Quantitative analysis of the hidden terminal problem in preamble sampling WSNs

C. Cano *, B. Bellalta, A. Cisneros, M. Oliver

NeTS Research Group, Dept. of Information and Communication Technologies, Universitat Pompeu Fabra, C/Tànger 122-140, 08018 Barcelona, Spain

ARTICLE INFO

Article history:
Received 20 August 2010
Received in revised form 10 March 2011
Accepted 10 April 2011
Available online xxxx

Keywords:
WSNs
MAC protocols
Hidden terminal
Performance evaluation

ABSTRACT

Collisions in preamble sampling WSNs cause a high waste of resources due to the mandatory transmission of a long preamble before every message. Moreover, when hidden terminals are present, collisions are even more noticeable since the vulnerability time is proportional to the long preamble duration. This effect reduces considerably the network performance as both the number of collisions and the resources spent in them are significant. The effects of hidden nodes in preamble sampling WSNs, in different case scenarios and under variable offered loads, are analyzed in this work. Results show that the impact of hidden terminals in the network performance is non-negligible, specially at medium to high loads. These results should be considered by MAC protocol designers in order to devise mechanisms to mitigate the hidden terminal problem.

1. Introduction

During the last few years several new medium access control (MAC) protocols have been designed to cope with the energy consumption constraint of wireless sensor networks (WSNs). These approaches were mainly motivated by the fact that the MAC layer controls the transceiver, that is the most energy consuming component of a sensor node. The common approach is to perform a low duty cycle operation in which sensor nodes sleep most of the time and wake up only to send or receive data. These approaches can be divided in two major categories [1]: (i) scheduled, where some kind of organization among sensor nodes is performed to decide when to sleep and wake up and (ii) unscheduled, where sensor nodes decide their duty cycle independently of the other nodes in the network.

Unscheduled MAC protocols are specially well suited for WSNs as precise synchronization is not needed, they are simpler and in most cases consume less energy than scheduled approaches. Among these, low power listening (LPL) (also known as preamble sampling) protocols have shown very good results in terms of energy consumption as they are capable of working at considerably low duty cycles (~1%). In this kind of protocols, sensor nodes wake up during a short period of time just to sample the medium: if the medium is empty they return to sleep but remain awake to receive the data if the medium is found busy. In the basic preamble sampling technique [2–4] a sensor node willing to transmit before sending the data it transmits a long preamble that overlaps with the listening time of the receiver, thus assuring it will be awake to receive the packet. Observe that at low loads the energy consumption is extremely reduced but, as soon as the load increases, collisions become very costly as the entire long preamble is involved.

The hidden terminal problem [5] is a well-known effect that occurs in wireless networks. The most simple case happens when two or more nodes, that are unable to sense each other, transmit simultaneously to a common receiver, causing a collision. In traditional wireless local area networks (WLANs), the most common way to alleviate this problem is the use of the request-to-send/clear-to-send (RTS/CTS) mechanism [6], that is in fact used in IEEE 802.11 [7] networks as an optional feature. Using this method, a sender first transmits an RTS message that is
replied with a CTS by the destination if it is ready to receive the data. As the length of the RTS is small (if compared to data messages common in WLANs), the vulnerability time is reduced, thus mitigating the effects of the hidden terminals. On the contrary, in preamble sampling WSNs, the vulnerability time is high due to the transmission of the long preamble, thus the hidden terminal is considerably more problematic than in WLANs. Note that if the RTS/CTS mechanism is used, the long preamble should be transmitted at least before the RTS message, but also before the CTS to guarantee the correct data transmission (otherwise, hidden terminals can be sleeping). This approach maintains the high vulnerability time but it also results in too much overhead for the typically small data messages found in this kind of networks.

Collisions from hidden nodes can be mitigated due to the capture effect, that is the ability of certain radios to correctly receive a message in spite of overlapping collisions (given that the difference of received signal strengths among them is sufficient). This effect, studied for the case of wireless sensor transceivers in [8,9] and also considered in this work, can help mitigating the hidden terminal problem, however, it is not capable of completely solving it as collisions among packets with similar received signal strengths can still happen.

Although the literature regarding wireless sensor networks is extensive, a detailed quantitative analysis of the hidden terminal problem in preamble sampling WSNs is still needed. It is usually considered that at the low loads that WSNs work, this problem is irrelevant. However, although some WSNs work at low traffic loads during most of the time, increases of the network load can suddenly happen caused by query dissemination or due to event detection at nearby sensor nodes. Moreover, the convergecast communication typical in WSNs shows higher loads in those nodes closer to the sink.

This work provides a clear understanding of the hidden terminal problem in preamble sampling WSNs in different scenarios and under a broad range of offered loads. The methodology of this study is based on both analytical modeling and simulations. A preamble sampling analytical model that captures the impact of hidden terminals in a set of key scenarios has been developed. Simulations complement the analytical outcomes in these key scenarios but also allow to evaluate the more complex scenarios studied in this work.

The rest of the paper is organized as follows: Section 2 reviews the related work of the hidden terminal problem in WSNs, then, in Section 3, the system considerations and assumptions are detailed. The analytical framework is presented in Section 4. After that, the quantitative analysis is performed in Section 5. Finally, Section 6 discusses the effects of hidden terminals in improved preamble sampling techniques and Section 7 provides some concluding remarks.

2. Related work

The hidden terminal problem [5] has been extensively studied, specially in IEEE 802.11 [7] networks. However, although hidden terminal effects are more problematic in preamble sampling WSNs, there have not been many efforts to study and quantify it. This problem has been considered, as well as some ideas to alleviate it, in the definition of some new preamble sampling MAC protocols.

The authors of [10] propose, in order to mitigate the hidden terminal problem, to increase the sensitivity of the sensor nodes. This technique, although effective to some extent, affects the performance of the network by reducing the throughput that can be achieved, apart from increasing the energy consumption of the sensor nodes (since higher sensitivity implies higher current draw).

Other works, such as [11], where a tree-like topology is considered, recommend to delay the transmissions each time a node overhears a message from its parent, assuming that its grandparent will forward the message after that. However, hidden terminal problems can also happen between independent generated messages, as will be seen throughout this paper.

Another proposed solution is to tune the back-off, increasing either the slot duration [12] or the contention window [13]. These techniques are effective for non-preamble sampling WSNs where message transmissions are comparable to back-off durations, however the back-off values that could alleviate collisions among hidden terminals in preamble sampling WSNs are so high that will extremely reduce the performance of the network.

