

Ad-hoc Extensions to the 802.15.3 MAC Protocol*

Samir Datta
WINLAB, Rutgers Univ.
sdatta@winlab.rutgers.edu

Ivan Seskar
WINLAB, Rutgers Univ.
seskar@winlab.rutgers.edu

Mustafa Demirhan
Intel Corp.
mustafa.demirhan@intel.com

Siun-Chuon Mau
WINLAB, Rutgers Univ.
siunmau@winlab.rutgers.edu

Dipankar Raychaudhuri
WINLAB, Rutgers Univ.
ray@winlab.rutgers.edu

Abstract

The paper describes the design and evaluation of Ad-Hoc extensions to the IEEE 802.15.3 Medium Access Control (MAC) layer for Wireless Personal Area Networks (WPANs). The proposed protocol allows communication between ad-hoc devices without the intervention of any central entity and, at the same time, ensures bounded delays for isochronous traffic. Ad-hoc communication is made possible without the hidden terminal problem. Features from both IEEE 802.15.3 and IEEE 802.11 standards are used - in particular the TDMA structure from 802.15.3 and RTS-CTS based contention from 802.11. The protocol includes certain other ingredients like a decentralized synchronization procedure using randomized beaconing, periodically interspersed contention periods, bit maps to convey reservation information and a mechanism to estimate and react to channel errors. The MAC has been simulated in ns-2 and simulation results are reported.

1. Introduction

This paper outlines the design and evaluation of a Medium Access Control (MAC) layer compatible with the IEEE 802.15.3 MAC [3]. The modified protocol allows for decentralized ad-hoc communication between devices, through contention. The hidden terminal problem is dealt with a mechanism similar to the RTS-CTS procedure. Additionally, it supports synchronous streams with packet deadlines and enables energy-constrained devices to sleep without the need for long waiting periods. There is also a mechanism to reduce control overhead. The proposed MAC adapts features from both IEEE 802.15.3 [3] and IEEE 802.11 [1] standards.

2. Outline of Ad-hoc mode for 802.15.3

The protocol uses Time Division Multiple Access (TDMA) across nodes without a central administering entity, with alternating periods of contention and contention free access. A randomized beaconing procedure, in which each device gets an equal (assuming uniform distribution of nodes) chance to transmit a beacon, is used in order to enable synchronization across the network.

During a contention period devices contend to reserve time for sending asynchronous data, similar to the contention in 802.11. However reservation is not for a time immediately after the negotiation but for a future slot in the contention free period. Data is sent during the contention free period in the agreed slots.

Contention for synchronous traffic occurs on a per flow rather than a per packet basis. The allocation is maintained until the stream is explicitly torn down. If there are persistent collisions with another stream there is provision for changing the reserved slots. Instead, if there are collisions with asynchronous traffic the node re-informs the other nodes about its reservation. Both these are done by monitoring the channel error rate in each slot using an Exponentially Weighted Moving Average (EWMA) procedure.

Notice that devices can sleep during the contention free periods if they are not sending or receiving data.

3. Related Work

The IEEE 802.11 [1] standard for Wireless LANs is well known. The IEEE 802.15 [2, 3, 4] series of standards for Wireless PANs have recently achieved prominence. These include IEEE 802.15.1 [2], the widely deployed Bluetooth standard and 802.15.4 [4], the low rate WPAN.

IEEE 802.15.3 [3] is the standard for high data rate WPANs. An 802.15.3 network consists of piconets managed by a device called the PicoNet Controller (PNC) (see

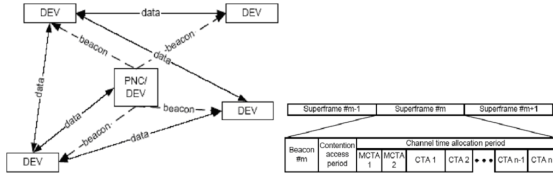


Figure 1. 802.15.3 Piconet elements and Superframe structure

Figure 1).

A piconet can associate with other piconets through a neighbor or child relationship. This provides extended coverage. The slot structure is TDMA with the PNC managing the slot allocation. Data communication is peer to peer. Slot lengths are variable and decided by the PNC (see Figure 1).

