Efficient QoS Provisioning at the MAC Layer in Heterogeneous Wireless Sensor Networks

M.Souil[a,*], A.Bouabdallah[a], A.E.Kamal[b]

[a] *UMR CNRS 7253 HeuDiaSyC, Université de Technologie de Compiègne, Compiègne Cedex F-60205, France*
[b] *Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa 50011, USA.*

Abstract

New emerging applications for wireless sensor networks, such as mission-critical and multimedia applications, require sensing heterogeneous phenomena, and that the network supports different types of QoS-constrained traffic at variable rates. Designing an efficient Medium Access Control protocol allowing these applications to work properly while coping with the limited resources of sensor networks is a challenging task. In this paper, we present a new Adaptive MAC Protocol with QoS support for Heterogeneous wireless sensor networks which provides high channel utilization with a hybrid and adaptive behavior, and integrates a new efficient prioritization scheme to provide QoS support and fair data delivery of heterogeneous traffic. The protocol design is presented along with performance results obtained through extensive simulations. A mathematical model is provided and applied to perform an analytical evaluation of AMPH. Performance modeling, analysis, and simulation results show the effectiveness of our solution.

Keywords: Medium access control (MAC), quality of service (QoS), wireless sensor networks (WSN), hybrid, heterogeneous.

1. Introduction

During the past ten years, wireless sensor networks (WSN) have drawn the attention of the research community, attracted by this new concept and the sum of challenges ahead. In the early days of wireless sensor networks, they were typically composed of a large number of identical nodes equipped with scalar sensors such as temperature, humidity and light sensors, and a radio chipset allowing communications between nodes. This new instrument was envisioned for a broad range of monitoring applications (continuous sensing, event detection, etc.) in several areas such as military, environment, industry and home [1].

Recently, the availability of low-cost new hardware such as CMOS cameras and microphones, accelerometers, gyroscopes, ECG and EMG sensors, coupled with technological improvements in wireless sensor nodes in terms of processing capabilities and memory, revolutionized traditional monitoring and sensing. The release of this new generation of nodes led to the emergence of many new promising applications such as multimedia surveillance networks, car traffic monitoring and advanced health care delivery [2, 3]. This new generation of applications has specific requirements compared with classic monitoring applications. Unlike most traditional wireless sensor network applications, new applications are often high data rate real-time applications and combine different sensing modalities, hence producing several types of traffic with various characteristics like multimedia streams, critical physiological data, emergency alerts, etc. Given the limited resources of wireless sensor networks, especially bandwidth, the QoS requirements of these applications point out the need for new networks which are capable of transporting large amounts of data, but also to perform real-time processing, correlation and aggregation of data originated from heterogeneous sources, and to provide QoS support.

The term QoS may have different meanings, depending on the context in which it is used: it may refer to the degree to which the system performs the functions required by the application/user, as well as the mechanisms implemented to provide this performance. Applications have QoS requirements and the network must provide QoS support. These two perspectives are interdependent [4]. Since different applications may have different requirements, no single QoS model can support all applications. In wireless networks, the MAC layer plays a key role in QoS provisioning. Since the radio channel is shared and cannot be accessed simultaneously by several nodes, network performance depends directly on the optimal management of this resource. MAC protocols can be classified into two categories: *contention-based* and *schedule-based*. Contention-based protocols like CSMA/CA are scalable with no strict time synchronization constraint, however, their performance degrades under heavy traffic because the probability of collisions increases. In schedule-based protocols such as TDMA, the channel is divided into time slots where each node has an exclusive access right to the medium during its time slot, hence avoiding collisions, idle listening, and overhearing problems. Nevertheless, time synchronization is required, and becomes challenging as the network size increases. Also, as nodes can only transmit during their own time slots, schedule-based protocols introduce latency and bandwidth under-utilization under light traffic.

[*] Corresponding author. Tel.: +33 344234645; fax: +33 344234477.
Email addresses: `marsouil@utc.fr` (M.Souil), `bouabdal@utc.fr` (A.Bouabdallah), `kamal@iastate.edu` (A.E.Kamal)

Few MAC protocols are designed to cope with different types of traffic and varying traffic load conditions. Designing an efficient MAC protocol for such networks is challenging. This is because such protocols must implement service differentiation mechanisms, intra-nodal and inter-nodal traffic prioritization, and an adaptive behavior according to traffic conditions. This motivates us to develop a new MAC protocol for wireless sensor networks with heterogeneous sensing capabilities.

In this paper we propose a new Adaptive MAC Protocol for Heterogeneous wireless sensor networks called AMPH. AMPH provides efficient delivery of heterogeneous traffic using service differentiation and traffic prioritization, and maximizes channel utilization by virtue of its hybrid adaptive nature. We evaluate the performance of AMPH using simulation by considering two classes of traffic: high priority real-time traffic and best-effort traffic. Simulation results show that AMPH enables efficient and fair delivery of both real-time and best-effort traffic according to their respective QoS requirements. The hybrid behavior of AMPH, which combines the strengths of contention-based and schedule-based protocols, outperforms contention-based protocols in terms of reliability and channel utilization.

The rest of this paper is organized as follows. The next section briefly introduces related research. A detailed description of AMPH design is presented in Section 3. Simulation results are discussed in Section 4. Section 5 presents the analytical model and reports on numerical results. Finally, Section 6 concludes the paper.

2. Related Work

Due to the unique resource constraints and application requirements of sensor networks, standard MAC protocols developed for wireless networks could not be used, as discussed in [1]. Therefore, several MAC protocols with different objectives were proposed. Initially, the main design goal of MAC protocols for WSN was to maximize network lifetime. A good survey of popular MAC protocols for wireless sensor networks is provided in [5]. As there is a wide variety of WSN applications whose requirements may be very different, it has become evident that no single MAC protocol can fit all applications. Indeed, several application-specific characteristics such as interactivity and reliability influence the network design. Thus, the underlying network must provide guarantees in terms of latency, bandwidth, and packet loss, just to name a few. There are many application-specific MAC protocols in the literature (e.g., delay-sensitive applications, bandwidth-hungry, mission-critical, etc.) as shown by the surveys in [6, 7, 8]. However, there are a few QoS-aware MAC protocols, i.e., protocols which aim to accommodate different types of QoS-constrained traffic and to adapt to varying traffic loads [9].

Saxena et al. [10] proposed a QoS MAC protocol for wireless multimedia sensor networks (WMSN), as multimedia applications commonly carry heterogeneous traffic with different QoS requirements. This protocol is based on a CSMA/CA approach and attempts to fulfill end-to-end delay and bandwidth requirements of three types of traffic (streaming video, real-time, and best-effort) using an adaptive contention window (CW) and a dynamic duty cycle for energy conservation. Service differentiation is achieved using multiple queues and a value of CW related to the traffic priority. Traffic of utmost importance will be assigned a small contention window to have a better chance of accessing the medium. CW size and duty cycle are adjusted according to network statistics such as transmission failures and dominant traffic type. A similar idea is pursued in the work of Yigitel et al. [11] which proposed a comparable protocol named Diff-MAC. Diff-MAC uses a different approach for intra-node packet prioritization and CW size adaptation. The MAC protocol in [10] implements one FIFO queue per class of traffic before the packets are scheduled for sending, whereas Diff-MAC provides a fair prioritization of packets within the same class based on the hop count metric of each packet and uses a weighted fair queuing (WFQ) method to control the relative throughput of each traffic class. Also, Diff-MAC continuously adapts the CW size while Saxena et al.'s MAC waits for the neighboring nodes to adjust it. Therefore, CW converges more quickly to its optimal size in Diff-MAC.

These two protocols use similar mechanisms to the IEEE 802.11e standard [12], particularly with respect to medium access prioritization. The hybrid coordination function (HCF) in the standard includes a method of channel access called Enhanced Distributed Channel Access (EDCA). EDCA defines four priority classes called access categories (AC): Background, Best-Effort, Video, and Voice. The priorities are implemented using contention windows. Voice and Video have smaller contention windows than Background and Best-Effort traffic in order to maximize the chance to transmit the priority traffic before delay-tolerant traffic. Saxena et al.'s MAC and Diff-MAC provide significant improvements over classic CSMA/CA approaches: they exhibit better performance in terms of throughput and latency. Diff-MAC also achieves fairness among the different traffic classes. However, although dynamic mechanisms enable the network to accommodate time-varying traffic loads, they introduce a significant complexity. Besides, contention-based protocols may not be efficient under high contention as RTS-CTS exchanges consume extra bandwidth. This overhead causes the channel utilization to be suboptimal.

Contention-free MAC protocols like TDMA perform better under heavy traffic loads. Indeed, scheduled transmissions allow avoiding collisions. Nevertheless, under low contention, TDMA leads to low channel utilization and high latency. Therefore, pure TDMA approaches are not suitable for variable traffic environment.

The limits of contention-based and contention-free MAC protocols have led to the development of hybrid MAC protocols which attempt to combine the advantages of both approaches. Z-MAC [13] protocol proposed by Rhee et al. is based on this paradigm: it dynamically adjusts its behavior between CSMA and TDMA depending on the level of contention in the network. During the setup phase, the nodes run the following operations: neighbor discovery, slot assignment, local frame exchange, and global time synchronization. The two-hop neighbor list is used as an input to the time slot assignment algorithm called DRAND [14]. This algorithm computes a schedule where two nodes within a two-hop communication neighborhood cannot be assigned to the same slot. When the setup phase is over, the transmission phase begins. Nodes can transmit during their own time slot, but they may also contend to use a slot that is not used by its owner, hence enhancing channel utilization. Before transmitting, nodes back off for a random time within a given contention window. When the backoff time expires, they run a clear channel assessment (CCA) to know if the channel is clear. The CW size is set in such a way that owners are always given a better chance of accessing the channel. This mechanism makes Z-MAC robust to synchronization errors. In case of clock drift, the performance of Z-MAC is similar to that of CSMA. To overcome the high overhead of RTS-CTS, this mechanism is not used in Z-MAC. Instead, Z-MAC implements two modes of operation: low contention level (LCL) and high contention level (HCL). When high contention is experienced, an explicit contention notification is sent causing the nodes to switch to HCL mode where nodes are no longer allowed to steal slots owned by two-hop neighbors. Z-MAC dynamically adjusts its behavior depending on the level of contention in the network, thus achieving high channel utilization. However, Z-MAC is not suited for heterogeneous applications since it does not implement any service differentiation mechanism and QoS provisioning. I-MAC [15] adds a prioritization scheme to Z-MAC and aims to take into account the traffic load for each sensor node according to its role in the network. Higher priority will be assigned to nodes having a lot of packets to send, such as cluster heads, thus allowing these nodes to have a better chance to access the medium than their low-priority neighbors, which improves the throughput of the former nodes. Four priority levels are implemented using custom CW sizes for each priority group. Although I-MAC reduces collisions, thus achieving a slightly better channel utilization than Z-MAC, it has not been designed to support QoS-constrained traffic either. Indeed, the prioritization scheme of I-MAC is only based on the amount of traffic of each node, whereas it should have also considered the traffic type in order to provide differentiated services. In addition, since the priority levels are fixed, a node cannot dynamically adapt its priority level in case of variable traffic conditions. Finally, this protocol may be hard to deploy over a large number of nodes. Nodes are assigned a fixed priority according to their role in the network, so this implies that nodes must be manually configured, unless they are able to infer their role in the network.

