Real-Time Communication in IEEE 802.11 Mobile Ad hoc Networks
A Feasibility Study

Barbara Hughes, Mark Gleeson, Marcin Karpinski, Raymond Cunningham and Vinny Cahill
Distributed Systems Group,
Department of Computer Science, Trinity College, Dublin, Ireland
{barbara.hughes, mark.gleeson, marcin.karpinski, raymond.cunningham, vinny.cahill}@cs.tcd.ie

Abstract

Achieving predictable communication latency in an ad hoc IEEE 802.11 wireless local area network necessitates an approach that overcomes the impact of the underlying non-deterministic contention-based medium access control (MAC) protocol. In this paper, we assess the feasibility of using a Time Division Multiple Access (TDMA) layer above the IEEE 802.11 MAC protocol in order to achieve such predictable communication latency.

We present the design and implementation of this TDMA layer and describe the influence of both the characteristics of the IEEE 802.11 MAC protocol and of the dynamics of an ad hoc network on its design. We show that our approach can yield the predictability and stability required to support real-time communication in a real ad hoc environment, typified by dynamic host mobility and varied offered load, subject to identified constraints.

1. Introduction

With increased research in ad hoc networks in recent years new application domains such as communication between mobile robots [10] and inter-vehicle communication [4] have evolved. Timely wireless communication is essential to allow applications in these domains to be realised.

The IEEE 802.11 wireless standards are dominant in the wireless local area networking (WLAN) community today, and recent surveys suggest that IEEE 802.11 standards (particularly g), will dominate the WLAN landscape for years to come [12].

The current IEEE 802.11 wireless ad hoc protocol employs a contention-based approach to medium access control (MAC). A carrier sense multiple access and collision avoidance (CSMA/CA) mechanism with a random backoff scheme are used to reduce contention for medium access and therefore packet collisions. However, due to the use of this contention-based MAC layer, the possibility of packet collisions is not eliminated. Moreover, the unpredictability of medium access control in IEEE 802.11 is exacerbated in a mobile ad hoc domain, where the absence of a fixed infrastructure means hosts themselves constitute the communication infrastructure - a host must act as both a router and an end host. As hosts move in and out of range of other hosts the link quality, network topology and number of hosts competing for wireless access changes dynamically [13]. To maintain consistent topological views, for example to provide real-time multi-hop communication, topology updates must somehow be communicated to other hosts, increasing the competition for wireless medium access and the potential for packet collisions [1]. Thus, IEEE 802.11 is unable to guarantee the timely access to the wireless medium that is required for real-time communication in a dynamic ad hoc network.

Reducing the unpredictability of contention-based MAC protocols in fixed networks has been the subject of previous research, for example, [2, 9], where contention can be avoided if competition to gain access to the medium is removed by a higher layer. One such approach is to employ a schedule-based MAC protocol, for example, Time Division Multiple Access, (TDMA), [11], where hosts negotiate amongst themselves the time at which each host should transmit. Using this method, if all hosts agree on and adhere to a negotiated schedule, competition for medium access is removed and predictable medium access latency achieved. TDMA is an established approach to providing timely communication in wired networks, where the number of participants is static and the network load remains within a known load hypothesis [6]. The dynamics of a mobile wireless ad hoc environment coupled with the underlying characteristics of the IEEE 802.11 MAC protocol, increase the complexity of maintaining correct and consistent operation of TDMA in this dynamic wireless domain.

In this paper, we assess the feasibility of this approach in a dynamic ad hoc network. We present the design and implementation of a TDMA layer above the IEEE 802.11

wireless ad hoc protocol and describe the influence of the characteristics of the IEEE 802.11 MAC protocol and the unrestricted dynamics of an ad hoc domain on the design of this TDMA layer. Furthermore, we show that our approach can yield the predictability and stability required to support real-time communication in a real world ad hoc environment, typified by dynamic host mobility and varied offered load.

In the next section, the contention-based approach to medium access used by IEEE 802.11 is described, and with the aid of experimental evaluation, the unpredictable packet transmission latency inherent in this approach is presented. Following this, we describe the TDMA protocol, and present the steps involved, and the design decisions required at each step, to implement TDMA as a layer above IEEE 802.11 in a real world wireless ad hoc domain to mitigate the characteristics of a contention-based MAC. In section 4, an experimental evaluation of the TDMA implementation is presented showing that predictable communication latency is possible, regardless of the unrestricted dynamics of the target domain, and follow this with a review of related work on attempting to achieve real-time guarantees in a wireless domain. We finish the paper with some conclusions and a discussion of future work.

2. Non-determinism with contention based medium access

Prior to describing our TDMA implementation, we provide a brief overview of the contention-based MAC protocol used by IEEE 802.11, and highlight, using real world experimental results, the sources of non-determinism that lead to the unpredictable transmission latency inherent in this approach.

In contention-based medium access, hosts with packets to transmit compete amongst themselves to gain access to the wireless medium. This competition occurs prior to the transmission of each packet. The IEEE 802.11 protocol for medium access control in ad hoc networks, the Distributed Coordination Function (DCF) [5], relies on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), to determine whether or not a mobile host may gain access to the wireless medium.

To use collision avoidance, a host with a packet to transmit senses the medium and, if it is busy waits a random amount of time (the backoff interval) prior to attempting the transmission. Each host uses a contention window (CW) and generates a random number between zero and CW as the backoff interval. A host freezes a countdown if a transmission from a different host is initiated and restarts the countdown when this transmission has completed. Only when the backoff counter reaches zero and the medium remains free for a DCF Inter Frame Space (DIFS) interval, may a transmission take place. However, if the medium is still busy at this time, CW is increased, and the countdown commences again.

