

A new binary conflict resolution-based MAC protocol for impulse-based UWB ad hoc networks

Ioannis Broustis^{1*,†}, Mart Molle¹, Srikanth Krishnamurthy¹, Michalis Faloutsos¹
and Jeffrey Foerster²

¹Department of Computer Science and Engineering, University of California, Riverside, CA 92521, U.S.A.

²Intel Corporation, Hillsboro, OR, U.S.A.

Summary

Ultra wide band (UWB) technology offers a promising high capacity solution for short-range wireless ad hoc networks, as in home networks or in wearable ad hoc networks. In this paper, we propose a novel multi-band MAC protocol for use in small ad hoc networks that deploy an underlying UWB based physical layer. In our approach, we divide the available UWB bandwidth into multiple simultaneously usable bands. In the absence of a sophisticated equalizer, the size of a slot for transmitting a UWB pulse is typically dictated by the *delay spread* of the channel. Therefore, using a wider frequency band to shorten the transmission time for each pulse does not increase the data rate in proportion to the available bandwidth. A multi-band approach that uses a plurality of bands that adhere to FCC specifications, with slightly elongated pulse durations, provides a solution that can effectively utilize the UWB spectrum. Our approach is based on the idea of conflict resolution using binary ‘something’/‘nothing’ feedback, which has not been widely studied in wireless and specifically in UWB networks. Our protocol unites binary conflict resolution and multi-band utilization to effectively utilize the available bandwidth. To ensure that our proposed approach is tightly knit with the underlying physical layer, we discuss physical-layer dependencies and the conformance to FCC-imposed emission limits. We evaluate our approach over extensive simulations. Our simulation results demonstrate the significant advantages of our approach over single-band solutions: the throughput increases significantly, and the number of collisions decreases considerably. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: ultra wide band (UWB); home networks; wearable ad hoc networks; medium access control; binary conflict resolution

1. Introduction

Ultra Wide Band (UWB) is a novel wireless short-range technology, which has been the focus of a

lot of interest in recent times [1–6]. UWB is especially attractive for high-bandwidth data applications in small areas such as in home networks or wearable ad hoc networks. However, due to its unique

*Correspondence to: Ioannis Broustis, Department of Computer Science and Engineering, University of California, Riverside, 900 University Ave., Riverside, CA 92521, U.S.A.

†E-mail: broustis@cs.ucr.edu

Contract/grant sponsor: NSF CAREER; contract/grant number: 0237920.

Contract/grant sponsor: NSF NRT; contract/grant number: 0335302.

properties,[‡] it is difficult to apply media access control (MAC) schemes that have been developed for more traditional wireless ad hoc networks, such as CSMA [7]. Our objective in this effort is to design a MAC protocol, which (1) fully exploits the capabilities of UWB communications and, (2) utilizes the available spectrum efficiently. Towards this, we design an approach that uses simple binary ‘something’/‘nothing’ feedback to avoid collisions among nodes contending for medium access. While physical layer technologies on UWB communications have been developed [4], MAC and higher layer technologies that enable the use of UWB in ad hoc networks are yet to mature [1]. The unique properties of UWB pose challenges to the design of a MAC protocol and require the MAC layer to be synergetic with the underlying physical layer. We present three of these practical challenges, which motivate our multi-band approach.

The first and most important motivating artifact for our multi-band approach is the associated flexibility in spectrum use and the interoperability with other networks. If a portion of the UWB bandwidth is being used by other coexisting services, the corresponding band can be avoided with a multi-band approach. Thus, UWB communications can coexist with other networks (such as IEEE 802.11a based networks), a definite requirement in urban, disaster recovery and military settings. For example, in the presence of an IEEE 802.11a network, the multi-band system can avoid using the band from 5.35 to 5.85 GHz.

The second motivating observation stems from the absence of carrier sensing capabilities in UWB. With impulse-based UWB, data is transmitted in the form of pulses[§] and there is no contiguous carrier, although these pulses are possibly modulated by means of a high frequency signal (referred to as the *pseudo-carrier*). Thus, the commonly used protocols that rely on carrier sensing are not necessarily applicable. In addition, the very limited number of UWB-based MAC protocols that have been proposed previously are based on arbitration via time-hopping on a single channel. Time-hopped sequences with a *short* spacing between the time-hops can lead to collisions, while *long* durations between time-hops can lead to high delays and low efficiency. Thus, the second key objective of our design

is to reduce collisions to the extent possible, without resorting to long time-hopping sequences.

The third motivation is an artifact of the wireless channel effects on UWB transmissions. With impulse-based UWB, pulses are subject to multipath *delay spread* due to which, multiple time-shifted copies of each transmitted pulse appear at the receiver. This delay spread causes *inter-symbol interference* (ISI), wherein the delayed copies of one pulse interfere with subsequent pulses [8]. In indoor settings, the magnitude of this delay spread is of the order of tens of nanoseconds. One approach to deal with ISI is the use of sophisticated equalization. However, this adds considerable hardware complexity to the transceivers and increases the synchronization overhead. In fact, impulse-based UWB communications already require a long acquisition time for nodes to be synchronized prior to communications [7], which becomes longer due to the training sequence overheads required with equalizers.^{||} Another approach to reduce ISI is to ensure that the spacing between the received pulses is larger than the delay spread; thus, the delayed copies of one pulse will not interfere with the next pulse.[¶] With this approach, as opposed to the width of a pulse, the *inter-pulse spacing* constrains the throughput of the channel. Thus, in this case, *a smaller bandwidth channel, which requires an elongated pulse duration, can yield a throughput comparable to that of a wider band for a fixed equalizer complexity*. Hence, we note that we can partition the UWB spectrum into multiple comparatively narrow frequency bands that are mutually orthogonal and can be used simultaneously, and thus, use the available spectrum more efficiently.

In this paper, we propose and develop a novel multi-band MAC protocol for use with UWB-based WPANs. Our design is based on three main concepts: (1) *Separation of control and data onto different bands*. (However, the separation is not pure as we will see later). Simply put, each data exchange begins with a *rendezvous* transaction on a control channel, where one node

[‡] We discuss these properties in the next section.

[§] Recent developments with OFDM and Multi-carrier CDMA use carrier based methods; the trade-offs between the use of impulse-based UWB and OFDM based UWB are discussed in Reference [16].

^{||} The WiMedia Alliance supports an OFDM-based specification [9] for UWB; the motivation for dividing the available spectrum into multiple bands is to overcome the need for complex equalization. *OFDM, however, requires complex signal processing in terms of complex inverse fourier transform computations. This makes the channel difficult to implement.* A MAC layer protocol for use with OFDM for ad hoc networks is yet to emerge.

[¶] For a given average power constraint, the peak power constraint also imposes restrictions on the pulse repetition frequency (PRF) as we will discuss later.

sends an explicit *request message* to the other, which signals its willingness to continue through a simple *partial-response message*. Thereafter, the two nodes switch their attention to the data bands and follow an algorithm (described later) to select one of them for the actual data exchange. This separation of function has two main advantages. First, since all nodes share a common unreserved channel only for short control messages, the contention on the shared channel is limited. Second, once a pair of nodes agrees to communicate on a data band, the communication can be continuous (no need for the use of time hopping sequences), and thus, it is highly efficient. (2) *Data-band selection using a distributed load-balancing scheduler*. We use a novel virtual last-come first-served (LCFS) algorithm by which node-pairs that have already completed their rendezvous phase wait until the algorithm assigns them to a free data channel. Our virtual stack mechanism has two major advantages. First, it maintains the separation between node-pairs according to the time at which they completed their initial rendezvous, so that two node-pairs will not be assigned to the same data band unless their respective partial-response messages collided on the request channel. Second, because the waiting node-pairs are served in LCFS order, our scheduling algorithm has the *limited sensing* property. This means that new nodes are free to enter the scheduling algorithm at any time without waiting to acquire any current state information (i.e., how many other nodes are already waiting in the queue) and without causing any disruption to the algorithm because of their ignorance. On the other hand, LCFS scheduling is often considered an unfair policy, because it increases the variance of customer waiting times by giving lower priority to the customers who have been waiting the longest. In our application, however, this unfairness issue is minimized because of the smoothing effect of having multiple servers (data bands) pulling customers from the common stack. (3) *A novel binary conflict resolution algorithm is used for collision avoidance on the data bands*. Since pulse-based UWB cannot easily support either carrier sensing or collision detection at the physical layer, there is no way for the MAC to abort a message transmission in the event of a collision. However, it is possible for the physical layer to provide the MAC with binary ‘something’/‘nothing’ feedback information at low cost through the transmission of a short, continuous stream of pulses called *beacons*. Thus if one or more nodes transmit a beacon in a particular time slot, then all receivers will detect ‘something;’ otherwise, all receivers will detect ‘nothing.’ The binary resolution algorithm (explained in Section 3) is

used to avoid collisions on the data channel by running a series of lotteries among the group of node-pairs whose partial-response messages collided in a single slot on the request channel. The result of each ‘round’ of this conflict resolution procedure is to allocate the data channel to exactly one ‘winning’ member of the group, and to cause the remaining group members to defer their message exchange until the next ‘round.’

