DYNAMMA: A DYNAmic Multi-channel Medium Access Framework for Wireless Ad Hoc Networks

[1]Venkatesh Rajendran, [1]Katia Obraczka and [1,2]J.J. Garcia-Luna-Aceves
[1]University of California at Santa Cruz, Santa Cruz, CA 95064, USA, email: {venkat,katia}@soe.ucsc.edu
[2]Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304, USA, email: jj@soe.ucsc.edu [*]

Abstract

This paper introduces a scheduled-access, multi-channel medium access control (MAC) framework for wireless multi-hop ad hoc networks (MANETs). The proposed framework dubbed DYNAmic Multi-Channel Medium Access, or DYNAMMA, features: (1) ability to dynamically adapt to application-specific traffic patterns, (2) collision-free, multi-channel operation, (3) energy efficiency, and (4) minimum signaling overhead.

We evaluate DYNAMMA through extensive simulations and compare its performance against scheduled-access (e.g., TRAMA [9]) and contention-based (e.g., 802.11) MAC protocols for different application scenarios. Our results show that DYNAMMA's ability to perform collision-free transmission over multiple channels significantly increases system capacity through higher channel utilization and spatial re-use. When compared to TRAMA, DYNAMMA's efficiency in terms of signaling overhead yields considerable energy savings as well as queueing delay reduction. We also present an implementation of DYNAMMA over an Ultra-Wideband (UWB) radio testbed. Our UWB testbed results indicate that DYNAMMA can achieve both high channel utilization (close to 90% for our experiments) and high energy efficiency (nodes, on average, sleep one third of the time).

1 Introduction

The last few years have witnessed significant advances in silicon manufacturing and fabless design which resulting in a steady decrease in form factor coupled with a steady increase in wireless communication capabilities. For instance, highly integrated, small foot-print, single-chip CMOS radios that can support very high data-rates (up to 480Mbps) are now commercially available [1]. Industry is moving towards multi-gigabit data rates using advanced coding/modulation techniques and new technologies [14, 20].

Such technological advances will enable new classes of applications for wireless multi-hop ad hoc networks (MANETs) that impose high throughput– and stringent quality-of-service (QoS) requirements. Some examples include multimedia streaming, large content transfer, emergency response operations, wireless storage area networks, etc.

As a direct consequence of these new physical layer (PHY) technological advances, the fundamental limits challenging development and deployment of high data-rate, QoS-sensitive applications have been moving from the PHY to the medium access control (MAC) layer. At the same time, these advanced PHY techniques often result in higher energy requirements. For instance, the use of multiple RF chains and advanced decoding techniques to improve PHY performance increase power consumption of the receive operation significantly.

As will become clear from the discussion in Section 2, which summarizes the current state-of-the-art in MAC research, there is an urgent need for novel energy-efficient MAC techniques that take advantage of available PHY capabilities to support high-date rate, QoS-sensitive applications. This was the main motivation behind the work presented in this paper, where we introduce the DYNAmic Multi-Channel Medium Access, or DYNAMMA, an energy-efficient, scheduled-access MAC framework for MANETs. Through its (1) ability to dynamically adapt to application-specific traffic patterns, (2) collision-free, multi-channel operation, and (3) minimum signaling overhead, the proposed framework dubbed , DYNAMMA achieves significant (1) increase in system capacity, (2) energy savings, and (3) queuing delay reduction.

[*]This work was supported in part by the US Army Research Office under grant W911NF-05-1-0246, and by the Baskin Chair of Computer Engineering. The opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the Department of Defense. This work has also been partially supported by NSF through awards ANI-0322441 and CNS-0534129.

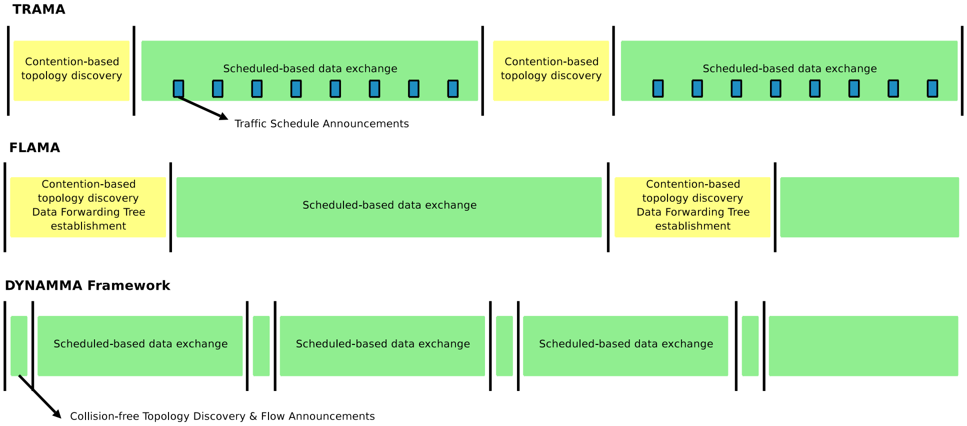

TRAMA
Contention-based topology discovery
Scheduled-based data exchange
Contention-based topology discovery
Scheduled-based data exchange
Traffic Schedule Announcements
FLAMA
Contention-based topology discovery Data Forwarding Tree establishment
Scheduled-based data exchange
Contention-based topology discovery Data Forwarding Tree establishment
DYNAMMA Framework
Scheduled-based data exchange
Scheduled-based data exchange
Scheduled-based data exchange
Collision-free Topology Discovery & Flow Announcements

Figure 1. Various approaches in MAC time slot organization

2 State-of-the-Art

Existing MAC protocols can be categorized as contention-, schedule-, or reservation-based. PAMAS [13] is one of the earliest contention-based proposals to address power efficiency in channel access. PAMAS saves energy by attempting to avoid over-hearing among neighboring nodes. To achieve this, PAMAS uses out-of channel signaling. Woo and Culler [19] address variations of CSMA tailored for sensor networks, and propose an adaptive rate control mechanism to achieve fair bandwidth allocation among sensor network nodes. In the power save (PS) mode in IEEE 802.11 DCF, nodes sleep periodically. Tseng *et al.* [17] investigated three sleep modalities in 802.11 DCF in multi-hop networks. The sensor-MAC protocol [21], or S-MAC, exhibits similar functionality to that of PAMAS and the protocol by Tseng *et al.*. Like the other approaches, S-MAC avoids overhearing and nodes periodically sleep. However, unlike PAMAS, S-MAC uses in-line signaling, and unlike modalities of the PS mode in 802.11 DCF, neighboring nodes can synchronize their sleep schedules. T-MAC [18] is an improvement over S-MAC that adapts the duty cycle based on traffic. However, synchronized listen periods increase channel contention significantly and also increases the overall noise floor during transmissions leading to degradation in link quality.