Using this collision avoidance technique, concurrent transmissions, and thus, packet collisions, are reduced. However, the possibility of collisions still exists amongst all hosts that finish their backoff countdown at the same time, perceive the medium to be free for the DIFS interval, and proceed to transmit. Thus, deterministic medium access is not possible in this scenario. In addition, given that the backoff countdown may be deferred a number of times, with each deferral reflecting the length of an ongoing transmission, it is not possible to predict the *actual* maximum backoff interval that will precede a packet transmission in an unrestricted ad hoc wireless environment.

Furthermore, using IEEE 802.11, a bounded number of retransmissions are attempted for each colliding packet, with each retransmission subject to the collision avoidance mechanism outlined previously. In non real-time communication, retransmissions are beneficial as the probability of *eventual* transmission is increased. However, in real-time communication, where transmission at a known deadline or within a known period, is required, a retransmission, i.e., a delayed transmission, may have adverse implications, e.g., out of date information may be worse than no transmission at all. Retransmissions compete for wireless medium access, increase the probability of packet collisions and the potential for unpredictable backoff intervals.

2.1. IEEE 802.11 medium access evaluation

To assess the effects of these IEEE 802.11 MAC characteristics a real world evaluation was conducted, using pairwise communication between four Pentium III notebooks running at 1 Gigahertz, with 512K L2 cache, 256MB RAM and wireless communication was achieved using Lucent ORiNOCO 11Mbit/s Gold cards. Each notebook ran an identical version of the Red Hat Linux 7.3 operating system. Unicast and broadcast results are presented in Figures 1 and 2 to highlight the difference in these transmission mechanisms. The broadcast results are the main focus of this paper.

All experiments were executed using fixed packet sizes of 512 bytes with varying offered load, where offered load is interpreted as the percentage of nominal bandwidth of the medium (11Mb/s) presented for transmission. For example, at 100% offered load 11Mb/s is presented for transmission. Figure 1, illustrates the impact of increasing offered load on achievable round-trip times. It is clear that as the volume of transmissions increase, to approximately 70% (in broadcast mode), the round trip times increase dramatically from that at lower offered load percentages. This variance may be attributed to the increase in prolonged and repeated backoff

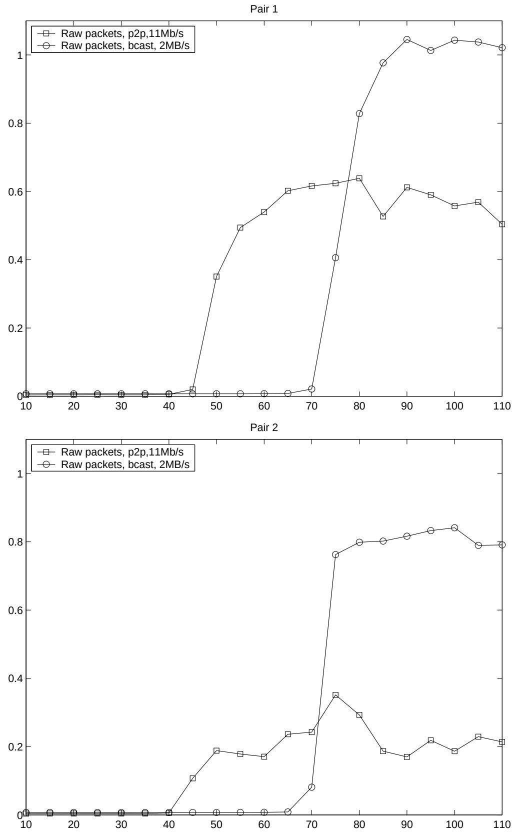

Figure 1. Round-trip time with four stations

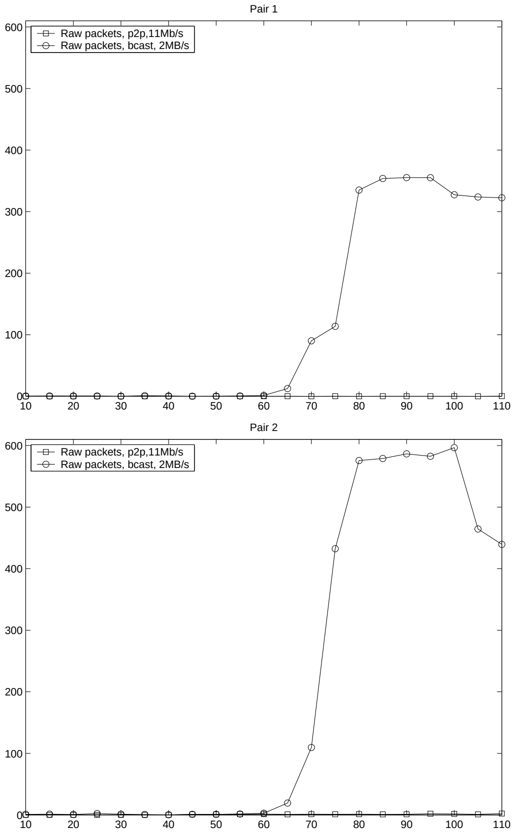

Figure 2. Round-trip packet loss with four stations

intervals caused by the increase in wireless activity at these higher offered loads. Given the nature of broadcast mode transmission, it is unsurprising that both wireless pairs are similarly effected. The duration and number of these back-off intervals is impacted by the state of the wireless environment when a transmission is required, i.e., the contention for the medium at this time, and thus, is not deterministic or suitable for real-time applications.

