BBN Technical Memorandum No. 1279

An Evaluation of the TSMA Protocol as a Control Channel Mechanism in MMWN

Job No. 11982000

26 April 2000

Prepared for:
Office of Research Administration and Sponsored Projects
The University of Texas at Dallas
2601 N. FLoyd Road
Technology Branch
Richardson, TX 75080

Prepared by:
BBN Technologies
10 Moulton St.
Cambridge, MA 02138
An Evaluation of the TSMA Protocol as a Control Channel Mechanism in MMWN

Rajesh Krishnan
James P.G. Sterbenz
BBN Technologies, 10 Moulton Street, Cambridge, MA 02138, USA
{krush, jgys}@bbn.com

26 April 2000

Abstract

This document reports the findings of the research effort at BBN Technologies to characterize the TSMA protocol, to analyze its performance as a control channel mechanism and to evaluate its applicability to BBN’s MMWN (Multimedia support for Mobile, multi-hop, Wireless Networks) Project. Specific contributions include:

- Evaluation of TSMA as a control channel protocol in MMWN.
- An acknowledgment scheme for TSMA that enhances average-case performance – increases throughput and decreases delay – while preserving the worst-case guarantees. Results of analytical and experimental performance evaluation are provided.
- Application of a transmit power control scheme developed at BBN that can control the topological properties of the network including the degree, which can potentially enhance TSMA performance in a given network or extend TSMA operation over a wider range of network scenarios.
- An implementation of the TSMA protocol, with performance enhancements using acknowledgments and topology control, within the MMWN testbed.
- General issues and design guidelines for the practical deployment of TSMA.

Keywords: Control Channel, Conflict-Free Protocols, Time Spread Multiple Access, Time Division Multiple Access, Carrier Sense Multiple Access, Mobile multi-hop Wireless Networks.

*This material is based upon work supported by the United States Army Research Office through a Subcontract issued by The University of Texas at Dallas under USARO Award No. DAAG55-97-1-0312 (UT Dallas Subcontract No, SC 97-12). Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of United States Army Research Office or The University of Texas at Dallas.
1 Introduction

The “Generic Control Channel Mechanism for Multi-hop Mobile Networks” project at the University of Texas, Dallas is funded by USARO under the DARPA GloMo program[10]. BBN Technologies is under subcontract to the University of Texas and BBN’s role in the project are the following:

- Evaluate TSMA vis-a-vis other control channel protocols in MMWN.
- Design performance enhancements to TSMA, specifically:
  - An acknowledgement scheme to enhance unicast traffic performance
  - Topology (degree) control by transmit power adjustment
- Implement TSMA and enhancements and integrate with the MMWN/CPT system software.
- Perform simulations to experimentally determine TSMA performance under different wireless network scenarios.
- Document and report the findings to the University of Texas, Dallas.

This document is the final report of BBN Technologies’ findings to the University of Texas, Dallas. In this section, we provide some background into TSMA and MMWN followed by a summary of contributions. We also provide a brief overview of how the rest of this document is structured.

TSMA

Time Spread Multiple Access (TSMA) is a wireless control channel protocol which was proposed by Chlamtac and Farago [1]. TSMA is a transmission scheduling scheme for a time-slotted system. TSMA has a number of unique mathematical properties which guarantee a contention-free slot for each node, to any one of its neighboring nodes, in every frame. The properties hold in mobile networks under certain assumptions of link bidirectionality, network size, network degree and mobility rate.

In TSMA, each node is assigned a unique transmission schedule such that the node has rights to transmit only in some slots in each frame. The schedules are determined by using an algebraic construction over finite fields (also known as Galois fields). Choosing a schedule involves choosing a prime number \( q \) and a polynomial degree \( k \) which, in turn, depend on \( N \), the size of the network in number of nodes and \( D \), the maximum degree of the network.

Equivalently the above design involves assigning each node with a unique polynomial of degree \( k \) over \( GF(q) \). The graph of the polynomial is mapped one-to-one to the schedule which results in a frame length of \( q^2 \) slots. Overlap of slot sets corresponds to common roots of the polynomials and a proper choice of \( q \) and \( k \) can ensure that a slot set cannot be covered by up to \( D^2 \) others. The non-covering condition can be expressed as \( q^2 \geq KD + 1 \) and the uniqueness condition can be expressed as \( q^{k+1} \geq N \) and together these represent the basic design equations for TSMA. From performance considerations, it is desirable to choose \( q \) and \( k \) such that \( q^2 \) is as small as possible\(^1\). In particular, a frame length of \( q^2 \ll N \) should be selected. Otherwise no performance improvement over static TDMA can be obtained.

The reader is referred to Chlamtac and Farago [1] for a detailed introduction to TSMA. Chlamtac, Farago and Zhang [2] extend the protocol further by using the technique of protocol threading which removes the degree constraint at the expense of longer schedules. Ju and Li [3] propose a different way to choose the TSMA parameters such that the schedule maximizes the minimum throughput per node. However this work does not significantly change either the length of TSMA schedules or the delay guarantee.