D-MAC [7] is a contention-based medium access protocol optimized for data gathering applications over unidirectional trees. It schedules transmissions at each hop so that the latency in data collection is reduced. However, D-MAC assumes fixed topology and does not allow multiple data gathering trees. It cannot adapt to other sensor network applications. All of the above mentioned protocols improve energy efficiency by avoiding idle listening. However, they waste energy in (1) collisions due to hidden terminals and (2) carrier-sensing.

In scheduled-access MACs, all nodes are time synchronized and access the medium using well-defined transmission schedules. Thus, Scheduled-access MACs [2,8,10–12] have become an attractive approach to medium access in MANETs due to their potential for improving channel efficiency and increasing energy savings.

The Traffic-Adaptive Medium Access (TRAMA) protocol [10] was the first proposal to implement energy-aware schedule-based medium access. TRAMA addresses energy efficiency by having nodes going into sleep mode if they are not selected to transmit and are not the intended receivers of traffic during a particular time slot. Besides its energy efficiency benefits, TRAMA's use of traffic information also makes it adaptive to the application at hand. However, TRAMA's adaptiveness comes at a price, namely the complexity of its election algorithm and scheduling overhead for announcing traffic information. It should be noted that schedule-based protocols exhibit inherently higher delivery delays when compared to contention-based approaches. In TRAMA, this is exacerbated by the need to propagate schedule information.

Unlike TRAMA [10]), FLAMA [8] does not require explicit schedule announcements during scheduled access periods. Alternatively, application-specific traffic information is exchanged among nodes during random access to reflect the driving application's specific traffic patterns, or *flows*. This allows FLAMA to still adapt to changes in traffic behavior and topology (e.g., node failure).

In both TRAMA and FLAMA, topology information is gathered during the random access period by exchanging signaling packets. Signaling exchange is based on contention-based channel access and is prone to collisions due to hidden terminals. As topology information is critical to establish collision-free transmission schedules, the random access period should be long enough to accommodate signaling packet retransmissions. During the random access period all higher-layer data arrivals are queued and all nodes have their radios in transmit or receive state. Hence, longer random access periods lead to proportional increase in power consumption and buffer requirements. Further, the frequency of the random access period directly impacts the amount of time needed for the network to re-configure whenever there is a topology change. However, higher random access period frequency means longer data delivery delays.

In FLAMA, the transmission schedules are established based on traffic flow information obtained during the random access period. This eliminates the overhead due to explicit traffic schedule announcements and thus improves channel utilization. However, the traffic characterization mechanism used in FLAMA is specific to data gathering applications.

The limitations described above motivate the need for improved topology and traffic discovery mechanisms that: (1) facilitate collision-free signaling exchange, (2) reduce power consumption and buffer size requirements, and still (3) allow for quick re-configuration and adaptability.

As described in detail in Section 3, DYNAMMA provides a flexible, low-overhead, collision-free signaling mechanism for gathering topology and traffic information. Traffic, which is characterized by a set of directed flows, and topology information is exchanged periodically in order to adapt to topological and traffic pattern changes. DYNAMMA's distributed scheduling protocol then uses topology and traffic information to schedule collision-free transmissions across multiple channels. Figure 1 illustrates the different approaches in scheduled-access MAC time slot organization.

Previous approaches to channel access scheduling, establish transmission schedules by electing the highest priority node as the transmitter. The intended receivers for the schedule are decided based on traffic schedule announcements [9] or by pre-establishing who the forwarding nodes are [8]. In DYNAMMA, application traffic is modeled using directed flows that can be directly derived from the destination-based queueing at the MAC layer. Each flow is represented by its arrival rate and its relationship with other incoming flows as explained in Section 3.

The WiMedia MAC targets UWB-based PHY [14] by defining a distributed, time-slotted medium access mechanism [14]. All nodes transmit beacons periodically and the medium access scheme is based on distributed reservations. Applications that require guaranteed service rates can take advantage of the reservation-based structure. However, static reservation-based approaches are not suited to applications with variable service rate. Reservation-based approaches may also lead to fairness problems and increased overhead in creating and maintaining reservations.

All previously mentioned protocols are designed to work with a single channel. Given that most commercially available radios to-date provide multiple orthogonal channels, protocols should make use of this feature to schedule parallel transmissions within a two-hop neighborhood, thus improving overall system capacity.

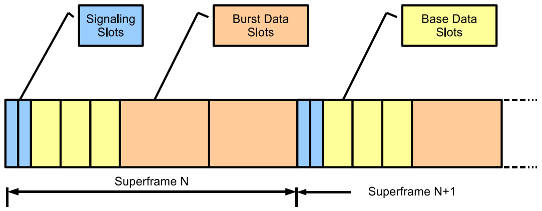


Figure 2. Time slot organization in DYNAMMA

The remainder of this paper is organized as follows. Section 3 describes DYNAMMA in detail and Section 4 presents performance results for DYNAMMA obtained from simulations and testbed experimentation. Finally, in Section 5, we present concluding remarks and directions for future work.

3 DYNAMMA

We summarize the notations used in the description of the DYNAMMA framework in Table 1.

DYNAMMA's time slot organization is illustrated in Figure 2. Time is divided into equally sized time units called *superframes*. DYNAMMA's superframe concept is similar to that of IEEE 802.15.3 [5] and WiMedia MAC [14].