An important observation, from Figure 2, is the increase in lost packets with the increasing offered load, which has a close correlation with the increasing round trip times in Figure 1. As the offered load increases, collisions increase leading to an increase in the volume of packet retransmissions, which in turn heighten the competition for access to the wireless medium, exacerbating the contention and collision problem further. In this scenario, there is a high probability that the retransmission limit, imposed by IEEE 802.11, will be reached prior to the successful transmission of the packet. If this is the case, the packet is dropped, and appears as a lost packet in Figure 2.

In summary, our results highlight the variability of the IEEE 802.11 MAC protocol in a dynamic ad hoc domain.

3. Implementing the TDMA layer

In TDMA, a slot (or the slot time), is a logical representation of a known and bounded period of time, the duration of which represents the maximum interval within which a specific host, i.e., the host allocated use of the slot, may transmit. The accumulation of a known number of slots of a specific size is a cycle. The maximum periodicity within which a specific slot will repeat is known, and calculated as: L = *slot_time* $\times\ (N-1)$, where N is the upper bound on the number of slots in a cycle. Thus, the maximum latency for a host with (at least) one slot allocated to have exclusive access to the wireless medium is also L.

A prerequisite for the correct operation of the TDMA protocol, i.e., packet transmission at a known time by a known host, is distributed agreement among participating hosts[1]on the division of the wireless medium into slots. In addition, each slot starts at a known time and lasts for a known and bounded duration. To achieve this prerequisite

[1]A participating host is executing the TDMA protocol and has at least one slot allocated for exclusive use.

the local clocks of all participating hosts must remain synchronised to a known precision. Furthermore, to guarantee ongoing correct operation of the protocol, all hosts must share a consistent view of the current time slot allocations and guarantee to transmit within the bounds of their allocated time slots only. Thus, there is an implicit requirement that a transmission is fully encapsulated in the allocated slot only.

To achieve these prerequisites for the operation of TDMA in a real IEEE 802.11 wireless environment, where the characteristics of the MAC protocol and the dynamics of an unrestricted mobile ad hoc environment, impact the timeliness and predictability achievable, is difficult. In the remainder of this section, the design of a TDMA implementation above the IEEE 802.11 MAC protocol is described.

3.1. System model

As described in this paper our TDMA implementation supports a known set of hosts where each participating host is bootstrapped with an initial slot pre-allocated, i.e., each host is guaranteed exclusive access to the wireless medium at least once in a cycle, although additional slots can be allocated dynamically to any host. The number of additional slots requested per host and the number of transmissions by a host per cycle is varied to support fluctuating offered load. Hosts are mobile but only one-hop communication is considered since multi-hop communication would require the use of a (real-time) routing protocol, which is beyond the scope of this study.

3.2. Slot design

A characteristic of the IEEE 802.11 MAC protocol, and particularly CSMA/CA, is that the medium must be free for a minimum of the DIFS interval, i.e., 50μs, prior to each transmission. Accommodating this characteristic in a slotted medium access method necessitates a guarantee that there is a minimum 50μs interval before the start of each slot during which the medium is idle. If this condition is guaranteed, the underlying CSMA/CA contention based protocol will allow transmission at the start of the next allocated slot is possible. To satisfy this constraint, a slot structure which guarantees a minimum of 50μs at the end of the slot, the *guard space*, within which no wireless transmission are allowed to take place, is used, see Figure 3.

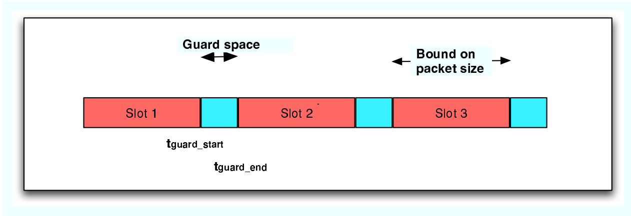

Figure 3. Design of a TDMA slot

An important requirement is that the slot time remaining when the guard space is discounted is of sufficient duration to ensure that the transmission of a packet is wholly encapsulated in that one slot. Thus, there is a requirement of a maximum bound on the packet size for transmission, i.e., the packet must be wholly encapsulated in one slot, with the guard space remaining free from any wireless transmissions. If a slot is guaranteed to encapsulate at most one transmission and the guard space is always free from transmissions, there should be no contention for the wireless medium during this transmission.

A scheduled transmission in a slot may be delayed by external wireless interference by other IEEE 802.11 hosts not participating in the TDMA protocol, or simply, other wireless devices transmitting in the transmission range of a participating host. Thus, satisfying the requirement that packets be fully encapsulated within a single slot is difficult given the unpredictable number and duration of backoff intervals of the underlying IEEE 802.11 MAC protocol resulting from such external interference.

Selecting a slot size that is sufficiently large to encapsulate the maximum number of IEEE 802.11 retransmission attempts (e.g., 4 for ORiNOCO cards), at maximum CW size for each attempt, reduces the probability of delayed transmissions. Using this approach there is a high probability that retransmissions will also be encapsulated in the scheduled slot with no impact on subsequent transmission in other slots. However, this approach still cannot provide a guarantee that a transmission will not be delayed. The *actual* backoff interval depends on how often the medium is perceived to be busy, during which the backoff counter for all other hosts is frozen. Therefore, the backoff interval cannot be determined in advance. Thus, even with a large slot size, delayed transmissions are possible.