The combination of topology transparency and a delay guarantee makes TSMA especially attractive to mobile networks, therefore, a performance evaluation in a realistic network is worthwhile. In this work we concentrate on evaluating TSMA as proposed in the original paper, but in the context of a control channel in MMWN, a network developed by BBN Technologies.

---

\(^1\)See Ju and Li [3] for a different design approach.
MMWN and DAWN

The Multimedia support for Mobile multi-hop Wireless Networks (MMWN) project at BBN Technologies was supported by DARPA’s GloMo program [10]. MMWN is a modular system of link and network layer algorithms to support distributed, real-time multimedia applications. MMWN has a scalable, hierarchical, virtual cellular architecture. The reader is referred to Ramanathan and Steenstrup [4] for more details on this work.

DAWN, the Density and Asymmetry-aware Wireless Networks [7] project at BBN Technologies, also supported by the GloMo program, adds a number of features to MMWN to better control the effects of density and asymmetry in wireless networks. Of special interest to TSMA are the topology control algorithms developed in DAWN.

The system software [6] for MMWN and DAWN was built using the C++ Protocol Toolkit (CPT) from Rooftop Communications Inc. The MMWN software can be loaded into embedded network hardware. Alternatively, we can run high-fidelity simulations using emulated radios and the same protocol code which runs in the real system. Currently the radio hardware in the MMWN testbed does not support slotted protocols, so the evaluation of TSMA was done using such high-fidelity simulations. The finite state diagram of our TSMA implementation in MMWN with and without acknowledgments is illustrated in Figure 1. CPT allows a layered protocol architecture that allows mixing and matching protocols at different layers easily. Our TSMA implementation can be used in the control channel layer instead of the built-in control channel protocols like CSMA.

The CPT-based MMWN simulation environment does not support hierarchical clustering and routing. Therefore the impact of TSMA on these network control functions were not studied.

CPT can support only up to 100 nodes. In practice, hardware and budget constraints limited detailed MMWN simulations to network sizes of 20–40 nodes. Furthermore, the MMWN simulation environment uses randomly generated network topologies and is not capable of generating networks with specific topological properties. These factors limited the range of the parameters that could be included in the study. Specifically, the experimental study of TSMA in randomly generated sparsely connected,
large networks that are favorable to TSMA was not feasible using detailed CPT-based MMWN simulations.

Contributions

The contributions from the project which are presented in detail in this document include:

- An implementation of the TSMA protocol with performance enhancements using acknowledgments and topology control and integration with the MMWN system software.
- An evaluation of TSMA as a control channel protocol in MMWN.
- An acknowledgment scheme for TSMA that enhances average-case performance – increases throughput and decreases delay – while preserving the worst-case guarantees. Results of analytical and experimental performance evaluation are provided.
- Application of a transmit power control scheme developed at BBN that can control the topological properties of the network including the degree, thereby enhancing TSMA performance in a given network and extending TSMA operation over a wider range of network scenarios.
- Other general issues and design guidelines for the practical deployment of TSMA.

Document Organization

The rest of the document is organized as follows. A performance evaluation of the TSMA acknowledgment scheme is provided in Section 2. A comparison of TSMA with CSMA is provided in Sections 3. Section 4 contains a characterization of TSMA performance degradation when the operating assumptions are violated. Section 5 describes how TSMA performance can be enhanced and its operation extended over a wider range of network scenarios by the use of transmit power control. A concluding summary and some suggested future extensions are provided in Section 6 and a list of bibliographic references is provided at the end.

2 Improving TSMA performance using ACKs

In the simplest incarnation of TSMA, every node is assigned a transmission schedule (frame) consisting of \( L = q^2 \) slots. Each node has a right to transmit in at most \( q \) slots as specified by the schedule. The schedules are assigned in such a way that the node is guaranteed at least one contention-free slot to each neighbor in each frame. Indeed there is such a slot in every \( L \) contiguous slots. However the location of the contention-free slot(s) for the given node is not fixed – it depends on the transmission schedules assigned to the current neighbors of the given node and their neighbors. Thus the location of the contention-free slot(s) changes with mobility. In order to ensure that a packet is transmitted successfully, we need to be sure that it is transmitted in the contention-free slot. This requires that the node must retransmit the packet in every slot in which it has a right to transmit (within \( L \) contiguous slots).

The performance of TSMA for unicast traffic can be enhanced by introducing acknowledgments. Chlamtac and Farago [1] note that if the destination node can instantaneously acknowledge receipt of the packet to the source node then the source node need not continue to retransmit the packets until the end of \( L \) slots. Instead, it can begin to transmit another packet that has been queued. Else, if there are no other packets to transmit, the node will remain quiet, thereby reducing contention with other nodes (during the remaining slots of the frame). In either case, this can result in an overall increase in throughput.

