

Adaptive Channel Access Mechanism for Zigbee (IEEE 802.15.4)

Vaddina Prakash Rao, and Dimitri Marandin

Abstract—The IEEE 802.15.4 (also known as Zigbee) is a new wireless personal area network (PAN) standard designed for wireless monitoring and control applications. The Zigbee standard is based on CSMA-CA for contention based medium access. In this paper a study of the Adaptive backoff exponent (BE) management of CSMA-CA for 802.15.4 is presented. The BEs determine the number of backoff slots that the device shall wait before accessing the channel. The power consumption requirements make CSMA-CA use fewer BEs which increase the probability of devices choosing identical BEs and as a result, wait for the same number of backoff slots in some cases. This often leads to degradation of system performance at congestion scenarios, due to higher number of collisions. This paper addresses the problem by proposing an adaptive mechanism to the current implementation of the backoff exponent management, based on a decision criterion. As a result of the implementation, potential packet collisions are reduced. The results of NS-2 simulations are presented, indicating an overall improvement in network performance.

Index Terms—Zigbee, IEEE 802.15.4, Backoff Exponent, MAC, Performance Evaluation

I. INTRODUCTION

THE wireless market has been traditionally dominated by high end technologies, and so far Wireless Personal Area Networking (WPAN) products have not been able to make a significant impact on the market. While some technologies like the Bluetooth have been quite a success story, in the areas like computer peripherals, mobile devices, etc, they could not be expanded to the automation arena.

This led to the specification of the wireless low data rate personal area networking technology, Zigbee (IEEE 802.15.4), for the home/industrial automation. It has received a tremendous boosting among the industry leaders and critics have been quick enough to indicate that no less than 80 million Zigbee products will be shipped by the end of 2006[13].

ZIGBEE is primarily designed for the wide ranging automation applications and to replace the existing non-standard technologies. It currently operates in the 868MHz band at 20Kbps in Europe, 915MHz band at 40Kbps in the USA, and the 2.4GHz ISM bands Worldwide at 250Kbps. Some of its primary features are:

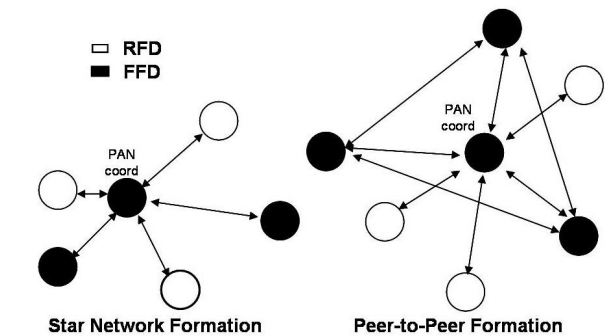


Fig. 1. Topologies

- Standard-based wireless technology
- Interoperability and worldwide usability
- Low data rates
- Ultra low power consumption
- Very small protocol stack
- Support for small up to excessively large networks
- Simple design
- Security and
- Reliability

In this paper we present a brief study of the CSMA-CA mechanism used in IEEE 802.15.4, with emphasis on the current implementation of the BE management which result in frequent packet collisions and degradation in system performance. We later move on to provide an algorithm called the Adaptive Backoff Exponent (ABE) which compliments the existing implementation with an improved and adaptive approach towards the problem source and reduces the probability of devices choosing identical number of backoff slots at collision rates, thus improving the systems performance considerably at these rates. NS-2[12] simulation results are provided to validate our claim of better performance.

II. ZIGBEE OVERVIEW

The IEEE 802.15.4 standard can be defined with a set of primitives. Primitives are services of each layer built over the services offered by the next lower layer[1]. These services are offered to the next higher layer or sublayer. There are 14 physical and 35 MAC layer primitives supported by Zigbee.

The low rate WPAN supports two types of topologies (Figure 1). They can form a star topology where the nodes can only talk to the coordinator and also a peer-to-peer topology where capable network nodes can route data. Several peer-to-peer networks can work together to form a mesh or cluster tree

Manuscript received October 10, 2006 and revised November 16, 2006. This is an extended version of the paper presented during NEW2AN conference, Saint Petersburg, May 2006.

Vaddina Prakash Rao is with Rohde & Schwarz Co. KG, Munich, Germany (e-mail: Prakash-Rao.Vaddina@rohde-schwarz.com)

Dimitri Marandin is with Technische Universität Dresden, Chair for Telecommunications, Dresden, Germany, (e-mail: marandin@ifn.et.tu-dresden.de)

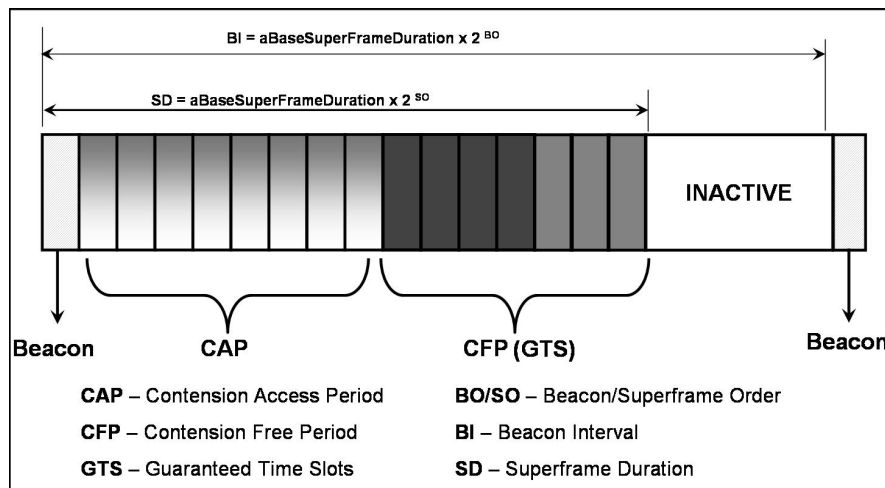


Fig. 2. The Superframe Structure

topologies. In this paper a star network topology is considered. Any FFD within a star network formation can either start a PAN by choosing a unique PAN identifier or request to join an already existing one.