3.3. Slot management

To maintain distributed agreement on the order of slot allocations and deallocations, updates to the management of slots must be totally ordered, so that two or more requests are guaranteed to be received and processed in the same order by all participating hosts. Guaranteeing totally-ordered message delivery in an ad hoc wireless network where the mobile hosts constitute the communication infrastructure and the connectivity and network topology changes dynamically over time, requires special consideration.

The approach employed is based on Cristian's synchronous atomic broadcast protocol [3]. To explain how the synchronous atomic protocol works, consider a host with one slot allocated that wishes another slot to be allocated to it. The mobile host creates a message requesting a slot, including information related to the atomic broadcast such as the mobile host's address, a sequence number and a timestamp. The mobile host then broadcasts this message a number of times by piggybacking this request onto data transmissions in its allocated slot.

A receiving mobile host stores this packet if it has not received it before based on the sequence number. In addition to storing the packet, the receiving host also rebroadcasts the packet in its allocated slot until the delivery time of the packet arrives.

The delivery time of the message is equal to the original timestamp plus the delay to delivery, Δ, which is a parameter of the atomic broadcast protocol. In this case, Δ is set as 2 cycles, thus, accommodating one rebroadcast transmission failure. When the delivery time of the message arrives, all the mobile hosts update their information consistently and allocate the mobile host a new slot.

Using this atomic broadcast protocol the probability that all participating hosts receive a slot management request is increased, but, given the characteristics of a dynamic ad hoc network, cannot be guaranteed, under the hypothesis of accommodating one lost message only.

3.4. Clock synchronisation

As noted, distributed clock synchronisation is a prerequisite for TDMA implementation. Distributed agreement on slot times, slot allocations and the continued consistency of transmissions in allocated slots are maintained relative to the precision of the local clocks of all TDMA participants.

One approach to clock synchronisation is to synchronise with a master node. The master station broadcasts the current clock time periodically[2]. Prior to calculating the value of the adjustment required by the local clock, the slave updates the time received from the master with the propagation delay to reach this host[3].

A clock synchronisation accuracy of within 50μs is desirable as this is the minimum gap within which the medium must be guaranteed to be free using IEEE 802.11 CSMA/CA, and thus, the minimum bound on the guard space at the end of each slot.

[2] Piggybacked with other transmissions.

[3] Propagation delay was calculated using experimental evaluation of the one-way latency of packet transmission in a benign environment.

Algorithm 1 Synchronise with master

1: $-25\mu s \leq primary\_bound \leq +25\mu s$
2: $secondary\_bound > primary\_bound$
3: $\Delta T = (t_{local} - t_{received})$
4: **if** $\Delta T < secondary\_bound$ and $\Delta T > primary\_bound$ **then**
5:    $t_{local} \pm primary\_bound$
6:    store $(\Delta T - primary\_bound)$
7: **else if** $\Delta T < primary\_bound$ **then**
8:    $t_{local} \pm \Delta T$
9: **else if** $\Delta T > secondary\_bound$ **then**
10:    no clock adjustment
11: **end if**

As **Algorithm 1** illustrates, a slave adjusts the local clock by, a minimum of ΔT, i.e., the calculated difference between the master clock time and the slave's local time, and a maximum, of the *primary_bound*. In the latter case, if Δ is greater than *primary_bound*, the difference of the two is stored and merged with the master clock time received in the next cycle. Lines 9-10, handle the case where there is a divergence between the master and slave clocks greater than the defined *secondary_bound*, e.g., due to a delayed packet. In this scenario, the clock of the slave is perceived as too far adrift of the master clock and no adjustments made.

Adjustments to a slave clock are always bounded, (by *primary_bound*) to achieve a smooth progression towards synchronisation. Using experimental evaluation, it was discovered that a local clock adjustment of $\pm$ 25μs was the maximum single adjustment possible to guarantee the correct operation of the underlying RTAI system, e.g., the functionality of ongoing periodic real-time tasks. In addition, in $<$ 1% of clock synchronisation cycles, a slave became sufficiently out of synhronisation with the master, to play no further part in the protocol. The higher the frequency of clock synchronisation packets from the master, i.e., the shorter the cycle time, the lower the clock drift possible from the master time.

3.5. Real-time Linux network subsystem

To implement timely medium access requires predictable behaviour at all phases of packet send and receive, thus, a further prerequisite is to implement a deterministic Linux network subsystem to achieve predictable asynchronous packet transmission and reception, and thus, a basis upon which the TDMA layer could be implemented.

A general-purpose Linux operating system has three main sources of unpredictability effecting packet transmission and reception: *(1)* a dependence on the dynamic allocation of `socket_buffers` for packet transmission and reception; *(2)* interruptible interrupt dispatching and interrupt service routine (ISR) execution, and *(3)* queuing received packets for future notification to higher layers. With RTAI [8] providing our real-time Linux environment, the next step was to implement a real-time network subsystem, RT-WLAN, to address these three sources of unpredictability.