Our approach to implement such an acknowledgement scheme is the following. We extend each data slot\(^2\) by including a

\(^2\)Since TSMA is used as a control channel mechanism in this study, the term “data” refers to control messages only and not to any other type of data. We use the term to differentiate between “data slot” and “ACK slot”.
mini-slot at the end of the data slot. An instance of such a scheme is illustrated in Figure 2. The destination node will use the mini-slot to send an acknowledgement to the source node, if it successfully received a unicast packet in the slot immediately preceding that mini-slot. On receiving this acknowledgement, the source node can stop retransmitting the given packet. Similar to the data slots, the mini-slot must also include a guard-band to accommodate the ramp-up and turn-around times for the radios and the maximum link-propagation delay. It must also include any time required to process acknowledgments.

![Figure 2: TSMA acknowledgement slots](image)

A number of other acknowledgement schemes are possible including for example, stop-and-wait, go-back-N and selective-repeat-request. These can be implemented also in conjunction with our scheme, at a higher layer like the logical link control layer. Our scheme is simpler and more suitable for implementation at the control channel layer. Furthermore, it closely mimics the acknowledgement scheme suggested by Chlamtac and Farago.

Our acknowledgement scheme preserves the TSMA guarantee of at least one contention-free slot to each neighbor per frame. Note however, that although an increase in throughput and decrease in delay is expected, the guarantee itself is not improved by the acknowledgments.

The mini-slot remains unused in the case when a broadcast or multicast packet is transmitted in the data slot. Broadcast and multicast traffic do not gain directly from the acknowledgement scheme presented in this section. However, since nodes stop retransmitting a given packet on receiving an acknowledgement, they free up bandwidth which can potentially be used to transmit any queued packets (including broadcast and multicast). So indirectly, broadcast and multicast performance is enhanced as well.

The total improvement in throughput depends on the fraction of the total traffic that has unicast semantics. Furthermore, the acknowledgement scheme comes with an overhead equal to the length of the mini-slot. Thus, if the network has a preponderance of broadcast or multicast traffic, our acknowledgement scheme can adversely affect the throughput.

We have implemented this acknowledgement scheme as part of the TSMA control channel layer in the MMWN testbed. Our implementation allows us to assign different TSMA schedules to nodes, to enable or disable acknowledgments and to vary the amount of unicast control traffic.

In the following subsections, we present an argument that our acknowledgement scheme preserves the guarantee, a rough performance analysis of TSMA performance using our acknowledgement scheme and an experimental evaluation of the scheme within the MMWN testbed.

---

3It is possible however to extend our acknowledgement scheme to broadcast and multicast, for example the source can select a specific destination to provide acknowledgments or the mini-slot can be made larger to accommodate multiple acknowledgments.
ACKs preserve the TSMA guarantee

In our acknowledgment scheme, a mini-slot is reserved after every data slot. If a unicast packet from a given node A is successfully received by the intended destination node B, then B sends an acknowledgment to A in the mini-slot immediately succeeding the data slot in which it received the packet from A. Acknowledgments are not queued or backlogged and the acknowledgement mini-slot is tied to the source-destination pair which successfully used the slot for transmitting a data packet.

We claim that our scheme preserves the TSMA guarantee under the standard TSMA assumptions\(^4\) from [1]:

- links are bidirectional,
- nodes cannot simultaneously transmit and receive in any given slot,
- there are no capture effects (i.e., nodes cannot successfully receive one of multiple colliding packets),
- network size and degree do not exceed the design values,
- the number of neighbor changes due to mobility is much fewer than one per frame, and
- packet loss occurs only due to contention.

Proving that ACKs preserve the TSMA guarantee involves two parts:

1. In the contention-free slot, the acknowledgment slot is also contention-free.
2. Although a packet and acknowledgment may be successfully exchanged in some slot other than a guaranteed contention-free slot and the succeeding mini-slot, such exchanges cannot be guaranteed for all nodes.

Under the standard TSMA assumptions, at least one contention-free slot to each neighbor in every frame is guaranteed for every node. Suppose a given node A transmits a packet to a destination node B in such a contention-free slot. In this slot, by design, none of A’s neighbors are expected to transmit (i.e. no primary conflict) and furthermore, none of B’s neighbors are expected to transmit (i.e. no secondary conflict) either. Therefore B will successfully receive the packet and transmit an acknowledgment in the succeeding mini-slot.

Suppose another node C also transmits a packet to yet another node D in the same slot. Let us assume that D successfully receives the packet from C. Clearly C can not be a neighbor to either A or B since this slot is guaranteed contention-free from A to B. Also, D cannot be a neighbor of A, otherwise C would have experienced a (secondary) conflict between messages from A and D.

It is however possible that D is a neighbor of B. Their transmissions can interfere at nodes that can hear both B and D. However since A cannot hear D, A will correctly receive B’s acknowledgment. This assumes that the medium is unguided and that the transceivers do not detect collisions while transmitting. Even if B’s acknowledgment were not received by A, it still does not affect the guarantee, only the average-case performance.