Every superframe consists of a fixed number of time slots. DYNAMMA's time slots can be of three different types, namely: *signaling slots*, *base data slots* and *burst data slots*. *Signaling slots* are used for neighbor/traffic information exchange, while *base data slots* and *burst data slots* are used for data exchange. The channel used for communication is dynamically assigned for every base– or burst data slot.

The duration of *base data slots* and *burst data slots* are fixed based on: the physical layer transmission rate, data packet size, number of data packets to be transmitted within the burst, channel switching time, and radio turn-on time. The duration of a *signaling slot* is based on the maximum signaling frame duration. The proposed superframe structure provides ample support and flexibility for neighbor discovery, traffic adaptation, and dynamic radio mode control to enable system-level energy optimizations.

3.1 Signaling Slot Assignment

Every node in a two-hop neighborhood is assigned its own signaling slot for collision-free signaling informa-

Table 1. DYNAMMA Notations

Number of channels, M	Total number of channel available.
One-hop Neighbors, $\mathbf{N1}(u)$	Set of neighbors of node u that are one-hop away.
Two-hop Neighbors, $\mathbf{N2}(u)$	Set of neighbors of node u which are two-hops away.
Active Flow Set, $\mathbf{AF}(u,t)$	Set of all flows that are active in the two-hop neighborhood of node u in timeslot t.
Required Access Slots, $S_r(f,n)$	Required number of access slots in superframe n for the flow f
Expected Access Slots, $E_r(f,n)$	Expected number of channel access slots in superframe n for the flow f
Channel Utilization Factor, $U(f,n)$	Channel utilization factor in superframe n for the flow f
Channel Utilization Threshold, TH_p	Channel utilization threshold for the flow class p

tion exchange. DYNAMMA's signaling slot assignment is similar to the beaconing slot assignment in the WiMedia MAC [14] [1]. A certain number of free signaling slots, called the *Announce Period*, are maintained for new nodes joining the network. The *Announce Period* is a parameter of the protocol that can be adjusted based on network topology dynamics.

Whenever a new node joins the network, it listens for a certain number of superframes to determine the network's current state, i.e., the start of the superframe, location of the signaling slots, the *Announce Period*, and the signaling slot assignments in the neighborhood. A node randomly selects a free slot in the superframe for signaling and announces it using the *Announce Period*. Signaling announcements allow for dynamic expansion of the signaling period based on the two-hop neighborhood size.

If multiple nodes join the network at the same time, more than one can choose the same signaling slot. A node can determine signaling slot assignment collisions based on the signaling packets transmitted by its one-hop neighbors and move to a different signaling slot to resolve the conflict.

During the initial join period, if the node did not find any signaling transmissions, it can start a superframe structure by selecting the start of the superframe and the location of the signaling slots. The node starts sending signaling packets in the selected signaling slot periodically. If two nodes start at the same time and start sending signaling packets at the same time, they may not discover each other. To prevent this, nodes are required to skip sending signaling packets periodically and listen for any activity during its signaling period.

If a signaling packet is not received from a neighboring node for a certain number of superframes, the node is considered to be inactive and is removed from the neighbor list. Time synchronization across the two-hop neighborhood is achieved using accurate timestamps provided by the PHY. Various known techniques (e.g., [3,4]) can be used for time synchronization.

Signaling frames are transmitted on a well-known channel every superframe. DYNAMMA's current framework can be easily extended to support dynamic channel hopping for signaling exchange. For example, the signaling channel for a particular superframe can be based on a pseudorandom function of the superframe identification number, or *superframeId*.

The total duration of the superframe used for signaling is directly proportional to the size of the two-hop neighborhood and duration of the *Announce Period*. This duration is much smaller when compared to that of TRAMA or FLAMA.

3.2 Traffic and Neighbor Discovery

All nodes are required to be active during signaling slots to gather neighbor– and traffic information. The following signaling information is encoded into the signaling packet: (1) the superframe identifier, or *superframeId*, (2) location of the signaling slot within the superframe, (3) one-hop neighborhood–, and (4) traffic information.

Traffic information is modeled as a set of one-hop flows [8] directed to– or originating from the node. A flow is nothing but a stream of packets originating from a node and destined to one of its one-hop neighbor(s). A flow, which can be unicast, multicast, or broadcast, is characterized by its originating node (or transmitter), destination node(s), and a unique flow identifier.

Flow information is gathered by each node as it propagates every superframe through signaling packets. This provides a flexible mechanism to adapt channel access based on current traffic characteristics.

The number of channel access slots required to service a flow is dependent on the flow's packet arrival rate and service rate. The flow arrival rates and service rates are dependent on the application traffic characteristics and the network topology. To improve channel utilization, channel access slots should not be allocated to a flow that does not have any data to send.

For this reason, in DYNAMMA, traffic flows are classified into different classes depending on their arrival and

[1] However, unlike WiMedia, there is no restriction on the position of the assigned signaling slot.

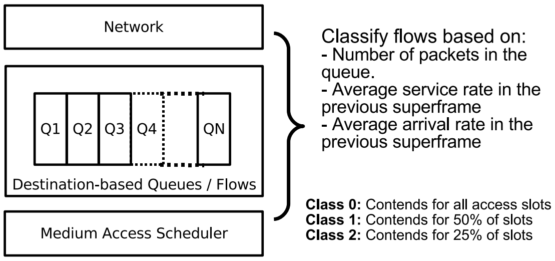

Network
Q1
Q2
Q3
Q4
QN
Destination-based Queues / Flows
Medium Access Scheduler
Classify flows based on:
- Number of packets in the queue.
- Average service rate in the previous superframe
- Average arrival rate in the previous superframe
Class 0: Contends for all access slots
Class 1: Contends for 50% of slots
Class 2: Contends for 25% of slots

Figure 3. Traffic classification

service rates. The number of channel access slots that a flow can contend is decided probabilistically based on its class identifier. In the current implementation we use three flow classes, with class identifiers ranging from 0-2. Class 0 flows are the highest-traffic flows and contend for all channel access slots in the superframe. Class 1 and class 2 flows are flows with reduced traffic and on average they contend for one-half and one-quarter of the superframe, respectively.