Ad-hoc communication, is difficult in the presence of just one PNC primarily because several reasons. Firstly, it is not possible for devices that are not PNC-capable to communicate in the absence of a PNC. Secondly, the devices may be spread over an area larger than reachable by one PNC. In 802.15.3 this would require the presence of multiple piconets (on the same channel). But bandwidth reuse is not possible with piconets linked by child or neighbor relationships because all devices share the same superframe, so that even if they are located in mutually non-interfering areas they cannot schedule simultaneous transmissions. Also, the hidden terminal problem is not dealt with.

The beaconing procedure in the proposed MAC is similar to the beaconing in the Independent Basic Service Set (IBSS) mode of the IEEE 802.11 standard [1]. Beaconing is a distributed procedure in which each member of the IBSS participates. Periodically (with a fixed time period) each node suspends its other activities and senses the channel. If it finds the channel not busy it sends a beacon, else it resumes the suspended activities. One assumption in the IBSS mode (and not in the proposed protocol) is that all the participating nodes are in hearing range of each other.

The concept of separating the contention and the contention free periods has been borrowed from satellite reservation systems[5]. In such systems each frame consists of a reservation period and a data period. Each node is allotted one slot per reservation period and if the node has a packet to send it uses the slot to reserve a data slot. The necessity of specifying the number of nodes a priori is a drawback in the context of Ad-hoc WPANs. Another disadvantage is that the uniqueness of reservation slots across the network precludes bandwidth reuse. Since some nodes may not hear each other, it is important that the length of the frame remains the same - this is difficult to enforce in a classical

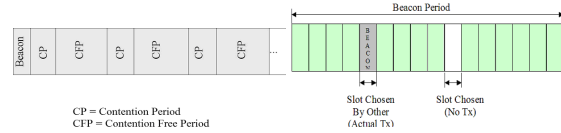


Figure 2. Structure of Superframe and Beacon

satellite reservation system without wastage of bandwidth. Therefore the RTS-CTS kind of reservation scheme, as outlined below, has been adopted.

4. Proposed Ad-hoc MAC

4.1. Global Synchronization of Ad-hoc devices

Channel time is divided into superframes that are shown in Figure 2. There is a TDMA structure across all devices. This information is maintained by a distributed beaconing procedure in which each device has a chance of sending out a beacon at the beginning of a superframe but only one device in a local region actually does. A device synchronizes its clock with the information contained in a heard beacon.

The beacon contains synchronization information in terms of a value signifying the time after which the TDMA part of the current superframe starts. The beacon period consists of a fixed number of slots each one of a duration equal to the time required to send a beacon. At the start of a beaconing period a node picks a random slot to send the beacon. The node transmits a beacon if and only if it does not receive one, before the slot arrives (see Figure 2).

Once the protocol has converged, each node knows the starting boundary of the beacon period and thus chooses a random slot for sending out a beacon. Since the wait is random, with high probability the minimum of the times chosen by all nodes is unique, only one node ends up sending the beacon. It is important to note that the random slot chosen by a node varies from one superframe to the next, and the source for generating random slots for a node is independent of the sources for other nodes. Also notice that the beacon contains the offset of the first contention period only. In the protocol, parameters like the number of slots per contention free period, the length of the slots, the number of contention free periods per superframe are all predetermined and globally known.

When a device wakes up and wants to join a network, it monitors a given channel for communication between other devices. If it does not hear anything for a long period (several multiples of its default superframe length) it sends out a beacon as in normal course of operation. If it hears a bea-

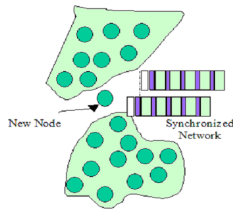


Figure 3. Merging of two internally synchronized networks

con, it synchronizes with that and if it hears a non-beacon packet it extends the wait before beaconing.

There is an apparent problem in the case when all the nodes are not in hearing range of each other. In this case there is a possibility of multiple beacons. If the node density is not very high, the chance for persistent collisions is low so eventually the nodes are able to synchronize. This is substantiated through simulations.