Based on the number of primitives supported by these nodes two types of devices are defined: the Full Function Device (FFD) and the Reduced Function Device (RFD). A FFD is a device that supports all the primitives, whereas the RFD is designed to support a subset of them[2]. Therefore, a FFD is capable of acting as a coordinator, or as a network node routing data to peer nodes or as a simple network node communicating only with the coordinator, whereas, the RFD is only capable of being an end node with communication to the coordinator. A FFD, functioning as a coordinator, acts as the principle controller of a network, and may provide services like synchronization to the devices under its network span. A network can choose to either synchronize with the coordinator (hence synchronizing with one another) or remain ignorant of its neighbors. Network synchronization can be achieved by periodic broadcast messages from the coordinator called, Beacons. Thus, in a beacon-enabled mode, all nodes receive synchronization information from the coordinator, whereas when operating in a nonbeacon-enabled mode, the coordinator will cease to issue beacons and the nodes wishing to transmit control/data messages, do so with CSMA-CA contention resolution.

In the beacon-enabled mode, the frame structure is referred to as the “Superframe”. It is the time period between two consecutive beacons and its length is represented by “Beacon Interval (BI)”. A superframe can consist of an active period and an optional inactive period (Figure 2). During the inactive period the coordinator will not interact with its PAN and will enter into a power-save mode. In the active period of the superframe the coordinator will communicate with its nodes by transmitting beacons. The length of the beacon interval is defined by the parameter “Beacon Order (BO)” as follows:

$$BI = SF_{min} \times 2^{BO}, \text{ where } 0 \leq BO \leq 14 \quad (1)$$

The length of the active period of the superframe, called the “Superframe Duration (SD)”, is determined by the variable “Superframe Order (SO)”, as follows:

$$SD = SF_{min} \times 2^{SO}, \text{ where } 0 \leq SO \leq BO \leq 14 \quad (2)$$

The value SF_{min} represents the minimum length of the superframe ($aBaseSuperFrameDuration$ in the standard). The IEEE 802.15.4 standard sets this duration to 960 symbols (a symbol corresponds to 4 bits in the 2.4 GHz and 1 bit in 868/915MHz).

When the BO and SO are equal there shall be no inactive portion. For $BO=15$, the coordinator will not transmit a beacon and the superframe order parameter is ignored. If $SO=15$ the superframe will not be active after the beacon.

The superframe duration is subdivided into 16 equal time slots. The first slot of all superframes shall always be a beacon. Following the beacon, starts the first time slot for devices to compete with each other to access the network. This is the “Contention Access Period (CAP)” (Figure 2). The WPAN also has provision to support network access free bandwidth access without any contention. This is represented by the “Guaranteed Time Slots (GTS)”. Any node that is allotted a GTS shall transmit its data without any contention from its peers, in the slot. The number of GTS’s in one superframe can vary anywhere between 0-7 time slots, and a node can be allocated more than one GTS. The coordinator at the start of every superframe issues a beacon describing this structure of the superframe. On receiving the beacon, the network nodes synchronize by extracting the details like CAP and the GTS start and end times, pending receive data, etc.

The IEEE 802.15.4 compliant devices use CSMA-CA to access the media for their data transmissions. The backoff time in this algorithm is implemented in units of time called ‘backoff slots’ where one backoff period is equal to $aUnitBackoffPeriod$ (= 20 Symbols, for IEEE 802.15.4). Before trying to sense a channel, a device shall wait for a specified random number of backoff slots within the interval $[0, (2^{BE} - 1)]$ defined by the

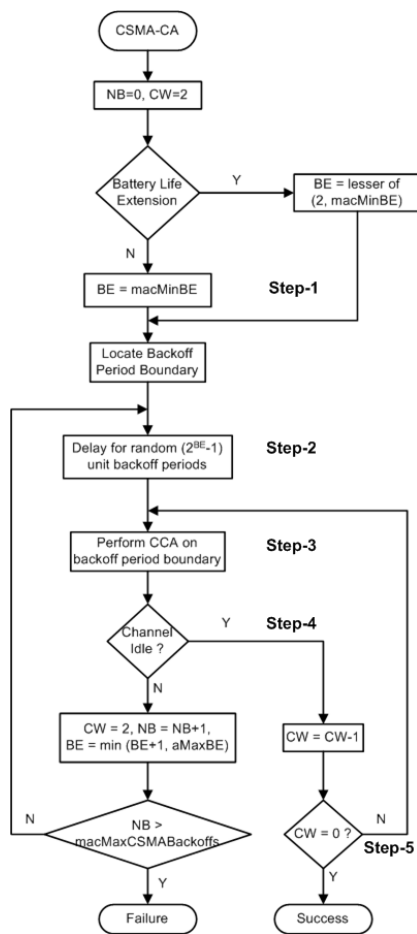


Fig. 3. CSMA-CA

variable BE . Each device uses a minimum BE value before the start of a new transmission and increments it after every failure to access the channel. BE determines the number of backoff slots the device shall wait before trying to access the channel.

Zigbee networks supports three types of data transmissions: direct, indirect and the guaranteed time slots. All nodes (including the coordinator) supporting direct transmission compete with each other using CSMA-CA to transmit the data, whereas in an indirect transmission, supported only by the coordinator, the messages are added to a list of pending transactions, which can be extracted at the discretion of the nodes. The guaranteed slots are the time slots of the superframe which are reserved to be used by a node.

A study of the system performance at collision scenarios reveals an exponential increase in the number of packet drops, for higher data rate operation. The drop in system performance can be attributed to the numerous packets with poor link quality. Link quality is an attribute to characterize the quality of each incoming packet. This measurement is implemented as Signal-to-Noise Ratio (SNR) estimation in most cases. Usually, poor link quality is a direct consequence of the hidden node problem. However, in the current scenario it has been assumed that there is no hidden node problem.

Because of the power consumption constraints, the BE of devices is never allowed to vary higher than the maximum value. As a result, devices often choose identical number of backoff slots and detect an idle channel. Thus, the devices on choosing an identical number of backoff slots transmit data/MAC-command packets without being aware that another node has also detected an idle channel. This lead to frequent confrontations among nodes which result in collisions, packet drops and as a consequence retransmissions, which affects the network throughput. It is this problem that is addressed in this paper.

III. CSMA-CA

The CSMA-CA algorithm is used to access the channel to transmit data or MAC command frames. The devices at first need to ensure that the channel is free before attempting to transmit their data. This is achieved by checking the idleness of the channel at random intervals of time. These random time units for which the devices shall wait for data transmission are called the *backoff* slots. One backoff slot is equal to $aUnitBackoffPeriod$ ($=20$ Symbols, for IEEE 802.15.4). Based on whether beacons are used/not used, the CSMA-CA will choose either a slotted or unslotted procedure. The unslotted version is also used in cases when the beacons could not be detected in the PAN. In the slotted mechanism, the backoff period boundaries of every device in the PAN will be aligned with the start of the beacon transmission. Hence all the devices will have their backoff slots aligned with each other. Where as in the unslotted mechanism there is no such restriction. In our simulation scenario only the slotted mechanism is considered.