All nodes maintain a set of destination-based queues (corresponding to outgoing flows), as shown in Figure 3. Flow classes are assigned based on the number of packets in queue, the average service rate and average arrival rate in the previous superframe. The channel access probability of a flow (f) can be approximated as $1/NumberOfContendingFlows$. The expected number of access slots for the flow, $E_r(f)$ is computed as the product of the channel access probability of the flow and the number of slots in the superframe. The required access slots for the flow, $S_r(f)$ is computed based on the current MAC layer queue parameters. Using the expected and required number of access slots, the fractional usage, $U(f) = S_r(f)/E_r(f)$, is computed for all outgoing flows. Flow classes are then assigned based on a threshold on the flow utilization factor. A flow belongs to class p, if $U(f) > TH_p$, where p is the smallest integer for which the inequality holds. For the current implementation, the class thresholds are fixed at $TH0 = 0.95$, $TH1 = 0.65$, and $TH2 = 0$.

Flow information is encoded in *flow bitmap* format to reduce overhead in flow announcement. The position of the flow in the bitmap is used as the *flow identifier* and the bit is set to 1 to indicate that the flow exists. The originating node identifier and the *flow identifier* are used to uniquely identify a flow. The destinations for the flow are determined based on the *flow identifier* and the ordering of the announced one-hop neighbor list. The flow destinations are known only for the one-hop neighbors, while the *flow identifier* is known for both the one- and two-hop neighbors.

The most significant bit of the bitmap is reserved and is used to indicate a broadcast flow. Multicast destination identifiers are sent as an extension of the one-hop neighbor list and the corresponding bit positions are used for announcing multicast flows.

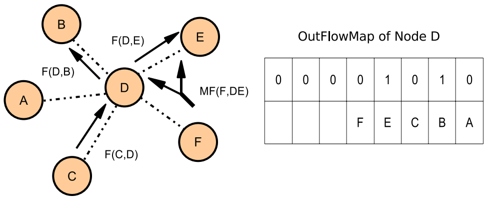


Figure 4. Traffic discovery

The bit-width of the *flow bitmap* determines the maximum number of flows that can be announced by a node. Figure 4 illustrates a simple out-flow bitmap and the corresponding node ordering for a node with two outgoing flows.

Nodes announce both their outgoing- and incoming (originating from a one-hop neighbor) flows. Additionally, nodes also announce all active outgoing flow identifiers of their two-hop neighbors (encoded as a bitmap). This provides all the information required to uniquely identify a two-hop originating flow and is required to avoid hidden terminal collisions.

3.3 Distributed Scheduling Algorithm

Flow and neighborhood information gathered using the signaling packet exchange are used by the distributed scheduling algorithm for establishing collision-free transmission schedules for base and burst data slots.

Whenever a new flow is added to the announcement, a node should ensure that the flow information is propagated to the two-hop neighborhood before activating the flow for distributed election and this can take up to two superframes.

At the start of the base data slot or burst data slot (say t in superframe n), every node executes the election algorithm to determine its state as transmitter, receiver, or sleeping by electing flows from the set of contending flows. The transmission channel is determined using a pseudo-random function (PRF). The algorithm ensures that the receivers of the elected flows are listening on the particular channel decided by the transmitter.

The steps involved in the election process at node u are described below:

- Gather all active contending flows $AF(u,t)$ for the current timeslot t. This includes all the outgoing flows of node u, all the outgoing flows of $N1(u)$, and all the outgoing flows of $N2(u)$ that are currently active. Class 0 flows are active for any timeslot t. For class 1 and class 2 flows, a random number is generated using a pseudo-random function $PRF(flow.srcId, t)$, which is used to decide if the flow contends in the current slot or not.

- Flow priorities are computed as $PRF(flow.srcId, flow.flowId, t, n)$ and the trans-

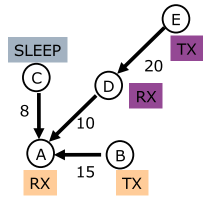


Figure 5. Transmission schedule establishment

mission channel for the flow is computed using $PRF(flow.srcId)\%M$.

- Flows are examined starting from the highest priority flow from the set $AF(u,t)$. The flow can be scheduled for communication if the transmission channel and the initiator/destination(s) are not allocated to a higher priority flow. If the node is the intended originator or destination of a flow that cannot be scheduled due to a conflict, the node can set the state to sleep mode. Additionally, the node elected as a receiver may enter sleep mode if no transmission starts within a preset idle time.

The pseudo code of the distributed election algorithm used in DYNAMMA is presented in Figure 6. The algorithm ensures that when a flow $F(X,Y)$ is activated in channel n, then: (1) any node that is one-hop to the node X cannot receive from in channel n to any other flows, (2) any node that is one-hop to the node Y cannot transmit in channel n, and (3) there is only one transmitter in channel n in the two-hop neighborhood.

As every node has limited flow and topology information, the set of inputs for the algorithm running on different nodes are different. Hence, there can be some inconsistency in the decision made by each node on a particular flow. It is important to ensure that these inconsistencies do not affect the correctness of the algorithm.

Figure 5 illustrates the transmission schedule establishment for a particular time slot. Flows are indicated using arrows and the numbers next to the flows are the flow priorities computed for the current time slot. The highest priority flow $F(E,D)$ is elected first and is assigned a transmit channel based on a pseudo-random function (PRF). The next higher priority flow $F(B,A)$ can be activated as long as the assigned channel is different from the channel assigned to $F(E,D)$. The flow $F(C,A)$ cannot be activated as the destination of the flow is already activated.

3.4 Correctness

In this section we establish that: DYNAMMA transmission schedules are collision-free, and DYNAMMA transmission schedules ensure that nodes do not transmit to a sleeping node or to a node listening on another channel.

For collision freedom, it is enough to ensure that two nodes in the in the two-hop neighborhood does not transmit during the same timeslot in the same channel. If two flows are contending for the same channel in the two-hop neighborhood, the flow with the highest priority is elected as the transmitter. This is ensured by steps 19 and 5 in the pseudo-code description of the algorithm. As there is only one transmitter in a particular channel in the two-hop neighborhood, transmission schedules established by DYNAMMA are collision-free.