Finally, there are solutions that group into clusters non-hidden nodes and use different listening times for each group, such as the work presented in [14]. However, they are addressed to scheduled approaches where synchronization is required.

3. System considerations

A preamble sampling WSNs has been considered. Therefore, sensor nodes sleep and wake up following the duty cycle. Periodically, sensor nodes wake up and sample the channel, if the medium is found idle, they go to sleep. Otherwise, if the medium is found busy, they remain active to receive the message. When a sensor node wants to send a message, it wakes up immediately and performs the medium access procedure. A long preamble that overlaps with the listening time of the receiver is transmitted before a message, thus assuring it will be awake during the data transmission. It has been considered that a long preamble is always sent before every packet, even for consecutive packets directed to the same destination. Moreover, the preamble sampling technique used in this work does not consider the adaptation of the duty cycle (and therefore of long preamble duration) to the traffic load. However, a discussion of how the hidden terminal problem can affect the performance in improved preamble sampling techniques is provided in Section 6.

Before accessing the channel, sensor nodes compute a random back-off between 0 and the Contention Window (CW). Assuming that the medium is slotted, the back-off...
time is decremented in one unit each slot time that the medium is detected free. Once the back-off expires, the node transmits the packet. If the medium is sensed busy during the back-off countdown, the back-off is frozen and it is restarted when the channel is detected free again. This medium access procedure is a carrier sense multiple access (CSMA) approach and it is the same as the one defined in the medium access control of IEEE 802.11 [7] networks. However, in this work a single-stage back-off is considered, this is, the CW value is not doubled at each transmission attempt as done using the binary exponential back-off (BEB) technique defined in the IEEE 802.11 MAC protocol.

It has been assumed that sensor nodes generate packets following a Poisson distribution with a constant packet length. In order to obtain a clear understanding of the problem only local communication is considered. This behaviour eliminates the variable traffic loads in the network and therefore the differences in the hidden terminal problem among different regions.

It has been considered that transmissions to the wireless channel suffer from path loss (due to the distance between transmitter and receiver) and slow fading. The shadow fading for each communication pair is considered symmetric and time invariant. Thus, the power at which frames sent by A are received at B is the same than from B to A, assuming that A and B use the same transmitting power.

In order to eliminate interfering factors in the analysis the wireless channel is assumed to be noise-free, therefore only the interferences produced by collisions cause transmission errors. If a collision occurs among packets of similar received signal strengths, all packets involved are lost. However, after a sensor node starts receiving a packet (or long preamble), overlapping collisions without corrupting the initial reception are possible (capture effect), given that the difference of signal strengths is higher enough. It has been considered that to capture a packet it must be received before the interfering signals. Therefore, if a receiver detects a signal, even if its considerably weak, it cannot resynchronize to a stronger signal that arrives later. This feature tries to model the fact that many radios cannot resynchronize to a stronger signal if they have already received a weaker one and it is a common assumption in many wireless simulators [15].

Considering a non-coherent FSK modulation and Manchester encoding (as used in Mica2 motes [16]), the packet error rate (PER) is calculated as shown in Eq. (1) [17].

\[
\text{PER} = 1 - (1 - \text{BER})^{2L}
\]

where \(L\) is the packet length (including headers) in bits and \(\text{BER}\) is the bit error rate, approximated as \(\text{BER} = \exp(-\text{SIR}/2)/2\), in absence of noise. The signal-to-interference ratio (SIR) can be calculated as the relation of received and interfering powers \(P_r / \sum P_{\text{int}}\). Note that the length of the long preamble is not considered as it does not provide any useful information and can, therefore, be corrupted.

Observe that the calculation above is only valid for complete interfering collisions (i.e., collisions that start during the preamble reception), however, in a hidden terminal scenario partial collisions should also be considered. In that case, the PER is calculated as:

\[
\text{PER} = 1 - \prod_{k=1}^{n_{\text{int}}} (1 - \text{BER}_k)^{2L_k}
\]

where \(n_{\text{int}}\) is the number of different interference regions of a packet, \(\text{BER}_k\) is the bit error rate of each region and depends on the number of interferences that affect it, and \(L_k\) refers to the number of bits that form each region. See Fig. 1 where an example of \(n_{\text{int}} = 2\) is shown.

Moreover, reception and carrier sense thresholds are considered. It is assumed that packets with received power over the reception threshold (sensitivity) can be correctly received. However, if the received power is below the reception threshold but higher than the carrier sense threshold, the packet cannot be understood but sensor nodes are able to detect that the medium is busy. Otherwise, if the received power is lower than the carrier sense

![Fig. 1. Example of total and partial overlapping collisions. In the example, Source Node 1 arrives at the receiver with the highest power. Its packet error rate is computed taking into account SIR1, that only affects the first part of the message and includes the interference from Source Node 2, and SIR2, that affects the last part and includes the sum of interferences from Source Nodes 2 and 3.](image-url)
threshold sensor nodes cannot detect that there is a transmission taking place.

A preamble sampling simulator that maps the above considerations and assumptions has been used for the evaluation. The simulator framework is based on the SENSE simulator [18], however it has been extended in order to include the preamble sampling MAC protocol, the shadowing propagation model and an improved capture effect feature (able to compute the SIRs under multiple overlapping collisions).

The simulator has been validated comparing its results with a Mica2 testbed. Fig. 2 shows simulation and testbed (average and 95% confidence intervals) results in a 5-node network and in a typical hidden terminal scenario in which 2 nodes hidden from each other transmit to a common receiver. In order to assure that both transmitters in the hidden terminal case cannot sense each other, it is not enough to check that they are unable to successfully receive packets as the carrier sense mechanism can still detect that there is a transmission in the medium. Thus, with the goal to guarantee that the hidden terminal problem is taking place the carrier sense mechanism has been deactivated, therefore they always found the medium idle even if the other node is transmitting. In these scenarios the ACKs and retransmissions have been disabled and the queue length at each sensor node is equal to zero. The rest of parameters (based on Mica2 motes [16]) are shown in Tables 1 and 2.

This section has presented the default system considerations and assumptions of this work. Most of them are defined in order to avoid factors that can interfere the quantification of the problem. However, the effect of the hidden terminal problem in a more realistic scenario considering periodic and event-based traffic profiles, convergecast communication and retransmissions enabled is also studied using simulations in Section 5.5.