Each device shall maintain three variables for each transmission attempt: Number of Backoffs(NB), Contention Window(CW) and Backoff Exponent(BE).

- NB : It is the number, the CSMA-CA algorithm was required to backoff while attempting the current transmission; this value shall be initialized to 0 before each new transmission attempt.
- CW : It is the contention window length, defining the number of backoff slots that need to be clear of channel activity before the transmission can commence; this value shall be initialized to 2 before each transmission attempt and reset to 2 when the channel is assessed to be busy.
- BE : It is the variable that determines the number of backoff slots a device shall wait before attempting to assess a channel's status. The number of backoff slots that a device shall wait is chosen randomly in the range of 0 to $(2^{BE} - 1)$.

The following steps are executed to access the channel (Figure 3):

- 1) *STEP-1*: The MAC sublayer shall initialize NB (to 0), CW (to 2) and BE (to $macMinBE=3$), and then locate the boundary of the next backoff period. The choice of the BE is also controlled by a variable called, "Battery Life Extension". It hinders the devices from using higher BE s, and thus effectively prevent them going into longer wait cycles. Therefore, in the standard CSMA-CA, when

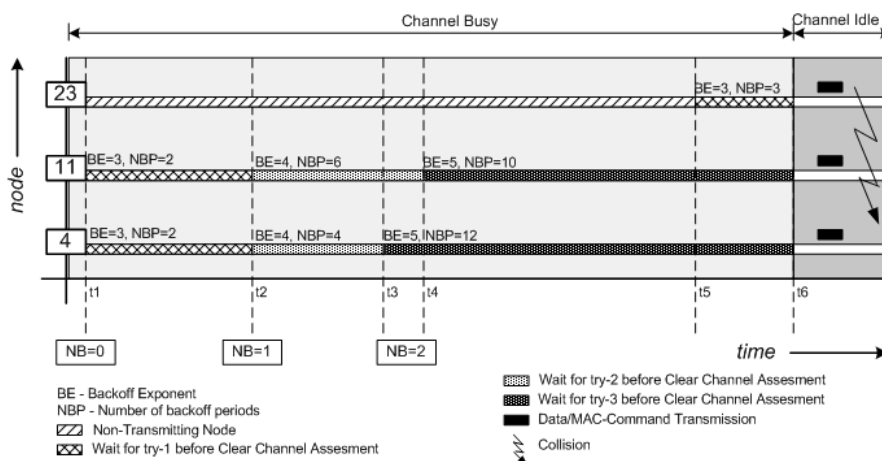


Fig. 4. Example collision scenario

this variable is set, the maximum value of BE can be only 2.

- 2) *STEP-2*: The MAC sublayer shall wait for a random number of complete backoff slots in the range of 0 and $(2^{BE} - 1)$.
- 3) *STEP-3*: After the completion of the backoff slots, the MAC shall request the PHY to perform a Clear Channel Assessment (CCA).
- 4) *STEP-4*: If the channel is assessed to be busy the MAC sublayer shall increment both NB and BE by one. It is ensured that BE is never greater than the maximum allowed value ($aMaxBE=5$). CW is also reset to 2. If the value of NB is less than or equal to the maximum allowed backoff attempts ($macMaxCSMABackoffs=4$), the mechanism shall proceed with *STEP-2* again. If not, the CSMA mechanism shall terminate with a Channel Access Failure status.
- 5) *STEP-5*: If the channel is determined to be idle, the mechanism shall ensure that CW is equal to zero. If not, it shall go back to *STEP-3* to the Clear Channel Assessment phase. If the contention window is 0, the MAC shall start transmission immediately.

IV. THE ADAPTIVE BACKOFF MECHANISM

A. Problem Statement

As already been discussed in the introduction, there is an exponential increase in the number of packet drops at high data rates. Zigbee devices are severely constrained in terms of power. Hence every effort is made to save as much energy as possible. During channel accesses the CSMA-CA algorithm is allowed to use only a very small range of backoff exponents ($macMinBE$ to $aMaxBE$), where the minimum BE , a device can support, is indicated by $macMinBE=3$ and the maximum by $aMaxBE=5$. Since the variable BE determines the number of backoff slots the devices shall wait to access the channel, the higher the value of BE , the longer the device will spend trying to access the channel in some cases. The longer wait adds up to the power consumption of the device. Therefore, a lower BE range is desired which will ensure that the devices

will never spend too much time waiting for channel access, which it is not even sure of. This often allows two or more devices ending up using the same number of backoff slots. As a result they detect an idle channel simultaneously and proceed with their transmissions which results in frequent collisions. An example scenario is described in the following paragraphs:

Let us assume a large set of devices are present in a star topology with a PAN coordinator and the others acting as end nodes. And a subset of these devices has data to be transmitted. Of all these transmitting devices, let us consider three nodes labeled 4, 11 and 23 for a brief study. At time t_1 , node-4 and node-11 have data to be transmitted and set their BE to the minimum value ($macMinBE$) of 3. Then they shall determine the number of backoff slots in the range $[0, (2^3 - 1)]$. Assume the devices choose to use 2 backoff slots. After the completion of 2 backoff slots they shall try accessing the channel's status. Other devices seem to be transmitting in this time and the channel is busy. So during this try they find the channel is occupied. Therefore, they increment their BE by 1. And again at time t_2 , they determine the number of backoff slots they need to wait in the range $[0, (2^4 - 1)]$. Let us assume the nodes 4 and 11 this time choose an unequal number of backoff slots of 4 and 6 respectively (Figure 4).

They wait for their number of backoff slots and node-4 exhausts its backoff slots first (Figure 4). Again let us assume a busy channel and node-4 shall increment its BE to 5, at time t_3 . Similarly node-11 after completion of its 6 backoff slots finds the channel is busy. So it increments its BE to 5 at t_4 . Now they shall use the backoff slots in the range $[0, (2^5 - 1)]$. Let us assume this time the nodes 4 and 11 use backoff slots of 12 and 10 respectively. This implies at any point of time (say t_5) when node-4 has completed 9 backoff slots, node-11 shall have completed 7. Let us again assume at the same time (t_5) node-23 is ready for its transmission and shall start off with a minimum backoff exponent of 3, which might lead it to use 3 backoff slots. Hence all the three devices, 4, 11 and 23 have 3 backoff slots to be completed, before they try to access the channel. And let us finally assume an idle medium

after t_6 , when the nodes complete their backoff slots. Hence all the three devices shall detect an idle medium at the end of their backoff slots, and shall attempt to transmit their data which would result in a collision.