3. AMPH Protocol Design

Our goal is to provide an efficient MAC protocol for heterogeneous wireless sensor networks. As more and more applications have heterogeneous sensing capabilities and require network support for different types of QoS-constrained traffic at variable rates, wireless sensor network support becomes a necessity. In this section, we present in detail the design of AMPH, our new adaptive MAC protocol for heterogeneous wireless sensor networks. The basic idea of our solution is similar to that of Z-MAC: we adopt a hybrid behavior which combines the strengths of both contention-based and schedule-based approaches to maximize the channel utilization. Our hybrid channel access method allows slot-stealing, thus achieving high channel utilization, and provides adaptability to variable traffic loads. We also introduce a new prioritization scheme which is designed to fulfill the requirements of real-time traffic. In the following subsections, we describe in detail the basic principles of AMPH along with its two main operation phases, setup and transmission.

3.1. AMPH Basic Principles

AMPH is a hybrid channel access method. It is mainly based on the time division principle, but nodes may transmit during any time slot in order to maximize channel utilization and minimize latency. Time is divided into several recurrent time slots of fixed duration. Nodes are assigned to time slots in such a way that no two nodes within a two-hop communication neighborhood are assigned to the same slot. More details about slot assignment are given in the *Setup* subsection. We call nodes assigned to a given slot *owners*. Otherwise, nodes are *non owners*. A cycle of N time slots constitutes a time frame, where N is the maximum number of time slots, i.e., equal to the maximum number of contenders within two hops.

As stated before, nodes may transmit during any time slot. We propose a new prioritization scheme which ensures that nodes with high priority traffic will be able to transmit ahead of low priority nodes in case of competition to access the channel. Our scheme also includes an intra-node arbitration mechanism so that priority packets take precedence over other packets as soon as they are created. We first explain the intra-node arbitration mechanism. Inter-nodes arbitration will be detailed subsequently.

AMPH supports two classes of traffic: real-time (RT) and best-effort (BE), and RT traffic takes precedence over BE traffic. We assume that the traffic class is statically set at the application layer. When a packet is submitted to the data link layer from the upper layer, a classifier checks whether the packet is real-time or best-effort and puts it into the appropriate packet queue. AMPH maintains two FIFO queues corresponding to the two classes of traffic, as shown in Fig. 1. We use a strict priority scheduler to set the next packet to send, so that RT traffic always has priority over BE traffic. Our scheduler systematically selects RT packets as long as the queue is not empty, then it continues with BE packets.

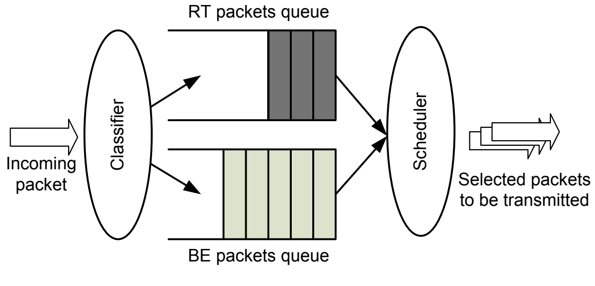


Figure 1: AMPH intra-node arbitration scheme

This scheduling mechanism allows to select RT packets for transmission ahead of delay tolerant BE packets. An additional mechanism is needed to organize channel access between competing nodes in order to guarantee that a node having RT traffic to send has higher chance to gain access to the medium than a node having BE traffic, hence ensuring that RT traffic queuing time is minimized. We propose a new arbitration scheme that provides low channel access delay for RT packets and fairness among nodes with traffic of the same class. Our arbitration mechanism uses timers called *backoffs* and operates as follows. Competing nodes pick a backoff value and wait for the backoff duration before trying to transmit. When the backoff timer of a node expires, it senses the medium by calling the CCA function of the PHY. If the PHY returns the channel status as idle, the node may start to send packets, otherwise it has to delay its transmission. As a result, the node that obtains the smallest backoff wins the contention and gains access to the medium. When the backoff of the other contenders expires, the channel will not be idle anymore, since the winner is currently transmitting, and they will back off again, using new samples of backoff durations. According to our design goals, RT traffic takes precedence over BE traffic, so nodes having RT packets to send should be able to access the channel ahead of nodes having BE traffic. In order to allow this behavior, nodes having RT traffic benefit from smaller backoffs than nodes with BE traffic which use longer backoffs. The contention window also depends on the role of the node: owner or non owner. Owners have priority over non owners. Since all nodes own a time slot, this system achieves a fair access to the channel among nodes having traffic of the same class. In addition, our mechanism allows non owners to steal the slots of owners when they have nothing to send, thus reducing channel access time and increasing channel utilization. Nodes having data to send pick the backoff value β in the appropriate contention window, according to the type of traffic selected by the scheduler and if they are owner or non owner. The contention windows form a non-overlapping interval set, as depicted in Table 1. Since the backoff is chosen randomly, the probability that contenders with similar conditions (non-owners having traffic of the same class) choose low backoff durations, and the collision probability will be low.

Owner + RT traffic	Interval A	$\beta \in [A_{min}, A_{max}[$
Non owner + RT traffic	Interval B	$\beta \in [B_{min}, B_{max}[$
Owner + BE traffic	Interval C	$\beta \in [C_{min}, C_{max}[$
Non owner + BE traffic	Interval D	$\beta \in [D_{min}, D_{max}[$
where $A < B < C < D$.		

Table 1: Contention windows corresponding to the role of the contender and the type of traffic it has to send

In Fig. 2, we depict an example scenario of two competing nodes u and v, where u and v both have RT traffic to send to the base station at the beginning of slot 0. Node u picks a backoff β_u in the interval A since it is the owner of the slot, and v picks its backoff β_v in the interval B. Since $\beta_u < \beta_v$, the backoff of node u expires first, so it runs a clear channel assessment (CCA) to determine if the channel is clear, i.e., that no nodes are currently transmitting. Node u finds the channel is idle, so it starts its transmission. When the backoff of node v expires, v also runs a CCA but as the channel is not idle anymore (node u is currently transmitting), it cannot transmit and has to wait for the beginning of the next slot (slot 1) to retry. As node v is the owner of slot 1, it will benefit from a small backoff. Therefore, it will be given the highest priority to access the channel. This example also illustrates how our backoff system ensures that AMPH is fair, i.e., that the medium is fairly shared among all nodes of the network. We can see that our arbitration mechanism guarantees that all nodes gain access to the channel at some point, in the worst case scenario, during their reserved time slot. Besides, due to the random nature of our scheme, all nodes of the same priority level have equal chance of stealing unused slots. The whole transmission process is described in Section 3.3.

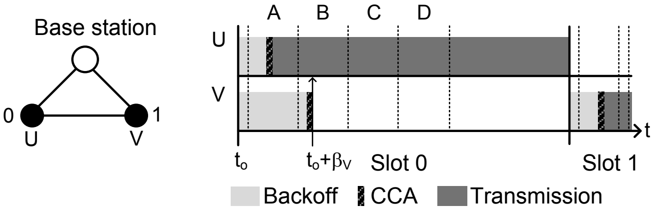


Figure 2: AMPH inter-node arbitration scheme

AMPH ensures that a maximum number of packets can be sent during a time slot in order to maximize the channel utilization. Indeed, transmitting a burst of packet is more efficient than transmitting only one packet per slot. It is not necessary to run the full transmission process for each packet and the overhead caused by the backoff mechanism is absorbed. The number of packets that can be sent into one burst depends on the packet size.

In our solution, we use a strict priority scheduler and a backoff mechanism. Both mechanisms always favor RT traffic. As a consequence, BE traffic may suffer from starvation. In order to avoid this situation, we arrange M frames among N in which BE traffic has priority over RT traffic, where N is the number of time slots in a frame and M is a parameter to adjust according to the amount of each type of traffic . During these particular time frames, the backoff values of BE traffic are smaller than those for RT traffic, so nodes having BE traffic have priority over node having RT traffic. This mechanism is optional and may be implemented only in networks with high data rate continuous RT traffic sources.

3.2. Setup

At startup, nodes enter a setup phase and they perform the following initialization actions: neighbor discovery, slot assignment, framing, and synchronization. Each node constructs its two-hop neighbors list which is used as an input for the slot assignment algorithm. The slot assignment problem is analogous to the graph coloring problem. In AMPH, slot assignment is performed using DRAND, an efficient distributed slot reuse scheduling algorithm also used in Z-MAC. DRAND ensures that no two nodes within a two-hop communication neighborhood are assigned to the same slot. For more details on DRAND operation, the reader may refer to [14]. The maximum number of slots defines the time frame length, and nodes synchronize their schedule at the beginning of the frame. When the setup phase is done, nodes begin their normal operation described in Section 3.3.

3.3. Transmission

As explained above, our protocol operates according to a specific time structure. The time is divided into recurrent time slots forming frames. The MAC routine occurs at the beginning of each time slot. Depending on whether the node has data to transmit or not, or whether it receives traffic from neighboring nodes, the node performs various operations. In the following, we explain the actions performed by a node during one time slot, especially during the transmission process. There are basically four possible scenarios:

- The node wants to transmit and the channel is idle,
- The node wants to transmit but the channel is not idle,
- The node receives data,
- The node has nothing to transmit and does not receive data.

We describe the operations of a node in these different scenarios by following the state transition diagram of AMPH given in Fig. 3.

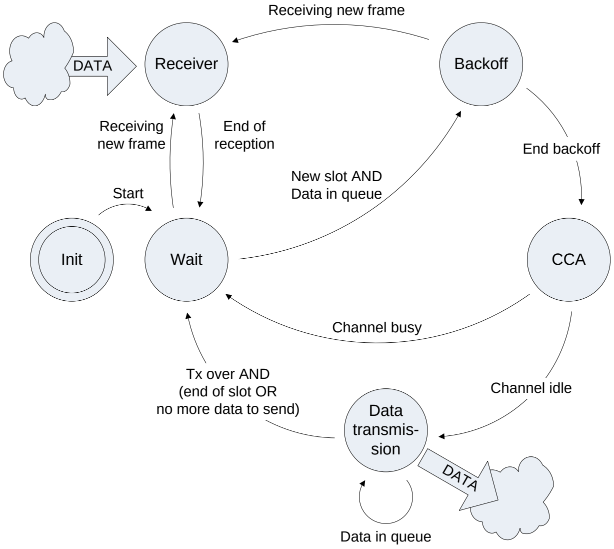


Figure 3: State machine of AMPH

Init. During the setup phase, the node is in the *Init* state. After the execution of the setup process, the node switches from the *Init* state to the *Wait* state.

Wait. The node ends up in *Wait* at the end of each time slot and stays in this state when it has nothing to do at the beginning of a new slot. The radio may be switched off if the following conditions are met: *the node has no data to send, and the node is not supposed to receive any data* (in a star topology for example, where every node can reach the base station directly).

Backoff. At the beginning of each time slot, if the node has packets to send, it enters the *Backoff* state and computes a backoff value β randomly within the corresponding window, as explained in Section 3.1. While waiting for the end of the backoff time, the node stays in the *Backoff* state. During backoff, the node listens to the radio channel in the event that it receives data. If so, it switches to the *Receiver* state.

CCA. When the backoff expires, the node switches to the *CCA* state and performs a clear channel assessment (CCA) to sense the channel. If the channel is idle, the node is allowed to begin the transmission and goes into the *Data transmission* state; otherwise it returns to the *Wait* state and waits for the beginning of the next slot to retry using the same process. As nodes listen to the radio channel during the backoff period, CCA is not necessary in this case. However, in a star topology were all nodes only communicate with the base station, the radio could then be turned off to save energy, hence CCA would be useful.