4. Analytical model

The development of analytical models helps in providing a better understanding of the network functionality and allows to study its performance in a broad range of scenarios with a lower computational effort if compared to simulations. Analytical models are also useful to perform optimization analysis that can provide the best operational parameters for a given network scenario. Therefore,

---

**Table 1**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data rate</td>
<td>19.2 kbps</td>
</tr>
<tr>
<td>Tx power</td>
<td>0 dBm</td>
</tr>
<tr>
<td>Sensitivity (reception threshold)</td>
<td>−99 dBm</td>
</tr>
<tr>
<td>Carrier sense threshold</td>
<td>−111 dBm</td>
</tr>
<tr>
<td>Supply voltage</td>
<td>3 V</td>
</tr>
<tr>
<td>Current consumption in Tx mode</td>
<td>24.5 mA</td>
</tr>
<tr>
<td>Current consumption in Rx/idle</td>
<td>17.6 mA</td>
</tr>
<tr>
<td>Current consumption in sleep mode</td>
<td>15 μA</td>
</tr>
</tbody>
</table>

**Table 2**

MAC parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of the data field (Ldata)</td>
<td>29 bytes</td>
</tr>
<tr>
<td>Headers (Lheaders)</td>
<td>11 bytes</td>
</tr>
<tr>
<td>Total data packet size (L)</td>
<td>40 bytes</td>
</tr>
<tr>
<td>Listen/sleep time</td>
<td>2.45/97.55 ms</td>
</tr>
<tr>
<td>Contention window (CW)</td>
<td>32</td>
</tr>
<tr>
<td>Slot time (σ)</td>
<td>417 μs</td>
</tr>
</tbody>
</table>
to understand how WSNs operate and provide further insights on the hidden terminal effects on the WSNs performance, an analytical model is presented. It captures the different traditional performance metrics (delay, throughput, etc.) and the energy consumption of the sensor nodes.

Each sensor node is modeled as a single M/M/1/K queue, assuming that: (i) packets are generated following a Poisson distribution with rate $\lambda$ packets/s, and (ii) the service time ($X$), the time since a packet of $L$ bits arrives at the head of the queue until it is released from it, follows an exponential distribution. From the M/M/1/K assumption, the offered load ($A$), the queue utilization ($\rho$) and the blocking probability ($P_b$) can be computed as shown in Eq. (3).

$$A = \lambda X, \quad \rho = A(1 - P_b), \quad P_b = \frac{(1 - A)A^K}{1 - A^{K+1}} \tag{3}$$

Considering that lost packets are not retransmitted, the service time can be calculated considering the average number of slots selected before each transmission attempt ($B$), the average slot duration ($\alpha$) and the duration of a transmission ($T$) as shown in Eq. (4).

$$X = B\alpha + T \tag{4}$$

Assuming that the ACK and RTS/CTS mechanisms are deactivated, the channel occupation duration of the node of interest transmission can be expressed as:

$$T = \frac{L_p + L}{r} \tag{5}$$

where $L_p$ is the length of the preamble and $r$ refers to the transmission rate.

Assuming that the back-off is uniformly distributed in the range [0 – CW], $B$ can be obtained as shown in Eq. (6).

$$B = \frac{CW - 1}{2} \tag{6}$$

The average slot duration ($\alpha$) is calculated considering the duration of the slot depending on the channel state, given that the node of interest is in back-off (Eq. (7)). Assuming that the channel is noise-free, it can only be in empty, successful or collision states with their corresponding probabilities $p_e$, $p_s$ and $p_c$.

$$\alpha = p_e\sigma + p_s(T_s + \sigma) + p_c(T_c + \sigma) \tag{7}$$

where $\sigma$ is the empty slot duration. The channel state probabilities as well as the channel occupation durations of other’s successful or collision transmissions ($T_s$ and $T_c$ respectively) are scenario-dependent. These metrics will be computed in Sections 4.1 and 4.2 for the cases where all nodes are in coverage range and in the presence of hidden terminals.

Thus, considering that the ACK mechanism is deactivated, the average throughput per node can be easily obtained as:

$$S = \rho \frac{L}{X} (1 - p) \tag{8}$$

where $p$ is the conditional collision probability. Its mathematical expression depends on each specific scenario and it will also be formulated in Sections 4.1 and 4.2 depending if all nodes are inside the coverage range or not.

Finally, the per-packet delay can be calculated as shown in Eq. (9).

$$D = \frac{EQ}{\lambda(1 - P_b)} \tag{9}$$

where EQ refers to the average number of packets in the queue:

$$EQ = \begin{cases} A - \frac{(K + 2)A^{K+2}}{(1 - A)^{K+2}} & \text{if } A < 1 \\ \frac{K+1}{2} & \text{if } A = 1 \end{cases} \tag{10}$$

The analytical model is solved using a fixed point approximation and the metrics obtained are used to compute the energy consumption, that is presented in Section 4.3.

4.1. Case 1: All nodes in coverage range

When all nodes are in coverage range, i.e., all sensor nodes ($n$) can sense each other, the conditional collision probability (assumed to be constant for all transmission attempts) can be computed as shown in Eq. (11). Given that the node of interest transmits, there will be a collision any time that any of the other $n - 1$ nodes in the network transmit in the same slot.

$$p = 1 - (1 - \tau)^{n-1} \tag{11}$$

Being $\tau$ the steady state probability that a node transmits in a random slot given that it has a packet ready to be transmitted:

$$\tau = \frac{\rho}{B + 1} \tag{12}$$

The channel state probabilities are related to the stationary probability that the rest of nodes (except the one that is in back-off) try to transmit in a given random slot, see Eq. (13). If none of the remaining nodes ($n - 1$) transmit, the slot remains empty, if only one transmits, it results in a successful transmission, and finally, if more than one transmit, the channel state results in a collision.

$$p_e = (1 - \tau)^{n-1}$$

$$p_s = (n - 1)\tau(1 - \tau)^{n-2}$$

$$p_c = 1 - p_e - p_s \tag{13}$$

Finally, and assuming that the ACK and RTS/CTS mechanisms are deactivated, the channel occupation durations can be computed as shown in Eq. (14).

$$T_s = T_c = T = \frac{L_p + L}{r} \tag{14}$$

4.2. Case 2: Presence of hidden terminals

Several multihop analytical models for wireless networks, with more emphasis on the IEEE 802.11 protocol, exist in the literature, being one of the most complete the one presented by Alizadeh et al. in [19]. However, as
the vast majority, they assume that hidden terminals enter in the contention phase synchronized. This assumption allows to model the conditional collision probability with the hidden terminals as \( p = 1 - (1 - \tau)^v \), where \( v = n_h T_v / \sigma \), being \( n_h \) the number of hidden terminals and \( T_v \) the vulnerability time. The same approach is also followed in the only, as far as the authors know, multihop analytical model of CSMA with preamble sampling [20].