The seriousness of the problem is clearly evident when many nodes transmitting at higher data rates are considered. The probability of at least two devices using the same number of backoff slots is increased. And the problem is magnified several folds when the current backoff exponent management is considered. Thus the odds of packet collisions are high at higher data rates which impact the throughput of network at these rates adversely. The insufficient range of backoff exponents from $3(\text{macMinBE})$ to $5(\text{aMaxBE})$, is the primary cause to blame for the poor performance at these rates. However, allowing the devices to use higher backoff exponents can impact the energy performance of the whole PAN as devices might increment their BE to use significantly higher backoff slots to sense the channel.

Therefore, we propose an algorithm which not only supports higher backoff exponents but also provides nearly same or better energy consumption while providing considerable improvement in the system's throughput performance.

B. Adaptive Backoff Exponent (ABE) Algorithm

The adaptive backoff exponent (ABE) algorithm is primarily based on three important principles. Firstly, it is the idea of providing a higher range of backoff exponents to the devices, to reduce the probability of devices choosing the same number of backoff slots to sense the channel. Secondly is to do away with a constant minimum backoff exponent (macMinBE) value as used in the standard CSMA-CA. In our algorithm, the minimum backoff exponent is variable; hence devices are not likely to start off with the same backoff exponent when they wish to start a data transmission. And thirdly, it is the way the minimum backoff exponent (macMinBE) is used. The algorithm implements a variable macMinBE , and the variation factor is chosen to be related to each node's contribution to the network traffic. Only devices that are involved in a transmission are taken into consideration. And devices that are not transmitting do not come under the purview of the algorithm.

As it can be observed, the algorithm does not interfere with CSMA-CA, but compliments it with an improved backoff exponent management. According to this algorithm, all devices that are contributing more to the network traffic are slapped with higher macMinBE 's, and devices which contribute less to the network congestion will use lower minimum backoff exponents. Therefore, devices with higher macMinBE are likely to wait longer than devices with lower macMinBE . At regular intervals (called Analyzing Cycles) the coordinator decides if it needs to implement a change in the macMinBE of the transmitting nodes. During the period of the analyzing cycle the coordinator shall keep track of each node's contribution to the network traffic, which shall be its decision criteria. If it finds an uneven distribution in network traffic contribution, it shall implement a change in the macMinBE

of the involved nodes. Also the devices are now allowed to use a higher range of backoff exponent ($\text{aMaxBE} = 7$). At the beginning of the transmission the devices shall start off with a macMinBE of 3. However, as time progresses and the node in question contributes unevenly to the network traffic, its macMinBE can be either decremented or incremented. Hence in the next transmission it shall use the new macMinBE . A detailed explanation of the algorithm follows. A star network formation with each transmission directed to the coordinator is assumed.

The algorithm is implemented in three phases.

- Analyzing phase
- Decision phase
- Implementation phase

The analyzing and the decision phases are implemented at the coordinator while the implementation phase is carried out at the end nodes involved in data transmissions.

1) *The Analyzing Phase:* The analyzing phase is repeated after every analyzing cycle. The analyzing cycle is the time during which the coordinator observes the contribution of each node to the network traffic. It is a variable that can be used to fine tune the algorithm to produce the best results. In the current simulations the analyzing cycle is taken as 3 Beacon Intervals.

At the initiation of data transmissions the devices shall start off by setting their macMinBE to 3 as specified in the standard CSMA-CA for 802.15.4. After the initialization, the devices shall continue with their data transmissions. During this period the coordinator sums the number of packets contributed by each node until it reaches the end of the analyzing cycle.

2) *The Decision Phase:* The decision phase follows the analyzing phase. Thus, at the end of the analyzing phase, the coordinator has enough information on each node's contribution to the network traffic. It shall now proceed to see if it is the time to apply a change in the macMinBE . If yes, it shall next decide which nodes should increment their macMinBE and which nodes should decrement it.

Before the steps involved in the Analyzing and Decision phases are explained, a brief introduction of the involved parameters is given.

- 1) *Group-1:* It is a group of devices/nodes whose macMinBE is to be decreased by 1. These are the nodes which contribute less to the network traffic compared to the other nodes.
- 2) *Group-2:* It comprises of devices/nodes whose macMinBE is to be increased by 1. These are the nodes which contribute more in comparison to the other nodes. When the number of transmitted packets (a measure of traffic) of Group-2 nodes differ from Group-1 nodes by at least PKT_DIFFERENCE number of packets, the change in the BE is performed. Otherwise, the algorithm proceeds to the next analyzing cycle.
- 3) *PKT_DIFFERENCE:* It is the variable which is the deciding factor for the application of the decision algorithm. If the difference between first node of

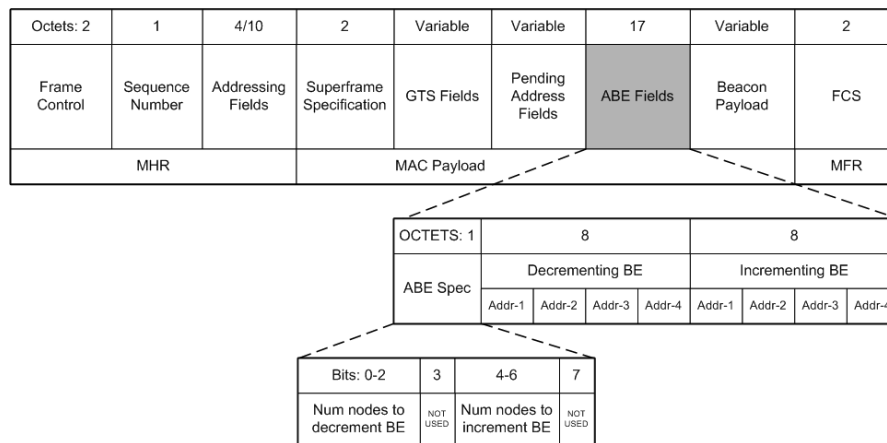


Fig. 5. Beacon Frame Format with ABE

Group-2 and the first node of Group-1 is more than *PKT_DIFFERENCE* number of packets, the decision algorithm is applied. If not, the node shall proceed with its normal beacon operation (Figure 6). To apply the decision algorithm further, the coordinator will indicate its decision in the beacon payload and transmits it. Our implementation of the algorithm uses a value of 5.

