireless environments shared with generic stations in industrial scenarios. In the future work, we intend to implement the GSC scheme in a network simulation software and the results will be compared with the HCCA function, showing the efficiency of the proposed scheme to deal with the industrial scenarios.

[2]In such a case, this group of stations do not set its NAVs to the end of the controlled phase.

Acknowledgment

The authors would like to give a special thanks to CAPES (Brazil) and GRICES (Portugal) agencies for their support.

References

[1] A. Willig, "Recent and Emerging Topics in Wireless Industrial Communications: A Selection," *Industrial Informatics, IEEE Transactions on*, vol. 4, no. 2, pp. 102–124, May 2008.

[2] O. Dolejs, P. Smolik, and Z. Hanzalek, "On the Ethernet use for real-time publish-subscribe based applications," *Factory Communication Systems, 2004. Proceedings. 2004 IEEE International Workshop on*, pp. 39–44, Sept. 2004.

[3] I. 802.11, *IEEE Standard for Information Technology - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 2003.

[4] I. 802.11a, *IEEE Standard for Telecommunications and Information Exchange Between Systems LAN/MAN: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications.*, 1999.

[5] I. 802.11b, *IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems LAN/MAN: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.*, 1999.

[6] I. 802.11g, *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.*, 1999.

[7] I. 802.11e, *IEEE Standard for Information Technology - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, 2005.

[8] L. W. Lim, P. Y. Tan, C. Apichaichalermwongse, K. Ando, and Y. Harada, "A Qos Scheduler for IEEE 802.11e WLANs," *Consumer Communications and Networking Conference, CCNC 2004*, pp. 199–204, 2004.

[9] J. Son, I. Lee, and H. Yoo, "An Efficient Polling Scheme for IEEE 802.11e," *IEICE Transactions on Communications, no. 12*, vol. E88-B, pp. 4690–4693, 2005.

[10] S. Mangold, S. Choi, G. Hiertz, O. Klein, and B. Walke, "Analysis of IEEE 802.11e for QoS support in wireless LANs," *IEEE Wireless Communications*, vol. 10, no. 6, pp. 40–50, Dec. 2003.

[11] D. Gao and J. Cai, "Admission Control in IEEE 802.11e Wireless LANS," *IEEE Network, 2005.*, 2005.

[12] Y. Xiao, "IEEE 802.11e: QoS provisioning at the MAC layer," *IEEE Wireless Communications*, vol. 11, no. 3, pp. 72–79, June 2004.

[13] C. Casetti, C.-F. Chiasserini, M. Fiore, and M. Garetto, "Notes on the Inefficiency on 802.11e HCCA," 2005.

[14] S.-C. Lo, G. Lee, and W.-T. Chen, "An efficient multipolling mechanism for IEEE 802.11 wireless LANs," *Computers, IEEE Transactions on*, vol. 52, no. 6, pp. 764–778, June 2003.

[15] R. Moraes, F. Vasques, P. Portugal, and J. Fonseca, "VTP-CSMA: A Virtual Token Passing Approach for Real-Time Communication in IEEE 802.11 Wireless Networks," *IEEE Transactions on Industrial Informatics*, vol. 3, pp. 215–224, 2007.