In the election process, receiver always listens to the highest priority flow that does not have any channel or transmitter conflicts. A lower priority flow involving this receiver (in the same or different channel), will not not be elected as the receiver (and/or channel) will be black-listed by steps 15 to 21 of the pseudo-code description of the algorithm. Hence, the receiver always listens to the exact flow that can be elected for communication. Note that the flow that the receiver is listening may not be chosen for activated by the transmitter due to a conflict that is hidden from the receiver. This does not affect the correctness of the algorithm and only impact the channel utilization.

A receiver enters sleep mode only if there are no incoming flows or the highest priority incoming flow has a channel or transmitter conflict that is not hidden from the transmitter. This prevents transmissions to a sleeping node.

4 Performance Evaluation

We evaluate the performance of DYNAMMA through simulations and testbed experiments.

4.1 Simulation Setup

In our simulation experiments, we compare the performance of DYNAMMA against both contention-based (represented by IEEE 802.11 DCF [6] and scheduled-access (represented by TRAMA [9]) MACs. Qualnet [16] is used as the simulation platform. The radio model employed is based on the WiMedia physical layer specification [14] for UWB networks. We implemented the radio model using BER lookup tables from Matlab simulations. The UWB physical layer is designed for short range, and high datarate applications. The physical layer supports different data rates ranging from $53.3 Mbps$ to $480 Mbps$ and seven channels (out of which only three are orthogonal). The radio range depends on the data rate and in our simulation we use the base rate of $53.3 Mbps$ with a radio range of about $20m$.

```
Compute $\mathbf{AF}(u,t)$ and sort $\mathbf{AF}(u,t)$ based on descending order of flow priorities.
Initialize $BlackListNodes = \emptyset$; $UsedChannelList = \emptyset$; $u.state = UNKNOWN$;
foreach ($flow \in \mathbf{AF}(u,t)$) begin
  if ($flow.srcId == u$) then : Outgoing flow
    if ($TXCHANNEL(u) \ni UsedChannelList$ && $flow.destId \ni BlackListNodes$) begin
        let $u.state = TX$; $u.txchan = TXCHANNEL(u)$; $u.txflow = flow$;
    else let $u.state = SLEEP$;
    endif
  else if ($flow.destId == u \,||\, flow.destId == ANY\_DEST$) then : Incoming flow
    if ( ($TXCHANNEL(flow.srcId) \ni UsedChannelList$) OR
     ($CONFLICTTX\ hiddenfrom\ flow.srcId$)) then
        if ($flow.destId \ni BlackListNodes$) then
            let $u.state = RX$;$u.rxchan = TXCHANNEL(flow.srcId)$; $u.rxflow = flow$;
        else $u.state = SLEEP$; endif
    else $u.state = SLEEP$; endif
  else if ($flow.srcId \in \mathbf{N1}(u)$) then : One-hop Originated Flow
    let $UsedChannelList = \{UsedChannelList, TXCHANNEL(flow.srcId)\}$;
    let $BlackListNodes = \{BlackListNodes, flow.srcId, flow.destId\}$;
  else : Two-hop or Three-hop Originated Flow
    let $UsedChannelList = \{UsedChannelList, TXCHANNEL(flow.srcId)\}$;
    let $BlackListNodes = \{BlackListNodes, flow.destId\}$;
    if ($flow.srcId \ni \{\mathbf{N1}(u), \mathbf{N2}(u)\}$) then set hidden usage flag; endif
  endif
if ($u.state == UNKNOWN$) then continue ; else break ;
end
```

Figure 6. DYNAMMA election algorithm pseudo-code

4.1.1 Traffic Generation

We consider two different traffic scenarios to illustrate DYNAMMA's application-awareness, i.e., its ability to perform well when subject to different application traffic patterns. In the first scenario, node traffic is statistically generated based on exponentially distributed packet arrivals. For our experiments, we vary arrival rates generating more or less traffic. We chose this traffic pattern as a way to stress-test protocol performance given that a node has flows to all its neighbors and thus contention for the channel is high (especially for flow-based election protocols like DYNAMMA). The second traffic scenario is based on a data gathering application in which all nodes periodically send data to a sink node. For this traffic pattern, data follows a reverse tree (from the leaves to the root) and thus exhibits less contention.

UWB radios are short-range with very tight SNR requirements when compared to a standard 802.11 radios. With short-range radios, the deployments are often hierarchical with small clusters of nodes and a "backbone" to provide connectivity across the groups. Hence, a topology size of 16 nodes with multiple hops is used for all simulation experiments. To ensure connectivity between nodes, a square grid placement with $18m$ separation is followed. For the data gathering application scenario, the data forwarding tree is hard-coded with a static route.

4.1.2 Protocol Parameters and Performance Metrics

We set DYNAMMA's parameters as follows. The duration of the *base data slot* is based on the duration of the maximum PHY payload size of 4095 bytes and SIFS ($10us$). The *burst data slot* is set based on a burst size of 2 transmitted with short preamble. Thus, the base and burst data slots are set to $638.125us$ and $1268.125us$, respectively. In our simulation experiments we followed a static assignment of signaling slots for the ease of implementation. The signaling slots are also grouped together in the superframe and each base slot into divided in to 16 slots. Hence, the signaling overhead is is one base slot per superframe. This can be reduced by using a dynamic signaling scheme, where slots are assigned dynamically based on the network size. The superframe consists of 16 base slots, 238 burst slots, and 16 signaling slots. We vary the the number of channels available for use in DYNAMMA from 1 to 3 in order to quantify the effect of multi-channel scheduling.

For a fair comparison, TRAMA's parameters are optimized for the high data rate physical layer. TRAMA's $SCHEDULE\_INTERVAL$ is set to be 100 transmission slots. The maximum size of a signaling packet is fixed at 96 bytes which results in a slot period of $28.25us$ with guard time to take care of switching. Transmission slots are fixed to support a maximum data fragment size 4095 bytes which results in a slot period of $630.75us$. The random access period is fixed to 10000 signaling slots (0.2825*s*) and is repeated once every 10000 transmission slots (6.3075*s*). TRAMA incurs overhead due to random access period every 6.3075*s* and no data communication takes place during this interval. This can lead to increased queueing drops in TRAMA during random access period.