1. **Predictable socket buffer allocation**
   Real-time memory management design must guarantee that memory is available and maintained in physical RAM. A `MemoryManager` module was implemented to create a pool of `RTsocket_buff` structures (a real-time `socket_buffer` abstraction), implemented as a fixed size doubly linked list which is guaranteed to remain in scope until explicitly removed. Using the `MemoryManager` interface, `RTsocket_buff` structures are available to all other modules in RT-WLAN.

2. **Timely interrupt handling**
   RTAI provides a framework for a timely interrupt dispatcher by diverting interrupt handling to RTAI and then Linux, by registering a real-time interrupt handler, which in the current context translates to diverting interrupts to RTAI and then PCMCIA. With timely interrupt dispatching available, the next step is to guarantee that interrupt service routines are timely. Thus, all interrupt service routines execute with interrupts disabled, guaranteeing that all interrupts are serviced to completion, with a predictable latency.

3. **Timely notification of packet reception**
   Eliminating unpredictable interrupt handling removes one source of uncertainty in packet reception. The implementation of a real-time queuing strategy removes the other. The `RTDeviceWrapper` module provides an interface to `RTnetif_rx`, the real-time version of `netif_rx`. Using this interface, a real-time memory pool, representing the pending packets, is maintained. This real-time memory pool is updated during the non-interruptible packet reception interrupt service routine, `RTorinoco_ev_rx`, within which immediate notification of packet availability is made to the higher layers. Combining real-time interrupt handling with predictable packet queuing provides real-time packet reception notification to higher layers.

3.6. Real-time transmission in an allocated slot

Assuming local clocks of all participants are synchronised to a known precision, all hosts have a consistent view of the current slot allocations and slot times, and a real-time network subsystem is available, the remaining consideration is how to transmit in an allocated slot.

Each participating host creates a real-time task that executes at the periodicity of the current slot time. For example, if the slot time is of 2ms duration, the real-time task also has a 2ms periodicity. Thus, the task is ready to transmit at the start of each slot. Furthermore, if the current slot is allocated to the local host a packet for transmission is always available to the task at the start of the slot time, and the transmission of the current packet is initiated in the current slot.

To summarise, using a combination of slot design, distributed slot management and clock synchronisation and interfacing with a real-time network subsystem, a TDMA layer above IEEE 802.11 was implemented, to reduce the influence of the characteristics of both the associated contention-based MAC and the dynamics of the targeted ad hoc environment.

4. Evaluation of the TDMA implementation

The evaluation of the TDMA implementation is in two parts. First, the evaluation of RT-WLAN is presented highlighting the predictability of the underlying network subsystem, and is followed by an experimental evaluation of the full TDMA implementation illustrating deterministic packet transmission and predictable slot allocation are achievable irrespective of the dynamics of the ad hoc environment, subject to identified constraints.

4.1. RT-WLAN evaluation

Our evaluation is based on experiments performed using the real-time `RTorinoco` network drivers and interfacing with RT-WLAN kernel modules. The RT-WLAN network subsystem was run on the notebooks used in the evaluation of IEEE 802.11 packet transmission in section 2. Each notebook executed with the kernel 2.4.20 patch and the RTAI release version 3.0 applied. Throughout, these experiments all packet transmissions are broadcast.

For our analysis of packet transmission, three sources of transmission latency prior to the packet leaving the wireless card were observed: *software latency*, incurred in the interval from the initial transmission request to the transfer of the packet to the device driver queue; *firmware latency*, incurred in the transfer of the packet to the physical wireless card, and finally, *communication latency*, incurred awaiting wireless medium access using the IEEE 802.11 contention-based MAC protocol.

Using RT-WLAN, the objective is to obtain a predictable software latency regardless of the current offered load. As shown in Table 1, this objective was achieved and maintained regardless of the increase in offered load, illustrated by the decreasing period between packet transmissions. These experiments were repeated in environments with high

Transmission period	Software	Firmware
20ms	384μs	120μs
5ms	387μs	122μs
3ms	389μs	124μs
1.5ms	388μs	159μs

Table 1. Average software and firmware latency

levels of wireless interference with similar results obtained with a maximum variance of $\pm 5\mu$ from the illustrated average cases. The firmware latency is not under software control, but as illustrated in our packet transmission analysis, a maximum variance of 30μs under high offered load is possible. Thus, a known and tolerable upper bound is available. With both predictable software and firmware latency achieved, the communication latency, i.e., the phase of packet transmission under the influence of the IEEE 802.11 MAC protocol, is the main source of unpredictability for wireless packet transmission latency that remains.

4.2. Communication within specified time bounds

To reduce the probability of delayed transmissions, as discussed in section 3.2, the slot size in these experiments is based on the observed average latency for packet transmission of a specific data packet size, over time. For example, using a packet size of 244 bytes, (the lower bound packet size in these experiments), the observed one-way latency, i.e., the time for a broadcast transmission to reach a participating host is, (on average), 1.791ms, which accumulates to a lower bound slot size of 2ms, when guard space is included.

In each experimental execution included in this section the number of available slots is fixed at 10. A designated master node, (host 1), is allocated slot 1 with remaining slots allocated to the slave hosts in a round-robin fashion, e.g., host 3 is allocated slot 3 etc. Both one-way and round-trip latency are measured. Each experiment commences with a broadcast transmission in the first slot by the master. To measure one-way latency the duration from the initial send invocation, by the master, to the reception of the transmission by any slave host is recorded. To measure the round-trip latency the reply by each slave host in the designated slot is included also. For example, assuming a slot size of 2ms, the earliest time at which host 3 can initiate a reply is in slot 3, i.e., a maximum of 4ms after the initial transmission by the master host.