Data transmission. Once a node reaches the state *Data transmission*, it is allowed to transmit. The node sends packets until either its queues are empty, or the time slot has expired. When the transmission is over, the node returns to the *Wait* state and awaits the beginning of the next slot. A similar transition to the *Wait* state happens when the node is in the *Receiver* state and reception is completed.

Receiver. In multihop networks, nodes may act as relay nodes and receive data from other nodes that need to be forwarded. Nodes have to listen for transmissions intended for them during the *Wait* and the *Backoff* states. Reception has priority over transmission. As soon as a packet reception begins, the node switches to the *Receiver* state. The node leaves this state when the reception is over and returns to the *Wait* state. No other event can interrupt the reception.

4. Simulation experiments and results

In this section, we study the efficiency of AMPH through simulation experiments. We describe our approach to perform this evaluation, then we analyze the relative performance of AMPH with Diff-MAC, which is the best competitor in the literature. We evaluate the channel utilization, the latency, and the reliability achieved by both protocols. Finally, we discuss the results and the ability of AMPH to support the high requirements of heterogeneous WSN applications.

4.1. Scenario and Simulation Parameters

The goal of our solution is to provide high channel utilization, efficient prioritization of real-time traffic, and fair data delivery in heterogeneous WSNs. In order to assess the performance of our protocol, we carried out extensive simulations for two different classes of traffic and we compared the results with those of Diff-MAC. We selected Diff-MAC as a basis for comparison since it is a well-known QoS-aware MAC protocol, and it is the closest protocol in the literature to our protocol. Like AMPH, Diff-MAC aims to meet the QoS requirements of heterogeneous traffic by providing differentiated services and fast delivery of the priority data. By using effective QoS mechanisms, it achieves high performance in terms of throughput and latency [11]. According to our study of the related work, Diff-MAC is currently the most efficient MAC protocol for heterogeneous wireless sensor networks. Our objective is to show the benefits of our hybrid channel access technique over a contention-based approach, as used in Diff-MAC, and to demonstrate the efficiency of our prioritization scheme. In order to evaluate the performance of AMPH, we examine the following metrics: *throughput*, *latency*, and *reliability*.

We used the MiXiM framework developed under the OMNeT++ network simulator [16] to simulate AMPH and compare it with Diff-MAC. Since our protocol is designed for heterogeneous WSNs with variable traffic load, we set up an example scenario similar to a multimedia monitoring application. We consider a wireless multimedia sensor network composed of nodes equipped with a video camera producing a continuous multimedia stream, and also with environment sensors which gather information such as temperature and luminous intensity. The application requires that the multimedia content is delivered in real-time, whereas light and temperature data are considered of secondary importance. In order to simulate this application scenario in OMNeT++, we implemented a custom application layer which generates two types of packets at different rates, corresponding to scalar data and multimedia content.

- *Simulation of scalar data:* to simulate the temperature and light measurements, our application layer generates small data packets (200 bits) whose packet inter-arrival times follow a Poisson distribution.

- *Simulation of multimedia content:* we assume that video cameras produce periodic video frames of 10,000 bits which are fragmented into 1,000 bit-long packets. In order to reproduce this traffic, our application layer periodically generates 10 packets of 1,000 bits each.

Diff-MAC implements three classes of traffic: BE, RT, and non real-time (NRT), which is an intermediary class of traffic for scalar data with higher QoS requirements than BE. As a consequence, our application layer tags one scalar data packet out of every two as an NRT packet. Since AMPH does not support this class of traffic, NRT packets are processed as BE packets.

Data generation rates are input parameters which are varied to evaluate the performance of AMPH and Diff-MAC under various traffic loads. The different traffic loads offered to the network are presented in Tables 2 and 3.

4.2. MAC Parameters

In the simulations, we set the duration of a time slot such that the owner of a time slot can send a complete video frame in one slot. Given that the size of one video frame is 10,000 bits, and assuming that the available bandwidth is 256,000 bps, the duration of a time slot must be at least 39.0625 ms. We set it to 40.96 ms to correspond to 128 time units of 0.32 ms, which is the duration of *aUnitBackoffPeriod*, the basic time period used in the IEEE 802.15.4 MAC. The size of the backoff intervals *A*, *B*, *C*, and *D*, expressed in time units, are provided in Table 4. Intervals *A* and *C* are only one time unit long, since there is no contention during these periods, unless the nodes are not synchronized. Additional parameters are shown in Table 5.

Mean inter-arrival time	Average packet rate
0.1 s	10 packets/s
0.05 s	20 packets/s
0.02 s	50 packets/s
0.01 s	100 packets/s

Table 2: NRT/BE traffic loads

Frame rate
0.1 frames/s
0.05 frames/s
0.02 frames/s
0.01 frames/s

Table 3: RT traffic loads

Interval	Duration (time units)
A	1
B	8
C	1
D	8

Table 4: Backoff intervals

Parameter	Value
RT packets buffer size	50 Kbits
NRT/BE packets buffer size	4 Kbits
Available bandwidth	256 000 bps
CCA duration	0.128 ms

Table 5: Additional simulation parameters

We implemented Diff-MAC according to the information provided in [9]. Since Diff-MAC adopts a CSMA/CA based medium access method, we adapted the implementation of CSMA/CA provided in MiXiM by adding the extra features of Diff-MAC: contention window size adaptation, and intra-node and intra-queue prioritization. Diff-MAC uses RTS/CTS and acknowledgments. Just as AMPH, Diff-MAC sends RT packets in a burst. The length of a burst corresponds to the number of fragments of one video frame.

4.3. Simulation results

We evaluated the performance of AMPH through extensive simulations using the OMNeT++ simulation engine and compared it to Diff-MAC. We simulated a network of eight multimedia nodes and a base station organized in a star topology where each node is within communication range of each other and we studied the relative performance of AMPH and Diff-MAC under various traffic loads. Each scenario is simulated

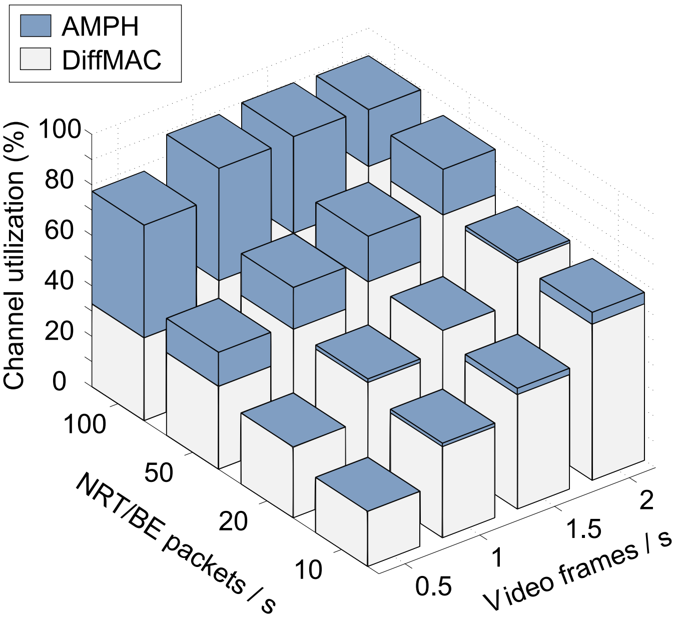


Figure 4: Comparative channel utilization

ten times with different seeds and the average was computed. In this section, we analyze the simulation results. We focus on the comparative channel utilization, average latency, and successful packet delivery ratio. The channel utilization is calculated as the throughput to channel capacity ratio. The definition of the latency is the time elapsed between the reception of a packet by the MAC layer and the transmission of this packet. The successful packet delivery ratio is calculated as the fraction of packets which were correctly received by the base station.

Since high throughput is necessary for high data rate applications such as multimedia applications, achieving high channel utilization is one of the primary goals of AMPH. In Fig. 4, we plotted the comparative channel utilization of AMPH and Diff-MAC. As shown in this figure, AMPH achieves better throughput than Diff-MAC in all scenarios, particularly when the traffic load increases. This confirms our hypothesis that the hybrid behavior of AMPH allows high channel utilization under variable traffic loads through the use of an efficient time division schedule which enhances the contention resolution. The ability to send multiple packets in one slot also contributes to maximizing the channel utilization, as well as the fact that we do not use control messages such as RTS / CTS or ACK.

AMPH also aims to provide fast data delivery for real-time and mission-critical applications. In Fig. 5, we show the average latency of RT traffic using Diff-MAC and AMPH. At low traffic loads, the latency is very small: ≈ 33 ms for Diff-MAC, and 45 ms for AMPH. Indeed, at low contention levels, nodes in Diff-MAC can access the medium almost immediately whereas in AMPH, the transmission process starts only at the beginning of a new slot. Nevertheless, the gap is not significant. When the traffic load increases, contention gradually increases and access to the channel becomes more difficult. Using AMPH, the latency of RT packets stays very low (≤ 70 ms), thus demonstrating the efficiency of our arbitration and QoS mechanisms. At the same time, the latency of Diff-MAC rises up to 330 ms.

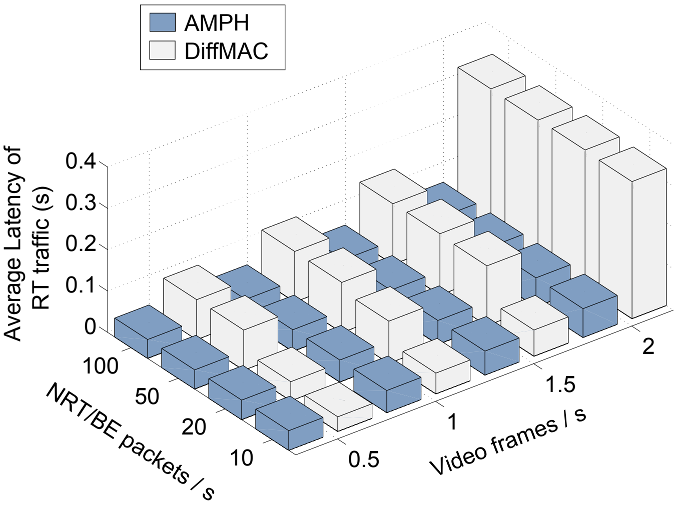


Figure 5: Comparative average latency of RT traffic

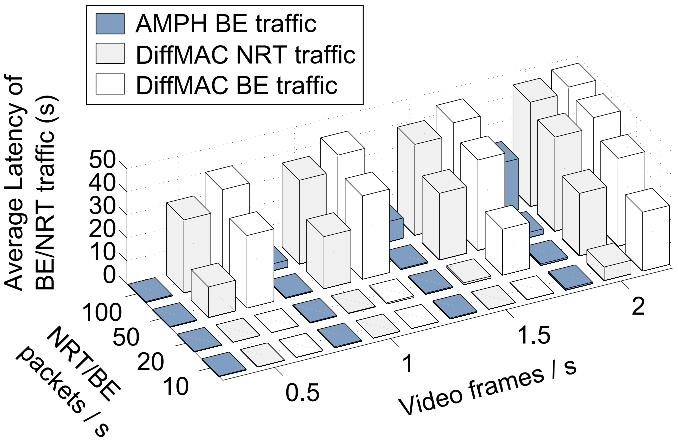


Figure 6: Comparative average latency of BE/NRT traffic

In Fig. 6, we plotted the average latency of BE traffic for Diff-MAC and AMPH. Diff-MAC supports two kinds of best-effort traffic: non real-time, NRT, and true best-effort, BE. NRT has higher priority than BE traffic. AMPH assimilates NRT to BE traffic. In almost all scenarios, the latency of BE packets using our protocol is less than one second. We notice that when the BE load is set to 100 packets/s, the latency of BE packets increases up to 22 s. However, it should be noted that the mechanism that favors BE traffic over RT traffic when the BE queue fills up was not implemented. This scenario shows that even under very high traffic conditions and with no special mechanism to favor BE traffic over RT traffic, BE traffic does not suffer from starvation. Globally, we notice that AMPH behaves very well, unlike Diff-MAC whose latency rises as soon as the traffic load reaches 50% of the available bandwidth.