We wish to state here that while we restrict ourselves to small ad hoc networks such as home networks and wearable networks (networks for which UWB technology is most appropriate) where all nodes can hear each other, we expect that our approach can be modified to support larger networks. We expect to pursue such design modifications in the future. We should also note here that we do not investigate what happens in the case of two co-located WPANs that interfere with each other. This inter-WPAN interference has been studied in Reference [2]. Finally, UWB communications are constrained by emission limits imposed by the Federal Communications Commission (FCC) [10]. In particular, FCC requires that the effective isotropic radiated power (EIRP) be no higher than -41.25 dbm/MHz. Our design conforms to FCC requirements both in terms of the average and peak emission power levels.

The rest of this paper is organized as follows. In Section 2, we provide the relevant background on UWB communications, discuss the physical layer dependencies and demonstrate the conformance of our design to FCC imposed regulations. In Section 3, we provide a detailed description of our MAC protocol. In section IV, we present our simulation framework, results and deliberate on the observations. Related work on the design of MAC protocols for use with UWB is discussed in Section 5. Finally, Section 6 concludes the paper.

2. Physical Layer Dependencies

In this section, we discuss the UWB physical layer and highlight its impact on the design of our protocol. Detailed descriptions of some of the aspects of UWB communications can be found in References [4,5,10].

2.1. Multi-Band Impulse-Based UWB Communications

UWB communications, as per the specifications of the FCC, use the spectrum from 3.1 to 10.6 GHz [10].

FCC defines UWB communications as those that use signals that span at least 500 MHz of absolute bandwidth or those that occupy a fractional bandwidth $W/f_c \geq 20\%$, where W is the transmission bandwidth and f_c is the frequency at the center of the band [3]. UWB systems have traditionally achieved these high bandwidths by using pulses that are of very short time duration; we refer to these as *impulse-based* UWB systems. A typical UWB pulse belongs to the family of *Gaussian* shaped doublets [3,4]; these shapes are generally used since they can be easily generated by hardware. Multi-band modulation facilitates the division of the 7.5 GHz of spectrum made available by the FCC into multiple smaller frequency bands. The spectrum allocated to each band must meet the FCC restrictions as mentioned earlier. With impulse-based UWB, the pulse shape is the primary characteristic that determines the distribution of energy in the frequency domain and therefore allows for the separation and thus, the simultaneous use of the bands. Depending on the spectrum of operation, the Gaussian pulse is modulated by a set of carriers that belong to the particular band. This center frequency component is typically referred to as the *pseudo-carrier*. We wish to point out that the center frequency components of the different bands must be separated sufficiently in the frequency domain to avoid inter-band interference effects. A detailed discussion of pulse shaping can be found in Reference [4]. In our simulations, we use a simple Gaussian shaped pulse and assume that appropriate modulating signals can be employed (as shown in Reference [4]) to facilitate the division of the bandwidth into multiple bands each of which is 500 MHz in bandwidth. The pseudo-carrier of the highest band is 10.35 GHz and that of the lowest band is 3.35 GHz.

2.2. Pulse Position Modulation

We use a commonly studied modulation scheme called Pulse Position Modulation or PPM [4]. We also assume the use of a rate 1/3 convolutional code [8] which in turn implies that the information in each data bit is encoded into three *encoded bits*. Each pulse represents an encoded bit, whose information content (i.e., whether it represents a '0' or '1') is determined by the *position* of the pulse within what we call a *chip time* T_c . If the pulse occupies the first part of the chip-time, it represents an encoded bit value of '0'; else, an encoded bit value of '1' is implied. We assume that a Viterbi decoder is deployed at the receiver [11] and

this enables the soft-decision decoding of the received information.

2.3. Time Hopping

Time hopping has been used in previous approaches for sharing a single frequency band among multiple users [1,12]. However, we use this approach only in the control band and not in the data bands as we explain later. In time hopping, a fixed number of chip-times are aggregated to form a *sequence frame*. The duration of each sequence frame is T_f , and thus the number of chip-times per sequence frame is T_f/T_c . Each transmitter sends a pulse in only one of the chip-times in each sequence frame. The specific chip-time is determined by the node's time hopping sequence (THS), which is typically generated via a pseudo-random number (PN) code. The distribution of PN codes (for making a node's THS known to its neighbors) has been the topic of a few efforts [13,14]. In our work, we assume that the PN code is a function of a node's identifier (possibly the MAC layer address). The generators of these PN code sequences are initialized at system set up. Nodes periodically use out-of-band techniques to announce the state of their PN code generators.** The technique is similar to the proposal in Reference [13].

Time hopping sequences may be either sender-based or receiver-based. In receiver-based time hopping, potential transmitters use the THS of the receiver when they attempt a transmission. In the sender-based case, the transmitter sends pulses based on its own THS. The sender-based strategy is robust. In this case, one of the two transmissions is perceived by the receiver as a useful signal, while the other contributes to multi-user interference. Note however that in this case the receiver must tune its hardware to the right code [15]. The receiver-based approach is much simpler to implement; however, one could encounter collisions between the pulses from different transmitters, directed towards the same receiver. It is possible for protocols to use both approaches, as in Reference [1].

The average spacing between successive transmissions as per the THS will have an effect on the achieved performance. With shorter spacing between the time-hops, the pulses could be sent at a faster

** These announcements are made in special frames that we refer to as *Availability frames*. We discuss this in a later section that describes our protocol.

rate;^{††} however, there is a higher possibility of collisions.^{‡‡} With longer spacing, the possibility of collisions is reduced; however, large delays could be incurred. With our scheme, as mentioned earlier, time hopping is only used for the transfer of short control messages; since these messages are infrequent and fairly short in duration (low load), the probability of experiencing collisions remains low even with a relatively short spacing between the time-hops.

2.4. Channel Impairments and Effects

We next discuss the effects of the wireless channel on UWB communications and the associated impact on our MAC protocol design. A signal typically experiences three types of channel impairments: pathloss, shadowing, and multipath effects. The pathloss is given by the Frii's law [3]:

$$\alpha = \left(\frac{c}{4 \cdot \pi \cdot d_{ij} \cdot f_c} \right)^2 \quad (1)$$

where c is the speed of light, f_c is the center frequency of the band and d_{ij} is the distance between the transmitter and the receiver (for wideband systems, the geometric mean of the upper and lower frequency limits of the pulse band is more accurate than using the center frequency in the Friis equation, but the center frequency is sufficient for this study). Note that, at a given distance d_{ij} , higher frequencies will experience *higher levels* of attenuation than lower frequencies. We ignore shadowing effects since we assume that transmissions are typically over short distances (≈ 10 m) and therefore do not experience shadow fading [16]. Due to the multipath nature of the wireless channel, UWB transmissions (high data rates) will experience *multi-path delay spread*. A receiver will receive multiple copies of a transmitted UWB pulse, each of which may have a different amplitude, phase, and delay. Beyond a certain delay threshold (an inherent characteristic of the channel being considered), the signal amplitudes may be considered negligible. This threshold is referred to as the delay spread of the

channel. For indoor environments, measurements have shown that the delay spread is of the order of tens of nanoseconds [16]. If the time-spacing between the UWB pulses is smaller than the the delay spread of the channel, copies of the transmitted encoded bit interfere with the subsequent encoded bits. This is called *inter-symbol interference* or ISI for short. Equalizers are typically used to combat ISI [11]. The higher the level of the ISI, the higher the complexity and sophistication of the required equalizer. Equalizers also require the transmission of a training sequence prior to information communication. This can be expensive in terms of the overhead consumed. With UWB transmissions, a preamble is needed to allow for the sender and receiver to synchronize prior to communications. By acquisition, we mean that the receiver learns how to recognize the presence of a pulse train in the presence of thermal or other noise factors. The aforementioned acquisition preamble is considered expensive in terms of overhead [7]. The deployment of a sophisticated equalizer will further increase the overhead costs incurred with UWB. Note that the attenuation experienced by the different pulse copies could differ. Thus, some copies may not cause a significant level of interference if they were to overlap with other transmissions. However, whether or not they have an effect is scenario specific. In order to completely eliminate the possibility of collisions, we require that pulse copies do not interfere with other subsequent transmissions. Another strategy for combatting ISI would be to use direct sequence CDMA in conjunction with a Rake receiver. However, the long codes with CDMA could still incur capacity penalties. The alternative that we explore in this work is to separate the pulses by at least the delay spread of the channel. Thus, the time-spacing between the pulses is chosen to be at least 30 ns^{§§} (delay spreads in indoor environments [16]). We recognize that by doing so, the pulse width could be increased to some extent since this is unlikely to interfere with future encoded bits. Increasing the pulse width allows for the use of lower pseudo-carrier frequencies and thus, facilitates the use of multiple frequency bands as discussed earlier.