The achievable round-trip latency per host are illustrated in Figure 4. As expected, the round-trip communication

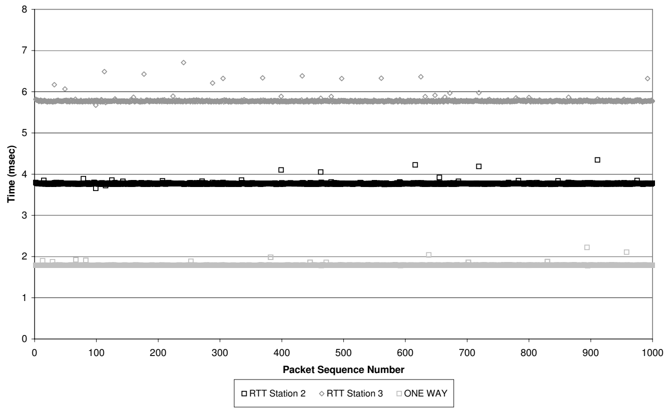

Figure 4. TDMA with 2ms slot times and 3 hosts

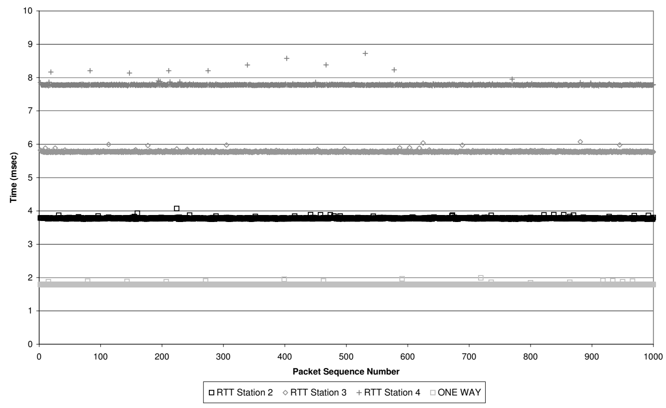

Figure 5. TDMA with 2ms slot times and 4 hosts

latency per slave host is within the theoretical bounds of twice the packet transmission send time, i.e., approximately 2(2ms)[4], and the latency to wait until the scheduled slot for each host, i.e., (N - 2) $\times$ 2ms slot size, where N corresponds to the start of the designated slot, and the master host slot is not included in the calculation. In Figure 5, the number of participating hosts was increased to 4 and the results obtained were comparable to those available for three participating hosts, highlighting that communication within known time-bounds is not effected by the number of participating hosts.

A small amount of jitter is in evidence in some of these results, e.g., host 3 in Figure 4. It is assumed that this jitter is attributed to the initiation of the underlying CSMA backoff

[4] The reply transmission does not alter the packet size.

procedure if there is any wireless interference in the transmission range of the participating TDMA hosts involved in these experiments. In this case, scheduled transmissions are delayed by extraneous wireless transmissions. To investigate the extent to which this jitter is related to wireless interference and the underlying non-determinstic backoff duration, the slot size was extended to 4ms, in Figure 6, with the previous packet size of 244 bytes maintained. As expected, the extension of the slot size increases the probability that a transmission, regardless of backoff duration, will be wholly encapsulated in one slot, thus, reducing the delay for scheduled slot transmission and the impact on subsequent slots as indicated by the substantial reduction of jitter for all hosts.

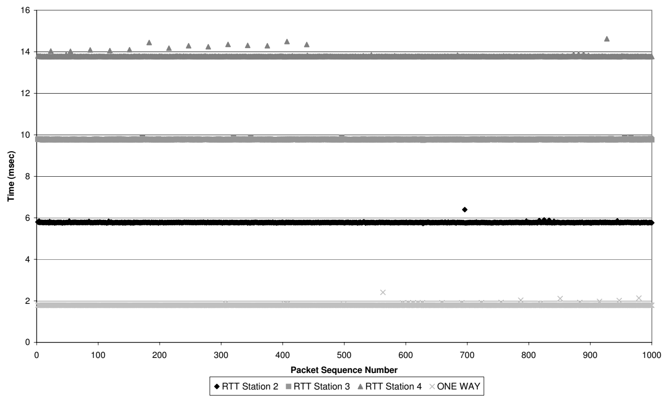

Figure 6. TDMA with 2ms packet size in a 4ms slot

The next phase of the evaluation was to determine the effect that dynamically varying offered load had on the achievable time-bounds. Similar to the interpretation of offered load in section 2, offered load is specified as a percentage of the nominal bandwidth available, which in this case translates to a number of slot allocations per participating host. For example, with a configuration of 10 slots available, 4 participating hosts and 10% offered load, each host is allocated at most one slot, at 110%, each host is allocated a maximum of 2 slots, i.e., 10 slots / 4 participating hosts. All previous results have been achieved with an offered load of 10%. Figures 7 and 8, illustrate the results obtained with an offered load of 90% and 110% respectively.

As can be clearly observed from these results, timely communication is independent of the current offered load, which is a direct contrast to the results obtained from the IEEE 802.11 evaluation in section 2. The TDMA implementation does not show a sensitivity to variances in offered load or number of participating hosts, and thus, is suitable for a dynamic ad hoc environment.

