Pilot Signal Assisted Ultra Wideband Medium Access Control Algorithm for Wireless Sensor Networks

Kamran Ayub, Valerij Zagurskis, Professional member IEEE

Abstract— Due to low-power transmission and robustness to multipath fading, Ultra Wideband at Physical layer is an ideal platform for WSN applications, but for the effective use: compatibility with MAC Layer is a challenge. In this paper we have proposed a compatible MAC technique, which well fits with the ultra-wideband and works well for indoor WSN-applications. The aim of research is to enhance the network throughput and reliability by scheduling mechanism. Our algorithm proposes a relatively new concept of Pilot Signal assisted synchronization that uses intelligent scheduling and consumes very low power. The possible application areas are indoor short range dense multipath environments.

Keywords—Medium Access Control, Pilot signal assisted communication, UWB, Wireless Sensor Networks.

I. INTRODUCTION

Use of UWB at physical layer has splendid capabilities like, extended range, very low power consumption, transmission with very short pulse (<1ns) and spectral occupancy of over 500 MHz. These properties make it an ideal candidate for physical layer. Before we discuss the use of UWB for WSN, let’s see the issues with the existing WSN MACs.

The major issues with the WSN MACs are collisions and energy wastage. In case of UWB-based MAC algorithms, collision is due to weak bonding (synchronization issue) that also tends toward energy loss. Existing MAC algorithms has long acquisition time and complex structure so they don’t fit well for UWB-Wireless Sensor Networks. In our approach we have used Transmitted Reference/Delay Hopped Impulse Radio (TR/DH IR-UWB) synchronization technique that improved synchronizations and reduces probability of collision. Furthermore use of multi-phase structure brings intelligent scheduling capability.

Our whole algorithm works around the Pilot signal which is an (TR/DH IR-UWB) impulse; it provides synchronization as well as communication control, and saves energy because no channel estimation is required.

Initially (in phase 1) algorithm works on first come first serve basis; as this is simplest structure for low channel requests. Once Channel requests increase to the level where collision occurs, algorithm triggers to the phase 2. Here in this phase, synchronization and scheduling is planned prior to the communication and each node gets access as per data size (needed to be transport). At the end of each round, priority is recalculated for the new schedule. If channel access requests drop to zero, algorithm triggers back to phase 1.

The rest of the paper is organized as follows. Section 2 focuses on the Pilot Signal assisted Synchronization technique. In Section 3 we defined our proposed algorithm. Performance is evaluated in Section 4. We conclude our paper with simulation results in Section 5.

II. PILOT ASSISTED SYNCHRONIZATION

There are many conventional methods for UWB synchronization, but in case of WSN, it needs to be very accurate usually at the level of pulse duration (which is very small). Due to the long acquisition time under multiple access environments formal synchronization methods do not work well for UWB-WSN. To overcome this complexity we have used a TR/DH synchronization technique is used Fig.1: shows high level topology diagram of single hope WSN-Cluster.

Transmitted Reference is a very simple and light weight synchronization technique; it is basically the transmission and reception of multiple pulses (doublets) in groups Fig.2, where individual pulses are separated by specific time gaps knows to the receiver. On the receiver side
correlation is performed with the delay version of itself. This method has better SNR (signal to noise ratio) compared to conventional schemes see “Spread Spectrum communication” [1].

In proposed method for the disassociation (with the delayed signal), Delay Hope (DH) code is used. 2nd order Volterra model [2] is used for the DH code design. As per research [2], [3] throughput performance improves when Delay Hopped codes are used with Transmitted Reference. In the 2nd phase CH (Cluster Head) broadcasts TR/DH UWB Pilot to all active nodes; here Pilot signal contains epoch. The “epoch” serves as a reference point for time measurement. Signals (from cluster nodes) to CH are synchronized with periodic cell. This clears ambiguity for Measurement. Signals (to cluster head). Signals (to CH) provided basic information (required for prioritization and scheduling) size of data, time of birth etc. Fig.3.

### III. PROPOSED ALGORITHM

**Assumptions:** Duration of Signals is more than Tr (Resolution Time of receiver). Time duration is proportional to priority. Network is based on Single hop, and single cluster, with all nodes have different priorities. Cluster has already been set-upped. We have assumed that there is no hidden or expose node problem. Round Robin proportional to priority. Network is based on Single hop, (Resolution Time of receiver). Time duration is

### A. Algorithm

| Initialization | Initially all active nodes will set their priority timer Tp = Tmax (Maximum allowed time for a node) Tmax=Ntr×Tpac+3tr \{where, N= number of Nodes, n= Receivers’ resolution time, Tpac= Size of data frame, Tp= Priority Time\} Input N, Tr, etc. |

![Fig.2. TR/DH Pilot pulse structure](image)

**B. Time Diagram**

For the better understanding of the algorithm, we have tested the worst case scenarios, that exhibit algorithm
behavior in case of collision. Here TX=Transmit, TP=Priority, RX=Receive, SL=Sleep, P=Pilot and A=Active. Let’s assume, 3 nodes n1, n2, n3 are active nodes and have data to transmit. Initially they set TP=Tm, and their priorities are n3=1, n1=2, n2=3. All will Listen (Rx) during TP and will send Pilot signal followed by Data for Transmission. Initially all were in series and no Collision detected Fig.4(a).