2.5. Conformance With FCC Regulations

The FCC regulations limit the effective isotropic radiated power (EIRP) to -41.25 dBm/MHz (Part 15 of the regulation) [3,10]; the power used, on average, per bit

^{††} The FCC regulations impose a limit on the pulse repetition frequency as will be discussed later.

^{‡‡} In Section 4, we explain in detail the cases in which pulse collisions can occur.

^{§§} Note that this translates to having a *chip-time* of 60 ns.

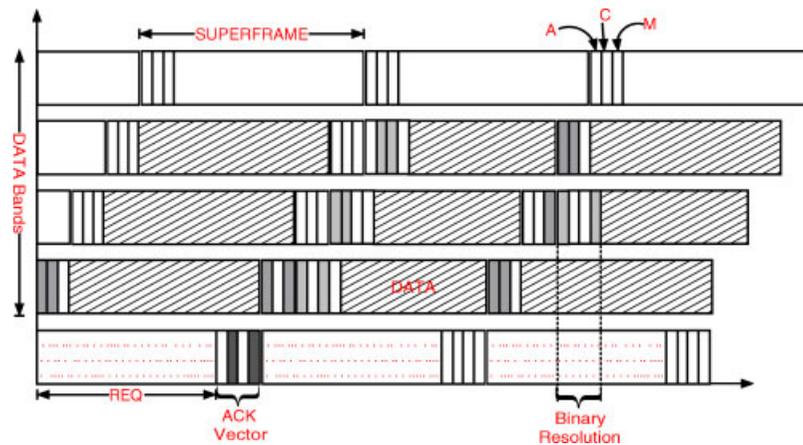


Fig. 1. The frame structure with our protocol.

cannot exceed this imposed limit. As discussed in Reference [22], to conform to this limit, we use low powers in each band. As we also discussed in Reference [22], the range is the lowest for the highest frequency band, with $f_c = 10.35$ GHz, and we assume this to be around 7 m. We set the range to 7 m for all of the other frequency bands and use the appropriately computed average transmission power for those bands. Clearly, the powers used will be lower than the FCC imposed limit. To summarize, with the settings as above, we conform to the FCC imposed restrictions on the EIRP. In addition to the imposed restriction on EIRP, the FCC also imposes a limit on the peak power that can be used for UWB transmissions. As specified in [10], if the average power limit is met and the frequency of pulse transmissions is higher than 1 MHz, the peak power limitation is also met. With our scheme, since the maximum distance between the pulses is 60 ns (the chip-time), the frequency is 16.67 MHz. Thus, our scheme conforms to the peak power constraint.

2.6. Coding and Higher Layer Abstractions

As mentioned earlier, we assume that a rate 1/3 convolutional code is used to help efficiently recover from errors if the only impairment is thermal noise. In our control band, we use a THS for arbitration as we will discuss later. This can in turn lead to collisions among transmissions. In order to provide a further level of robustness on this band, we employ a *repetition code* of 2, that is, we repeat twice the output of the convolutional encoder. In our simulations, we assume the presence of the convolutional encoder and decoder and do not implement them. Instead,

we use the bit error rate of 10^{-7} and discard bits at this rate.

2.7. Time Synchronization

Our approach requires the division of time into frames, which implies that communicating nodes must be synchronized in time. We assume that synchronization is achieved by the use of previously proposed methods for this purpose [17,18].

3. The Multi-Band MAC Protocol

In this section, we describe in detail, our multi-band MAC protocol.

3.1. Two-Dimensional Frame Structure

Figure 1 illustrates how we partition the total allocated spectrum into B disjoint frequency bands. For an FCC-compliant system, we can assume that $B \leq 15$ since the total available spectrum is 7.5 GHz and each transmission must span at least 500 MHz of bandwidth. Band 1 is called the *Req-Band*, which is shared by all nodes during the initial rendezvous phase described below in Subsubsection 3.2.1. The remaining $B - 1$ bands are called *Data Bands*, which are allocated to an individual node pair for a private data exchange as described in Subsubsection 3.2.4. Each frequency band is further partitioned in time to form a sequence of fixed-length *superframes*. All bands use the same superframe length, but the superframe boundaries for adjacent bands are offset

Table I. The effect of context-dependent physical-layer channel encoding on net data rate.

Encoding context			Name	Symbol	Test value	Description
Req	Data	Beacon				
✓	✓		<i>Pulse slot</i>	τ_p	30 ns	Time allocated to one pulse transmission, including delay spread
✓	✓		<i>Chip time</i>	τ_c	60 ns	Two consecutive pulse slots, the <i>occupied</i> pulse slot encodes the binary chip value
✓			<i>Spreading factor</i>	η	6	One <i>sequence frame</i> $\equiv \eta$ <i>chip times</i> ; each transmitter following a time-hopping sequence uses only one <i>chip time</i> per <i>sequence frame</i>
✓	✓		<i>Encoding rate</i>	$1/\rho$	1/3	A rate $1/\rho$ convolutional encoder expands each data bit into a sequence of ρ chips
✓	✓		<i>Repetition rate</i>	ν	2	A rate ν repetition code repeats every chip sequence generated by the convolutional encoder ν times to improve robustness
		✓	<i>Beacon length</i>	β	30	Number of <i>consecutively occupied</i> pulse slots to broadcast binary ‘something’ feedback to all nodes
2160 ns	360 ns	900 ns	Overall transmission time per bit of MAC-layer data (based on test parameters)			

by fixed time shift.^{|||} Moreover, the Req-Band and Data Bands use different internal structures for their respective superframes, as shown in Figure 1.

Req-Band superframes consist of a *Request Period* followed by *Response Period*. An unpaired sender (e.g., X) that wishes to establish a new data exchange session sends a *request message* to its intended receiver (e.g., Y) during the Request Period. The receiver returns an active acknowledgement during the Response Period to complete the rendezvous. Since the Req-Band is shared by all nodes, the MAC employs more robust (but less time-efficient) physical-layer encoding schemes during the Initial Rendezvous phase to reduce the likelihood of rendezvous failure (as shown in Table I). In particular, every request message sent during the Request Period must follow the receiver’s time-hopping sequence. Thus, multiple node pairs can complete the initial rendezvous phase during the same superframe as long as their respective time-hopping sequences do not overlap excessively. Similarly, the Response Period is partitioned into a V -slot *ACK Vector*, such that each receiver must transmit a *beacon* in its assigned ACK-Vector slot (e.g., $hash(Y)$ for receiver Y) to accept an incoming request. A beacon is a control message that consists of a block of $\beta \gg 1$ pulse slots, all of which must be either empty (to represent the ‘nothing’ control message) or occupied by β consecutive pulses (to represent the ‘something’ control message).

^{|||} By staggering the superframe boundaries across different bands, a node executing the virtual stack algorithm described in Subsubsection 3.2.2 only needs to monitor one band at a time. The time required for a node to switch its receiver from one band to another is approximately 4 ns [19].

This simple ‘all-or-nothing’ encoding means that beacon transmissions exhibit the following *Partial Response Property*:

Suppose one or more nodes Y_1, \dots, Y_k all transmit beacons at the same time. Then, because of the fixed and highly redundant beacon format, all listening nodes will correctly receive ‘something’ without the need for prior synchronization with the associated sender(s). However, since every beacon is exactly the same (i.e., it carries neither source or destination addresses nor other identifying information), the listening nodes have no way to determine which node(s) sent the beacon.

Data Band superframes start with the *public* triplet of $A/C/M$ beaconing slots, which are used for ‘something’/‘nothing’ control signals that must be visible to all nodes currently executing the band-selection phase. The remainder of the superframe is reserved for *private* communications among those nodes already assigned to this band. However, the private portion of the superframe has two possible formats. Because of the Partial Response Property, a node pair must first claim ownership of its chosen data band before beginning the actual data exchange. This Data Band Claiming process treats the private portion of the superframe as a sequence of beaconing slots, where each active node pair bids for control of the Data Band by transmitting a pattern of beacons.

3.2. Operation of the MAC

We now describe the operation of our multi-band MAC in detail, while focusing on the steps required for completing a successful data exchange. For

concreteness, our description will focus on a running example of the event sequence when transmitter node X , say, carries out a successful data exchange with receiver node Y , say.