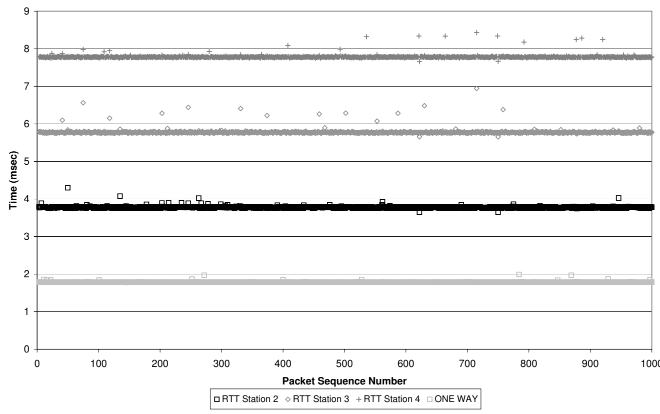

Figure 7. TDMA with 2ms slot time and 90% offered load

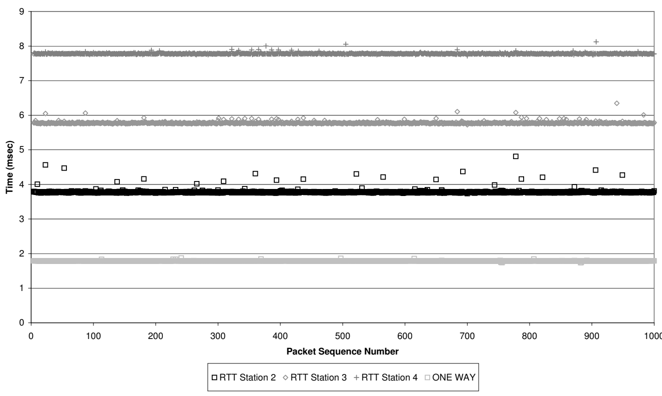

Figure 8. TDMA with 2ms slot time and 110% offered load

4.3. Packet loss

In the complete suite of experiments executed there was a total of 81,000 packet transmissions, of which only 6 packets were lost, i.e., $< .01\%$. This result is in stark contrast to the results obtained in the IEEE 802.11 evaluation, section 2, where the packet loss is sensitive to both the current offered load and competition for the wireless medium.

In addition, of the 81,000 packet transmissions, 222, i.e., .27%, of them experienced a delay prior to reception at the slave host which was sufficiently large to miss the transmission of a reply in the slot allocated to this slave host. Resulting in the failure to transmit a reply from this host.

Given the round-robin allocation of slots, host 2, experiences the least latency prior to the required transmission of a reply in the allocated slot and has the highest probability of a delayed packet reception impacting the ability to reply in a slot. It is unsurprising that all 222 lost transmissions due to delays are associated with host 2.

4.4. Slot allocation within known time bounds

Further experiments were executed to determine the time-bounds achievable for dynamic slot allocation. In this case, the slot allocation request is initiated by a higher layer at a random time within the TDMA cycle.

In these experiments, a cycle is 24ms, encapsulating 12 slots of 2ms duration. Although the slot request may be made at any time in a cycle, the initiation of the atomic broadcast protocol for the slot allocation request, must occur in the slot allocated to the requesting host.

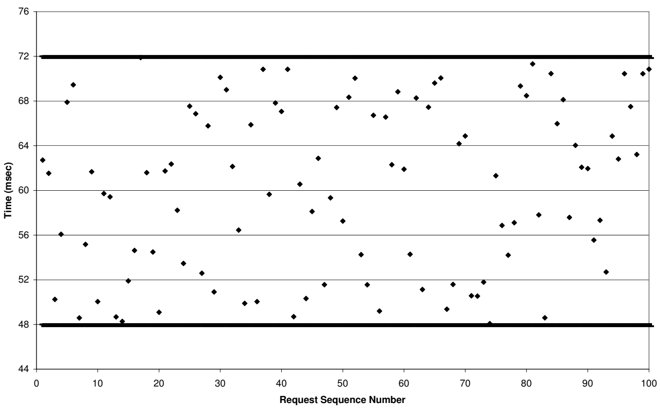

Figure 9. Slot allocation within known time-bounds

Figure 9, illustrates that the achievable lower bound on slot allocation is 2 cycles, (48ms) and the upper bound is 3 cycles (72ms). The difference between the bounds is associated with the worst-case latency from the initial slot allocation request to the start of the atomic broadcast protocol.

For example, if the slot allocation request is available at the start of the allocated slot and this slot is currently granted exclusive medium access, the atomic broadcast protocol is started with the transmission of a piggybacked transmission in this slot. The latency for slot allocation in this case is representative solely of Δ in the atomic broadcast protocol, i.e., 48ms. The worst-case latency for slot allocation occurs if the initial slot request is available in the slot following an allocated slot of the requesting host. Thus, the earliest time the atomic broadcast protocol can start is 1 cycle from the initial slot allocation request, when the allocated slot is granted exclusive medium access again. In this case the worst-case latency of 3 cycles is obtained.

The number of slots allocated to hosts coupled with the dispersion of these slots in the cycle effects the time-bounds for individual slot allocations, e.g., with more than one slot allocation there is an increased probability that the atomic broadcast protocol will be started within the worst-case bound of one cycle, but does not effect the upper and lower bounds achieved.