Tsertou et al. first claimed and studied the inaccuracy of this assumption [21]. They show that the model becomes problematic when the vulnerability time is higher than the average back-off duration due to the time difference of the back-off phase among hidden terminals. Observe that this is the case of preamble sampling WSNs in which the vulnerability time includes the long preamble transmission that is usually large (in the range of thousands of ms). They also provide a different approach to model the effect of hidden terminals, however, although the model is quite complex, only saturation conditions and a network formed by just two transmitting and hidden nodes are considered.

The analytical model of hidden terminals presented in this work follows the same approach presented in Section 4.1 but the conditional collision probability is computed without assuming synchronization among hidden nodes. This has been done by defining three main assumptions: (i) all source nodes in the network are hidden terminals of each other, (ii) hidden terminals do not have transmitting neighbours and (iii) the maximum back-off is smaller than the data packet transmission time.

Observe that assumptions (i) and (ii) are the same than the ones in [21] and hold for WSNs if the load of the network is low as the effect of hidden terminals dominates among the effects of neighbouring nodes. The extension of the analytical model to a general network is a challenge due to the aforementioned desynchronization among hidden terminals. As will be explained later, the approach presented here assumes that there will be a collision always that a hidden node has something to transmit. Note that this assumption does not hold in a general network, where a sensor node has transmitting neighbours and can, therefore, reduce its transmission attempts due to communication in the neighbourhood. Moreover, in this case, the average time spent in back-off is highly difficult to compute because of the possible continuous overlapping collisions that also happen due to the desynchronization effect.

Considering that the source nodes do not have transmitting neighbours and that they are hidden terminals among them, the conditional collision probability can be computed as shown in Eq. (15). Assuming that the maximum back-off is smaller than the data packet transmission time, the probability that a packet transmission fails is: (i) the probability that any other hidden terminal \( (n_h) \) has something to transmit (or it is, in fact, transmitting something) when the data packet transmission starts or (ii) it generates a packet in the time interval \( t_{col} \) after the start of the data packet transmission, where \( t_{col} = \frac{1}{r} - Br \). Refer to Fig. 3 where a schematic representation is provided.

\[
p = 1 - ((1 - \rho)e^{-\lambda t_{col}})^{n_h} \tag{15}
\]

In this specific case, since the transmitting nodes do not overhear any other transmission, the channel states probabilities of the source nodes are simply:

\[
\begin{align*}
    p_e &= 1 \\
    p_s &= 0 \\
    p_t &= 0
\end{align*} \tag{16}
\]

Finally, the successful transmission duration is, as shown in the previous case, \( T_s = (L_p + L)/r \). The rest of metrics are obtained using the equations in Section 4. Note that, without loss of generality, these metrics are computed for the transmitting nodes only, not the receiver.

4.3. Energy consumption

The total energy consumption of a sensor node can be divided in four parts: (i) the energy spent to transmit and (ii) receive messages, (iii) the energy wasted in overhearing, and (iv) the energy spent in duty cycle (sleeping and waking up in inactive periods):

\[
e = e_{tx} + e_{rx} + e_{ov} + e_{dc} \tag{17}
\]
Let \( N \) be the total number of messages a node sends during a time of observation \( T_{\text{obs}} \). It can be approximated as \( N \approx T_{\text{obs}}/(1 - P_3) \) when the queue size is small. Thus, the total energy spent to transmit \( N \) messages is computed following Eq. (18), taking into account the energy needed to transmit the long preamble and the data packet and the energy spent in idle mode during the empty slots of the back-off countdown (the busy periods will be included in the receiving and overhearing energy consumptions).

\[
e_{\text{tx}} = N \left( E_{\text{idle}}(p_B \sigma) + E_{\text{rx}} \left( \frac{p + L}{T} \right) \right) \quad (18)
\]

where \( E_{\text{idle}} \) and \( E_{\text{rx}} \) are the energy consumptions of being in idle and transmission modes respectively. Observe that Eq. (18) accounts for that messages that are successfully sent, discarded due to maximum retry limit and those that collide.

For Case 1 where all nodes are in coverage range, the energy consumptions receiving and in overhearing are shown in Eqs. (19) and (21). However, for Case 2, where source nodes do not have transmitting neighbours, the energy consumptions receiving and in overhearing of the source nodes are zero (\( e_{\text{rx}} = e_{\text{ov}} = 0 \)).

When all nodes are in coverage range it has been considered for simplicity that each node receives \( N_i \) packets destined to it, where \( N_i \) is the total number of messages a node successfully sends during a time \( T_{\text{obs}} \), i.e., \( N_i = T_{\text{obs}}/S/L \). Therefore, the total energy consumption to receive those messages is computed as the average energy to receive the preamble and the data packet for each received message:

\[
e_{\text{rx}} = N_i \left( e_p + E_{\text{rx}} \left( \frac{L}{T} \right) \right) \quad (19)
\]

where \( E_{\text{rx}} \) is the energy consumption of being in reception mode and the parameter \( e_p \) refers to the average energy spent receiving the long preamble. If a node has something to transmit it will be listening to the channel, receiving the entire long preamble of any other transmission in the medium. Otherwise, the node will be in duty cycle mode and, on average, it will wake up in the middle of the other’s long preamble transmissions, receiving on average half of the preamble transmission:

\[
e_p = E_{\text{rx}} \frac{L}{T} \left( \rho + \frac{1}{2}(1 - \rho) \right) \quad (20)
\]

Note that the colliding packets have not been considered in Eq. (19). It is assumed that the unsuccessful transmissions of a node collide with those that are destined to it and that the probability to collide with more than one packet can be neglected.