In general, since in any frame we cannot guarantee another slot in which A can successfully transmit a packet to B, the overall guarantee cannot be improved using acknowledgments. Whenever A successfully transmits a packet to B, this implies that none of A’s neighbors could have successfully received a different packet. So B’s acknowledgment to A will also be received successfully.

In summary, the above proves that our scheme implements the acknowledgments assumed in the original TSMA paper [1] and preserves the TSMA guarantee under the conditions assumed in the paper. Specifically, the above proof holds only under the assumption that packets and acknowledgments are lost only due to contention. Other losses due to jamming, fading, buffer overflows and loss of the contention-free slot due to mobility must be corrected by higher layers.

\(^4\)In practice, some or all of these assumptions can be violated, but hopefully with a low probability. Therefore, end-to-end reliability must be implemented at higher layers.
Bounds on TSMA throughput improvement using ACKs

In this section, we present a very rough analysis of how much increase in throughput can be expected from acknowledgments.

Without acknowledgments, the maximum throughput per node is limited to $C/q^2$, where $C$ is the channel capacity and $q$ is the prime number chosen for computing the TSMA schedule. If there are $N$ sources, the maximum total throughput for the network is $NC/q^2$.

Let $\alpha$ be the ratio of the length of the ACK slot to the length of the data slot. Due to the ACK overhead, the capacity of the channel is reduced to $(1-\alpha)C$. Therefore with ACKs, the throughput per node is limited to $(1-\alpha)C/q^2$ if the traffic is composed exclusively of broadcast or multicast packets.

Suppose the traffic is exclusively composed of unicast packets. Without a priori knowledge of the topology and the node schedules, we can expect the contention-free slot, on average, to be located at the middle of the frame. Therefore, we can expect the maximum throughput per node to be at least $(1-\alpha)2C/q^2$. In other words, we can expect the throughput to be increased by a factor of two on average. Also, since in each sub-frame of $q$ slots, a given node can have at most one slot in which it has a right to transmit, the maximum throughput per node cannot exceed the weak upper bound of $(1-\alpha)C/q$.

When there is a mix of unicast and broadcast traffic, we need to have sufficient unicast traffic in order to balance the gain in throughput from ACKs with the overhead due to the ACKs. Suppose the fraction of unicast traffic in each node is $\gamma$. We require that,

\[
(1-\alpha)(2C/q^2)\gamma + (1-\alpha)(C/q^2)(1-\gamma) \geq (C/q^2)
\]

\[
(1-\alpha)(1+\gamma) \geq 1
\]

\[
\gamma \geq \alpha/(1-\alpha).
\]

Note that the delay due to contention measured in the number of slots is the same whether we use ACKs or not. With ACKs, the delay measured in time units can be higher due to the overhead of mini-slots. However, the significant portion of the delay is expected to be due to queueing at the nodes. Analysis of this queueing delay is more involved. With ACKs, unicast packets will require less than a frame on average. So the average service time of the queue is decreased and therefore, the total waiting time can be expected to decrease as well.

Experimental Results

In this section, we describe experiments on the MMWN testbed to evaluate the improvement in TSMA performance with the use of acknowledgments. We used a static 20-node randomly generated topology, shown in Figure 3, which has a maximum degree of 8 and an average degree of 5. We chose a TSMA schedule with 121 slots with each data slot equal to 1599 $\mu$s and the mini-slot equal to 299 $\mu$s. The data rate of the radio was set to 4200 Kbit/s.

A number of traffic generators at arbitrary locations in the network were used to generate VC requests in a shuffle-periodic fashion. The rate of VC requests from these traffic generators were changed to control the offered load to the network. We ran three cases at each load, each with a different control channel protocol – CSMA, TSMA and TSMA with acknowledgments. Discussion of CSMA performance with respect to TSMA is deferred to Section 3.

Since we seek to evaluate TSMA strictly as a control channel mechanism, once a VC is set up successfully, it is discarded immediately and no data is sent over it. In MMWN, VC setup signalling traffic has unicast semantics while the only other control traffic involves neighbor discovery and link state routing packets, both of which are broadcast. The general MMWN design admits a hierarchical, virtual cellular architecture, and control messages for hierarchical clustering and routing have been specified. However, the CPT-based MMWN implementation that was used in this study does not implement hierarchical clustering and routing functions.

\footnote{In all our experiments, the fraction of unicast traffic was much higher than that of the broadcast traffic and so this was not an issue.}

\footnote{A VC request is generated in every period $T$, but perturbed uniformly within $T \pm \delta$, for a given $\delta < T/2$.}
Measurements are made at the interface between the network layer and the control channel layer. We consider two types of metrics: The first type relates to control channel performance as observed by the user of the network. We consider the number of VCs setup successfully and the average VC setup delay normalized by the hop distance of the VC path. The second type relates to raw throughput and delay for unicast packets as experienced at the interface of the network and the control channel layer. The control channel layer queue implements a first-in-first-out discipline.