- 4) *numBeaconIntervals*: The number of beacon intervals which comprises one analyzing cycle. It is the time period after which the need for an adjustment of *macMinBE* is analyzed. This is determined as a number of beacon intervals after which this analysis is to be conducted. In our implementation of the algorithm it is initialized to a value of 3.
- 5) *numReceivedPkts[i]*: The array used to maintain the count of the number of packets contributed by each transmitting source node-*i* to the network traffic.
- 6) *nodeIndices[i]*: The array holding the node indexes of the sources (*i*) which are transmitting to the coordinator. The order of arrangement of the nodes in this array and their contributions indicated by the array *numReceivedPkts[i]* is always matched. Arranging the contributions array in any order should make a corresponding adjustment to this array.
- 7) *ABE Specification Fields*: The specification field of the Beacon Payload with 1 octet in size indicate the number of nodes whose *macMinBE* needs to be decremented and the number of nodes whose *macMinBE* is incremented. See Figure 5.
- 8) *addrList[8]*: The list of short addresses (2 octets) of devices whose *macMinBE* is incremented or decremented. The first four node addresses indicate nodes whose *macMinBE* is decreased by 1, while the end 4 node addresses indicate nodes whose *macMinBE* is increased by 1. See Figure 5. The index of the array indicates the number of nodes to which the algorithm can be applied in one cycle (discussed later).

The analyzing phase and the decision phase can be broadly explained with the following steps (Figure 6). As a first step, the variables *PKT_DIFFERENCE* and *numBeaconIntervals*

are initialized respectively as indicated in their definitions above. And the variable, *aMaxBE* is assigned the value 7. Note that the values are not absolute and the variables are meant to be tuned to produce efficient results. The indicated values have been used in the simulations that follow.

These steps are conducted at the coordinator:

- 1) *STEP-1*: Firstly the coordinator waits for a packet reception. If a packet is received, proceed to *STEP-2*.
- 2) *STEP-2*: Check if the received packet is a data packet. If not, go back to *STEP-1*. If yes, follow *STEP-3*.
- 3) *STEP-3*: Increment the packet count for the source node of the packet. Check if the analyzing cycle is complete. If not, go back to *STEP-1* else follow the next step *STEP-4*.
- 4) *STEP-4*: Arrange the nodes in order of the number of packets contributed by each node, so that the least contributor stays in the first place and the major contributor is placed at last.
- 5) *STEP-5*: Check if the number of contributors is more than 8. If yes, go to *STEP-6*. If the contributors are less than 8, go to *STEP-7*.
- 6) *STEP-6*: The first four nodes of the contributors list form Group-1. And the last 4 nodes come under Group-2.
- 7) *STEP-7*: The first half of the nodes form Group-1 and the second half the Group-2. Check if the number of contributors is even. If odd, arrange the middle node to either Group-1 or Group-2. If the difference in the packet contribution of this node to the packet contribution of the first node is greater than *PKT_DIFFERENCE*, the node is listed under Group-2 else it comes in Group-1. Figure 7 describes the division of the flows into groups in more detail.
- 8) *STEP-8*: Check if the difference between the first nodes of Group-2 and Group-1 is more than *PKT_DIFFERENCE*. If yes, include Group-1 nodes in the decrement fields of the beacon payload and Group-2 nodes in the increment fields. Similarly, include the number of nodes in Group-1 and Group-2 in the specification fields. If the difference in the first nodes is not significant, set the specification fields to 0. This indicates

small traffic rates and hence indicates little variation in traffic contribution by nodes. Therefore, the algorithm proceeds with the normal operation.

Having done the above steps, the arrays, numReceivedPkts[i] and nodeIndices[i] are reset to prepare them for the next analyzing cycle.

3) *The Implementation Phase:* The coordinator has taken its decision and has indicated it in the beacon fields, and the beacon is transmitted. Now the node shall act according to the coordinator's indication. The following steps specific to ABE implementation are followed at every node that receives the beacon. These steps are indicated in Figure 8.

- 1) *STEP-1:* Upon beacon reception, the node extracts the ABE Beacon Payload Specification fields.
- 2) *STEP-2:* See if the Beacon Payload Specification field is set to 0. If yes, continue with normal operation. If no, extract the number of decrements and number of increments.
- 3) *STEP-3:* Check if its address matches to the address list in the first four fields of the *addrList*. If yes, decrement its *macMinBE* by 1, and check if it is equal to 0. If yes, set it to 1. If its address is not listed in the first four address list, go to STEP-4.
- 4) *STEP-4:* Check if its address matches to the address list in the last four fields of the *addrList*. If yes, increment its *macMinBE* by 1. If its address is not listed, continue with the normal beacon operation.

V. BEACON MODIFICATION

The algorithm utilizes additional space within the beacon payload to convey the coordinator's decision on the *macMinBE* adjustment. Hence additional space is required in the MAC frame to indicate the number of incrementing/decrementing nodes and their respective node addresses. The algorithm uses 1 octet to indicate the number of increments and decrements and a maximum of 16 octets to indicate the short addresses of the devices (1 short address is 2 octets long) (Figure 5). Therefore, depending on the number of decrementing and incrementing nodes, a maximum of 17 octets are used for this mechanism. Since every beacon is transmitted in the first slot of the superframe, independent of the contention access and the contention free periods, there is no effect on the bandwidth utilization.