Similarly, the energy consumption due to overhearing is computed as the energy spent receiving the rest of messages. Assuming that each node sends and receives \( N_i \) messages, the number of successful transmissions that are overheard are \( (n - 2)N_i \). On the other hand, and assuming also that a collision occurs between 2 nodes, the number of observed collisions is \( (n - 2)((N - N_i)/2) \). Then, the total energy consumption in overhearing is computed as follows:

\[
e_{\text{ov}} = (n - 2) \left( N_i + \frac{(N - N_i)}{2} \right) \left( e_p + E_{\text{rx}} \left( \frac{L}{T} \right) \right) \quad (21)
\]

Finally, the time that a node remains inactive (\( T_{\text{inactive}} \)) it performs a low duty cycle operation, listening and sleeping according to the duty cycle as shown in Eq. (22).

\[
e_{\text{dc}} = T_{\text{active}} \left( E_{\text{idle}} \left( \frac{T_{\text{listen}}}{T_{\text{ci}}} \right) + E_{\text{sleep}} \left( \frac{T_{\text{sleep}}}{T_{\text{ci}}} \right) \right) \quad (22)
\]

where \( E_{\text{sleep}} \) is the energy consumption in sleep mode and \( T_{\text{ci}} \) is the check interval:

\[
T_{\text{ci}} = T_{\text{listen}} + T_{\text{sleep}} \quad (23)
\]

The time that a node remains inactive can be obtained subtracting the time in which the node is performing some kind of operation (\( T_{\text{active}} \)) to the total observation time: \( T_{\text{inactive}} = T_{\text{obs}} - T_{\text{active}} \). Observe that, \( T_{\text{active}} \) can be computed considering the time that a node is transmitting, receiving and in overhearing.

\[
T_{\text{active}} = T_{\text{tx}} + T_{\text{rx}} + T_{\text{ov}} \quad (24)
\]

Note that \( T_{\text{tx}}, T_{\text{rx}} \) and \( T_{\text{ov}} \) can be computed as \( e_{\text{tx}}, e_{\text{rx}} \) and \( e_{\text{ov}} \) respectively but accounting for the time instead of the energy:

\[
T_{\text{tx}} = N \left( p_B \sigma + \frac{L}{T} \right) \\
T_{\text{rx}} = N_i \left( \frac{1}{2}(\rho + \frac{1}{2}(1 - \rho)) + \frac{1}{T} \right) \\
T_{\text{ov}} = (n - 2) \left( N_i + \frac{(N - N_i)}{2} \right) \left( \frac{L}{T} \left( \rho + \frac{1}{2}(1 - \rho) \right) + \frac{1}{T} \right) \quad (25)
\]

The validation of the analytical model can be found in Sections 5.1 and 5.2 where the results from the analytical framework are compared to simulations in a single hop network and in the presence of hidden terminals without transmitting neighbours.

5. Quantitative analysis

This section presents the quantitative results of the hidden terminal problem in different case scenarios. Three reference scenarios are considered: a single hop network (for reference purposes), the \( n_h \)-hidden (where a set of transmitting hidden terminals send data to a receiver), a grid and a network formed by nodes located randomly in the area. The random scenario is evaluated also under more realistic assumptions like event-based traffic profiles, convergecast communication and retransmissions enabled.

The single hop, the \( n_h \)-hidden and grid scenarios allow to carefully evaluate the problem. For this reason, it is important to fix the specific topologies, with this goal the carrier sense threshold is considered equal to the reception threshold and only the path loss model has been used. The path loss exponent is considered to be equal 4 while the log-normal standard deviation of the shadowing propagation model (used in the random scenario) has been set to 7 dB [22]. The carrier sense threshold (used also in the random scenario) has been set 12 dB lower than the sensitivity.

Table 3 summarizes the considerations of the three basic scenarios (single hop, \( n_h \)-hidden and grid), the random scenario and the more realistic random scenario while Ta-

Please cite this article in press as: C. Cano et al., Quantitative analysis of the hidden terminal problem in preamble sampling WSNs, Ad Hoc Netw. (2011), doi:10.1016/jadhoc.2011.04.009
bles 1 and 2 in Section 3 show the hardware and MAC parameters considered. The queue length ($K$) has been set to 10 packets.

Table 3
Summary of considerations for each scenario. The abbreviations PL and SW refer to Path Loss and Shadowing respectively.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Basic</th>
<th>Random</th>
<th>Realistic Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic profile</td>
<td>Poisson</td>
<td>Poisson</td>
<td>Periodic and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>events</td>
</tr>
<tr>
<td>Communication model</td>
<td>Local</td>
<td>Local</td>
<td>Convergecast</td>
</tr>
<tr>
<td>Carrier sense</td>
<td>&lt;Sensitivity</td>
<td>&lt;Sensitivity</td>
<td>&lt;Sensitivity</td>
</tr>
<tr>
<td>Retx and ACKs</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

5.1. Single hop scenario

The first scenario is based on a single hop network formed by $n$ nodes transmitting data to a receiver. The set of nodes are located in the area assuring that all of them are inside the coverage range of the others. This scenario will be used as a reference of the WSNs performance when no hidden terminals are present. As all nodes are in coverage range no capture effect is possible. As stated in the simulator description, no capture effect is considered if collisions occur at the same instant (one propagation time after the initial reception is considered also the same instant). Simulation results of this scenario will show the mean value of each metric obtained from a simulation run with a simulation time equal to 100000 s.

Fig. 4. Performance results in the single hop scenario.
Fig. 5. 1, 2 and 3-hidden scenarios.

(a) Total throughput per node. The inset shows a zoom of the lower loaded region ($0 \leq \lambda \leq 1$ packets/s).

(b) Energy consumption/byte received

(c) Delay

(d) Conditional collision probability

Fig. 6. Performance results in the $n_h$-hidden scenarios.
The performance results are depicted in Fig. 4. It shows the results obtained using the simulator and the analytical model presented in Sections 3 and 4.1 respectively. As can be observed, the network saturation point is clearly seen in the figures, note that it appears at lower loads as the number of nodes increases. Observe, in Fig. 4a, that shows the achieved throughput as well as the offered load, how the network is able to support without losses a load of more than 1 packet/s with \( n = 5 \) but less than 0.5 packets/s with \( n = 15 \). The rest of metrics, energy consumption per byte successfully sent (Fig. 4b), delay (Fig. 4c) and conditional collision probability (Fig. 4d) show a similar tendency with the one observed in the throughput. Note also that the energy per byte clearly depends on the number of active nodes, more transmitting nodes implies a higher number of collisions and also an increased overhearing cost. Apart from the quantitative results, it is worth noting how the simulation and the analytical model closely match.