Another problem occurs when two networks, which are internally and separately synchronized, come into contact with each other, either through the introduction of a new node or through mobility (see Figure 3). The two networks need to agree upon a common schedule so both of them can function correctly. This is solved by assuming that all nodes in a synchronized network share a common key, which is distributed through beaconing. When a node hears a beacon from another node (possibly outside its own beacon period) it compares the received key with its own key and synchronizes with the superframe of the higher key. Synchronization implies that the node adjusts the start of its beaconing period with that of the other node. This prevents cycling between schedules by a node. Another problem arises when a synchronized network becomes partitioned into two parts due to mobility or when an intermediate node leaves, and then later merges together again. Partitions can become desynchronized even though they share a key. To avoid such a scenario the protocol may maintain two time stamped keys. When the duration of a key expires, the node sending the beacon updates the key to a randomly chosen value. When a node receives a beacon it replaces its keys with the two most recent ones among the ones it had and has received. If the expiry time of the keys is chosen to be significantly longer than the time for propagation of beacon information across the network then synchronized nodes always have at least one key in common. If it is kept shorter than time for clock drift, then even with partitioning and merging, the keys would be different.

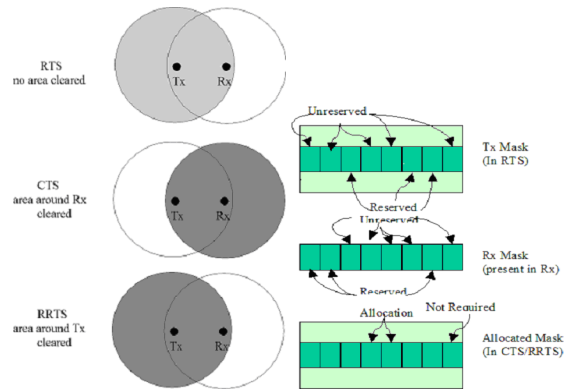


Figure 4. Modified RTS/CTS and Bit-maps

4.2. Contention

In order to reduce the packet transmission latency the superframe has a number of contention periods in each of which contention for Data packets to be transmitted in the next Contention Free Period, takes place (see Figure 2).

A time slot is reserved for a data packet through a modified RTS-CTS mechanism. A device (Tx) wanting to obtain a time slot sends out an RTS listing its free slots in the current and next superframe. The recipient (Rx), on getting the RTS and comparing with its own schedule, determines a free time slot and sends back a CTS containing this information. Each overhearer of the CTS (apart from Tx) marks the slot contained there as in use so that it doesn't use it itself. Potentially, this leaves some devices near Tx, that didn't hear the CTS, unaware of the scheduled slot (as this information was not contained in the RTS unlike in the usual 802.11). Tx then sends out another message called the Repeated RTS or RRTS containing this information (see Figure 4).

4.3. Use of Bit-maps to list free/allocated slots

Rather than specify one particular slot in the RTS, the Tx specifies a number of free slots, one of which may be chosen by the Rx for the data packet under consideration. This is done by sending a bit map in the RTS with 0s indicating which slots are free. Tx might send bit maps corresponding to several contention free periods depending on how sure it wants to be about successfully negotiating one slot. Each bit map is tagged with the position of the Contention Free Period in the superframe. For simplicity, reservations are restricted to be in the current or the next superframe only.

One drawback of the MAC as described so far, is that in the per packet contention for asynchronous traffic, there are at least 3 packets (viz. RTS/CTS/RRTS) generated for

each data packet. This would be fine for short control packets; however with a high bandwidth a significant fraction of the time is wasted on control overheads. This effect can be reduced by contending for several packets at once.

4.4. Provision for Synchronous Traffic

Per packet contention for synchronous traffic is wasteful, and would cause variable and unbounded delays. The alternative is per stream contention. The originating device for the flow traffic reserves the bandwidth over the whole superframe through an RTS-CTS-RRTS exchange carrying information about the superframe.