3.2.1. Initial Rendezvous Phase

Initially, nodes begin executing the MAC protocol from either the *idle state* (e.g., node Y , which currently has no outgoing data waiting to be sent) or from the *unpaired transmitter state* (e.g., node X , which needs send a block of data \mathcal{D} to some adjacent node Y). In our MAC protocol, *unpaired transmitter node* X is responsible for actively initiating the data exchange session with its target receiver Y . Thus, X begins the process by generating an d -bit randomly-generated \mathcal{R}_X , which will be used by Y during the subsequent data-band claiming phase, and identifying the time-hopping sequence $\text{THS}(Y)$ that is monitored by its target receiver. During the Request Period for the next Req-Band superframe (superframe s , say), node X transmits a request message to node Y containing payload \mathcal{R}_X , according to $\text{THS}(Y)$. If the appropriate ACK-Vector slot during the Response Period contains a ‘something’ beacon, node X assumes that its initial rendezvous was successful and advances to the Band Selection phase, described later in this section. Otherwise, node X assumes that its rendezvous attempt failed, and waits for a random backoff delay before repeating the process.

Conversely, *idle receiver node* Y simply listens passively to the Req-Band in accordance with $\text{THS}(Y)$, to see if any unpaired transmitter wishes to establish a data exchange session with Y . If superframe s carries such a request message, then node Y records the identity of its new session partner (i.e., transmitter X) and the randomly-generated session priority \mathcal{R}_X in its payload, then completes their initial rendezvous by transmitting a ‘something’ beacon in ACK-Vector slot $\text{hash}(Y)$ of the Response Period and advances to the Band Selection phase, described later. Otherwise, node Y simply continues to listen for incoming request messages during each subsequent superframe.

Since the ACK-Vector length is much smaller than the total number of nodes in the network, the remaining steps of our MAC protocol must be robust enough to handle the inherent ambiguity of seeing a ‘something’ beacon in a particular ACK-Vector slot. For example, suppose that unpaired transmitters X_1 and X_2 both send request messages during superframe s , and that the *hash* function coincidentally assigns their respective receivers Y_1 and Y_2 to the same ACK-Vector slot. In other words, the Partial Response Property means

that the observed outcome of an Initial Rendezvous attempt during superframe s may be different for the two participating nodes:

If unpaired transmitter X_1 sends a request message to target receiver Y_1 during superframe s , then X_1 concludes that its initial rendezvous was successful if *any* node(s) transmitted ‘something’ in ACK-Vector slot $\text{hash}(Y_1)$. However, receiver Y_1 will not conclude that their initial rendezvous was successful unless *it* sent ‘something,’ even if ACK-Vector slot $\text{hash}(Y_1)$ contains ‘something.’

As we will see below, our protocol allows the transmitter to quickly discover such ‘false rendezvous’ errors, and to restore normal operations without disrupting any other data exchanges nor wasting a data band by allocating it a failed data transfer session.

3.2.2. Data-band selection phase with the virtual stack

After successfully completing an Initial Rendezvous, each node joins the *Virtual Stack*, where it waits for its assignment to an available Data Band. During this time, the waiting nodes acquire all the necessary information to synchronize their own local state to the evolution of the global Virtual Stack merely by *passive listening for beacon transmissions* in the ACK-Vector slots for the Req-Band Response Period, and in the public *A/C/M* control slots for each Data Band. This passive listening is used to maintain a *local stack-depth* variable, say $q_N(t)$ for node N , which marks the node’s current priority in the band-selection process. Note that N makes no attempt to learn any global properties about the Virtual Stack because data bands are assigned in LCFS order. To simplify the design of the subsequent Data-Band Claiming Phase, the local-stack update rules ensure that at any time t , the difference in local stack-depths for any pair of nodes executing the Virtual Stack Algorithm is always an *even* number. Therefore, node N always knows that it is $\lceil q_N(t)/2 \rceil nd$ in line for receiving an available Data Band at time t .

Upon completion of the Initial Rendezvous Phase, node N enters the Data-Band selection Phase by initializing $q_N(t) \leftarrow 2$, to indicate that N is now first in line for receiving an available Data Band—as one would expect given our ‘limited-sensing’ LCFS scheduling policy. Thereafter, N updates $q_N(t)$ after each beaconing slot according to the following rules: (i) During the Response Period for each subsequent Req-Band superframe, N listens to every ACK-Vector slot to see whether it contains ‘something.’ If so, N

updates its current stack depth via $q_N(t) \leftarrow q_N(t) + 2$. Otherwise, $q_N(t)$ remains unchanged. Note that the value of $q_N(t)$ is either unchanged, or increases by exactly 2 when N executes this rule. (ii) During the public control slots for each subsequent superframe on Data Band b , N listens to its A and C control slots to see if both of them contain ‘something.’ If not, Data Band b may be a candidate for reassignment; so N decrements its stack depth via $q_N(t) \leftarrow q_N(t) - 1$ and changes its value to an *odd* number. Otherwise, Data Band b must be occupied by an ongoing data transfer session that will last for at least one more superframe; so N leaves $q_N(t)$ unchanged. (iii) If $q_N(t)$ is an *odd* number, then N listens to the M control slot on Data Band b to see if it contains ‘something.’ If not, then Data Band b is *available* and N decrements its stack depth as per $q_N(t) \leftarrow q_N(t) - 1$. Otherwise, Data Band b is *not available* because some previously-assigned nodes are still trying to claim it though another round of the Binary Conflict Resolution Procedure (described next); so, N increments its stack depth via $q_N(t) \leftarrow q_N(t) + 1$ and moves on. Note that $q_N(t)$ is returned to an *even* number when N executes this rule. However, the net effect of examining the $A/C/M$ control slots for Data Band b can either restore $q_N(t)$ to its previous value, or decrease it by exactly 2. (iv) If $q_N(t) = 0$ at the end of the $A/C/M$ control slots for Data Band b , N terminates the Virtual Stack Algorithm by selecting band b , and then, advances to the Claiming Phase.

Now consider what happens when several nodes jointly execute the Virtual Stack Algorithm after simultaneously completing the Initial Rendezvous Phase. Define $\mathcal{V}(s, v)$ as the set of all nodes that began executing the Virtual Stack Algorithm at the end of ACK-Vector slot v of Req-Band superframe s . Recalling our previous discussion of the Partial Response Property for beacons, we conclude that if $\mathcal{V}(s, v) \neq \emptyset$ then, it should contain at least one node Y_i that transmitted ‘something’ in ACK-Vector slot v in response to receiving a valid request message in superframe s . $\mathcal{V}(s, v)$ should also contain all nodes X_j that transmitted request messages to a target receiver for which the hash value, $hash() = v$. As long as every node in $\mathcal{V}(s, v)$ hears the same sequence of beacon transmissions,^{¶¶} it does not matter how many nodes there are in $\mathcal{V}(s, v)$. They will all update their respective

local stack-depth variables in lock-step until they terminate the Virtual Stack Algorithm by choosing the same Data Band b' in superframe $s' \geq s$ and advance to the Claiming Phase. On the other hand, any nodes that leave the network during the Data-Band Selection Phase will disappear from $\mathcal{V}(s, v)$, and any node that fails to update its local stack-depth correctly because of transmission errors will find itself in the wrong group $\mathcal{V}(s_0, v_0)$. Therefore, the set of nodes $\mathcal{A}(s', b')$ that is actually assigned to Data Band b' during superframe s' could be different from $\mathcal{V}(s, b)$, and might contain mismatched transmitter or receiver nodes.

3.2.3. Data-band claiming with the binary conflict resolution algorithm

Let us now restrict our attention to the set of nodes $\mathcal{A}(s', b')$ that were newly assigned to Data Band b' at the end of the $A/C/M$ control slots in superframe s' . Notice that all of these nodes share a common stack-depth of zero—in contrast to local stack-depths ≥ 2 for all other nodes executing the Virtual Stack Algorithm at this time. Ideally, $\mathcal{A}(s', b')$ consists of exactly one pair of matching nodes, viz., the transmitter X and its chosen receiver Y in our running example. In this case, the ideal strategy would allow X and Y to begin their Data Transfer Phase immediately. Unfortunately, if we are not so lucky $\mathcal{A}(s', b')$ then may also include some unpaired transmitters and/or multiple pairs of matching nodes, and this simple strategy would cause a Data Band collision. Because UWB does not support carrier sensing or collision detection, each Data Band collision would waste the entire superframe, and force the affected nodes to abort the Data Transfer Phase and enter the backoff state. Therefore, our MAC protocol inserts a *Data Band Claiming Phase* between the Data Band Selection and Data Transfer phases, which provides two methods for avoiding data-band collisions.

First, we eliminate all potential interference due to any unpaired transmitters in $\mathcal{A}(s', b')$ by introducing a role-reversal ‘trick’ in which *no transmitter can send data without an invitation from its target receiver*. This ‘invitation’ consists of seeing the randomly-generated session priority from the payload of its own initial request message form the *winning bid* during this Claiming ‘round.’ Second, we minimize the possibility of interference between multiple receivers in $\mathcal{A}(s', b')$ by using a Binary Conflict Resolution Algorithm to

^{¶¶} This symmetric feedback assumption is very reasonable because of the highly redundant, fixed structure of each ‘something’ beacon, provided every node in the ad hoc network is within hearing range of the others. This is true if the nodes are all located in a small area that is approximately 10 m \times 10 m.