The algorithm, however, cannot be implemented under all conditions. The slot duration of the superframe should be big enough to accommodate a beacon along with the ABE fields. To find out the threshold value of SO , below which the implementation of ABE is not possible under any circumstances, a formula to determine the slot duration is derived:

$$\begin{aligned}
 T_{slot} &= \frac{SD}{N} \\
 &= \frac{SF_{min} \times 2^{SO}}{N} \\
 &= \frac{ST_{min} \times N \times 2^{SO}}{N}
 \end{aligned}$$

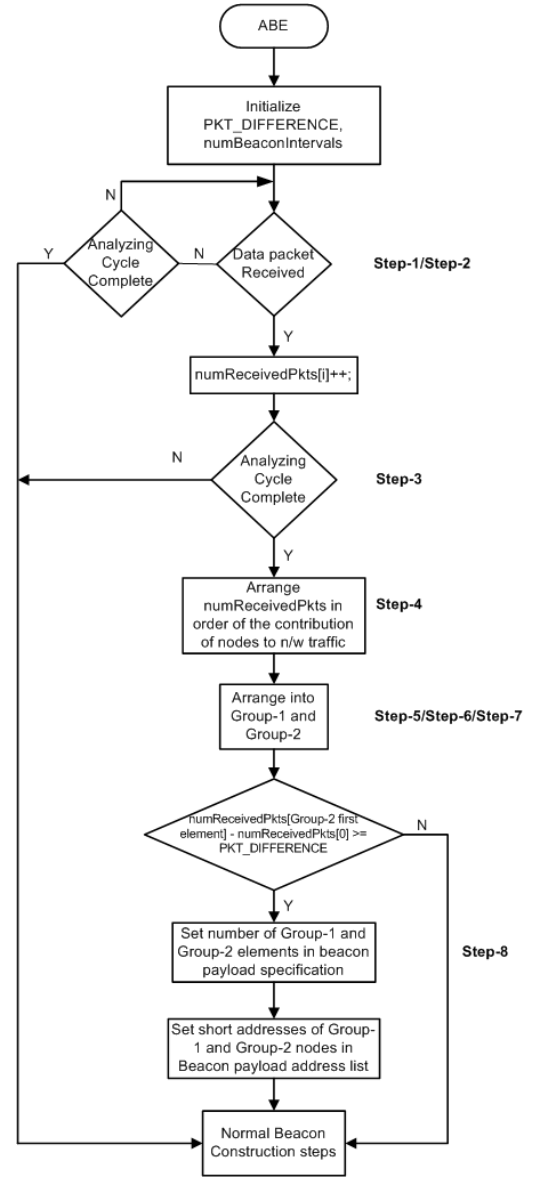


Fig. 6. ABE Analysis and Decision Phases

$$= ST_{min} \times 2^{SO} \text{ Symbols} \quad (3)$$

, where

T_{slot} : length of each superframe slot (in symbols)

SD : length of the superframe (in symbols)

SF_{min} : minimum length of the superframe (The standard defines it as *aBaseSuperframeDuration*)

ST_{min} : minimum length of the superframe slot (Standard value *aBaseSlotDuration*=60 Symbols)

N : number of slots in the superframe (Standard value *aNumSuperframeSlots*=16)

Using Figure 5, we can calculate that including the implementation of the algorithm, the beacon has a minimum size of $(2+1+10+2+17+2)=34$ bytes. It is noted that this calculation assumes that no node has a GTS allocation and no pending address fields are present. Therefore, the first slot of the

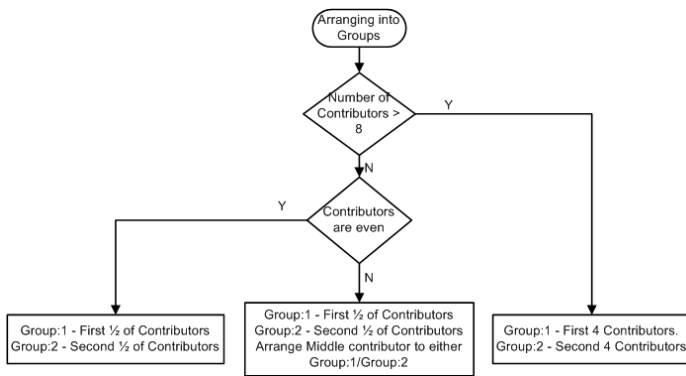


Fig. 7. Arrangement into Groups

superframe where the beacon is transmitted should be able to accommodate at least (34+PHY Header) bytes in order to support the implementation of ABE. Also a second restriction regarding the maximum MAC service data unit (SDU) applies. The maximum length of the MAC SDU that the PHY can accept is 127 bytes. Therefore all additions to the algorithm should be within this range.

With reference to the above, calculating the slot length for $SO=1, 2,$ and 3 in bytes for the $868/915\text{MHz}$ where 1 Symbol is represented by 1 bit, we get:

$$T_{slot} |_{SO=1, 868/915\text{MHz}} = 15 \text{ bytes} \quad (4)$$

$$T_{slot} |_{SO=2, 868/915\text{MHz}} = 30 \text{ bytes} \quad (5)$$

$$T_{slot} |_{SO=3, 868/915\text{MHz}} = 60 \text{ bytes} \quad (6)$$

Observing the above details it is concluded that the algorithm cannot be successfully applied when used for $SO \leq 2$, since it does not satisfy the minimum slot duration required to transmit a beacon. However, for $SO \geq 3$, an ABE implementation is possible, even though it cannot guarantee the use of all or partial use of GTS slots and pending address fields.

VI. PERFORMANCE ANALYSIS

Several metrics can be used to grade the performance of a technology against the elements of wireless networking. Some of these metrics have been carefully chosen to give an idea of behavior and the reliability of Zigbee networks. The analysis focuses only on data transmission, and hence the metrics are applied only to the data packets. This analysis is focused on four metrics as defined here:

- 1) *Network Throughput R*: It is a measure of the amount of data transmitted from the source to the destination in a unit period of time (second). The throughput of the entire network is the summary of the throughput of all nodes involved in data transmissions.
- 2) *End-to-End Delay D*: It is the average of the time taken for all data packets to reach the destination node successfully.
- 3) *Delivery Ratio r*: It is the ratio of all the transmitted packets that are successfully received by their destination nodes. A successful transmission is one where a