A node monitors the error rate of each slot in which it transmits synchronous packets. If the error rate is very high for a slot then a scheduling conflict with another stream is likely and the node changes its allocation by sending out an RTS with a cancel-mask and a new demand for the number of slots canceled, which is responded to by the receiver using the usual RTS-CTS-RRTS exchange.

Two nodes with conflicting flow allocations may try to cancel their allocations simultaneously. To prevent such race conditions, nodes check for the error rates of their slots at randomly distributed intervals. Instead of a conflict between two synchronous streams there may be conflict between synchronous and asynchronous traffic e.g. if the entering node does not know about an existing stream and schedules asynchronous traffic. Due to the aperiodic nature of the asynchronous traffic it is likely that there will not be persistent collisions in a slot but the error rate will nevertheless be high. To take care of this problem, the node with the flow initiates the RTS-CTS-RRTS transaction with no actual allocation taking place, just the current allocation for the stream being repeated. This would inform the errant node about the existing stream and would update its tables accordingly. This mechanism also helps in neutralizing the effects of lost CTS/RRTS packets for synchronous streams. Previously, the flow was exposed to interference from nodes, which had missed the flow CTS/RRTS packets but this effect is reduced due to their retransmission.

The error rate of a slot is maintained through an Exponentially Weighted Moving Average (EWMA) mechanism. Essentially, the error rate should decay exponentially if ACKs for packets are being received correctly while it should increase if the ACKs are not received. Thus for a parameter λ , which controls the rate of decay, the error rate ϵ of a slot is modified as follows: $\epsilon := \lambda\epsilon + (1 - \lambda)$, when a packet is transmitted and $\epsilon := \lambda\epsilon - \lambda(1 - \lambda)$, when the packet is successfully acknowledged.

One also needs to define two thresholds ϵ_1, ϵ_2 , such that if the error rate goes beyond them then corrective action should take place, i.e. if $\epsilon \geq \epsilon_1$, then a change in reservation should be initiated and if $\epsilon_1 > \epsilon \geq \epsilon_2$, then the RTS-CTS-

RRTS transaction should be repeated.

5. Determination of Parameters and Results

The proposed extensions above has been simulated in ns-2 based on a previous 802.15.3 model. Protocol and simulation parameters are specified in Tables 1 and 2 respectively.

Table 1. Protocol Parameters

Property	Value
Slot Duration	138 μ s
Bandwidth	100 Mbps
Superframe Length	30 ms
Transmission Range	25 m
Beacon Slot Length	17 μ m
#Beacon Slots/Beacon	16
#Contentions/Superframe	5
#Slots/Contention Free Period	32
Contention Duration	25% of Superframe (excluding beacon)
Max. #slots contended/RTS (Asynchronous Traffic)	8
Synchronous Traffic Parameters: $\lambda, \epsilon_1, \epsilon_2$	0.95, 0.5, 0.1

A simple radio model in which a node can receive a packet only within the transmission range is used. Thus there can be no interference from a node outside the transmission range while simultaneous transmissions inside the transmission range will always cause corrupted packets.

There is no mobility in the simulation. The static topology is generated by picking points with uniformly random coordinates in a rectangular grid.

Traffic is local i.e. it is between adjacent neighbors in the topology. UBR Traffic has exponentially distributed inter packet intervals and packet sizes. In the Synchronous Mode, end nodes for the CBR traffic have been chosen depending on the topology to keep the load evenly distributed.

5.1. Simulation Results

We list the parameters and the way we select their values. *Duration of Data Slot*: Determined so as to allow exactly one packet of maximum length to be transmitted and the ACK to be received. *Number of Data Slots per Contention Free Period*: Chosen to be 32. *Length of Contention Period*: The length of the contention period expressed as a percentage of the superframe (excluding the beaconing period) is determined by considering Figures 5-6. From the figures it is clear that Contention = 25% is a good choice. *Number of Contentions per Superframe* This parameter affects the performance of the MAC layer in the following