These dimensions match that of a typical home network or a wearable ad hoc network.

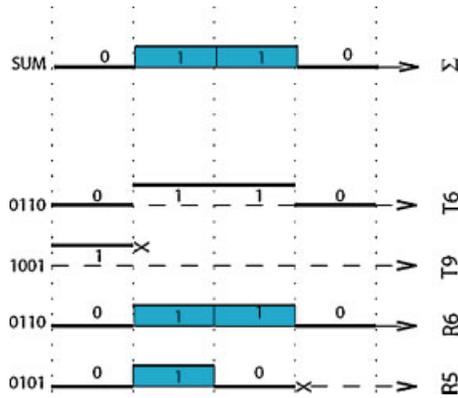


Fig. 2. One 'Round' of the binary conflict resolution algorithm using 4-bit session priorities.

serialize their respective 'invitations' so that *only the highest-remaining random session priority* is broadcast during each Claiming round. Thus, even if $\mathcal{A}(s', b')$ contains several matched pairs of nodes because of an ACK-Vector collision, their respective data transfer sessions will not collide unless they chose the same random session priority.

During each Claiming round, all of the active receivers in $\mathcal{A}(s', b')$ execute a d -step Binary Conflict Resolution Algorithm to determine which of them holds the *largest* d -bit session priority. These steps consist of a sequence of d beaoning slots, through which all nodes in $\mathcal{A}(s', b')$ compare their *own* session priority to the *winning* priority, one bit at a time starting from the most significant end, to determine which of them holds the winning value.

The operation of a four step Claiming round is shown in Figure 2, that includes two transmitters (with priorities 9 and 6) and two receivers (with priorities 6 and 5). Each node is represented by its own sequence of beaoning slots running down the page (labeled T9, T6, R6, and R5, respectively). The right-most timeline (labeled Σ) shows that the winning session priority is just the logical 'OR' of all transmitted beacons. At each step, the timeline for each active node is marked with the corresponding bit of its session priority. If that priority bit is '1' then an active transmitter looks for a beacon (shown as a hollow box) whereas an active receiver sends a beacon (shown as a shaded box). As soon as an active node detects a beacon mismatch (i.e., a transmitter does not find a beacon when it is expecting one, or a receiver finds a beacon when it did not transmit one), the node immediately drops out of this round (shown by a \times symbol at the end of the slot). Notice that node T9 has the highest session priority, but it drops out after the first step when it expects a bea-

con but sees none (i.e., all active receivers hold smaller priorities so its initial rendezvous must have failed). Similarly, node R5 drops out after the third step when it expects an idle control slot but sees a beacon (i.e., its priority is below the maximum); however, it will try again in the next round. Finally, nodes T6 and R6 find a match at every step and thus succeed in claiming the Data Band and immediately advance to the Data Transfer Phase. Meanwhile, all other nodes in $\mathcal{A}(s', b')$ that hold session priorities than are *strictly smaller* than the winning value remain with Data Band b' until the current Data Transfer Phase terminates at the beginning of superframe s' (where the A and C control slots do not contain beacons for the first time), at which time all remaining receivers in $\mathcal{A}(s', b')$ transmit a beacon in the M control slot to start another Claiming round.

3.2.4. Data transfer phase

At the end of each 'round' of the Binary Conflict Resolution Algorithm, Data Band b' has now been claimed by a subset of the receivers in $\mathcal{A}(s', b')$ that share the same random number for their session priority. With very high probability, we can assume that this set consists of a single receiver Y and its associated transmitter X . In that case, X transmits its first quantum of data to Y during the remainder of the current superframe. If Y is able to correctly receive this data, it transmits a beacon in the A control slot of the next superframe to complete their initial handshake. Thereafter, X can continue the data transfer session by transmitting a beacon in the C control slot, followed by more data in the remainder of the superframe. Conversely, if Y cannot complete the initial handshake with its associated transmitter X (e.g., because X left the network, or its data packet collided with another transmission), then Y forfeits its claim to the Data Band.

If their initial handshake succeeds, then the data transfer from X to Y continues as long as Y transmits 'something' in every A control slot (acknowledging reception of the previous data packet), X transmits 'something' in every C control slot (offering to continue the session), and X fills the remainder of each superframe with its next data packet. Eventually, the session ends when the A control slot and/or the C control slot is empty for the first time.

4. Simulation Results

We present the evaluation of our idea through simulations using a C++ simulator that we have developed by extending a previous simulation effort [2]. Our

focus is on the performance at the MAC layer. Thus, we assume that data is injected at the MAC layer and that the transmissions of a node are intended for a neighbor. Nodes are distributed over a restricted region of interest so as to reflect either home or wearable environments. Hence we assume that all nodes form a clique topology, in which each node is capable of reaching all other nodes. In our simulations, we use assumptions and conventions that are widely used in UWB studies and try to incorporate as many realistic details [1,2], as possible.

4.1. Comparisons

We compare our scheme with a single-band approach in order to demonstrate the benefits of our multi-band scheme. In a nutshell, the single-band approach is based on using a single band with time hopping as the basic means of access. We do not assume the presence of an equalizer and hence, the pulses are spaced apart as in the multi-band approach. One might think that the single band approach is disadvantaged to a large extent; while this is true to some sense and it is intuitively clear that the multi-band approach can yield a significant increase in the achievable throughput especially when the number of users is small, the comparison quantifies the achievable gains. Furthermore, we provide some sample results wherein we eliminate some of the collision effects in the single-band approach (the approach we take for doing this is discussed below); this provides a fair comparison of the two approaches. Note here that we do not compare our approach with the previous approaches on multi-band media access in ad hoc networks; this is because they are based on carrier sensing and, therefore, inappropriate for direct comparisons.

4.2. The Single Band Approach

With the single band approach, data and control packets use the entire 7.5 GHz bandwidth (whereas up to $B - 1$ simultaneous users can transmit data packets on different bands during the same superframe in our multi-band scheme). The approach is *loosely* based on the approach in Reference [1].*** Initially, the nodes exchange the control messages (as with our protocol) to establish a handshake. If the handshake is successful,

*** We note here that it is not our purpose to compare our scheme against the MAC protocol presented in Reference [1]. That scheme uses rate adaptation in order to alleviate interference effects, a technique which we do not incorporate—even though we could.

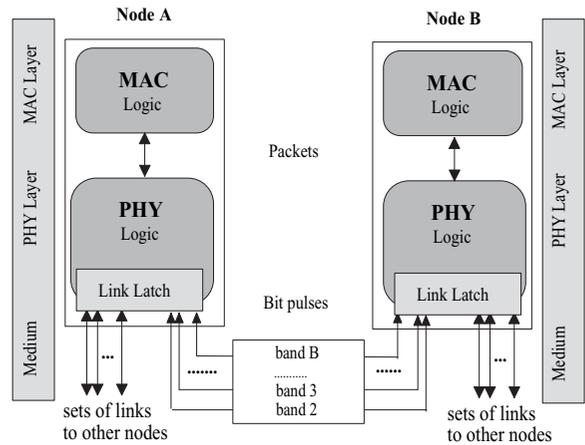


Fig. 3. Our simulation implementation platform.

the nodes switch to a unique THS, on which they communicate. However, note that the bandwidth is shared among the plurality of users and the data transmissions will also have to compete with the transmission of the control information. This would put the single-band approach at a distinct disadvantage, especially at low loads. In order to avoid giving our scheme an unfair advantage, we provide a version of the single band approach where we *magically* eliminate the effects of pulse collisions on the reception of data packets; when the communicating nodes switch to the unique predetermined THS (mentioned above) to exchange data packets^{†††} they communicate collision free. Note that this assumption now shifts the unfair advantage to the single band case, since many more than $B - 1$ simultaneous data transfers could be supported if the requests get through. One can envision this to be akin to using a perfect equalizer, which is calibrated during the reception of a request packet, to eliminate the ISI during the reception of the following data packet. Note however, that with both the *collision-free* version of the single-band *and* the multi-band approaches, pulse collisions may occur during the initial handshake wherein a request is transmitted as per the receiver's THS. In our plots, we label the more realistic single-band approach as simply *single-band*; we label the collision free version of the approach as CF-Single-Band.

4.3. Simulator Implementation Details

In our implementation, the physical layer consists of a number, m , of sets of virtual links as shown in Figure 3.

^{†††} A similar single-band scheme is described in Reference [1].