5.2. \( n_h \)-Hidden scenario

This scenario is formed by a receiver, located in the middle, and a variable number (from 2 to 4) of source nodes. All source nodes are hidden from each other and they do not have any transmitting neighbour (see Fig. 5). The term \( n_h \)-hidden refers to the case where there are \( n_h + 1 \) source nodes, therefore from each node point of view the number of hidden terminals is \( n_h \). As the distance from the different source nodes to the receiver is the same and the path loss model is used, no capture effect is possible (i.e., all packets that suffer a collision are lost). Simulation results of this scenario will also show the mean value of each metric obtained from a simulation run with a simulation time equal to 100000 s.

These scenarios are helpful in order to quantify the effect of hidden terminals when they do not reduce their transmission attempts because of communication in the neighbourhood. These scenarios can also be seen as a simplification of a random scenario at very low load conditions, when the effect of hidden terminals dominates among the effect caused by neighbours. Observe that this scenario obeys the assumptions stated in Section 4.2 where the analytical model in presence of hidden terminals is described.

Results are shown in Fig. 6. Note that although the number of hidden terminals is low, the network starts showing losses at an offered load equal to 0.4 packets/s per node, see Fig. 6a. In the same figure it can be observed that when the collision probability is higher than 50% (Fig. 6d), the throughput shows an inflection point and starts to decrease with the offered load until it reaches...
zero, meaning that the successive overlapping collisions prevent the successful reception of any packet at high loads. Regarding the energy consumption per byte (Fig. 6b), it decreases with the number of hidden terminals only due to the small number of collisions since the effect of overhearing is not present. Observe also that as there is no overhearing, the delay, shown in Fig. 6c, is the same for the different configurations. The conditional collision probability (Fig. 6d), as well as the other metrics, show the same tendency found in the throughput. Notice also in this scenario, the close match between the analytical model and the simulations.

As a conclusion of the results of this scenario, it can be said that the effect of hidden terminals is non-negligible even at the lower loads considered, but when the load is considerable, the network is unable to provide any service at all.

5.2.1. $n_h$-Hidden vs. single hop

An interesting result is to compare the performance obtained in the $n_h$-hidden scenarios with the single hop ones (all nodes inside the coverage range of each other) when the number of sources nodes is the same. This gives an idea of the performance decrease of having a hidden terminal instead of a neighbour. Note that it has been considered that channel errors occur only due to collisions since a noise-free channel has been assumed, therefore the distance between the transmitter and the receiver is not affecting the results.

Fig. 7 shows the throughput decrease comparing the one obtained in the single hop scenario for 2, 3 and 4 source nodes with the obtained in the different $n_h$-hidden scenarios. It can be observed that, even for the lower loads, the reduction is quite important. For instance, at 1 packet/s, the decrease is more than 10%, 20% and 30% with 1, 2 and

![Total throughput per node](image1)

![Energy consumption/byte received](image2)

![Delay](image3)

![Conditional collision probability](image4)

Fig. 10. Performance results in the grid scenario.
3 hidden terminals respectively. At higher loads, the decrease is equal to 100% since the throughput in the \( n_h \)-hidden scenarios tends to zero.

5.2.2. \( n_h \)-Hidden with capture

In order to evaluate how the capture effect can mitigate the hidden terminal problem, source node A of the scenario shown in Fig. 5 has been deliberately moved in order to be closer to the receiver. The differences among the received signal strengths are such that the packet error rate of a packet coming from node A that suffers a collision affecting the entire message (even if all of the \( n_h \) hidden nodes collide) is negligible.

The increase in the aggregated throughput of the \( n_h \)-hidden scenarios with capture is shown in Fig. 8. In this specific case, in which only the packets coming from node A can be captured, the gain is not very high (less than 10%) for offered loads lower than 1 packet/s due to the reduced hidden terminal transmission attempts. Thus, these hidden nodes will reduce their transmission attempts as they have to contend with their neighbours to access the channel, increasing the back-off duration accounting for the possible successful transmissions or collisions between their neighbours.

In this scenario, due to the equidistant distances among neighbours and the propagation model used, the capture effect is also not possible.

The results, compared to the ones obtained in the 3-hidden scenario, are depicted in Fig. 10. Related to the throughput (Fig. 10a), it can be seen that for loads lower than 4 packets/s, the results of the grid follow the same tendency than the results of the 3-hidden scenario. However, the throughput achieved is smaller due to the increased back-off duration since source nodes should wait until transmissions in their neighbourhood finish. For loads higher than 4 packets/s a higher throughput, if compared to the 3-hidden case, can be observed. The reason of this improvement is a reduction of the probability that a hidden terminal transmits during a given packet transmission caused by communication in its neighbourhood (it finds the medium busy). As previously seen in the other scenarios, the results of energy consumption, delay and conditional collision probability, depicted in Figs. 10b–d respectively, show a clear similarity with the tendency observed in the throughput.

As a concluding remark, similar to the 3-hidden scenario, losses are noticeable even at the lower loads considered. However, a slight improvement is found at high loads due to the reduced hidden terminal transmission attempts.

5.4. Random scenario

In this scenario a variable number of sensor nodes (100, 200 and 300) are randomly placed in a 500 × 500 m² area, resulting in three different densities (\( \bar{\gamma}_{100} = 4 \times 10^{-4} \), \( \bar{\gamma}_{200} = 8 \times 10^{-4} \) and \( \bar{\gamma}_{300} = 12 \times 10^{-4} \) nodes/m² respectively). In order to avoid routing effects, each node sends data to a randomly selected neighbour, see Fig. 11. Only metrics computed at inner nodes are analyzed, with the goal of avoiding border effects. In this scenario the mean and 95% confidence intervals of a set of 10 independent simulations (with nodes placed in different positions) will be provided. Each simulation run has a simulation time equal to 1000 s.

The aim of this scenario is to analyze how the performance of the network changes by increasing the average number of hidden terminals. However, as the number of hidden terminals per communication pair is modified by changing the density, the total neighbours per node also vary. This effect complicates the comparison among the different configurations as changes in the number of hidden terminals also imply a variation in the number of neighbouring nodes.