Figs. 7 and 8 plot the successful packet delivery ratio performed by AMPH and Diff-MAC for all types of traffic. Reliable data delivery is an important requirement, especially for critical and real-time applications, where packet loss decreases the information quality. However, for some high-throughput

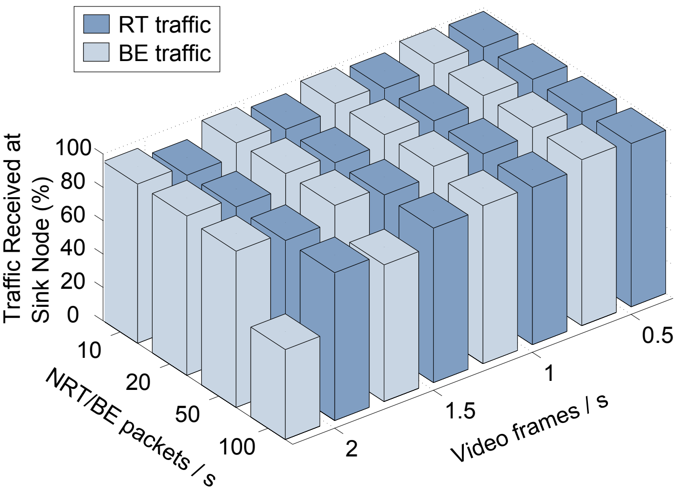


Figure 7: Comparative successful packet delivery ratio of AMPH

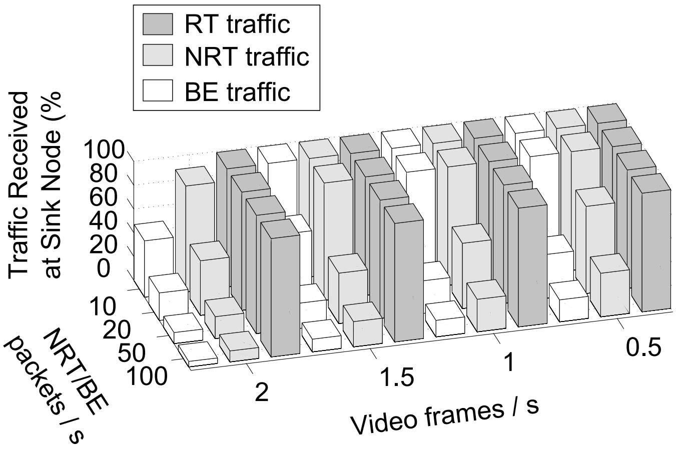


Figure 8: Comparative successful packet delivery ratio of Diff-MAC

traffic such as multimedia streaming, some packet loss can be tolerated up to a certain extent without affecting the playback quality. Additionally, coding techniques can be used to mitigate the effect of packet loss. Our simulation results show that AMPH achieves high reliability, although it does not implement RTS/CTS exchanges or packet loss recovery techniques. For real-time traffic, in the worst case scenario the reliability is 89%, and the average reliability is approximately equal to 94%, thus demonstrating that AMPH is very reliable for this class of traffic. AMPH is not only reliable for RT traffic but also for the BE traffic, since simulation results show that the average reliability of BE traffic is approximately equal to 94%. However, we notice that when the RT frame rate is equal to 2 frames/s and the BE traffic load is also set to the maximum load level, the reliability drops to approximately 50%. In this scenario, the traffic load causes nodes to encounter buffer overflows. Regarding Diff-MAC, the offered reliability for RT traffic is almost equal to 100%. Diff-MAC outperforms AMPH, but at the cost of poor throughput. As for NRT and BE traffic, packet loss increases as the traffic load grows. The two reasons for that are that packets are dropped when either they have reached the maximum number of transmission attempts, or when buffer overflows. According to the preferential treatment of NRT traffic over BE traffic, AMPH suffers lower losses. Globally, we can say that AMPH outperforms Diff-MAC under NRT/BE traffic, since for about half of the experiment, the reliability of Diff-MAC is lower than 50%.

4.4. Conclusions

In this section, we performed extensive simulations in order to demonstrate the performance of AMPH and compare the results with our closest competitor in the literature named Diff-MAC. The results have shown that AMPH outperforms Diff-MAC in terms of channel utilization and latency for both classes of traffic RT and BE. As for reliability, Diff-MAC offers almost a 100% reliable RT packet transmission, but at the cost of poor throughput, whereas AMPH experiences limited packet loss (≤ 10%) while not wasting bandwidth with control messages. We had previously demonstrated in [17] that AMPH outperforms CSMA/CA. These new experiments also tend to confirm the superiority of our hybrid behavior over contention-based solutions. In conclusion, our protocol effective fair service differentiation and QoS mechanisms minimize real-time traffic latency and prevent best-effort traffic starvation. The time division schedule enhances the contention resolution leading to high channel utilization and reliability. Hence, AMPH provides efficient QoS provisioning for heterogeneous traffic for a new generation of promising applications with high QoS requirements such as multimedia, tracking, and health care applications.

5. Modeling and Performance Analysis

In this section, we provide an analytical model of our MAC protocol AMPH. The mathematical model allows the evaluation of the MAC latency by estimating the probability that a node begins a transmission within a given time and also estimates the data delivery reliability by deriving the probability of success of a transmission attempt. In addition, our model shows how the network size and the distribution of traffic (proportion of RT and BE traffic) affect the performance of AMPH. We first introduce our approach for developing the model along with some definitions and design assumptions, then we explain in detail the formulation of our mathematical model, and finally we provide the analytical performance study of AMPH.

5.1. Model Assumptions, Reference Scenario and Notations

The design of our model follows a similar approach to that of Buratti et al. [18], where the authors provide an analytical model for evaluating the performance of the non-beacon enabled mode of the IEEE 802.15.4 standard [19]. The model provided by Buratti et al. allows the evaluation of the probability that a given sender node succeeds in accessing the channel, and that the sink receives the transmitted packet. Similarly, the goal of our model is to estimate the channel access time and the data delivery ratio of AMPH in order to perform an analytical evaluation of its performance in terms of latency and reliability.

Furthermore, we aim to analyze the impact of the network size and the traffic distribution. In what follows, we present some assumptions made in the model design along with the notations used in the formulation of the model, then we provide a short reminder on AMPH operation.

Topology. We consider N nodes organized in a star topology and a sink which does not transmit data which is located at the center of the star. We assume that all nodes are within radio range of each other, and therefore the hidden terminal problem does not occur. Nevertheless, collisions may occur if two or more nodes sense the channel at the same time, find the channel idle and start their transmissions simultaneously.

Traffic. Our model is designed to allow the performance evaluation of the two types of traffic supported by AMPH: real-time (RT) and best-effort (BE).

Packet size. Although AMPH may transmit several packets during one time slot, we only take into account the transmission of one packet, since it is sufficient to provide the MAC latency. As a consequence, the packet size does not affect the results.

Resolution time. In the definition of our model, the time is discrete and the resolution time is equal to *aUnitBackoffPeriod*, the base time unit in the IEEE 802.15.4. We call *aUnitBackoffPeriod* a time unit, and one time unit is equal to 0.32 ms.

The notations and symbols used in the definition of our model are summarized in Table 6.

Symbol	Meaning / Definition
N	Network size
$P\{T_i^j\}$	Probability to begin a transmission at time unit j of slot i
$P\{S_i^j\}$	Probability to be sensing at time unit j of slot i
p_s	Probability of success of a transmission
$p_{b_i^j}$	Probability to find the channel busy at time unit j of slot i
$p_{f_i^j}$	Probability to find the channel free at time unit j of slot $i = \bar{p}_{b_i^j}$
$p_{u_i^j}$	Probability that the transmission started in (i, j) is unique
V_{S_i}	Vector containing the probability of being in each sensing state in slot i
b_v	Backoff value computed for a given node at the beginning of each slot
β_A	Upper limit of the contention window A (cf. Fig.11)
β_B	Upper limit of the contention window B (cf. Fig.11)
β_C	Upper limit of the contention window C (cf. Fig.11)
β_D	Upper limit of the contention window D (cf. Fig.11)

Table 6: Summary of notations

5.2. Formulation of the Mathematical Model

The objective of our model is to derive expressions of the following metrics:

- The probability that a node begins its transmission in a given slot i at the time unit j which is denoted as $P\{T_i^j\}$
- The success probability for a transmission, i.e the probability that a node succeeds in transmitting a packet and that no collision occurs which is denoted as p_s

In order to compute these metrics, we analyze in detail the transmission process of a specific node denoted as the target node. According to the operation of AMPH, a node achieving the transmission process can be in one of the four states represented in Fig. 9: Backoff, Sensing (S), Transmitting (T) or Idle. Idle is the default state when a node waits for the time slot boundary. At the beginning of a new time slot, a node having data to send computes a backoff value, waits for the backoff to expire, and senses the channel. After sensing, if the channel is found idle, the transmission begins immediately. Otherwise, the transmission is delayed and the node has to wait until the beginning of the next time slot before trying to transmit again.

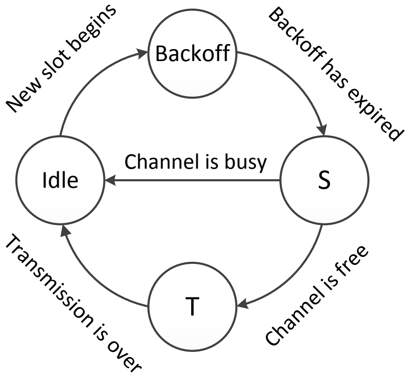


Figure 9: Full state-transitions diagram

From this analysis, we notice that the transmission of a packet is conditioned on the fact that the channel is free or busy. Evaluating the probability that the target node starts a transmission at a given time is equivalent to modeling the channel status when the node senses the channel, since we can deduce both the probability that a node begins transmission at an arbitrary time, t, given the probability that it was sensing the channel at $t-1$, and the probability to find the channel free at this moment.

In order to better describe the transitions between the sensing states over time and the transmitting states, we provide in Fig 10 the different possible sensing and transmitting states from slot 0 to a generic slot i, and the possible transitions from one state to its successors.