The first set of results are the number of successful VCs setup per second for different offered loads (Figure 4) and the average VC setup delay normalized by the hop distance of the VC path (Figure 5). We see that acknowledgments considerably increase the VC throughput of the network. Acknowledgments lower the delay at medium loads only.

The second set of results shows the number of unicast packets delivered per second for different offered loads (Figure 6) and the average packet delay (Figure 7). We see that acknowledgments considerably increase the packet throughput as well. In contrast to the VC setup delay, the packet delay shows a marked improvement when using acknowledgments.

The 90-th percentile of the hop-normalized VC setup delay and the packet delay are plotted against offered load in Figures 8 and 9 respectively. These curves show a trend similar to their respective averages.

To summarize, in this section, we have presented the following ideas:

- Without acknowledgments, the average-case performance of TSMA is identical to the worst-case guarantee and this implies a considerable performance penalty.
- We proposed and implemented a simple and practical acknowledgment scheme for TSMA which preserves the worst-case guarantee.
- We experimentally demonstrated that our acknowledgment scheme significantly improves the average case performance (increases throughput and decreases delay) when there is a preponderance of unicast traffic.
- We also provide analytical bounds on the performance improvement expected from our acknowledgment scheme.
3 Comparison of TSMA with CSMA

In this section, TSMA is compared with CSMA. The comparison is somewhat subjective, since the two protocols differ in many ways. CSMA is an asynchronous, random access protocol with collisions, whereas TSMA is a slotted protocol with scheduled conflict-free access. CSMA offers no guarantees, whereas TSMA offers a guarantee on delay and throughput. Both are topology transparent. CSMA does not depend intrinsically on the network size or degree whereas TSMA depends on both.

The TSMA protocol requires slot synchronization across all nodes in the network, therefore, a global time synchronization protocol such as Network Time Protocol (NTP) [5] is required. CSMA does not require synchronization and has lower implementation complexity.

In radio networks, contention-freedom does not imply guaranteed delivery. Therefore, scenarios with high losses due to interference and fading can result in deteriorated performance. Therefore, reliability must be implemented in the higher layers for both CSMA and TSMA.
In the experiments described in Section 2, we ran a CSMA case for each run of TSMA and TSMA with acknowledgments. This enables us to simultaneously determine how TSMA with and without acknowledgments compare to CSMA in MMWN.

The first set of results are the number of successful VCs setup per second for different offered loads (Figure 4) and the average VC setup delay normalized by the hop distance of the VC path (Figure 5). The VC setup latency will depend on the number of hops and the transmission delay in each link. For the graphs considered the variations in transmission delay are negligible. However, different pairs of nodes have different hop distances between them and the choice of node pairs can significantly affect the total VC setup time. In order to compare the VC setup time uniformly, we divide the total VC setup time by the hop distance.

We see that while TSMA with acknowledgments offers a maximum throughput of 20 VCs per second, CSMA has a maximum throughput that exceeds 80 VCs per second. The average VC setup delay with TSMA is several orders of magnitude higher than that of CSMA.

The second set of results shows the number of unicast packets delivered per second for different offered loads (Figure 6) and the packet delay (Figure 7). Again we see that CSMA is superior to TSMA with or without acknowledgments. However the maximum raw throughput for TSMA is not reached within the range of loads offered in the experiment. This leads us to the conclusion that the VC throughput is so low with TSMA primarily because the packet queuing delay is so high (several seconds) that VCs time out and perhaps some link adjacencies are lost as well. Increasing the timeouts can result in filling up of the VC table whereas decreasing the timeouts can cause VCs over multiple hops to fail. The 90-th percentile of the hop-normalized VC setup delay and the packet delay are plotted against offered load in Figures 8 and 9 respectively. These curves show a trend similar to their respective averages.

In Figure 6 more unicast packets are delivered with TSMA than with CSMA at offered loads of less than 110 VCs/second, even though the number of VCs set up successfully is higher in the case of CSMA. This is because the number of packets generated per VC request is higher in the case of TSMA. This can happen due to VC signaling timeouts from excessive queueing delays. Additionally, due to delayed beacon packets with TSMA, adjacencies are occasionally lost. This leads to longer routes and therefore more hops and more unicast packets. Therefore, even though fewer VCs are actually requested (and set up) with TSMA than with CSMA, more packets may be generated and delivered.

Thus we observe that with TSMA the maximum VC throughput occurs much before CSMA reaches its unstable region. The maximum delay with CSMA is so much smaller than the delay guarantee (and the actual delay) provided by TSMA. TSMA also reduces the available capacity at each node. Therefore, even at light loads queueing delays begin to dominate in the case of TSMA. This is true even in small networks of reasonable degree. In larger networks, we expect this effect to be even more pronounced since TSMA will require even longer schedules.

In the experiments described in Section 2, we ran a CSMA case for each run of TSMA and TSMA with acknowledgments. This enables us to simultaneously determine how TSMA with and without acknowledgments compare to CSMA in MMWN.