The number of neighbours per node and hidden terminals per communication pair can be approximated by considering the path loss propagation model (although the shadowing model is used in this scenario). The average neighbours (those node that can sense the target node transmissions, i.e. in the carrier sense range) per node is

![Fig. 11. Random scenario.](image-url)
obtained as: \( n = \gamma \pi r^2 - 1 \), where \( r = 98 \text{ m} \) given the transmission power, propagation model and carrier sense threshold considered. To compute the average number of hidden nodes, the exclusive area of two circles of the same radius \( r \) separated a distance \( d \) is:

\[
A_e = \pi r^2 - \left( 2r^2 \cos^{-1} \left( \frac{d}{2r} \right) - \frac{d}{2} \sqrt{4r^2 - d^2} \right)
\]  

(26)

The distance between the two centres that makes the inner area equal to the outer one is \( d = r/\sqrt{2} \), therefore the average number of hidden terminals (average number of nodes in the exclusive area) can be computed as \( n_e = \gamma A_e \).

Solving for the different densities considered, we have that there are on average 11 neighbours and 5 hidden terminals when there are 100 nodes in the network, 23 and 10 when there are 200 and 35 and 15 when 300 nodes are considered.

Fig. 12 shows the results with the capture effect activated (w capt.) and deactivated (w/o capt.). It can be observed that the throughput achieved (Fig. 12a) is significantly smaller than the offered load for the entire range of loads considered. Even for the lower loads (see Fig. 13) the percentage of lost throughput is significant. For instance, with 100 nodes without the capture effect enabled, this difference is approximately 30% for only 0.2 packets/s offered load and more than 80% for 1 packet/s (a bit less if the capture effect is activated). Observe, once again, the similarities of the tendency among the different metrics (energy per byte, delay and conditional collision probability, Figs. 12b–d) and the throughput. It is important to note the considerable improvement obtained in the different metrics, except the delay (as retransmissions are not considered), when the capture effect is activated, however, the low performance is maintained.
The configuration with 100 source nodes can be compared to the single hop case with \( n = 10 \) (Fig. 4) as the number of neighbours is approximately the same. However, in the random case the average number of hidden terminals per communication pair is 5 while in the single hop case there are no hidden terminals present. Observe that the maximum achievable throughput differs in 80 bits/s (approximately a 53% decrease) considering the capture effect case. Moreover, losses appear at offered loads lower than 0.1 packets/s while in the single hop case losses start at 1 packet/s. Therefore, having 5 hidden terminals (on average) affects the performance of the network noticeably, even for the lower offered loads.

As concluding remarks it should be pointed out that, as already seen in the previous scenarios, the effects of hidden terminals are important at high loads but they are also considerable at lower loads. Moreover, it has been shown that the capture effect provides considerable improvements when there are hidden terminals present although the global performance of the network is still significantly low.

5.5. Random scenario with event-based and periodic traffic profiles

While the rest of the scenarios evaluated in this work provide a simple framework to quantify the problem, the goal of this scenario is to evaluate the hidden terminal effect in a more realistic network. The scenario of interest is based on the previous random scenario but with the following extensions:

- Two traffic profiles are considered: periodic low rate messages and event-based data coming from event detection.
- The ACK mechanism and retransmissions are enabled.
- Convergecast communication is considered: sensor nodes send all their data to the sink.

It has been considered that sensor nodes send messages periodically to the sink, this traffic profile is based on a typical monitoring application. At each sensor node the time between periodic message generation is constant and it has been set equal to 1 h. Moreover, if sensor nodes detect an event they notify the sink immediately by sending an event-based message. The event radius is considered equal to 150 m, therefore nodes with a distance to the event centre smaller than this value will detect the event and will generate a message. Communication from the sensor nodes to the sink is based on the shortest path. Routes to the sink are precomputed in order to avoid routing messages to interfere with the simulation. The capture effect has been enabled, however if a packet transmission fails at the MAC layer (the ACK is not received) sensor nodes wait a random back-off and retry transmission given that the maximum retry limit (set to 5) has not been reached. The ACK length has been set equal to 5 bytes and it has been considered that a receiver waits 11\( \sigma \) to respond with an ACK after a data packet reception.

A 200-node network in which sensor nodes continuously send periodic messages is simulated. After 5400 s of the simulation start an event is generated in a specific position (see Fig. 14). In that moment, sensor nodes inside the radius of the event try to send a message to the sink (placed in the middle of the topology). Results of 4 independent simulations (with different placement of sensor nodes) with a duration of 6000 s are shown in Fig. 15. It shows the number of received and sent packets of each traffic profile. It can be observed that the periodic messages are nearly all received (only ‘Simulation 3’ and ‘Simulation 4’ show 2 and 4 periodic packets lost respectively).
while the event-based messages suffer from a higher loss rate. Although not all the messages must be received at sink to reliably detect an event [23], the reliability obtained is considerably low (a 32.3% on average). This low reliability in the event-based messages is caused by the increment of contention among those sensor nodes detecting the event, both, nodes in coverage range and hidden terminals.

The results of this scenario reflect for a more realistic case the conclusions obtained by the quantification analysis of the more controlled scenarios. An increase of the network load causes a performance reduction due to higher contention among neighbour sensor nodes but also due to the high vulnerability time in collisions among hidden terminals.

6. Effects of hidden terminals in improved preamble sampling techniques

From the first definition of preamble sampling [2,3], several extensions to improve its performance have been proposed. However, for simplification purposes, only the basic preamble sampling technique with CSMA medium access has been studied in this work. Thus, the goal of this section is to discuss how the hidden terminal problem can affect those improved preamble sampling techniques presented in the literature. The main preamble sampling improvements can be divided into three broad categories: (i) the division of the long preamble into series of small short packets, (ii) the reduction of the long preamble by remembering the receiver wake up time and (iii) the adap-
tation of the duty cycle (and thus, the preamble) to the traffic load.

The division into small and short packets of the long preamble (for instance, the approaches presented in [24,25,10]) allows a reduction of overhearing by including the destination ID of the data packet, therefore those nodes that do not need to receive the message can go to sleep. Moreover, some techniques leave idle spaces between the short packets to allow the transmission of an early ACK that stops the transmission of the preamble. Observe that although considerable benefits can be obtained in terms of energy consumption, the hidden terminal problem affects in a similar way as if the long preamble is transmitted. Some benefits can be obtained with the transmission of the early ACK since it reduces the size of the preamble and therefore the vulnerability time, however, collisions can still happen during the transmission of (on average) half of the preamble and the data packet as interfering nodes can be sleeping during the transmission of the early ACK.