A transmission may begin in slot i at the time unit j only if the channel was not busy when the sender node sensed the channel at time unit $j-1$. Given that the probability of being in a sensing state in (i, j) is denoted as $P\{S_i^j\}$ and the probability

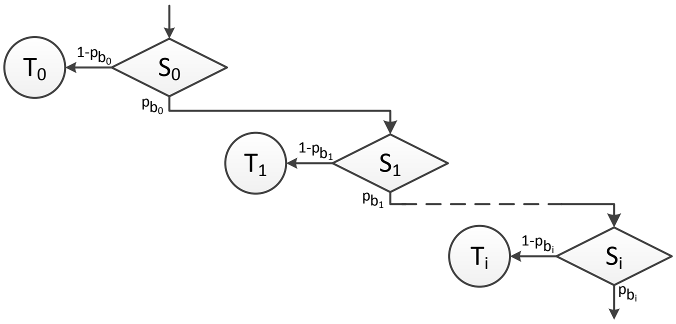


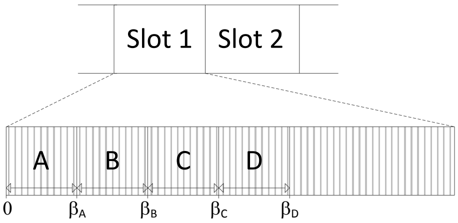
Figure 10: Representation of the transitions between Sensing and Transmitting states

that the channel is found busy in (i, j) is denoted as $p_{b_i}^{j}$, the probability to begin a transmission in (i, j) denoted as $P\{T_i^j\}$ is:

$$P\{T_i^j\} = P\{S_i^{j-1}\} \cdot (1 - p_{b_i}^{j-1}) \qquad (1)$$

Since $P\{T_i^j\}$ only depends on the probability to be in the sensing state and to find the channel free, our model aims to determine all the possible sensing states and the associated probabilities to find the channel free. In the following, the sensing states are denoted as S_i^j, where i represents the slot number and j the time unit at which the node carries out the CCA function. As the CCA duration is less that one time unit, we assume that it is performed during the last $0.128ms$ of the backoff b_v, so $j = b_v$.

The backoff is modeled as follows. The backoff time value b_v is uniformly distributed in contention windows which depends on the type of traffic that the target node wants to send and if it is the owner of the current slot. The contention windows are non-overlapping intervals set as shown in Fig. 11.



$A = [\ 0,\ \beta_A\ )$	Owner and RT
$B = [\ \beta_A,\ \beta_B\ )$	Non Owner and RT
$C = [\ \beta_B,\ \beta_C\ )$	Owner and BE
$D = [\ \beta_C,\ \beta_D\ )$	Non Owner and BE

Figure 11: Backoff contention windows

The value of b_v can be any number between 0 and β_D, thus enabling the following sensing states: S_i^0, S_i^1, ..., $S_i^{\beta_D}$. However, the behavior of the protocol is unchanged for values of b_v belonging to the same contention window. Therefore, it is possible to group the sensing states according to the values of j: the sensing states S_i^j where $j \in A$ are grouped in the meta state S_i^A, the sensing states S_i^j where $j \in B$ are grouped in the meta state S_i^B, etc. We depict a state-transition diagram of the meta sensing states in Fig 12.

A node in the sensing state can become, at the next time unit, either transmitting if the channel is free, or idle if the channel is found busy (cf. Fig 10). If the node fails to access the channel, the node will retry to access the channel at the next time slot and compute a new backoff value according to its new role and type of traffic. The diagram represents the feasible transitions from all the possible sensing states in slot i to the possible sensing states in slot $i + 1$.

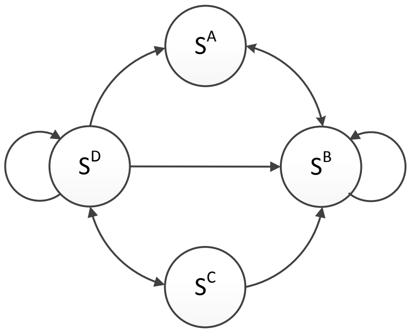


Figure 12: State-transition diagram of a generic node

The transition from state S_i^j to state $S_{i+1}^{j'}$ depends on three parameters :

- The probability to find the channel busy in (i, j) $p_{b_i}^{j}$
- The role of the node in slot $i + 1$
- The type of traffic the node has to transmit at the beginning of time slot $i + 1$

The transition probability from state S_i^j to state $S_{i+1}^{j'}$ depends only on the first parameter $p_{b_i}^{j}$, as explained below. The other two parameters determine which meta state the transition leads to. Indeed, the role of the nodes evolves and in addition, they can receive RT packets from upper layers anytime. As we want to strictly favor RT traffic over BE, if a node fails to access the channel to transmit a BE packet in slot i and receives a RT packet in the meantime, in slot $i + 1$ the node will be in the sensing state S^A or S^B, while it was in S^C or S^D in i. The sending process of the BE packet is interrupted. However, in the model, we consider the process of sending a given packet from beginning to end. As a consequence, all transitions from states S^C and S^D to states S^A and S^B are impossible. We represent the remaining possible transitions in Fig. 13 and we further provide the associated transition probabilities, according to the role and type of traffic of the node in slot $i + 1$.

We denote by $P\{S^A|S^B\}$ the transition probability from state S_i^j where $j \in B$ to $S_{i+1}^{j'}$ where $j' \in A$. In Table 7, we give the transition probabilities of all possible transitions according to the type of traffic that the target node wants to transmit and its role at slot $i + 1$.

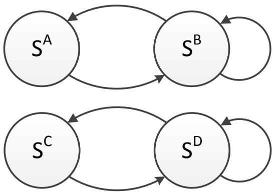


Figure 13: Simplified state-transition diagram

Node parameters	Transition probability
Node with RT traffic, owner at slot $i+1$	$P\{S^A \mid S^B\} = p_{b_i}^B$
Node with RT traffic, non owner at slot $i+1$	$P\{S^B \mid S^A\} = 0$ $P\{S^B \mid S^B\} = p_{b_i}^B$
Node with BE traffic, owner at slot $i+1$	$P\{S^C \mid S^D\} = p_{b_i}^D$
Node with BE traffic, non owner at slot $i+1$	$P\{S^D \mid S^C\} = p_{b_i}^C$ $P\{S^D \mid S^D\} = p_{b_i}^D$

Table 7: Transition probabilities

5.3. Calculation

In the previous section, we have formulated the basis of the mathematical model. In what follows, we explain in detail the calculation of the various elements provided during the model definition: the probability that the target node is sensing, the probability to find the channel busy, and the probability that the transmission starts and is successful.

5.3.1. Calculation of the probability of sensing at the next slot

Let V_{S_i} be a vector formed of the probability that the target node is in one of the four meta sensing states at time slot i.

$$V_{S_i} = \left\{P\{S_i^A\}, P\{S_i^B\}, P\{S_i^C\}, P\{S_i^D\}\right\} \tag{2}$$

The probability $V_{S_{i+1}}$ that the target node ends up in the four sensing states at time slot $i+1$ is:

$$V_{S_{i+1}} = V_{S_i} \cdot Trans \tag{3}$$

where $Trans$ is a state-transition matrix. The process is a chain, however, it is not a Markov chain since our process is not memoryless. Indeed, $Trans$ depends on the history of the node, as we explain herein after.

The possible transitions from S_i^j to $S_{i+1}^{j'}$ are determined by the role of the node in slot $i+1$ (owner or non owner), but if the node has already been owner in the current frame, it cannot be owner anymore in this frame, and therefore, states S^A and S^C are no longer accessible. In order to reflect this evolution of the role of the node, we represent the transition probabilities as three distinct transition matrices: $Trans_1$, $Trans_2$, and $Trans_3$. The computation of $V_{S_{i+1}}$ through Equation 3 uses one of these three transition matrices depending on the following scenarios:

- $Trans_1$ is used when the target node has not been owner yet and is not the owner of the next slot
- $Trans_2$ is used when the target node is the owner of the next slot
- $Trans_3$ is used when the target node has already been owner in the current frame

According to Table 7, these matrices can be written as:

$$Trans_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & p_{b_i}^B & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{b_i}^D \end{pmatrix} \tag{4}$$

$$Trans_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ p_{b_i}^B & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & p_{b_i}^D & 0 \end{pmatrix} \tag{5}$$

$$Trans_3 = \begin{pmatrix} 0 & p_{b_i}^A & 0 & 0 \\ 0 & p_{b_i}^B & 0 & 0 \\ 0 & 0 & 0 & p_{b_i}^C \\ 0 & 0 & 0 & p_{b_i}^D \end{pmatrix} \tag{6}$$

The probability $P\{S_{i+1}^{j'}\}$ that the target node fails to access the channel in (i, j) and ends up in the sensing state in $(i+1, j')$ is expressed as:

$$P\{S_{i+1}^{j'}\} = V_{S_{i+1}}(j') \tag{7}$$

In order to initialize the computation process, an initialization vector which describes the role and the type of traffic that the target node has to send is necessary. Let V_{S_0} be the vector which represents the state of the target node at slot 0.

$$V_{S_0} = \left\{P\{S_0^A\}, P\{S_0^B\}, P\{S_0^C\}, P\{S_0^D\}\right\} \tag{8}$$

The possible values of V_{S_0} are represented in Table 8.

Target node parameters	Value of V_{S_0}
Owner with RT traffic	$\{1, 0, 0, 0\}$
Non owner with RT traffic	$\{0, 1, 0, 0\}$
Owner with BE traffic	$\{0, 0, 1, 0\}$
Non owner with BE traffic	$\{0, 0, 0, 1\}$

Table 8: Possible values of the initialization vector V_{S_0}

5.3.2. Calculation of the probability to find the channel busy

The status of the channel when a node senses the channel determines if it may start to transmit or not. If the channel is found busy, this means that another node is already transmitting. Therefore, the node must delay its transmission, otherwise a collision will ensue. In AMPH, once a node gains access to the channel, it transmits as many packets as it can before the end of the slot. As a consequence, when one node starts transmission, the other nodes will find the channel busy for the rest of the slot. In this part, we aim to compute the probability that the target node finds the channel busy in (i, j) which denoted by $p_{b_i^j}$. In order to simplify the formulation of p_b, we express this probability as the opposite of the probability that the channel is free, which denoted by p_f, and $p_b = 1 - p_f$.

We compute $p_{f_i^j}$ from the point of view of the target node. The probability that the target node finds the channel free depends on the type of traffic that the target node wants to transmit, and on its role during the current slot (owner or non owner). $p_{f_i^j}$ also depends on the traffic of other sender nodes which are competing to transmit during the current slot, i.e, act as contenders. The probability that a contender wants to send RT traffic is denoted as p_{rt} and the probability that it wants to send BE traffic is denoted as p_{be}. The probabilities p_{rt} and p_{be} are considered to be constant over time.

We split the calculation of $p_{f_i^j}$ into four steps according to the role and the traffic type of the target node, i.e., if j belongs the contention window A, B, C or D.

- $j \in A = [0,\beta_A)$
  $j \in A$ when the target node is the owner of the current slot and has RT traffic to send. In this case, it has the highest priority to access the channel, so no other node can use from a smaller backoff. Therefore, it is impossible for another node to start transmitting before the target node, and $p_{f_i^j} = 1$.

- $j \in B = [\beta_A,\beta_B)$
  $j \in B$ when the target node has RT traffic to send but is not the owner of the slot. If the owner of this slot did not have RT traffic in its sending queue or had no traffic at all, the target node can still contend for channel access. However, other nodes can also have RT traffic to send and compete to access the channel.