Table 2. Simulation Parameters

Property	Value	Variation
#Nodes	25	
Bandwidth	100 Mbps	
Simulation Area	90 × 90	
Mobility	None	
Traffic Type	UBR CBR + UBR	
Traffic Rate for CBR Traffic	4.9 Mbps/stream	4.9-7.8 Mbps/stream
Synchronous Traffic Over-Reservation	0%	5%-25%
Synchronous Traffic Deadline	75 ms	
Error Rate	2.5e-6	

ways: Increasing the number of contentions per superframe improves the performance of the MAC in terms of goodput, delay and packet loss (see Figure 7). This is because the fraction of beaconing period is reduced so a larger fraction of time is spent in contending for and sending data. Increasing the number of contentions per superframe increases the superframe length (the interval between two beaconing periods) leading to slower convergence of beaconing (see Figure 8). This leads us to choose number of contentions per superframe = 5 as doing well on both the previous factors. *Number of Beacon Slots:* From Figure 9 it is clear that the number of beacon slots = 16 gives an almost constant convergence time with varying load. Larger number of beacon slots would increase the fraction of time for beaconing without getting significant advantage. *Asynchronous Deadline:* from Figure 10 it is clear that an asynchronous deadline of 125ms gives high goodput over the infinite deadline case with only a small effect on the delay or packet loss. *Maximum number of asynchronous slots contended per RTS:* From Figure 11 it is clear that there isn't a large variation in the goodput/delay/packet loss based on this parameter, thus 8 is a reasonable choice. Figure 8 shows the goodput, delay and packet loss characteristics of a combination of synchronous and asynchronous traffic. Here the selected CBR sources for synchronous traffic, have one connection per node for a total of 12 connections. It is clear that goodput for synchronous traffic is virtually unaffected by the presence of asynchronous traffic. The behavior is similar for delay and packet loss. Conversely, asynchronous traffic is certainly affected by the amount of synchronous traffic present as evidenced by the curves for three different loadings of synchronous traffic and as expected the good-

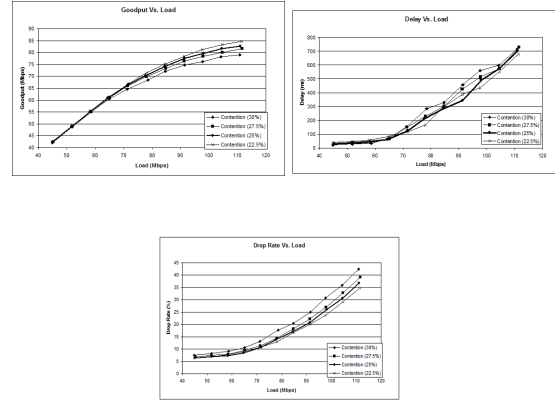


Figure 5. Goodput, Delay, Packet Loss Vs. Load param. on Contention (no deadline)

put decreases and the delay and packet loss increase with the increase in the synchronous traffic.

6. Conclusion

The paper describes a MAC protocol that combines features from 802.15.3 and 802.11. The protocol is decentralized, allows for ad-hoc communication between devices and avoids the hidden terminal problem. It permits synchronous traffic, without per packet contention and allows energy-constrained devices to go into a sleep mode when they are not communicating. Simulations show that the goodput, delay and packet losses are reasonable for random topologies. Synchronous traffic is virtually unaffected by the presence of asynchronous traffic in terms of these parameters. Rapid beaconing convergence to a stable TDMA mode, is observed.

References

- [1] IEEE. Std. 802.11 1st ed. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
- [2] IEEE. Std. 802.15.1-2002 ed. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs).
- [3] IEEE. Std. 802.15.3-2003 ed. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs).
- [4] IEEE. Std. 802.15.4-2003 ed. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs).
- [5] I. M. Jacobs, R. Binder, and E. V. Hoversten. General Purpose Packet Satellite Networks. *Proc. IEEE*, pages 1448–1468, 1978.