The protocols that remember the wake up time of the receiver and send only the necessary preamble to cope with the clock drifts (some of the most well-known approaches are [26–28]) allow a notable reduction of the duration of the long preamble and thus, the vulnerability time. For these cases in which hidden terminals transmit to receivers that have different wake up times, the hidden terminals effects are, therefore, decreased if compared to the transmission of the entire long preamble. However, note that as all nodes (hidden terminals or not) that want to transmit to a common receiver will try transmission a few instants before it wakes up, the probability of collision increases. Moreover, if all nodes wake up at approximately the same time, as defined in [27], the probability of correctly receiving a packet in presence of hidden terminals decreases considerably.

The preamble sampling protocols that can, to some extent, lighten the effects of hidden terminals are those that are capable of adapting their behaviour to the traffic load (for example, different approaches of traffic load adaptability are the works presented in [25,29–31]). Some of them adapt the duty cycle, and therefore, the duration of the long preamble to the traffic load while others define hybrid protocols in which as soon as the load increases synchronous approaches are adopted. However, how to rapid and efficiently estimate the load of the network is a challenge in common low-capability sensor nodes, even more difficult when event-based traffic profiles are present as sporadic increases of the load can happen due to event detection at nearby sensor nodes.

There are also some approaches like the more bit of WiseMAC [26] or the piggybacking defined in [32] that allow to suppress the preamble for consecutive transmissions to the same recipient. These techniques allow to reduce the channel occupation duration of successful transmissions but they are not focused on addressing hidden terminal problems.

Finally, although not belonging to the preamble sampling MAC protocol category, asynchronous receiver-initiated approaches like the RI-MAC [33] and A-MAC protocols [34] are clear competitors of preamble sampling. In these approaches sensor nodes wake up periodically and inform (by sending a message) the others. A node with a packet to transmit remains awake until the reception of the receiver’s wake up notification and then, it transmits the message. These protocols have the disadvantage of consuming higher energy when the network load is low if compared to preamble sampling. In receiver-initiated protocols a notification is always sent at each wake up, however in preamble sampling sensor nodes have just to sample the channel during a short time interval. On the contrary, when the network load is high, receiver-initiated protocols perform better since the long preamble is not needed. When there are hidden terminal problems, these protocols, similarly to the protocols that remember the wake up time of the receivers, can achieve some improvements when the packets are directed to different receivers, with different wake up times. However, in presence of hidden terminals trying to transmit to the same receiver the problem is considerable since they will all try to start the transmission just after the reception of the notification. However, these approaches normally define that a receiver can notify transmitters the back-off to use after inferring a collision among them, thus alleviating the problem.

7. Conclusions

In this work, the effects of hidden terminals in preamble sampling WSNs have been studied. The performance of the network has been evaluated by means of a preamble sampling simulator and an analytical approach. The analytical framework presented in this work is the first approach to model the hidden terminal effects in preamble sampling WSNs considering the desynchronization among them and it has shown a good accuracy compared to simulation results.

Results for different kinds of scenarios and traffic profiles have shown that the hidden terminal problem is of high importance at medium to high loads, that can be caused for instance by event detection. It has been also illustrated that the capture effect cannot substantially improve the overall performance of the network.

These results should be considered by designers of MAC protocols in order to develop mechanisms to lighten those effects, specially if increments of the network load are expected in specific moments (for instance due to event detection or query dissemination) or regions (those nodes closer to sink transmit more data). From the results obtained, two main recommendations to MAC designers can be provided. The first one is the consideration of adaptive approaches capable of reducing the vulnerability time (for instance decreasing the long preamble transmission duration) as soon as the network load increases, thus mitigating the negative effects of hidden terminals from medium to high loads. Another recommendation that can be derived is the avoidance of too aggressive retransmissions. Although the effect of retransmissions is not deeply studied in this work, it can be devised that some retransmission techniques could lead to continuous and overlapping collisions due to the desynchronization among hidden terminals, making the network unusable.

Please cite this article in press as: C. Cano et al., Quantitative analysis of the hidden terminal problem in preamble sampling WSNs, Ad Hoc Netw. (2011), doi:10.1016/jadhoc.2011.04.009
References


Cristina Cano obtained the Telecommunications Engineering Degree at the Universitat Politecnica de Catalunya (UPC) in February 2006. Then, she received the TICMA (Information, Communication and Audiovisual Media Technologies) Master Degree from the Universitat Pompeu Fabra (UPF) in June 2007. Currently, she is working on her Ph.D. on Information and Communication Technologies at UPF. Her research interests include Wireless Networks, Wireless Sensor Networks, Quality of Service and MAC layer design.

Boris Bellalta received his MSc degree in Telecommunications from the Universitat Politecnica de Catalunya (UPC) in 2002 and the Ph.D. in Information and Digital Communications from the Universitat Pompeu Fabra (UPF) in 2007, where he combined the Ph.D. studies with a full-time assistant professor position. Since 2007, he is a post-doc researcher and full time lecturer at UPF. His main research interests are in the area of traffic engineering, queuing models and wireless communications.
Andrés Cisneros received his BSc in Telecommunications Engineering from the Universitat Pompeu Fabra (UPF) in 2009. Since 2008, he is an assistant researcher at Networking Technologies and Strategies (NeTS) research group at UPF. He has been involved in different research projects related to WLANs (EDCA optimization with TCP and VoIP traffic) and to WSNs (Testbed evaluation of MAC and routing protocols). He is currently pursuing for a MSc in Telecommunications Engineering from UPF.

Miquel Oliver is the head of the Networking Technologies and Strategies (NeTS) research group at UPF (Universitat Pompeu Fabra). He got a degree in Telecommunications Engineering (1994, UPC), a degree in Business Administration (2009, UOC) and a Ph.D. in Electrical Engineering (1999, UPC). He held a postdoctoral position at the Wireless Information Networks Laboratory (WINLAB) within Rutgers University (NJ, USA). He has been involved in several research projects related to wireless access network neutrality. His research topics have been evolving from radio resource management for cellular systems, quality of service in wireless data networks, IP mobility, radio access protocols including wireless sensor networks and distributed peer-to-peer protocols. ICT impact upon society as well as business models for incoming technologies and networks have been also researched during the last few years. Prof. Oliver has authored more than 40 papers in journals and international conferences, two patents and has been the director of five Ph.D. theses since 2006.