4.2 Simulation Results

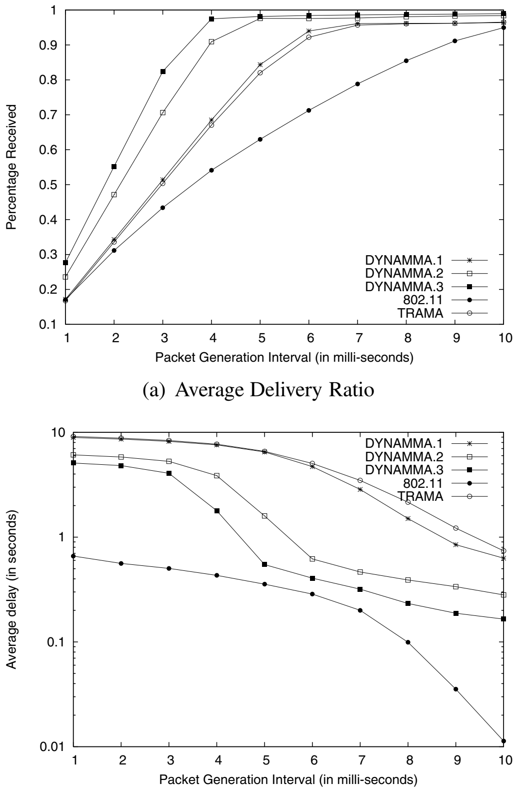
4.2.1 Synthetic Traffic

In this scenario, all nodes generate unicast traffic to a randomly selected next-hop node. The data generation interval is varied from 1*ms* to 12*ms* and the results are averaged over several runs. Figure 7(a) shows the average packet delivery ratio at each node and Figure 7(b) shows the average per-hop queuing delay. The main observation here is that the major source of packet loss in scheduled-access MAC protocols is packet drops due to buffer overflows whereas packet losses in 802.11 are mostly due to hidden terminal collisions. Hence, packet delivery ratio decreases with increased offered load for all the protocols. Due to collision-freedom, both TRAMA and DYNAMMA exhibit higher delivery ratio than the 802.11.

DYNAMMA with single channel (DYNAMMA-1) achieves better queueing delay when compared to TRAMA and this leads to improved delivery ratio. This is a considerable improvement given that one of the main drawbacks of scheduled-access MACs is their inherent queuing delay. This is why DYNAMMA's queuing delay is higher than that of the 802.11.

As we increase the number of channels (DYNAMMA-2 and DYNAMMA-3 for 2– and 3 channels, respectively), there is an improvement in the delivery ratio and queuing delay, more pronounced when the offered traffic is high. However, we noticed that as we increase the number of channels, the efficiency of the scheduling algorithm decreases due to transmitter/receiver conflicts i.e. flows cannot be scheduled as the transmitter or receiver is already a part of another flow on a different channel. However, due to the limited topology and flow information, there is no guarantee that the neighboring flow will be activated. Hence, there is a potential for wasted channel access slots in which none of the flows are active.

Figure 9(a) shows the percentage of time spent by nodes in sleep mode. DYNAMMA implements idle receive timeouts, i.e. the receiver is turned off if the transmission does not start within a timeout. This leads to increased energy savings in DYNAMMA when compared to TRAMA. We also notice that the energy savings decreases as we increase the number of channels. This is due to the fact that nodes spend more time transmitting or listening when more channels are available for communication.



(a) Average Delivery Ratio

(b) Average Queueing Delay

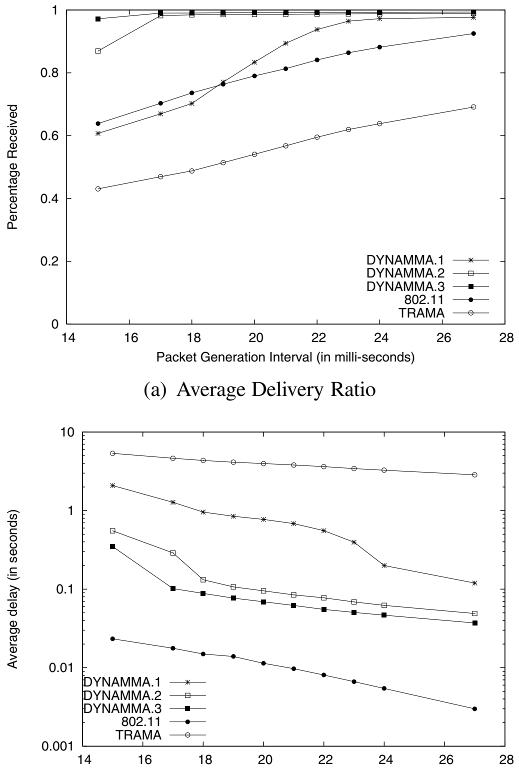
Figure 7. Synthetic Traffic

4.2.2 Data Gathering Application

In this scenario, we place a data sink in one corner of the grid. All nodes generate traffic that is routed to the sink using a data forwarding tree. The routing table for data forwarding is hard-coded for this scenario. The goal of this experiment is to analyze the performance of DYNAMMA when there is a regular traffic pattern. The results highlight the application adaptiveness of DYNAMMA when compared to TRAMA.

Figure 8(a) illustrates the average delivery ratio at the sink node and Figure 8(b) shows the average per-hop delay. TRAMA suffers heavily in this topology due to the high per-hop queueing delay. Periodic random access periods also affect packet delivery and a large number of packets are dropped due to MAC layer queue overflow. This leads to a significant decrease in delivery ratio at the sink. DYNAMMA's inline signaling mechanism and the ability to adapt traffic announcement improve queueing delay significantly when compared to TRAMA. DYNAMMA also outperforms 802.11 in average delivery ratio due to its collision-free scheduling algorithm.