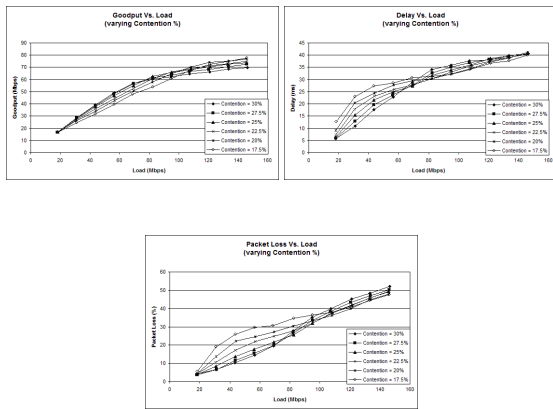


Figure 6. Goodput, Delay, Packet Loss Vs. Load param. on Contention (125ms deadline)

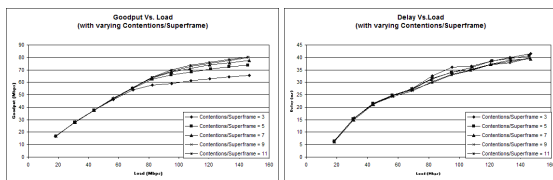


Figure 7. Goodput, Delay Vs. Load param. on Contention/Superframe (125ms deadline)

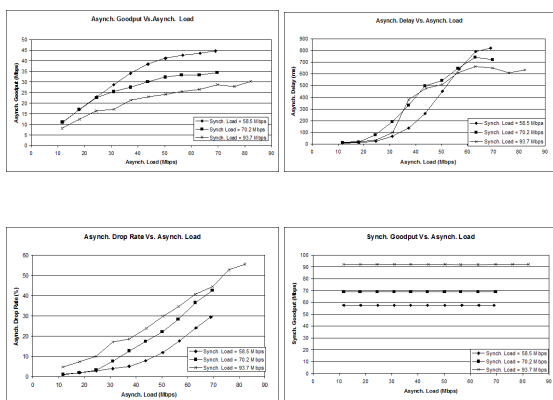


Figure 8. Asynch. Goodput, Delay/ Packet Loss and Synchron. Goodput Vs. Asynch. Load param. on Synchron. Load

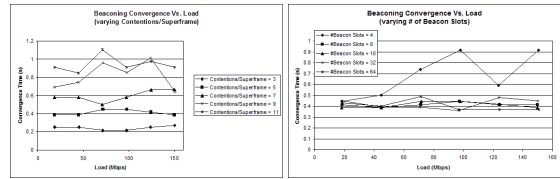


Figure 9. Beaconing Conv. Time Vs. Load param. on Contentions/Superframe, #Beacon Slots/Superframe

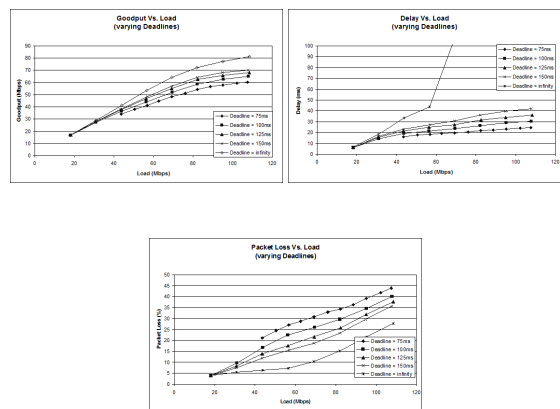


Figure 10. Goodput, Delay, Packet Loss Vs. Load param. on Asynch. Deadline

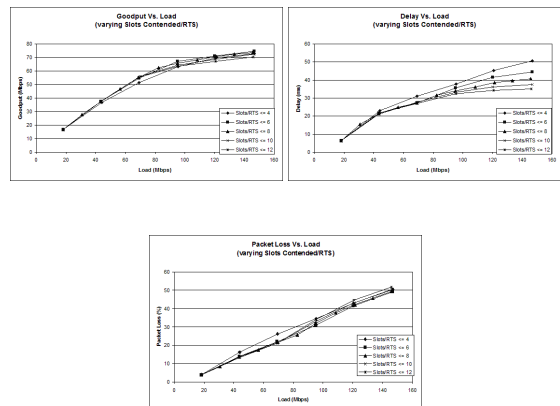


Figure 11. Goodput, Delay, Packet Loss Vs. Load param. on #Slots contended for/RTS