  For $j = \beta_A$, only the owner of the slot can be transmitting, so $p_{f_i^{\beta_A}}$ is equal to the probability that the owner had no RT traffic to send, i.e.:

  $$p_{f_i^{\beta_A}} = 1 - p_{rt}$$

  For the other values of j in B, the channel can be found free if the channel was already free in $j = \beta_A$ and no node started to transmit between $\beta_A + 1$ and j. The probability that at least one node started a transmission between $\beta_A + 1$ and j is equal to the probability that its backoff value was in $[\beta_A + 1, j]$ and that there were RT packets in its queue. Given that the total number of nodes in the network is equal to N, in the worst case scenario, the number of contenders in this scenario is $N - 2$ (total number of nodes minus the target node and the owner), and since $P\{b_v \in [\beta_A + 1, j]\} = \frac{j-\beta_A}{\beta_B-\beta_A}$, we have:

  $$\begin{aligned} p_{f_i^j} &= p_{f_i^{\beta_A}} \cdot (1 - p_{rt} \cdot \frac{j-\beta_A}{\beta_B-\beta_A})^{(N-2)} \\ &= (1 - p_{rt}) \cdot (1 - p_{rt} \cdot \frac{j-\beta_A}{\beta_B-\beta_A})^{(N-2)} \end{aligned}$$

- $j \in C = [\beta_B,\beta_C)$
  A node whose backoff value belongs to the interval C is the owner of the slot and does not have RT traffic. No other node can compete to have access to the channel in C, but a transmission may already be in progress if at least one of the remaining nodes had RT traffic to send. The probability that the channel is found free for $j \in C$ is equal to the probability that no other node had RT traffic to send in slot i:

  $$p_{f_i^j} = (1 - p_{rt})^{(N-1)}$$

- $j \in D = [\beta_C,\beta_D)$
  $j \in D$ means that the target node only has RT traffic to send and is not owner of the slot, therefore, it has the lowest priority to access the channel. Nevertheless, it is still possible to find the channel free. The channel can be free in $j = \beta_C$ if no node had RT traffic, and if the owner of the slot did not have BE traffic either:

  $$p_{f_i^{\beta_C}} = (1 - p_{rt})^{(N-1)} \cdot (1 - p_{be})$$

  For the remaining possible values of j, i.e., for $j \in [\beta_C + 1, \beta_D - 1]$, the probability that the channel is free is equal to the probability that the channel was already free in $j = \beta_C$ and no node started to transmit between $\beta_C + 1$ and j:

  $$\begin{aligned} p_{f_i^j} &= p_{f_i^{\beta_C}} \cdot (1 - p_{be} \cdot \frac{j-\beta_C}{\beta_D-\beta_C})^{(N-2)} \\ &= (1 - p_{rt})^{(N-1)} \cdot (1 - p_{be}) \cdot (1 - p_{be} \cdot \frac{j-\beta_C}{\beta_D-\beta_C})^{(N-2)} \end{aligned}$$

We computed the probability that the target node finds the channel free for all possible values of j, at a generic time slot i. Since $p_{f_i^j}$ only depends on p_{rt}, p_{be}, N, and CW size, and given that all these values are constant over time, p_f is identical for every slot ($\forall$ i):

$$p_{f_0^j} = p_{f_1^j} = (...) = p_{f_{N-1}^j}$$

Given that $p_b = 1 - p_f$, we can deduce the probability to find the channel busy from the previous calculations. As a result, the value of p_b for all i and j is:

$$p_{b_i^j} = \begin{cases} 0, & \text{for } j \in A \\ 1 - (1 - p_{rt}) \cdot (1 - p_{rt} \cdot \frac{j-\beta_A}{\beta_B-\beta_A})^{(N-2)}, & \text{for } j \in B \\ 1 - (1 - p_{rt})^{(N-1)}, & \text{for } j \in C \\ 1 - (1 - p_{rt})^{(N-1)} \cdot (1 - p_{be}) \cdot (1 - p_{be} \cdot \frac{j-\beta_C}{\beta_D-\beta_C})^{(N-2)}, & \text{for } j \in D \end{cases} \tag{9}$$

5.3.3. Calculation of success probability

The success probability, p_s, is the probability that the target node successfully transmits a packet to the base station, i.e., that the node succeeds in accessing the channel and no collision occurs during the transmission. Collisions may occur if two transmissions start at the same time, which happens when two nodes select the same backoff value and sense the channel simultaneously. In order to compute p_s, we compute the probability that the transmission started by the target node in (i, j) is unique, which denoted by $p_{u_i}^{j}$.

$p_{u_i}^{j}$ is equal to the probability that no contender gets the same backoff value as the target node. We compute $p_{u_i}^{j}$ according to the possible values of j.

- For $j \in A$ and $j \in C$
  In this case, since the target node is the owner of the slot, it is the only one that can compete for channel access during these intervals. As a result, we have:

  $$p_{u_i}^{j} = 1$$

- For $j \in B$
  Only nodes with RT traffic to send can select a backoff value in this interval apart from the owner of the slot.

  Let G_i be the event "the i^{th} contender gets the same backoff value as the target node". The sample space is $\Omega = B$ and $|\Omega| = \beta_B - \beta_A$. We have:

  $$P\{G_i\} = p_{rt} \cdot \frac{1}{\beta_B - \beta_A}$$

  The probability $P\{G_i^C\}$ that the i^{th} contender does not get the same backoff value as the target node is $1 - P\{G_i\}$. In order to compute the probability that the transmission of the target node is unique, none of the contenders should pick this value. The probability that no contender picks the same backoff value as the target node is:

  $$p_{u_i}^{j} = (1 - p_{rt} \cdot \frac{1}{\beta_B - \beta_A})^{(N-2)}$$

- For $j \in D$
  The transmission attempt is unique if no contender got the same backoff value as the target node, in the event that the contenders had BE traffic and no RT traffic (otherwise their backoff would have been in B). We use the same method as above, the only difference is that the sample space is $\Omega = D$ and $|\Omega| = \beta_D - \beta_C$. Also, only nodes with BE traffic and no RT traffic can be contenders. We have:

  $$p_{u_i}^{j} = (1 - p_{rt})^{(N-1)} \cdot (1 - p_{be} \cdot \frac{1}{\beta_D - \beta_C})^{(N-2)}$$

The probability that the transmission of the target node is unique can now be summarized as:

$$p_{u_i}^{j} = \begin{cases} 1, & \text{for } j \in A \\ (1 - p_{rt} \cdot \frac{1}{\beta_B - \beta_A})^{(N-2)}, & \text{for } j \in B \\ 1, & \text{for } j \in C \\ (1 - p_{rt})^{(N-1)} \cdot (1 - p_{be} \cdot \frac{1}{\beta_D - \beta_C})^{(N-2)}, & \text{for } j \in D \end{cases} \tag{10}$$

Finally, we obtain p_s using the following relation:

$$p_s = \sum_{i=0}^{N-1} \Big(\textstyle\sum_{j=0}^{\beta_D - 1} \big(P\{T_i^j\} \cdot p_{u_i}^{j} \big) \Big) \tag{11}$$

5.4. The Algorithm

In order to compute all the target performance metrics, we provide the following algorithm. It allows the evaluation of the probability that the target node starts a transmission within a given time and the associated probability of success, according to the following parameters. Through these parameters, we will subsequently analyze the performance of our protocol in depth under various scenarios.

- The type of traffic of the target node
- The traffic of contenders p_{rt} and p_{be}
- The network size N

Id and V_{S_0} are initialization data. Id is the identifier of the target node, and also the slot number to which it is assigned. V_{S_0} describes the initial state of the target node and is initialized according to Id and the type of traffic that we aim to study. We listed all possible values of V_{S_0} earlier in Table 8.

Algorithm 1

```
Input: $N$, $Id$, $p_{rt}$, $p_{be}$, and $V_{S_0}$
Output: $P\{T\}$ and $p_s$
  Compute $p_b{}^j$ ∀ $j$ according to (9)
  Compute $Trans_1$, $Trans_2$, and $Trans_3$ according to (4), (5), and (6)
  Compute $P\{T_0^j\}$ ∀ $j$ according to (1)
  for $i = 0 \rightarrow N - 1$ do
      for $j = 0 \rightarrow \beta_D - 1$ do
          Compute $P\{S_i^j\}$ according to (7)
          Compute $P\{T_i^j\}$ according to (1)
          Compute $p_{u_i}^{j}$ according to (10)
          Compute $V_{S_{i+1}}$ according to (2) and (3)
      end for
  end for
  Compute $p_s$ according to (11)
  return $P\{T\}$ and $p_s$
```

Our algorithm assumes that the target node starts the transmission process at slot 0. However, in real operation the transmission process starts as soon as the packet is received from upper layers and the slot number can be any value between 0 and $N - 1$. Since the probability of transmission and the probability of success depend on whether the target node is owner or not, the performance results are highly influenced by the slot number assigned to the target node. For instance, if we consider that the target node is the owner of slot 0 and that it wants to transmit RT traffic, then $P\{T_0\} = 1$. However, if the target node

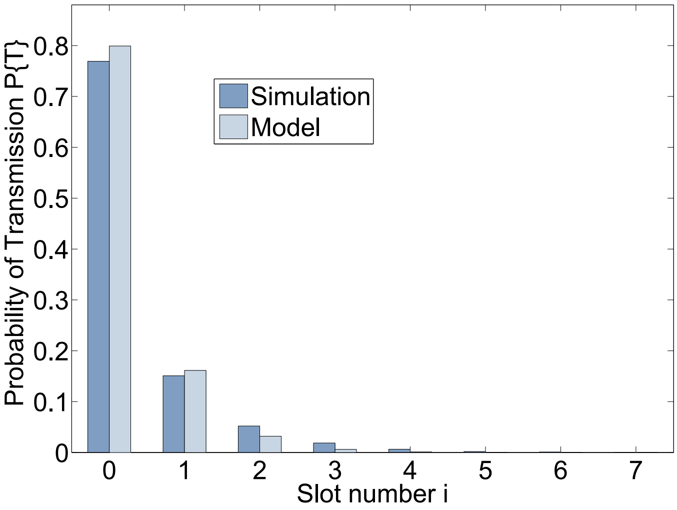


Figure 14: Probability of transmission at the i^{th} attempt (slot) where $p_{rt} = 0,07$

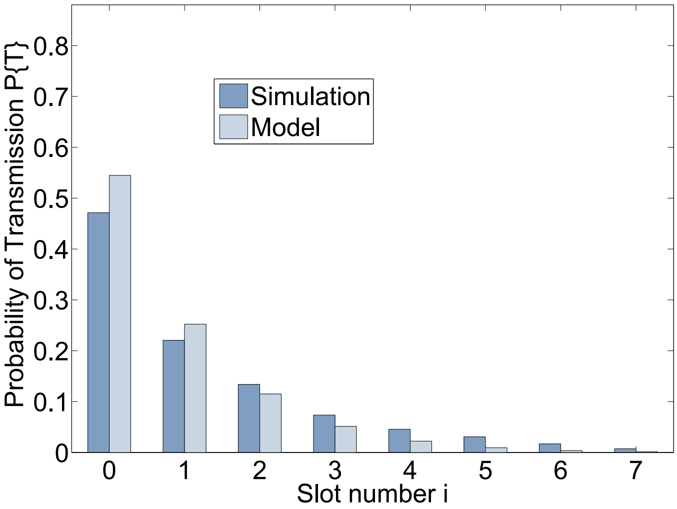


Figure 15: Probability of transmission at the i^{th} attempt (slot) where $p_{rt} = 0,19$

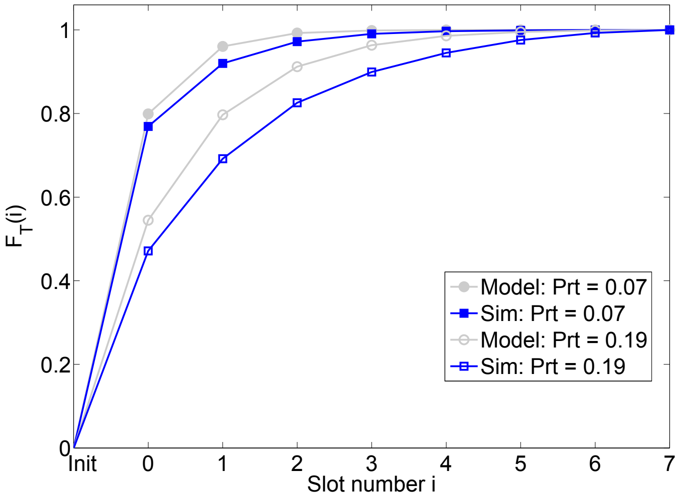


Figure 16: Cumulative probability distribution function $F_T(i)$ obtained through simulations and through the mathematical model for different values of p_{rt}.