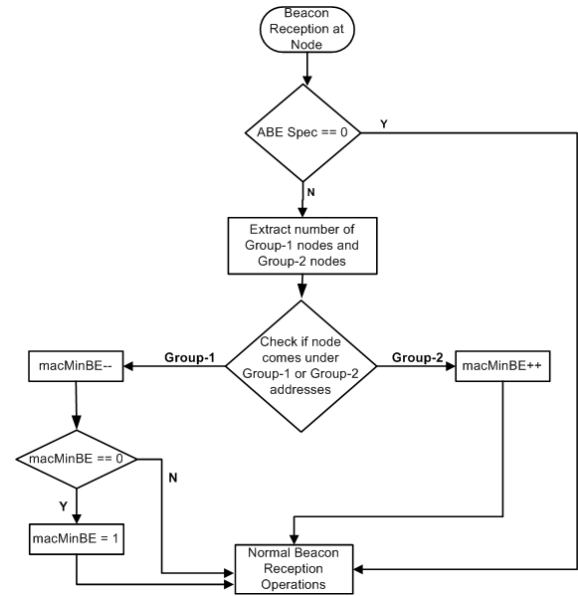


Fig. 8. Implementation Phase

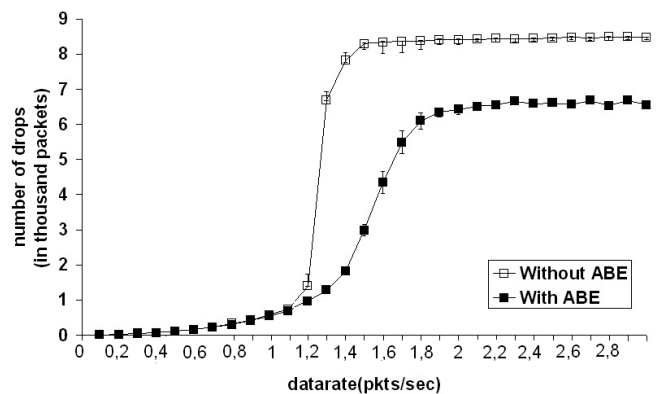
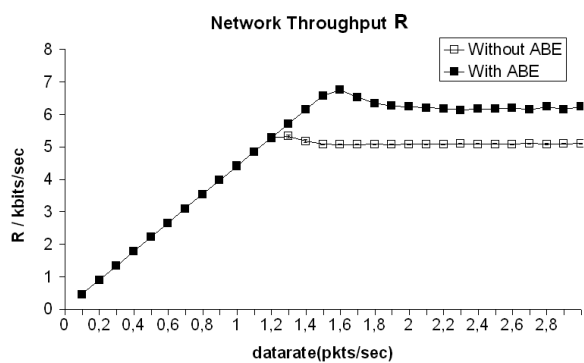
data packet transmitted from the source is successfully received by its destination, irrespective of the number of drops and retransmissions of the data packet.

- 4) *Energy Consumption E*: It is the average of the percent of the initial energy consumed by each node.

The simulations are conducted on a star network topology, with 15 nodes and 8 CBR traffic flows. In our simulations only the end nodes are allowed to transmit data to the coordinator (using direct transmission) while the coordinator shall have no data to the nodes (no indirect transmission). Each data packet has a length of 70 bytes. A Drop Tail queue is used. The nodes are all placed at a distance of 10m from the coordinator. Two-Ray ground propagation model with an OmniAntenna is used. All simulations are conducted with the Network Simulator-2 using the software modules provided by [4].

The graphs in Figure 9 highlight a comparative analysis of the two methods of backoff management. They indicate the improvement brought about by the application of the Adaptive Backoff Exponent algorithm. The performance metrics are measured as a function of the traffic load, generated by each node. The results are produced with 95% confidence levels, to express the accuracy of the simulations.

With reference to the graphs, it can be said that usage of the Adaptive Backoff Exponent algorithm effectively moves the congestion area to higher data rates. The areas, where extreme numbers of collisions are detected, result in performance degradation of the network. It is these areas which mark the beginning of network congestion. At traffic rates below the congestion areas, the performance of the system without ABE is similar to that of the system applying ABE. At lower rates the number of nodes contending for channel access at any given time is smaller than the number of nodes accessing it at higher rates. At higher traffic rates, more number of nodes try to access the channel and the insufficient distribution of



nodes with significant difference in contribution to network load is selected and their *BEs* are changed while the rest of the devices shall not be affected. Although the algorithm does not affect the other nodes in the first analyzing cycle, they cannot remain immune to its effects. In the first cycle, the nodes which had contributed more were punished with higher *BE* values. The second group of nodes are allowed to decrease their *BE*. As a result of this imbalance, the devices with higher *BEs* shall contribute less in the next analyzing cycle. This allows the nodes not affected by the algorithm in the first cycle to be affected in the next cycle.

The main part of the algorithm is to select eight nodes with the widest variation in traffic contribution. However, when the node density is higher, the current implementation of the algorithm may not be very effective. But it is to be noted that this implementation can be easily extended to support higher number of nodes. But when the number of nodes to be addressed within the beacon payload is too high, then support of extra nodes can be implemented at the expense of other features like GTS. While some applications do not need GTS allocations, some do. Therefore, an efficient application of ABE depends purely on the type of application and the type of scenario.

It is assumed in this implementation that the coordinator can analyze the traffic contribution from every node. Since this contribution is measured at the coordinator by summing the number of packets received from each node in the analyzing cycle.

The implementation of ABE is primarily restricted because of two reasons: Firstly, due to GTS allocations and secondly, because of pending address fields in a two way transmission (direct and indirect). In a two way communicating network the nodes transmit to and receive from the coordinator. If the coordinator has data to be transmitted to a node, it should indicate the node address in the pending address fields of the beacon and transmit it. The node on receiving the beacon extracts the pending address fields and if it finds its address listed, requests the coordinator for its data. These indications in the beacon of all the pending data occupies large amount of the beacon payload. The matter is even complicated when the PAN contains large number of nodes, since the possibility of several nodes expecting data from the coordinator is high.

A PAN operating at low superframe orders (say 3 at 868MHz/915Mhz) has a slot duration of 60 bytes, and the minimum beacon duration including the ABE implementation is 34 bytes. Even though a successful implementation of the algorithm is possible, allocating GTS slots and/or pending address fields may not be always possible. A definitive statement cannot be made because the number of fields required depends on the network scenario. Therefore, in a scenario where, let us say, there are 4 GTS allocations and no pending data at the coordinator, can be successfully implemented. Whereas, in a scenario where there are 5 GTS allocations and 4 nodes need to extract data from the coordinator, the resultant cannot fit in the slot. Therefore, it is realized that a free implementation of GTS and pending address fields may not be always possible

under these cases.

The ABE in the current form chooses to analyze only eight nodes of the PAN, in one analyzing cycle. Even though this implementation can support large PANs, a fair analysis of all the nodes may not be guaranteed. In these cases, extending the analysis of the number of nodes in one analyzing cycle is necessary. The algorithm can be readily extended to support more number of nodes. But again, the implementation of the GTS and pending address fields, when operating the PAN with small SO, poses restrictions.

Another problem, when operating at short superframe durations, using GTS and pending fields, is the size of the beacon, which is limited to 127 bytes. Allocations in the beacon cannot exceed this value. Therefore, an implementation of ABE can again be restricted.