The first set of results are the number of successful VCs setup per second for different offered loads (Figure 4) and the average VC setup delay normalized by the hop distance of the VC path (Figure 5). The VC setup latency will depend on the number of hops and the transmission delay in each link. For the graphs considered the variations in transmission delay are negligible. However, different pairs of nodes have different hop distances between them and the choice of node pairs can significantly affect the total VC setup time. In order to compare the VC setup time uniformly, we divide the total VC setup time by the hop distance.

We see that while TSMA with acknowledgments offers a maximum throughput of 20 VCs per second, CSMA has a maximum throughput that exceeds 80 VCs per second. The average VC setup delay with TSMA is several orders of magnitude higher than that of CSMA.

The second set of results shows the number of unicast packets delivered per second for different offered loads (Figure 6) and the packet delay (Figure 7). Again we see that CSMA is superior to TSMA with or without acknowledgments. However the maximum raw throughput for TSMA is not reached within the range of loads offered in the experiment. This leads us to the conclusion that the VC throughput is so low with TSMA primarily because the packet queuing delay is so high (several seconds) that VCs time out and perhaps some link adjacencies are lost as well. Increasing the timeouts can result in filling up of the VC table whereas decreasing the timeouts can cause VCs over multiple hops to fail. The 90-th percentile of the hop-normalized VC setup delay and the packet delay are plotted against offered load in Figures 8 and 9 respectively. These curves show a trend similar to their respective averages.

In Figure 6 more unicast packets are delivered with TSMA than with CSMA at offered loads of less than 110 VCs/second, even though the number of VCs set up successfully is higher in the case of CSMA. This is because the number of packets generated per VC request is higher in the case of TSMA. This can happen due to VC signaling timeouts from excessive queueing delays. Additionally, due to delayed beacon packets with TSMA, adjacencies are occasionally lost. This leads to longer routes and therefore more hops and more unicast packets. Therefore, even though fewer VCs are actually requested (and set up) with TSMA than with CSMA, more packets may be generated and delivered.

Thus we observe that with TSMA the maximum VC throughput occurs much before CSMA reaches its unstable region. The maximum delay with CSMA is so much smaller than the delay guarantee (and the actual delay) provided by TSMA. TSMA also reduces the available capacity at each node. Therefore, even at light loads queueing delays begin to dominate in the case of TSMA. This is true even in small networks of reasonable degree. In larger networks, we expect this effect to be even more pronounced since TSMA will require even longer schedules.
The large packets in our experiments were link state updates about 350 bytes in length. VC setup messages were unicast packets about 60 bytes long. The TSMA delay is affected by this variation since the slot length is designed to accommodate the long link state packets. We must note that the MMWN system currently does not have support for packet fragmentation and reassembly at the control channel layer. Therefore with TSMA, shorter packets pay an overhead since each slot has to accommodate the maximum packet size expected. CSMA, due to its random access nature, does not suffer from this. This may skew our comparisons. However it is not clear whether performance will improve if fragmentation and reassembly were implemented. Large packets will have to pay the overheads due to a fragmentation and reassembly header and associated processing. Furthermore, in a real radio network, there would be packet losses due to other causes like interference and fading, and fragmentation and reassembly at the control channel layer may affect the performance adversely. The impact of fragmentation and reassembly on TSMA performance requires further study.

4 Degraded TSMA operation

Degraded operation refers to the case in which one or more of the operating assumptions for TSMA are violated, with regard to network size, mobility rate, link bidirectionality or degree.

Exceeding the number of nodes used as the design value for computing the TSMA schedules implies that multiple nodes will have to be assigned the same transmission schedule. If two such nodes with identical schedules become first or second hop neighbors then both their throughputs will be very seriously affected, however other nodes will remain unaffected. If we can guarantee that two nodes assigned the same schedule will never come within two hops of each other for the lifetime of the network, such degraded operation will pose no problems. However this guarantee can be made only for static networks. In practice, we believe that the network designer can usually exercise better control of the size of the network and provision against this situation.

If the rate of mobility is so high that there are multiple neighbor changes in every frame, a situation may arise in which a given source is never able to exercise the guaranteed contention-free slot to a given destination. This can happen especially if mobility occurs in deterministic patterns and the load is very high. For example, suppose in the first half of each frame, the source has a particular set of neighbors which result in the guaranteed contention-free slot to fall in the second half of the frame. Just around mid-frame, the first set of neighbors is replaced by a second set resulting in the contention-free slot to occur in the first half of the frame. Around the end of the frame, the sets of neighbors are swapped again and the cycle can continue indefinitely.

We note however that this example is not likely in normal operation. Based on the application, mobility rates and patterns are often known to the network designer a priori. Although this example illustrates a specific TSMA vulnerability to mobility, high mobility can affect other control channel protocols and several higher level protocols as well. Most network control protocols, such as neighbor discovery and routing will not be able to keep up with mobility rates consisting of multiple link changes within a frame.