This number is equal to the number of bands; each set of links has a separate buffer and connects a node with its neighbors. As a result, a node has m links with a neighbor node, each representing a different band. The MAC layer of the transmitter delivers the packet to the appropriate link of the appropriate band. The physical layer component converts the bits to pulses, which will be transmitted through this link. The channel characteristics, discussed earlier in Section 2 are applied and distort the transmission. The receiver picks each pulse, decodes a set of pulses that form a bit (if possible), and stores the bit in a buffer. A bit may be discarded either due to a collision (elaborated below) or due to its being corrupted by thermal noise as discussed in Section 2. When a set of bits that form a packet have been received correctly, the packet is re-constructed and delivered to the receiver's MAC layer.

4.4. Simulation Scenarios

4.4.1. Network layout

Nodes are mobile and form a single-hop ad hoc network as in a home or in a wearable computing system. They all use the same, maximum transmission power. We use video traffic (described later). This UWB technology is especially suited for such traffic.^{***} We vary the number of nodes from 6 to 30. As mentioned in Section 2, the maximum range of a transmitter is considered to be 7 m. The total number of bands in the multi-band system is 15. A transmitter selects a receiver randomly.

4.4.2. Frame structures

Every sequence frame consists of six T_c frames (chip-times). The duration of the superframe is set to 6250 chip-times, which is approximately equivalent to a successful packet exchange; the A , C , and M control slots are accounted for. This frame duration is also long enough to contain a successful request transmission (which occupies 4320 chip-times) as well as 64 ACK slots following it. Each ACK slot is 30 chip-times long.

4.4.3. Traffic characteristics

We import bursty MPEG-4 medium and high quality video traffic in our simulation experiments. We have obtained the video traces from [20]. Table II summarizes the traffic characteristics. We fragment a video

Table II. MPEG-4 medium and high quality video traffic characteristics.

Metric	Medium	High
Minimum frame size (bytes)	26	72
Maximum frame size (bytes)	8511	16745
Mean frame size (bytes)	1338	3800
Mean bit rate (bits/sec)	267000	770000
Peak bit rate (bits/sec)	1702200	3300000
Compression ratio	28.401	9.92

frame into 250-byte parts when the frame is larger than 250 bytes. The fragments are then sequentially inserted into the nodes' queues. Request packets are 15 bytes long; this is in accordance with the size of control packets used with other wireless protocols (such as with the IEEE 802.11 MAC protocol) [21].

As mentioned earlier, our decision to import this kind of traffic stems from the fact that it is delay-sensitive and because UWB networks are a possible solution for wirelessly transmitted media. The human eye must see a video frame within 40 ms of when it saw the preceding frame. If not, video jitter is perceived. Since delayed (for more than 40 ms) video packets are useless, the network will drop them. While we have tested both the single-band and the multi-band schemes with very high quality MPEG-4 video traffic, below we present results mostly with medium quality video traffic. We also show packet drops, when high quality video traffic is considered. The behavioral trends with the two kinds of traffic were similar.

4.4.4. Pulse collisions and bit errors

We assume the presence of a rate 1/3 convolutional encoder for the data bands and accordingly, three pulses represent a single bit. In the control band or in the case of the single band system, in order to provide robustness to collisions, we use a repetition code of 2, that is, each bit (comprising of three pulses) is repeated twice. Thus, in these bands, six pulses form a bit. For the type of receiver that we assume, a *pulse collision* occurs when two or more pulses (of sufficient energy) arrive during the same T_c period, in the same band. A bit is received in error, when all of the pulses that make up the bit either collide, or are corrupted due to thermal noise. At this point note that we take into consideration the *receiver's power sensitivity*. In particular, the receiver maintains an upper and a lower power threshold. If the received power falls below the lower threshold, the receiver does not take into account the received energy. If the received power

^{***} As an example, one could have a plurality of video cameras sending images to a set of processing receivers.

(within a T_c) exceeds the upper threshold, the receiver will assume that a collision has occurred. Finally, if the received power falls between the lower and the upper threshold, the receiver will assume a correct reception. Furthermore, in our physical layer implementation we take into account the *average signal attenuation*, which has an impact to the received signal power. As explained earlier, the nodes' time hopping sequences are not orthogonal and this can cause pulse collisions. Let us assume that two pulses arrive at a given receiver, *at the same pulse slot*. If these pulses are attenuated to some extent, or if their phases have changed, then their simultaneous reception *may not* trigger a collision at the receiver. This is because their additive received power may not exceed the receiver's upper power threshold. In that case, the receiver will assume that it correctly received a pulse. Note here that the receiver correlates the received signal with a reference signal, and does not check upon the shape of the received pulse; the correlation reflects the received energy. This is because the transmitted signal is expected to be distorted (by the channel and filters) anyway.

4.4.5. *Back-off policy*

With our backoff algorithm (discussed in Section 3) for packet retries, we set the initial back-off to a randomly chosen value between 0 and 5 superframes. After each retry, the maximum value increases by 2, until it reaches a maximum of 15. We have varied these values and the results obtained demonstrate behavioral traits that are similar to those considered in our sample set presented here. The packet is discarded if, after 15 attempts, a node is unable to deliver it to its intended neighbor.

4.4.6. *Providing for consecutive packet transmissions*

With any given reservation, we allow a transmitter to send at most two consecutive packets to its receiver. This would in some sense amortize the preamble and request costs over a larger transmission. It also provides a level of fairness, we restrict this number to two, to prevent the dominance of a channel by a single communicating pair. The overall simulation time is 15 million chip-times T_c .

4.5. Results

We only present a sample set of results since the behavioral trends captured by this sample set are representative of a more exhaustive set.

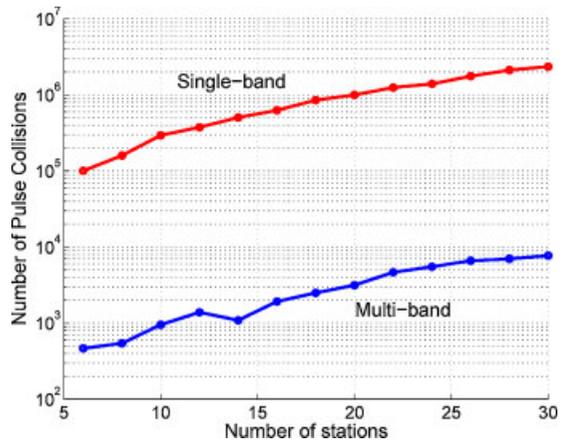


Fig. 4. Number of pulse collisions, for medium quality MPEG-4 traffic. Logarithmic scale is used.

In Figure 4, we plot the total number of pulse collisions for each approach as a function of the number of nodes in the network. We observe that our protocol decreases the number of pulse collisions by more than two orders of magnitude, as compared with the single-band approach. The reason is that, in our protocol, data packets are transmitted practically free of collisions, since they are exchanged on an exclusively reserved data band. In contrast, in the single-band case, packets suffer frequent collisions due to overlaps between nodes' THSs.

In Figure 5, we plot the bit error rate averaged over the observations from all the nodes in the network as a function of the number of nodes. We observe a much higher bit error rate in the single-band system, again, a direct result of collisions of data packets. In particular, the difference in bit error rates increases with the number of nodes in the network. With a few nodes (6 to 10),

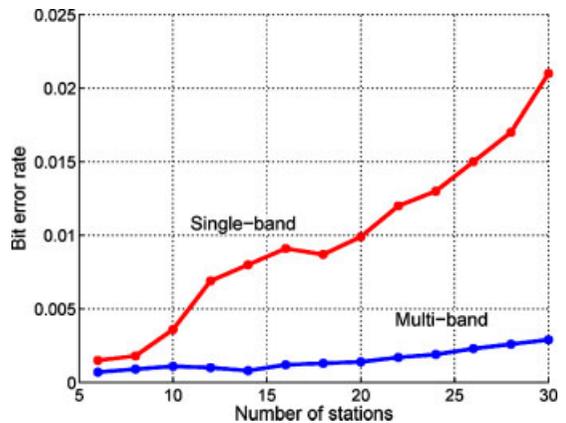


Fig. 5. Bit error rate in the network, for medium quality MPEG-4 traffic. The multi-band scheme outperforms.

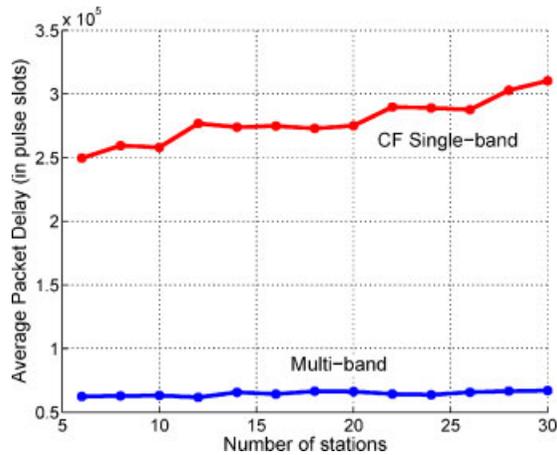


Fig. 6. Average video packet delay. Medium quality MPEG-4 traffic is considered.

collisions are not very frequent with the single-band approach. As mentioned earlier, we have $6 T_c$ chips in a T_f frame. This implies that in the ideal case, there are at most six nodes that are transmitting in parallel without causing interference to each other; this ideal case also happens when their THSs do not overlap. However, there is always some amount of overlap among THSs; as the number of nodes increases, more overlaps take place. This degrades the performance of the single-band scheme.