Thus, it is concluded that the implementation of the algorithm can be successfully implemented under all conditions yet limited in certain ways only to be decided by the application scenario. However, when operating with short superframe duration it is advised that no GTS be supported and data transmission from the coordinator to the nodes be avoided.

VII. CONCLUSIONS AND FUTURE WORK

Zigbee is a promising new PAN technology. With its features of reliable short range communications at low data rates and ultra low power consumption, it has created a market for itself. However, minor disadvantages do not allow it to perform at its ultimate capability. One such problem, the inefficient backoff management at higher rates, is addressed here, and a modification to the Adaptive Backoff Exponent is proposed. The simulation results indicated the improvement that can be obtained by applying ABE.

The ABE is implemented by making three primary changes to the existing backoff exponent management. Using a higher value of "maximum BE (*aMaxBE*)", making the minimum BE(*macMinBE*) variable and finally using an adaptive mechanism to maintain these values. The algorithm is simple, easy to implement, and effective. Its strength lies in the fact that it can be fine tuned to suit to the application.

The analyze and decision phases of the algorithm can also be used for an efficient active/inactive period management based on the traffic rate to achieve a better system performance. It can be applied to check the network load and change the superframe duration appropriately so as to vary the active and inactive periods of the superframe to suit the requirements of the network under the working traffic conditions. Modifying the algorithm to support the analysis of more than eight nodes in one analyzing cycle and to simulate a large PAN with multiple nodes, with varying factors like shorter and longer superframe slots, support to GTS and/or data transmission from the coordinator to the node, can be a good issue for further investigation. Observing the adaptability of the algorithm under varying traffic or network conditions can also be an interesting topic.

REFERENCES

- [1] IEEE computer society, *Wireless MAC and PHY Specifications For LR-WPANS*, IEEE, 2003.
- [2] Jianliang Zheng and Myung J. Lee, *Will IEEE 802.15.4 Make Ubiquitous Networking a Reality?: A Discussion on a Potential Low Power Low Bit Rate Standard*, IEEE Communications magazine, pp. 140-146, June 2004.
- [3] Gang Lu, Bhaskar Krishnamachari, Cauligi S. Raghavendra, "Performance Evaluation of the IEEE 802.15.4 MAC for Low-Rate Low-Power Wireless Networks," IEEE, pp. 701-706, 2004.
- [4] <http://ees2cy.engr.cuny.cuny.edu/zheng/pub/>, "Zigbee Software Modules," City College of New York.
- [5] <http://www.energizer.com/>, "Energizer Engineering DataSheet," Energizer Battery Co.
- [6] Chris Evans-Pughe, "Bzzzz .. Is the ZigBee wireless standard, promoted by an alliance of 25 firms, a big threat to Bluetooth?," IEEE Review, March 2003.
- [7] "What exactly is ... ZigBee?," IEE Communications Engineer, August/September 2004.
- [8] Jeff Grammer, CEO, Ember Inc, "Zigbee Starts To Buzz," IEE Review, November 2004.
- [9] Sinem Coleri Ergen, "ZigBee/IEEE 802.15.4 Summary," September 10, 2004.
- [10] "ZigBee. Wireless Technology Made Simple," RF & WIRELESS, May, 2005.
- [11] Pete Cross, "Zeroing in on ZigBee," Circuit Cellar, Issue 176, March 2005.
- [12] <http://www.isi.edu/nsnam/ns/>, "The network simulator (V2)"
- [13] ABI Research, "The ZigBee Wave: Some Hesitate, Some Plunge In," ABI Research, September 2004.
- [14] Patricia McDermott-Wells, "What is Bluetooth," IEEE, pp. 33-35, December 2004/January 2005.
- [15] Y. Liu, M. Lee, and T.Saadawi, "A Bluetooth Scatternet Route structure for Multi-hop Ad Hoc Networks," IEEE JSAC, vol. 21, no. 2, pp. 229-239, Feb 2003.
- [16] Joseph M. Kahn, and John R. Barry, "Wireless Infrared Communications," IEEE, Vol. 85, NO. 2, pp. 265-298, February 1997.
- [17] ZENSYS Inc., "Article: Z-Wave, The Wireless Control Language," www.zen-sys.com.
- [18] Mikhail Galeev, "Article: Home networking with Zigbee," www.embedded.com, April 2004.
- [19] Kevin Fall, Kannan Varadhan, "The NS Manual," VINT Project, May 17, 2005.
- [20] Eitan Altman, Tania Jimenez, "NS Simulator for beginners," Lecture Notes, Univ. de Los Andes, Sophia-Antipolis, December 4, 2003.
- [21] Qingshan Shan, Ying Liu, Gareth Prosser, David Brown, "Wireless Intelligent Sensor Networks for Refrigerated Vehicle," IEEE 6th CAS Symposium on Emerging Technologies, May 2004.
- [22] Kwyro Lee, "MICROS: An Experimental Coin-sized, Low Cost, Low Power CMOS ZigBee-like Radio at 2.4 GHz as an Ubiquitous Network Node," 24th International Conference On Microelectronics (MIEL 2004, VOL 2, Serbia And Montenegro, 16-19 MAY, 2004.
- [23] Nick Baker, "ZigBee and Bluetooth, Strengths and Weaknesses for Industrial Applications," IEE Computing and Control Engineering, April/May, 2004.
- [24] David Steed, Jr., MaxStream, "The Power of Sensitivity Improving range with receiver sensitivity,".



Vaddina Prakash Rao, was born in the southern indian state of Andhra Pradesh in 1980. He received his Bachelor's in Electrical Engineering from the Osmania University in the year 2002. After that he moved to Germany, to the Technische Universität Dresden(TUD) to pursue a Masters degree in Electrical Technology, in December 2005. The work on "Adaptive Backoff Exponent" is mainly based on his thesis work at TUD. He currently works as a Software Developer at Rohde & Schwarz, Munich, Germany.



Dimitri Marandin, was born in Moscow, Russia in 1978. He received his diploma in Computer Engineering from Institute of Information Technologies at Moscow State University of Railway Engineering(MIIT) in June 2000. Following work experience with the Russian Scientific Research and Design Institute of Informatization, Automation & Communication (VNIIS MPS-Russia) in Moscow in 2001/2002, he has joined the Chair for Telecommunications at Technische Universität Dresden(TUD) in May 2001. His research interests

include sensor networks, ad-hoc networks and wireless networks.