The presence of unidirectional links can be accommodated within the existing TSMA framework by using the maximum of the interference in- and out-degrees of the network while computing schedules. Therefore, this case gets subsumed by the following discussion.

Exceeding Maximum Degree

Operating assumptions for TSMA can be violated during the course of normal operation by exceeding the maximum network degree assumed by the design. It is extremely difficult to predict the maximum degree of any mobile network. Although we can use results based on randomly generated graphs (see Section 5) as approximate guidelines to select suitable design values of the network degree, such values will be often violated during normal operation. Consider for example, a conference hall or a football stadium where hundreds or thousands of users with PDAs with an “always-on wireless Internet” capability are located close together, or for example, in a defense scenario in which a tactical squad may come together for reporting/briefing and then disperse for operations.
It is easy to see that designing for the worst-case network degree is impractical. However, maintaining a low degree is desirable for higher spatial reuse. With TSMA (and several other slotted protocols) the maximum per-node throughput and packet delay both depend on the length of the schedule. Performance is acutely sensitive to the design value of the maximum degree (frame length varies as $O(D^2)$), and there is a significant incentive to be able to accurately determine the lowest acceptable maximum degree for a given network.

What if we operate the network in the degraded mode when the degree constraint is violated? The guarantee of at least one contention-free slot for each node in each frame will no longer hold. However, this does not mean that the network will immediately become unusable. At light loads, all neighbors will not have packets to transmit simultaneously and packets can be transmitted successfully.

This observation leads to another interesting possibility. What if we always operate the network with a schedule corresponding to a degree less than the maximum network degree? Clearly, at light loads we can expect a reduction in delay, but with high loads and a large number of sources, the throughput and delay should both be affected adversely.

To characterize this form of degraded operation, we conducted simulations using the MMWN testbed on a fully-connected 8-node graph. The actual routing topology may occasionally be less than fully connected when packets are lost or delayed since the neighbor discovery protocols declare a link down when beacon packets are lost. We tested three scenarios: In the first one we used a 49-slot TSMA schedule with 4 sources over a range of offered loads. Next we used 8 sources to generate a range of offered loads with the same schedule. The 49-slot schedule does not afford the guarantee since the degree constraint is violated. In the third scenario we used a 121-slot TSMA schedule which will afford the guarantee of a contention-free slot to every node in every frame. The VC setup throughput is plotted against the offered load in Figure 10. The results show that the TSMA guarantee comes at a very high price, operating the network with an aggressive schedule which may violate the guarantee but perform significantly better and in fact provide lower delays than that guaranteed by the worst case design.

5 Improving TSMA performance using Power Control

In wireless packet radio networks, keeping the degree low has a number of advantages [11]. Keeping the degree low leads to increase in network capacity due to spatial reuse. In some slotted control channel protocols, keeping the degree low allows shorter frame lengths to be used, leading to higher throughput and lower delays for each node. A smaller network degree also implies fewer links; in networks using link state routing protocols, with fewer links, the link state update overhead is significantly reduced.

In wireless networks, degree can indicate the number of peers formed by the routing protocol at a given node; alternately, degree can indicate the number of links discovered by the neighbor discovery protocol at a given node. Instead, in this dis-
In order to design a TSMA schedule for a given wireless network, we need to know the size of the network and the maximum network degree. In the worst case, the network degree will be one less than the total number of nodes in the network which leads to impractically large TSMA schedules. However on average the network degree will be much smaller. To determine reasonable values of the network degree, let us consider the case when a number of identical wireless nodes are uniformly distributed in a unit square on the Euclidean plane. Let us assume an ideal case where there is no fading or interference and the medium is isotropic. Let these nodes be assigned the same transmit power level which is the minimum required to ensure that the network is connected. In our case the transmit power is related to the transmit radius by an inverse square law. The minimum transmit radius that renders the network connected can be easily determined by conducting a binary search in the range 0 to $\sqrt{2}$.

We ran simulations with 2, 5, 10, 20, 50, 100 and 256 nodes (1000 samples of each) uniformly distributed in a unit square on the Euclidean plane. We determined the average $\mu$ and standard deviation $\sigma$ of the network degree when the nodes transmit at the minimum power that is necessary to keep the network connected. This is plotted in the graph in Figure 11. Two reasonable designs for the TSMA schedule would be to use $[\mu]$ or $[\mu + \sigma]$ as the maximum degree.

Figure 11: Average and Standard Deviation of the Degree of Connected, Minimum Fixed-Radius Neighborhood Graph of Random Sites in [0,1] X [0,1]. Each data point is averaged over 1000 runs.
The reader will observe that the average degree (of the network when all nodes use the same minimum power required to keep the network connected) exceeds the “magic” number of 6 recommended by Kleinrock and Silvester[11], even for small graphs of 20 nodes. It is not known if the expected value of the maximum degree of the fixed-power connected network is bounded, as the network size goes to infinity (see Philips et. al [9]).