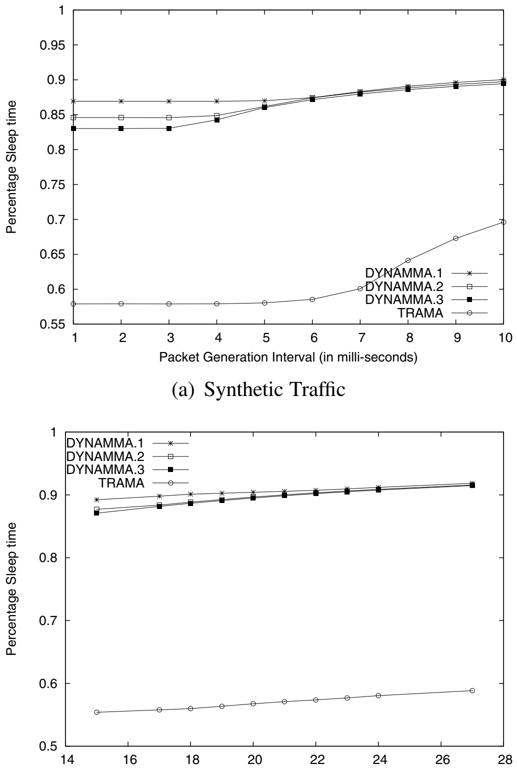
Figure 9(b) compares the energy savings of DYNAMMA and TRAMA. DYNAMMA significantly increases energy



(a) Average Delivery Ratio

(b) Average Queueing Delay

Figure 8. Data Gathering



(a) Synthetic Traffic

(b) Data Gathering

Figure 9. Energy Savings

savings as the scheduling algorithm ensures that nodes sleep when they are not part of an active flow. As we increase the number of channels, the delivery ratio and queueing delay improve due to multiple transmission schedules in the two-hop neighborhood on orthogonal channels. As expected, we see a big improvement in the performance as we go from one channel to two channels. However, beyond two channels the improvement is limited due to transmitter/receiver conflicts.

In single channel scheduling approaches, it is not necessary for the receiver to know the exact transmitter in the one-hop neighborhood. It is enough to ensure that the receiver is listening for transmissions and any node in the one-hop neighborhood of the receiver can transmit. However, in the multi-channel scheduling the channel used for communication depends on the transmitter. Hence, the receiver should exactly know the transmitter in the one-hop. This restriction prevents any arbitrary node that is one-hop to the receiver re-using the medium and leads to some reduced channel re-use.

4.3 Testbed Experiments

The main goal of the testbed experiments is to provide a proof-of-concept implementation of the DYNAMMA framework on a high data-rate physical layer such as UWB. In our setup, we used a Xilinx evaluation board with a Virtex-II pro FPGA. The FPGA has a PowerPC(PPC405) hard macro that can be clocked up to 350MHz and the board supports several customizable interfaces.

The platform uses the UWB radio (RTU7010) daughter board developed by Realtek Semiconductors as shown in Figure 10. The radio implements the WiMedia physical layer standard [14] and supports data rates from $53.3Mbps$ upto $480Mbps$. The radio is controlled through the MAC-PHY interface (MPI) specified by WiMedia [15].

MAC functions are partitioned into hardware and firmware components. The hardware component implements all the time critical functions such as radio mode control, transmit/receive scheduling, and transmit/receive DMA. The firmware component implements all the protocol functions, signaling, queue management, and the flow-based distributed scheduling. The hardware also provides accurate timestamps for packet transmission and reception.

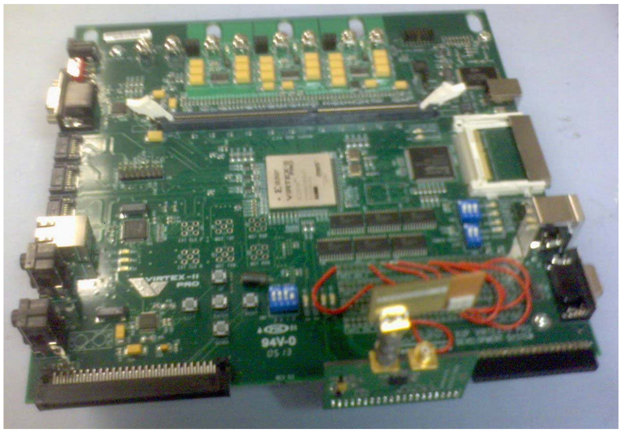


Figure 10. Xilinx FPGA Evaulation Board

The hardware scheduler is responsible for switching the radio mode at appropriate times with resolutions upto 15ns.

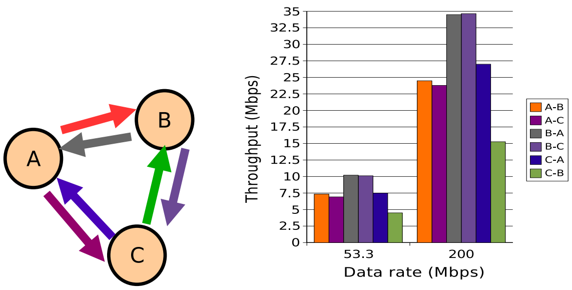
The initial join sequence is implemented as discussed in Section 3.1. The base slot duration is fixed at 644us such that the maximum payload size (4095 bytes) can be supported at the base data-rate (53.3$Mbps$). With in the same duration, a burst of transmissions can be supported with higher physical layer data-rate. This eliminates the need for burst data slots and makes the implementation easier. For example with the same base slot duration, a 3400 byte data burst of size 4 can be supported at 200$Mbps$. The signaling slot duration is 161us i.e. one fourth of the base slot. The superframe is made up of 16 signaling slots and 256 base slots, totaling to 167.440ms.

Time synchronization in the order of us is required to maintain synchronization with the neighbors. The clock drift that can be tolerated over a superframe is in the order of us. To achieve this, we use a 66MHz crystal with less than 10ppm offset.

Time synchronization is performed every superframe based on the timestamp of the received signaling information from neighbors. UWB radios are short range and hence, the propagation delay is negligible when compared to the required synchronization resolution. This facilitates the use of one way timestamps for time synchronization. A node always synchronizes to the slowest neighbor in the one-hop neighborhood.

The difference between the actual signaling packet arrival timestamp and the expected signaling packet arrival timestamp is used to determine the clock offset of a neighbor. If the difference is positive, then the neighbors clock is slower than this nodes clock. If this is the slowest node in the neighborhood the node delays its superframe by this offset.