Next we report the observed average video packet delay in the network. This delay is the duration between the instance that a video packet arrives to the MAC layer queue of a node, until the instance that it is completely reconstructed at its destination. In Figure 6, we plot the average packet delay as a function of the number of nodes in the network with MPEG-4 medium quality video traffic. We observe that in our protocol, packet delays are lower by a factor of 4 to 6, as compared with the delay incurred with the single band scheme. More significantly, note that the average packet delay is constant with our multi-band scheme! This suggests that the delay is not very sensitive to the number of participating nodes (or stations) in the network. This provides us with the desirable feature of low jitter. In contrast, the delay increases in the single-band scheme, as the number of nodes increases. We point out here that we did not observe any video packet losses in the multi-band scheme; since our system rejected those packets that did not meet our jitter constraint, this suggests that the jitter requirements were almost entirely satisfied. In the single-band case, with more than 16 users, we observed that video packet losses become more and more frequent. Recall that as the number of

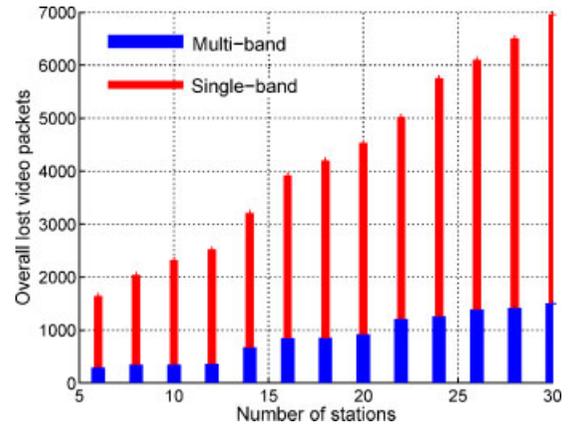


Fig. 7. Comparison between the two schemes concerning the overall lost video packets. High quality MPEG-4 traffic is considered.

nodes increases, the network load increases as well; this resulted in excessive jitter and caused the drop-page of packets. Next, we discuss these measurements in detail.

We next look at the total number of lost video packets, the observations being over the duration of the simulation. Note from Figure 7, that our multi-band protocol performs better than the single-band approach by almost a factor of 7. Furthermore, notice that this improvement is over an *unfairly* advantaged single-band system, which can support any number of simultaneous data transmissions as long as their request handshake was successful. In spite of the advantage, with the single-band system pulse collisions are more frequent, and because the amount of time that a packet remains in the queue is extremely long and in many cases longer than 40 ms; given the requirements of the video application, the network simply drops packets that are delayed by more than 40 ms. One could argue that the problem of pulse collisions could be solved by increasing the number of T_c chips in the T_f frame. This would decrease the number of collisions, however, it would considerably increase the average packet delays even more (this was corroborated by our simulation experiments).

Finally, we wish to observe the utilization of bands. For this, we consider two simulation scenarios, in which we have 16 nodes and 28 nodes in the network, respectively. Because our MAC protocol involves a staggering process for the band selection, the frequency bands are not uniformly utilized. On the contrary, the lower data bands are mostly used, while the utilization of upper ones is rare. The ideal case for 16 nodes would be to have 8 bands being used at the same time, since

Table III. Band utilization for 16 nodes in the network (not all bands are used).

Data band	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Utilization	1049	894	832	764	540	206	22	0	0	0	0	0	0	0

8 transmitter–receiver pairs can be formed at most at the same time. However, we observe that *at most* 5–7 bands are used in parallel at any time. To illustrate this with an example, we provide Table III. We observe that the first data band is mostly used, while data bands 8 to 14 are never used. Taking this observation further, we performed the same experiment with 28 nodes. We observed that even though all of the bands were used, the upper ones are utilized only for a couple of times during the simulation. From these experiments, we make conclusions both about the fairness of the virtual stack, as well as for the overall bandwidth utilization. As for the fairness of the virtual stack algorithm, we conclude that, since the upper bands are under-utilized, there is almost always some bandwidth available for a session to take place. Hence, nodes do not stay in the virtual stack for a long time. Even though the LCFS process is in general considered to be unfair, in our scheme this does not seem to cause an unfair situation. Furthermore, as we mentioned above, in some cases, the upper bands are never used. This implies that there are cases wherein we can reduce the number of bands in the system. As a result, in cases of interference from other co-located wireless systems (e.g., 802.11a networks), even if we avoid using a set of bands, this will not degrade the overall performance of the network. Finally, in our scheme fairness is ensured by limiting the maximum number of packets transmitted upon gaining a band.

5. Related Work on UWB Networks

There is very little prior work on the design of a MAC protocol for multi-band UWB-based wireless networks that support ad hoc communications. The only work that we are aware of is our prior work in Reference [22]. In this previous effort, however, no binary conflict resolution algorithm was used. There have been some studies on single-band MAC layer solutions for ad hoc networks.

5.1. Previous Ad Hoc UWB Schemes

Le Boudec *et al.* [1,12,23] propose a scheme that uses dynamic channel coding. The scheme uses two

types of THS: a receiver-based THS and an invitation-based THS. After the successful transmission of a request using the receiver-based THS, the pair switches to a *unique* invitation-based THS and uses this THS for the duration of the session. In Reference [24] the authors describe approaches towards the development of a THS-based MAC protocol for radio resource sharing in UWB ad hoc networks. Ding *et al.* [7] study issues related to channel acquisition unsuitable for UWB networks. In Reference [25], all nodes share a THS and the receiver broadcasts an invitation, as per this sequence. Potential transmitters compete during a contention period, to lock on to the receiver. In Reference [26], the authors propose a full-duplex access scheme for impulse-based UWB networks. A theoretical treatment on optimal routing, scheduling, and power control appears in Reference [27].

5.2. WPAN Configurations

Most other studies consider master-slave configurations [2,28,29]. The IEEE 805.15.3a task group proposal [28] for media access control is based on the notion of piconets. The master-slave configurations inevitably cause the master node to be a bottleneck. The WiMedia Alliance (MBOA), supports a UWB specification that is based on an OFDM approach [9]. However, the use of OFDM requires (i) frequent complex inverse fast fourier transform computations [4] (ii) simultaneous receiver synchronization with multiple carriers. Yomo *et al.* [2] study the interference between distinct Wireless Personal Area Networks (WPANs) that operate in a master-slave configuration.

5.3. Reservation Based MAC Protocols

The use of reservations for arbitrating access to a plurality of orthogonal bands has been considered in wireless and satellite networks [16,30]. However, the presence of a centralized arbiter (a satellite or base-station) makes allocation much easier as compared to allocation in ad hoc networks. Recently, the use of multiple bands in ad hoc networks that use the IEEE 802.11 MAC protocol has been considered in Reference [31]. However, carrier sensing is possible with IEEE 802.11

and the issues related to MAC access are different from those that arise due to the use of UWB.

6. Conclusions

In this paper, we propose a novel multi-band MAC protocol for use with impulse-based UWB ad hoc networks. The design of our protocol is motivated by the following factors: (a) the use of a multi-band approach provides an inherent flexibility in operation to coexist with other wireless networks, (b) arbitration methods based on the use of time-hopped sequences suffer from inefficiencies due to collisions or large delays, and (c) in the absence of a complex equalizer, due to the effects of the multipath delay spread, the entire UWB spectrum cannot be efficiently utilized by a single band approach. The key idea in our design is the use of a smart variant of the binary conflict resolution algorithm to efficiently handle the allocation of the multiplicity of bands. We perform extensive simulations to demonstrate that our protocol achieves extremely high throughput and much lower latencies as compared to a single band approach wherein no equalizer is available.

Acknowledgement

This work is supported in part by the NSF CAREER Grant No. 0237920 and the NSF NRT grant No. 0335302.