5. Related Work

In order to support real-time applications within current Linux and IEEE 802.11 systems appropriate real-time extensions and abstractions are essential. A number of approaches have been proposed to support prioritized transmissions for real-time IEEE 802.11 traffic. For example, in [14], the IEEE 802.11 standard is modified to provide prioritized MAC access whilst allowing real-time and non real-time traffic to co-exist. In [7], Kanodia et al. present a distributed priority scheduling technique that involves all stations monitoring all data and control packets for piggybacked priority tags to determine a schedule that represents a stations priority level relative to other stations. The IEEE 802.11 backoff scheme is modified to incorporate this schedule. Both of these approaches necessitate changes to the underlying IEEE 802.11 standard, which implies that firmware changes are made as MAC functions are normally hardcoded on a wireless card. Given the widespread deployment of IEEE 802.11, an approach to utilise this widespread penetration without further firmware change is more generally applicable.

Bluetooth [11], is a centralised Time Division Multiplex (TDM) system, where a designated master controls access to the wireless medium and determines which host may transmit within a fixed 625μsec time slot. Similarly, in IEEE 802.16 (WiMax) [11], a base station (BS), controls access to the wireless medium for both downstream (BS to host) and upstream (host to BS) communication, where medium access is divided into time slots or related to QoS specifications, respectively. Both of these protocols achieve predictable medium access latency by adopting a centralised solution. However, given the potential for a highly dynamic network topology and mobility of participating hosts a centralised solution is not applicable in an ad hoc domain.

6. Conclusions

In this paper, the design and implementation of a TDMA layer above the IEEE 802.11 wireless ad hoc protocol was presented and the influence of the characteristics of the

IEEE 802.11 MAC protocol and the dynamics of an ad hoc domain on the design of this TDMA layer described.

Using experimental results from a real implementation, the feasibility of this approach was evaluated. It was shown that communication and dynamic slot allocation within known time-bounds in a mobile ad hoc network is achievable, subject to some identifiable constraints.

Of particualr significance is the fact that any external interference from wireless devices that are not participating in the TDMA protocol (including non-IEEE 802.11 devices) may cause the transmissions of participating hosts to be delayed. Selecting a slot size that is sufficiently large to accommodate the backoff procedure of IEEE 802.11 is one approach to addressing this issue but is expensive in terms of unused bandwidth reserved for retransmissions that may not be required. A tradeoff between bandwidth utilisation and increasing the probability of maintaining time-bounded communication is necessary. Delays due to external interference may also contribute to loss of clock synchronisation as may lost synchronisation messages.

Using an atomic broadcast protocol, the probability of consistent slot management amongst all participating hosts is increased, but, given the dynamics of the ad hoc environment, is only guaranteed under a certain failure hypothesis (e.g., one lost message in the implementation described here). Extending Δ in the atomic broadcast protocol will increase the tolerance to lost messages, but with a cost of increased latency for all slot management decisions.

References

[1] S. Chakrabarti and A. Mishra. QoS Issues in Ad Hoc Wireless Networks. *IEEE Communications Magazine*, 39(2):142–148, February 2001.

[2] S. Crane, B. Tangney, and J. Moreau. Enforcing Determinism in a CSMA/CD Local-Area Network. *Microprocessing and Microprogrammng, The Euromicro Journal*, 26(3):205–212, October 1989.

[3] F. Cristian. Synchronous Atomic Broadcast for Redundant Broadcast Channels. *Journal of Real-Time Systems*, 2:195–212, 1990.

[4] H. Hartenstein, B. Bochow, A. Ebner, M. Lott, M. Radimirsch, and D. Vollmer. Position-aware Ad Hoc Wireless Networks for Inter-vehicle Communications: the Fleetnet Project. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc01)*, pages 259–262, 2001.

[5] IEEE. *IEEE Std. 802.11 2nd Edition - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE, August 2005.

[6] J. Kaiser and M. Mock. Implementing the Real-time Publisher/Subscriber Model on the Controller Area Network (CAN). In *2nd IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, May 1999.

[7] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. W. Knightly. Distributed Multi-hop Scheduling and Medium Access with Delay and Throughput Constraints. In *MOBICOM*, pages 200–209, 2001.

[8] P. Mantegazza, E. Bianchi, M. Angelo, D. Beal, and K. Yaghmour. *DIAPM-RTAI Programming Guide 1.0*. DIAPM, September 2000.

[9] G. D. Pallas, G. I. Papadimitriou, and A. S. Pomportsis. A QoS Approach to Hybrid TDMA with Heuristic Traffic Shaping for Time Critical Application Environments. In *ISCC*, pages 901–906. IEEE Computer Society, 2005.

[10] J. D. Sweeney, R. A. Grupen, and P. Shenoy. Active QoS Flow Maintenance in Controlled Mobile Networks. In *Proceedings of the Fourth International IEEE Symposium on Robotics and Automation (ISRA)*, Queretaro, Mexico, August 2004.

[11] A. S. Tanenbaum. *Computer networks: 4th edition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 4th edition edition, 1988.

[12] U. Varshney. The Status and Future of 802.11-Based WLANs. *IEEE Computer*, 36(6):102–105, 2003.

[13] K. Wang and B. Li. Efficient and Guaranteed Service Coverage in Partitionable Mobile Ad Hoc Networks. In *IEEE Joint Conference of Computer and Communication Societues (INFOCOM02)*, pages 1089–1098, 23-27 June 2002.

[14] H. Ye, G. C. Walsh, and L. Bushnell. Real-time Mixed Traffic Wireless Networks. *IEEE Transactions on Industrial Electronics*, 8(5), Oct. 2001.