Signaling packets with flow– and neighborhood information are prepared at the start of the superframe inside the superframe start interrupt handler. Once the node establishes the superframe structure, the periodic time slot interrupt timer is activated. During the time slot interrupt, the distributed scheduling algorithm is executed using current flow– and neighborhood information. As this can take some time for processing, we use the result of the algorithm to schedule the radio mode for the next time slot. This allows for a large processing time to establish the transmission schedules.



(a) Topology (b) Per-flow Throughput

Figure 11. Test-bed Experiments

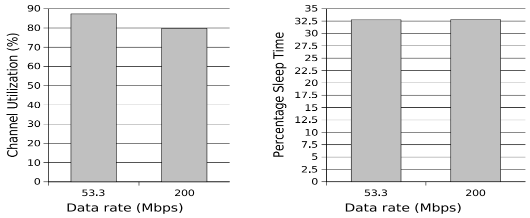


Figure 12. Channel utilization and energy savings

A simple topology of three nodes as shown in Figure 11(a) is considered and a single channel is used for communication. Nodes join the network one by one and each node establishes an outgoing flow to all one-hop neighbors. Data is generated at the MAC layer and the saturated throughput is measured for all the flows. All nodes stop and report their statistics after 10000 superframes. The experiment is repeated with 53.3$Mbps$ and 200$Mbps$ physical layer data-rates. For 200$Mbps$ data-rate, packets are transmitted as a burst of size 4 during every channel access slot.

Figure 11(b) presents the per-flow throughput attained by the flows. As illustrated in Figure 12, DYNAMMA achieves very high channel utilization 87.33% and 79.8% for 53.3$Mbps$ and 200$Mbps$ data rates respectively. The channel utilization is slightly lower of 200$Mbps$ data-rate and this is due to the fact that the overhead due the header increases as the data-rate is increased. As expected, nodes on an average sleep for one thirds of the time as illustrated in Figure 12.

5 Conclusions

In this paper we introduced DYNAMMA, a framework for DYNAmic Multi-channel Medium Access. DYNAMMA models application behavior using directed flows and schedules collision-free transmissions across multiple channels. DYNAMMA reduces energy consumption by switching the radio to low-power standby node whenever the nodes are intended participants of the flows.

We compared DYNAMMA's performance against TRAMA and 802.11 by extensive simulations for a two different application scenarios. It is evident from the simulation results that significant energy savings (upto 90%) with higher delivery ratio when compared to TRAMA and 802.11. DYNAMMA's multi-channel approach improves the channel utilization while reducing energy consumption. DYNAMMA can adapt to application traffic using minimal signaling exchange in the form of flow bitmaps.

We also presented a prototype implementation of DYNAMMA on a UWB MAC testbed. Testbed results indicate that DYNAMMA framework can achieve very high channel utilization (upto 87.33%) with considerable energy savings on a high data-rate physical layer technology.

Acknowledgment

The authors would like to thank **Wionics Research - Realtek Group** for providing the UWB radios, Lower-MAC IP and the FPGA hardware for the test-bed experiments. The authors would like to thank Turgut Aytur, Ran-Hong Yan, Ping-Wen Ong, Stephan ten Brink and Ravi Shankar Mahadevappa for their inputs in the UWB testbed.

References

[1] T. Aytur, H.-C. Kang, R. Mahadevappa, M. Altintas, S. ten Brink, T. Diep, C.-C. Hsu, C.-C. Lee, R.-H. Yan, and B. Razavi. A fully integrated UWB PHY in 0.13um CMOS. In *Solid-State Circuits Conference, Digest of Technical Papers, ISSCC 2006*, Feb 2006.

[2] L. Bao and J. J. Garcia-Luna-Aceves. A new approach to channel access scheduling for ad hoc networks. In *The seventh annual international conference on Mobile computing and networking 2001*, pages 210–221, 2001.

[3] E.D.Kaplan. *Understanding GPS: Principles and Applications*. Artech House, 1996.

[4] J. Elson and D. Estrin. Time synchronization for wireless sensor networks. In *IPDPS 2001*, April 2001.

[5] IEEE. IEEE Standard for Information Technology - Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for high rate Wireless Personal Area Networks (WPANs). IEEE Std 802.15.3-2003.

[6] IEEE. IEEE Standard for Information Technology: Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications. ANSI/IEEE Standard 802.11, 1999 Edition, 1999.

[7] G. Lu, B. Krishnamachari, and C. Raghavendra. An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks. In *Int. Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, Santa Fe, NM, Apr. 2004.

[8] V. Rajendran, J. J. Garcia-Luna-Aceves, and K. Obraczka. Energy-efficient, application-aware medium access for sensor networks. In *Proceedings of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems*. IEEE, 2005.

[9] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient collision-free medium access control for wireless sensor networks. In *Proceedings of the first international conference on Embedded networked sensor systems*, pages 181–192. ACM Press, 2003.

[10] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, 12:63 – 78, Feb 2006.

[11] S. Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 5(2):81–94, 1999.

[12] R. Rozovsky and P. R. Kumar. SEEDEX: a MAC protocol for ad hoc networks. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 67–75, New York, NY, USA, 2001. ACM Press.

[13] S. Singh and C. Raghavendra. PAMAS: Power aware multi-access protocol with signaling for ad hoc networks, 1999.

[14] High rate ultra wideband PHY and MAC standard. ECMA Internal Standard ECMA-368, December 2005.

[15] MAC-PHY interface for ECMA-368. ECMA Internal Standard ECMA-369, December 2005.

[16] Scalable networks, http://www.scalble-networks.com.

[17] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh. Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks. In *Proceed-ings of the IEEE Infocom*, June 2002.

[18] T. van Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the first international conference on Embedded networked sensor systems*, pages 171–180. ACM Press, 2003.

[19] A. Woo and D. Culler. A transmission control scheme for media access in sensor networks. *ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom) 2001*, 2001.

[20] Y. Xiao. IEEE 802.11n: enhancements for higher throughput in wireless LANs. *Wireless Communications, IEEE*, 12:82 – 91, Dec 2005.

[21] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *IEEE Infocom 2002*, June 2002.