References

1. Le Boudec JY, Merz R, Radunovic B, Widmer J. A MAC protocol for UWB very low power mobile ad-hoc networks based on dynamic channel coding with interference mitigation. EPFL Technical Report ID: IC/2004/02, 01-26-2004.
2. Yomo H, Popovski P, Wijting C, Kovacs IZ, Deblauwe N, Baena AF, Prasad R. Medium access techniques in ultra-wideband ad hoc networks. ETAI, September 2003, Ohrid, FYrom.
3. Roy S, Foerster JR, Somayazulu VS, Leeper DG. Ultrawideband radio design: the promise of high-speed, short-range wireless connectivity. *Proceedings of the IEEE*, February 2004.
4. Ghavami M, Michael LB, Kohno R. *Ultra Wideband Signals and Systems in Communication Engineering*. John Wiley and sons: Chichester, 2004.
5. Win MZ, Scholtz RA. Impulse radio: how it works. *IEEE Communications Letters* 2(1), January 1998.
6. Win MZ, Scholtz RA. Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple access communications. *IEEE Transactions of Communications*, April 2000.
7. Ding J, Zhao L, Medidi S, Sivalingam KM. MAC protocols for ultra-wide-band wireless networks: impact of channel acquisition time. In *SPIE-ITCOM Conference*.
8. Gitlin R, Hayes JF, Weinstein S. *Data Communication Principles*. Plenum Press: New York, 1992.
9. Multiband OFDM Physical Layer Proposal for IEEE 802.15.3a, September 2004, <http://www.multibandofdm.org>
10. Federal Communications Commission: Revision of Part 15 of the Commission's Rules Regarding Ultra-Wideband Transmission Systems, First Report and Order, ET Docket 98-153, 04-2002.
11. Proakis JG. *Digital Communications*, 4th edn. McGraw-Hill: New York, 2001.
12. Le Boudec JY, Merz R, Radunovic B, Widmer J. DCC-MAC: a decentralized MAC protocol for 802.15.4a-like UWB mobile ad-hoc networks based on dynamic channel coding. *Broadnets*, 2004.
13. Garcia JJ, Aceves L, Raju J. Distributed assignment of codes for multihop packet—radio networks. *IEEE MILCOM 1997*, Vol. 1, pp. 450–454.
14. Sousa ES, Silvester JA. Spreading code protocols for distributed spread-spectrum packet radio networks. *IEEE Transactions on Communications* 1998, COM-36, pp. 272–281.
15. De Nardis L, Di Benedetto MG. Medium access control design for UWB communication systems: review and trends. *Journal of Communications and Networks*, December 2003.
16. Pahlavan K, Krishnamurthy P. *Principles of Wireless Networks: A Unified Approach*. Prentice Hall: Englewood Cliffs, NJ, 2002.
17. Romer K. Time synchronization in ad hoc networks. In ACM MOBIHOC 2001.
18. Meier L, Blum P, Thiele L. Internal synchronization of drift-consistent clocks in ad-hoc sensor networks. In ACM Mobihoc 2004.
19. Discrete time communications. IEEE 802.15.3a 480 Mbps wireless personal area networks: achieving a low complexity multi-band implementation, white paper, January 2003.
20. <http://www.tkn.ee.tu-berlin.de/research/trace/trace.html> (accessed 03/10/2005)
21. ANSI/IEEE 802.11 Standard, 1999 Edition.
22. Broustis I, Krishnamurthy SV, Molle M, Faloutsos M, Foerster J. A multiband MAC protocol for impulse-based UWB ad hoc networks. *Proceedings of IEEE SECON*, 2005.
23. Merz R, Le Boudec JY, Widmer J, Radunovic B. A rate-adaptive MAC protocol for low-power ultra-wide band ad hoc networks. *Proceedings of Ad-Hoc Now*, 2004.
24. Cuomo F, Martello C, Baiocchi A, Fabrizio C. Radio resource sharing for ad hoc networking with UWB. *IEEE Journal on Selected Areas in Communications*, December 2002.
25. Hicham A, Souilmi Y, Bonnet C. Self-balanced receiver-oriented MAC for ultra-wide band mobile ad hoc networks. *International Workshop on UWB Systems*, June 2003.
26. Kolenchery S, Townsend J, Freebersyser J. A novel impulse radio network for tactical military wireless communications. In *IEEE MilCom 1998*.
27. Radunovic B, Le Boudec JY. Optimal power control, scheduling and routing in UWB networks. *IEEE Journal on Selected Areas in Communications*, September 2004; 22(7): 1252.
28. IEEE 802.15.3 MAC standard.
29. Cuomo F, Baiocchi A, Capriotti F, Martello C. Radio resource optimisation in an UWB wireless access. *Proceedings of IST Mobile Communications Summit 2002*, Thessaloniki, Greece, June 17–19 2002, pp. 723–727.
30. Krishnamurthy S, Liu C, Gupta V. Medium access control protocols for satellite communications. In *Internetworking and Computing over Satellite Networks*, Zhang Y (ed.). Kluwer, 2003.
31. So J, Vaidya N. Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. *ACM MOBIHOC*, 2004.

Authors' Biographies



Ioannis Broustis received his Diploma in Electronics and Computer Engineering from Technical University of Crete in 2003, and his M.Sc. from University of California, Riverside, in 2005. Currently he is a Ph.D. candidate at the University of California, in Riverside. His research

interests include Wireless Networking, especially mobile ad hoc networks, concentrating on the Medium Access Control (MAC) Layer and the Network Layer. Mostly he deals with wireless testbeds, network modeling, simulation, and implementation, especially for wireless cross-layer schemes.



Mart L. Molle received the M.S. and Ph.D. degrees in Computer Science from UCLA in 1978 and 1981, respectively. Between 1981 and 1994, he was on the faculty of the Department of Computer Science at the University of Toronto. In 1994, he joined the University of California, Riverside where he is a Professor of Computer Science and Engineering,

and served as the Department Chair from 1999 to 2002. Dr Molle's research interests include the performance evaluation of protocols for computer networks and of distributed systems. He is particularly interested in efficiently-solvable analytical modeling techniques, fundamental performance limits, applications of queueing theory, and scheduling theory to distributed algorithms, and model validation through measurement and simulation. He has served as an Editor for the IEEE/ACM Transactions on Networking, and as a Task Force Chair for the IEEE 802.3 Ethernet Standards Working Group.



Srikanth V. Krishnamurthy received his Ph.D degree in electrical and computer engineering from the University of California at San Diego in 1997. He received his B.E.(Hons.) degree in electrical and electronics engineering and the M.Sc (Hons.) degree in physics with distinction from Birla Institute of Technology and Science, Pilani, India in 1992 and the Master of Applied Science degree in electrical and computer engineering from Concordia University, Montreal, Canada in 1994. From 1994 to 1995, he was a Graduate Research Assistant at the Center for Telecommunications Research, Columbia University,

and served as the Department Chair from 1999 to 2002. Dr Molle's research interests include the performance evaluation of protocols for computer networks and of distributed systems. He is particularly interested in efficiently-solvable analytical modeling techniques, fundamental performance limits, applications of queueing theory, and scheduling theory to distributed algorithms, and model validation through measurement and simulation. He has served as an Editor for the IEEE/ACM Transactions on Networking, and as a Task Force Chair for the IEEE 802.3 Ethernet Standards Working Group.

New York. From 1998 to 2000, he was a Research Staff Scientist at the Information Sciences Laboratory, HRL Laboratories, LLC, Malibu, CA. Currently, he is an Assistant Professor of Computer Science at the University of California, Riverside. His research interests span CDMA and TDMA technologies, medium access control protocols for satellite and wireless networks, routing and multicasting in wireless networks, power control, the use of smart antennas and security in wireless networks. Dr. Krishnamurthy has been a PI or a project lead on projects from various DARPA programs including the Fault Tolerant Networks program, the Next Generation Internet program and the Small Unit Operations program. He is the recipient of the NSF CAREER Award from ANI in 2003. He has also co-edited the book 'Ad Hoc Networks: Technologies and Protocols' published by Springer Verlag in 2004. He has served on the program committees of INFOCOM, MOBIHOC, and ICC and is Associate Editor for ACM MC2R.



Michalis Faloutsos is a faculty member at the Computer Science Department in the University of California, Riverside. He got his bachelor's degree at the National Technical University of Athens and his M.Sc and Ph.D. at the University of Toronto. His interests include, Internet

protocols and measurements, multicasting, cellular, and ad-hoc networks. With his two brothers, he co-authored the paper on powerlaws of the Internet topology (SIGCOMM'99), which is in the top 15 most cited papers of 1999. His work has been supported by several NSF and DAPRA grants, including the prestigious NSF CAREER award. He is actively involved in the community as a reviewer and a TPC member in many conferences and journals.



Jeffrey Foerster joined Intel in August 2000 as a Wireless Researcher with Intel Architecture Labs in Hillsboro, Oregon. He is currently focusing on future short- and medium-range wireless technologies, including Ultra-wideband (UWB) technology and related regulations, system design, and performance analysis.

Prior to joining Intel, he worked on Broadband Wireless Access (BWA) systems and standards (IEEE 802.16). He received his B.S., M.S., and Ph.D. degrees from the University of California, San Diego, where his thesis focused on adaptive interference suppression techniques for CDMA systems.