2nd Round: At 27, two new nodes(n4, n5) entered, and sent Pilot signal which got collision with existing n2’s Tx. At this stage Phase 2 will be activated.

2nd Round (Ideal Case): Nodes will Tx during TP and will listen (Rx) same amount of time (as TP) & (after that) will go to Sleep (will remain in Sleep till have another data). All nodes get scheduled at the scheduled time; in this Round node n3 is not active as it has nothing to send, so no time was allocated to n3. Fig.4(b).

2nd Round (Tx outside the TP): Nodes are Tx-ing during TP and Listen (equivalent time), and then go to Sleep. n3 (assumption) could not complete Tx within its TP, and Tx-ing in vulnerable window, in fact it’s n2’s turn (n3’s Tx will be blocked). Fig.4(c).

2nd Round (Tx outside the TP)-Successful: Assume n4 could not complete its Tx within TP, will continue Tx in vulnerable window, as there was no other node scheduled for that time, it successfully completes its transmission(Tx) Fig.4(d).

We have used fixed transmitter power; similarly we have blocked the dynamic channel coding part.

A. Flow Chart

![Flow Chart](image)

IV. PERFORMANCE EVALUATION

We have used Mixim (Omnnet++) based simulation-platform, that gives UWB-physical layer model as well as media access layer[4], it well suits our situation (model is modified as per requirement). Major parameters we have used are traffic rate (λ) and N (network nodes in a cluster).

Performance was compared with two very close competitor’s “LEACH and U-Mac” protocols [5], [6]. U-Mac was simulated as per the model defined in [7].

We have set max and min time for generating a hello message (=10&1Sec) respectively, beside that MSI Threshold =15%, and interference threshold=50%, tx power and tx rate was adjusted as per 802.15.4a conditions. Pmax (Maximum power) was selected as 15dBm, N=25(nodes) structured under single-hop (for our algorithm case). We have focused on the simple case but in future we can implement this under multi-hop environment, & multiple types of traffic (as per algorithm assessment); common peer-to-peer and a sink-based traffic were selected [8].
(λ = from 0.25 calls/s to 5 call/s) is used. Table I summarizes the main simulation values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (No of nodes)</td>
<td>5-25</td>
</tr>
<tr>
<td>P (Transmission power)</td>
<td>-15dBm(39μW)</td>
</tr>
<tr>
<td>Traffic rate (Poisson)λ</td>
<td>0.5-5.0 calls/s</td>
</tr>
<tr>
<td>Payload size</td>
<td>160 bytes</td>
</tr>
<tr>
<td>Simulation area (los)</td>
<td>25X25</td>
</tr>
<tr>
<td>α (path loss exponent)</td>
<td>2.5</td>
</tr>
<tr>
<td>η (background noise + interference)</td>
<td>2.5 × 10^{-17}mW/Hz</td>
</tr>
<tr>
<td>γ (threshold)</td>
<td>5dB</td>
</tr>
<tr>
<td>T (pulse repetition time)</td>
<td>Modifiable</td>
</tr>
<tr>
<td>R (bit rate)</td>
<td>850kbps</td>
</tr>
</tbody>
</table>

The performance of our Algorithm was compared in terms of overall Network throughput, Packet delay and power consumption. We have selected two close competitors, LEACH and U-MAC [5], [6].

V. SIMULATION RESULTS AND SUMMARY

In Fig. 6(a), the overall achievable throughput in the network is shown. Although we have not measured multi-hop case based on single hop topology throughput is excellent and remains very stable. We can see that as the number of nodes increases, both UWB based MACs beat the traditional LEACH this is because of admission control. The performance of LEACH quickly drops as the number of interference sources increase; this is due to the weak synchronization.

Average power consumption: For this calculation we have used the chip-level energy-model described in [9], we compared among simulated protocols Fig. 6(b). It can be seen that our Algorithm and U-MAC has almost similar energy consumption at startup, but increasing the number of nodes, increases the power consumption (e.g. due to scheduling queue). In heavy network conditions (e.g. n = 25) the difference is quite large and the average power usage is significantly less than competitors Fig. 6(b).

The average Packet delay is in-fact end-end delay of a packet from source to destination. Results for the packet latency are shown in Fig. 6(c). The y-axis shows the average packet delay, whereas the x-axis is the number of nodes (N). As we can see that the average delay in case of our Algorithm is about 5 msec at start (when 5 nodes are active), and does not cross to 10msec even in busy situation. In general we can say that in our algorithm’s case nodes experience the lowest delay compared to other algorithms.

REFERENCES