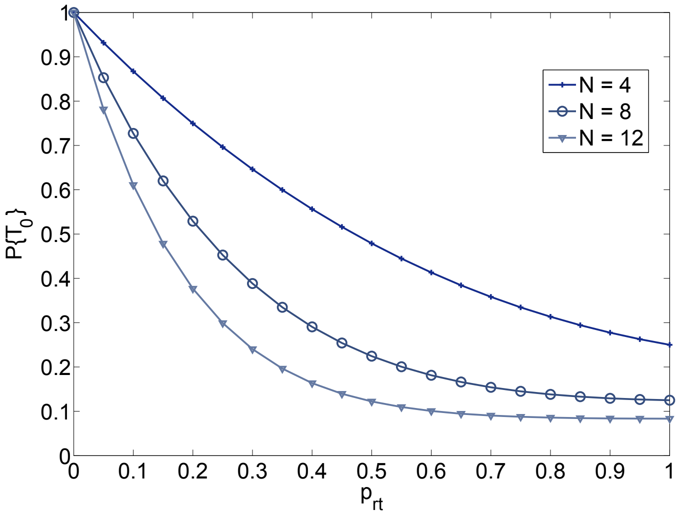


Figure 17: Probability of transmission at the first attempt $P\{T_0\}$ $(i = 0)$ as a function of p_{rt} obtained through the mathematical model for different values of N.

is non owner, $P\{T_0\}$ takes smaller values, since it has to contend with the other nodes to gain access to the channel. In order to evaluate the average performance, we run the algorithm with each possible value of Id and compute the arithmetic mean of $P\{T\}$ and p_s.

5.5. *Numerical Results and performance analysis*

In this last part, we use the model to analyze the performance of AMPH through the study of the behavior of one node in a given network of N nodes. Numerical computations are carried out using Matlab. Equivalent scenarios are performed in the simulator OMNeT++ in order to validate the model. As explained in the previous section, the following results are the average obtained by running Algorithm 1 with each possible value of Id. Since we aim to demonstrate the efficiency of the QoS mechanisms of our protocol, in a first phase, we study the performance of AMPH for the RT traffic class. First, we analyze the transmission probability and we derive the MAC latency for RT traffic, then we assess the data delivery ratio of RT traffic through the probability of success. The performance of AMPH regarding BE traffic is discussed subsequently.

5.5.1. *Transmission probability and latency of RT traffic*

In Figs. 14 and 15, we plotted the probability of transmission $P\{T\}$ as a function of i, for different values of p_{rt} while N is set to 8. Fig. 16 represents the corresponding cumulative probability distribution function $F\{T_i\}$. As for Fig. 17, it plots the probability of transmission at the first attempt as a function of p_{rt} for different values of N.

First, we can observe that simulation results and results from the mathematical model do not exhibit significant differences, and therefore, the model is validated. Secondly, the cumulative function shows that the probability of transmission reaches 1 for $i = N - 1$, thus demonstrating that AMPH ensures that when a node has RT traffic to send, it will succeed in access-

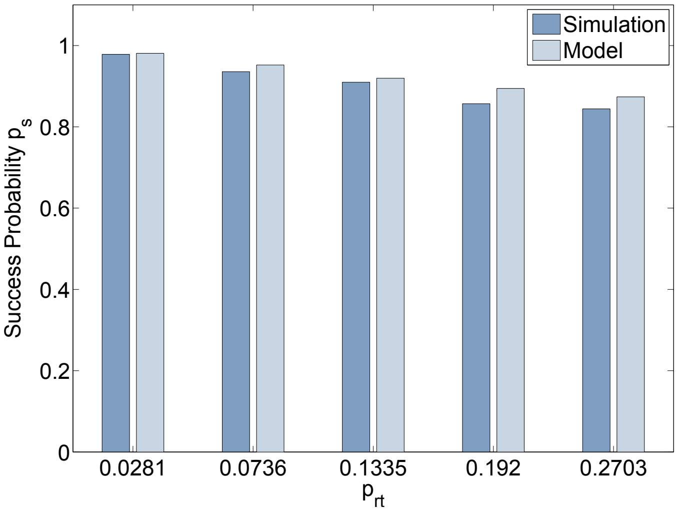


Figure 18: Transmission success probability of a RT packet obtained through simulations and the mathematical model for different values of p_{rt} and $N = 8$.

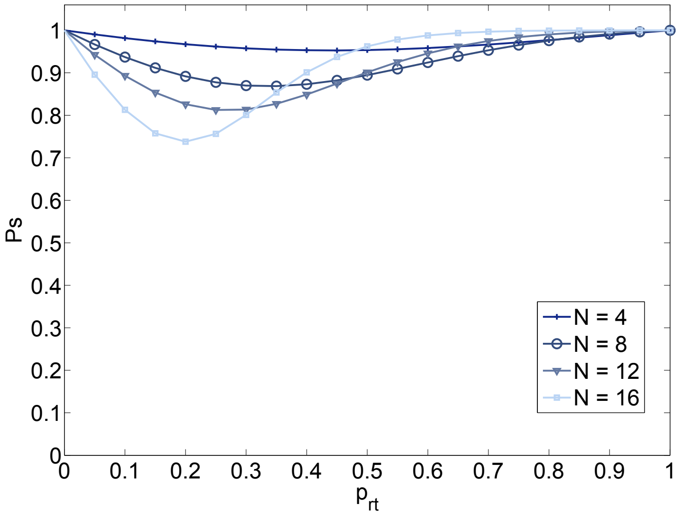


Figure 19: Transmission success probability of a RT packet as a function of p_{rt} obtained through the mathematical model for different values of N.

ing the channel before a time frame has elapsed, in the worst case after $N - 1$ attempts. Also, the cumulative probability distribution function indicates that AMPH minimizes the channel access delay, since it shows that the probability of transmission is high as from low values of i, i.e., from the first transmission attempts. Finally, Fig. 17 shows how AMPH ensures high probability to transmit at the first attempt, even if $P\{T_0\}$ decreases as p_{rt} increases. Nevertheless, the drop is not sharp for low values of N. Indeed, for $N = 8$ and $p_{rt} = 0.19$, $P\{T_0\} \approx 0.5$, which is very satisfactory. In addition, this figure points out that $P\{T_0\}$ is bounded, as for $p_{rt} = 1$, $P\{T_0\} = 1/N$.

In summary, the results show that AMPH guarantees that the latency of RT traffic is minimized and bounded by the duration of one time frame. This analysis confirms the trend observed earlier by the simulation results.

5.5.2. *Success probability of RT traffic transmissions*

The following figures depict the results of the evaluation of the probability of success p_s obtained through simulations and the mathematical model for RT traffic. In Fig. 18, we plotted simulation results of p_s and corresponding numerical results obtained through the mathematical model in order to evaluate the data delivery ratio of AMPH. In this experiment, we fixed N to 8. As would be expected, the success probability decreases as the probability that contenders have RT traffic increases. Indeed, the probability that a neighboring node tries to send RT traffic during an empty slot increases, therefore, the probability of collision rises accordingly. We observe a very slight difference between the analytical and simulation results. Therefore, our model is accurate.

In Fig. 19, we plot different values of p_s obtained through the mathematical model as a function of p_{rt} for different values of N, in order to study the influence of the traffic of contenders and of the network size on the data delivery ratio. According to what we observed in the previous figure, p_s decreases as p_{rt} increases, but surprisingly, it increases again past some value of p_{rt}, which depends on the size of the network. In fact, this behavior is normal given that when p_{rt} grows, it is more likely that other nodes have RT traffic to send during their own slot, thus the target node will not find empty slots to steal and will have to wait for its own time slot to perform its sending attempt, in which it is impossible that a collision occurs.

We notice that the reliability of AMPH deteriorates as N grows for medium values of p_{rt}. Indeed, the larger the network, the more the reliability decreases, as p_u decreases exponentially as a function of N. Nevertheless, if we look at these results from the temporal point of view, the collision probability can also be seen as the probability that the target node accesses the channel without waiting for its own slot, thus improving latency. There is a trade-off between reliability and latency. In our solution, we chose to focus on latency, since high-throughput traffic like multimedia traffic is relatively loss tolerant but not delay tolerant. Also, the reliability in small networks or for low values of p_{rt} remains fully acceptable: below a 10% packet loss, coding techniques can compensate [20]. If one may want to use AMPH under unfavorable conditions, it is entirely possible to implement a safe mode that would be triggered when excessive degradation of reliability occurs, where the base station sends acknowledgments when the sender transmits during the time slot of another node. We decided not to implement this technique, since in addition to minimizing latency, we also aim to maximize throughput and not to waste it by using multiple control messages.

5.5.3. *Transmission probability and latency of BE traffic*

In order to provide a comprehensive overview of the performance of AMPH, we also evaluate $P\{T\}$ and p_s for BE traffic. As a first step, we consider a network with no RT traffic.

In Fig. 20, we study the probability of transmission of BE packets by representing the cumulative function $F_T(i)$ of simulation and analytical results of the evaluation of $P\{T_i\}$ for different values of p_{be}. During this experiment, N was set to 8. We can see that the results obtained through simulations and the mathematical model are very close, which means that our

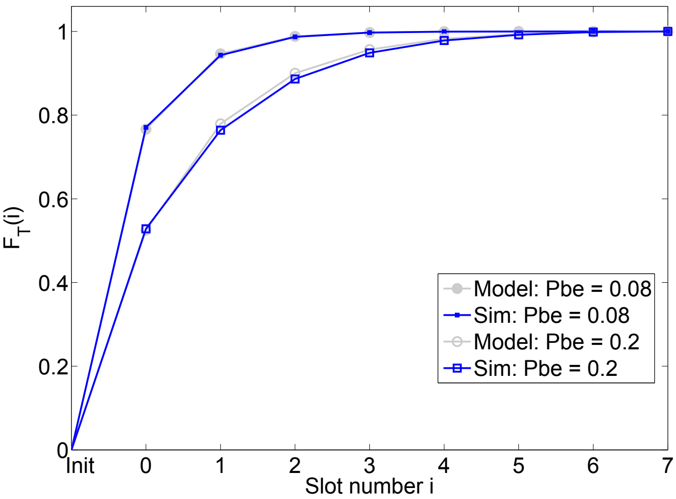


Figure 20: Cumulative function $F_T(i)$ as a function of i obtained through simulations and through the mathematical model for different values of p_{be} where $p_{rt} = 0$.

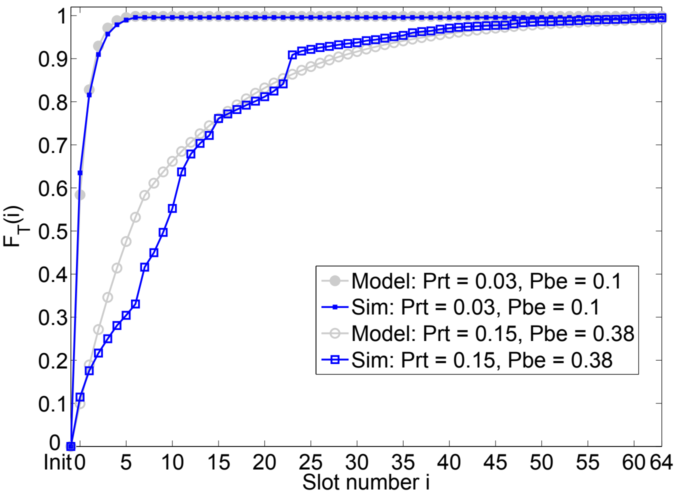


Figure 21: Cumulative function $F_T(i)$ as a function of i obtained through simulations and through the mathematical model for different values of p_{rt} and p_{be}.

model of the BE traffic is also accurate. With no RT traffic in the network, the latency of BE traffic is similar to that of RT traffic. Indeed, the mechanism to access the channel is the same, but with an extra access overhead, since the backoff values are a little bit larger.

In the following experiments, we introduce RT traffic in order to analyze its impact on the latency of BE packets. As in the previous figure, Fig. 21 shows the cumulative function of the transmission probability as a function of i. We still observe a good agreement between both simulation and analytical results. As p_{rt} increases, the probability to transmit BE traffic within a minimum number of time slots decreases, so the latency of BE traffic increases accordingly, but it remains acceptable. For instance, when $p_{rt} = 0,15$, $P\{T_{25}\} > 0,8$, i.e., there is more than 80% chance that the transmission happens before $i = 25$, which gives a MAC latency of $25 * 40,96$ (slot duration) $\approx$ 1s.

In Fig. 22, we plot $F_T(i)$ for larger values of p_{rt}, namely $0,25$ and $0,5$, and different values of p_{be}, in order to further analyze the impact of the traffic of contenders on the MAC latency for BE packets. The results show that p_{rt} is the key parameter regarding BE latency. We can see that the parameter p_{be} also affects the results but its influence on $P\{T\}$ is smaller. For high values of p_{rt}, the transmission probability of BE packets is poor. In this case, the anti-starvation mechanism implemented in AMPH is highly desirable in order to increase the transmission probability of BE packets.

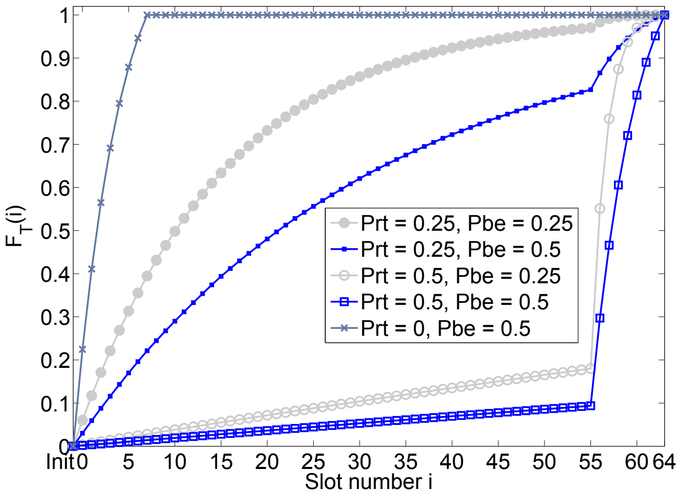


Figure 22: Cumulative function $F_T(i)$ as a function of i obtained through the mathematical model with larger values of p_{rt}.

Fig. 22 also demonstrates the importance of our anti-starvation mechanism regarding the latency of best-effort traffic. When p_{rt} is high, the chances to transmit BE traffic drop. Our mechanism allows that, in M frames among N, BE traffic has priority over RT traffic. In this experiment, N was set to 8 and M to 1, so one frame among eight is arranged to favor BE traffic over RT traffic. Since in a star topology, the maximum number of slots is equal to the number of nodes in the network, the size of the frame is equal to 8. The switch in the priorities occurs at the 8^{th} frame, then 56 slots have elapsed ($7 \cdot 8$ slots). We can see the discontinuity in the figure at $i = 56$, from where the transmission of BE traffic is favored. At the end of this frame, we notice that $F_T(i)$ reaches 1, hence proving that the latency of BE packets is also bounded. As a consequence of our anti-starvation mechanism, the maximum MAC latency for BE traffic is $N^2 \cdot$ *slot duration*. This mechanism is optional and may be triggered only when the BE queue reaches a certain threshold, and the occurrence frequency of special frames can be adjusted according to the traffic conditions through the parameter M. However, the more often the special frame occurs, the less bandwidth remains for RT traffic. A trade-off must be found between BE latency, RT latency, and the throughput required by each traffic class.

5.5.4. *Success probability of BE traffic transmissions*

Finally, we consider the probability of success when transmitting BE packets. In Fig.23, we plot p_s for BE traffic for different values of p_{be}, $N = 8$ and $p_{rt} = 0$. The results were obtained through simulations and using the model. We observe

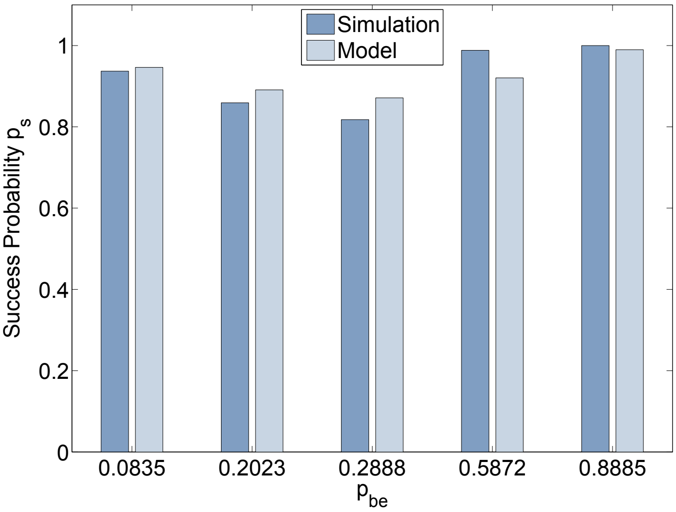


Figure 23: Transmission success probability of a BE packet obtained through simulations and the mathematical model for different values of p_{be} and $N = 8$.

the same behavior as for the probability of success of RT traffic transmissions. As p_{be} increases, p_s decreases, until the probability to find an empty slot falls and then p_s starts to rise. We can see that the success probability of BE traffic transmissions is high, which confirms that AMPH achieves high reliability for BE traffic as well, as we observed through the simulation results in Section 4. In the worst case, when $p_{be} = 0.2888$, p_s stays above 0.8. Considering that BE traffic is redundant in most applications (e.g., temperature measurements of several sensors in the same area), the impact of a limited packet loss is negligible.

6. Conclusion

In this paper, we proposed AMPH, a new adaptive MAC protocol for heterogeneous wireless sensor networks with fair service differentiation, high throughput and QoS support. The simulation and analytical results have demonstrated that AMPH's hybrid behavior outperforms contention-based protocols, like CSMA/CA and Diff-MAC, in terms of channel utilization, latency and reliability. AMPH effective fair service differentiation and QoS mechanisms minimize real-time traffic latency and prevent best-effort traffic starvation. The time division schedule enhances the contention resolution leading to high reliability. Hence, AMPH enables efficient delivery of heterogeneous traffic for a new generation of promising applications with high QoS requirements.

Through extensive simulation experiments and formal analysis, we studied the performance of our protocol using a star topology network. Although this network configuration fits many applications (for example medical monitoring applications), as a future work we intend to generalize the protocol to be suitable for multi-hop networks. A large scale implementation raises new challenges as global clock synchronization and the hidden terminal problem. Enhancements can also be achieved through more dynamic mechanisms like adaptive backoff values, slot duration and number of frames reserved to avoid best-effort traffic starvation. Furthermore, we plan to implement AMPH on our Imote2 platform in order to validate AMPH performance through experimentation.

Acknowledgments

This work was carried out in the framework of the Labex MS2T, which was funded by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (2002) 393–422.

[2] I. F. Akyildiz, T. Melodia, K. R. Chowdhury, A survey on wireless multimedia sensor networks, Computer Networks 51 (2007) 921–960.

[3] H. Alemdar, C. Ersoy, Wireless sensor networks for healthcare: A survey, Computer Networks 54 (2010) 2688–2710.

[4] D. Chen, P. K. Varshney, QoS Support in Wireless Sensor Networks: A Survey, in: Proc. of the 2004 International Conference on Wireless Networks (ICWN 2004), Las Vegas, Nevada, 2004.

[5] I. Demirkol, C. Ersoy, F. Alagöz, MAC protocols for wireless sensor networks: a survey, Communications Magazine, IEEE 44 (4) (2006) 115–121.

[6] Z. Teng, K.-I. Kim, A Survey on Real-Time MAC Protocols in Wireless Sensor Networks, Communications and Network 2 (2) (2010) 104–112.

[7] P. Suriyachai, U. Roedig, A. Scott, A Survey of MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks, IEEE Communications Surveys & Tutorials (2011) 1–25.

[8] S. Ullah, B. Shen, S. Riazul Islam, P. Khan, S. Saleem, K. Sup Kwak, A Study of MAC Protocols for WBANs, Sensors 10 (1) (2010) 128–145.

[9] M. A. Yigitel, O. D. Incel, C. Ersoy, QoS-aware MAC protocols for wireless sensor networks: A survey, Computer Networks 55 (8) (2011) 1389–1286.

[10] N. Saxena, A. Roy, J. Shin, Dynamic duty cycle and adaptive contention window based QoS-MAC protocol for wireless multimedia sensor networks, Computer Networks 52 (13) (2008) 2532–2542.

[11] M. A. Yigitel, O. D. Incel, C. Ersoy, Design and implementation of a QoS-aware MAC protocol for Wireless Multimedia Sensor Networks, Computer Communications 34 (16) (2011) 1991–2001.

[12] IEEE 802.11e WG, Medium Access Control (MAC) Enhancements for Quality of Service (2005).

[13] I. Rhee, A. Warrier, M. Aia, J. Min, Z-MAC: a Hybrid MAC for Wireless Sensor Networks, IEEE/ACM Transactions on Networking 16 (3) (2008) 511–524.

[14] I. Rhee, A. Warrier, J. Min, L. Xu, DRAND: distributed randomized TDMA scheduling for wireless ad-hoc networks, in: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '06, New York, NY, USA, 2006, pp. 190–201.

[15] I. Slama, B. Shrestha, B. Jouaber, D. Zeghlache, A Hybrid MAC with Prioritization for Wireless Sensor Networks, in: 33rd IEEE Conference on Local Computer Networks (LCN 2008), 2008.

[16] OMNet++ network simulator, <http://www.omnetpp.org/ >.

[17] M. Souil, T. Rault, A. Bouabdallah, A New Adaptive MAC Protocol with QoS support for Heterogeneous Wireless Sensor Networks, in: 17th IEEE Symposium on Computers and Communications (IEEE ISCC 2012), Cappadocia, Turkey, 2012.

[18] C. Buratti, R. Verdone, Performance Analysis of IEEE 802.15.4 Non Beacon-Enabled Mode, Vehicular Technology, IEEE Transactions on 58 (7) (2009) 3480–3493. doi:10.1109/TVT.2009.2014956.

[19] Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (Phy) Specifications for Low-Rate Wireless Personl Area Networks (LR-WPANS) (2006).

[20] S. Ye, M. Ouaret, F. Dufaux, M. Ansorge, T. Ebrahimi, Error resiliency of distributed video coding in wireless video communication, in: Proc. SPIE 7073, Applications of Digital Image Processing XXXI, 2008.