This motivates the use of Autonomous Topology Control, the goal of which is to assign different transmit power levels to different nodes so that a connected network of desired properties can be formed; in this case, we wish to lower the degree.

We ran simulations to evaluate the advantages of using transmit power control in conjunction with TSMA. For our evaluation we chose a very simple simple heuristic existing in the MMWN system for power control. The pseudocode of this algorithm is given in Figure 13. This heuristic uses only the number of links formed by the node using a neighbor discovery protocol. When the number of links exceeds a maximum threshold, the power is successively dropped by a fixed number of dB until the number of links falls below the maximum threshold or until the minimum transmit power is reached. Similarly, when the number of links drops to below a minimum threshold the power is raised. The algorithm makes no effort to keep the network connected. Note that the MMWN system supports other more sophisticated algorithms [8].

We conducted experiments using the MMWN testbed on the 20-node randomly generated topology illustrated in Figure 12. For each offered load we compared CSMA, TSMA with an aggressive schedule of 49 slots, TSMA with a 49-slot schedule with topology control, and TSMA with a 121-slot schedule that can accommodate the maximum degree observed in this topology.

Figure 14 shows the aggregate VC throughput against different offered loads. The 121-slot schedule performs very poorly, whereas, the 49-slot schedule performs more or less identically with or without topology control at low loads. At high loads, the performance with topology control is marginally better. Note that in all cases, CSMA outperforms TSMA. Figure 15 shows the delay per hop for VC setups against different offered loads. The average delay with the 121-slot schedule is much higher than that of the 49-slot schedule. This suggests that operating the network with an aggressive schedule that potentially violates the degree can lead to better performance. Topology control can help prevent excessive performance degradation at high loads.
AdjustPowerTimeout = 5 seconds
StartupPowerLevel = 30 dB
MaximumPowerLevel = 40 dB
MinimumPowerLevel = 10 dB
DeltaPerEpoch = 3 dB
DegreeHighThreshold = 6
DegreeLowThreshold = 2

Psuedocode

Upon activation:

    set operationalPower = StartupPowerLevel
    set PwrTimer(random(0.5*AdjustPowerTimeout, 1.5*AdjustPowerTimeout))

PwrTimer expires:

    if my degree is greater than DegreeHighThreshold
    reduce power level by DeltaPerEpoch, unless you would go below Minimum

    if my degree is less than DegreeLowThreshold
    increase power level by DeltaPerEpoch, unless you would go above Maximum

    set PwrTimer(random(0.5*AdjustPowerTimeout, 1.5*AdjustPowerTimeout))

Figure 13: Topology Control Algorithm

Figure 14: Number of VCs setup successfully per second versus offered network load with and without topology control

Figure 15: Average VC setup delay per hop versus offered network load with and without topology control
Therefore, further examination of this aspect of TSMA performance is recommended.

For each of the four cases, the raw unicast packet throughput and packet delay are plotted respectively in Figure 16 and Figure 17. The curves are very similar to the VC throughput and average VC setup delay curves. Finally the 90th percentile of the VC setup delay per hop and the unicast packet delay are plotted in Figures 18 and 19 respectively which are similar to the curves of the corresponding averages.

While these results are not yet conclusive, the DAWN project at BBN has achieved dramatic gains in throughput for other protocols including CSMA [8] by using datagram traffic in larger networks and more sophisticated topology control algorithms. Therefore, further examination of this aspect of TSMA performance is recommended.

Figure 16: Number of unicast packets delivered per second versus offered network load with and without topology control

Figure 17: Raw unicast packet delay versus offered network load with and without topology control

Figure 18: 90th percentile of hop-normalized VC setup delay with and without topology control

Figure 19: 90th percentile of unicast packet delay with and without topology control
6 Conclusions

In this project, we implemented the TSMA protocol with some enhancements and integrated it with the MMWN system software. We also evaluated the performance of TSMA and the enhancements vis-a-vis CSMA. As noted earlier, the study is limited to the range of parameters allowed by the simulation environment. Specifically, large sparse network topologies that are favorable to TSMA could not be simulated.

Within the range of parameters considered in the simulations, we observed that CSMA performs better than TSMA, both in terms of throughput and delay. However we demonstrated that acknowledgments significantly improve the performance of TSMA when there is a preponderance of unicast traffic.

The guarantee of a contention-free slot in every frame at the control channel level provided by TSMA comes at a performance penalty. However, simulations demonstrate that we can trade-off the TSMA guarantee for improved performance.

Topology control can be used to reduce the network degree. However, in our experiments, we constrained the parameters of the algorithm to avoid network partitioning of the randomly generated topology that was used. In turn, this limited the number of admissible solutions afforded by the randomly generated topology. We did not conclusively demonstrate the benefits of topology control for TSMA.

Unfortunately, more extensive experimentation was not possible since the high fidelity MMWN simulations limited both the maximum network size and the number of experiments that could be considered within the time constraints of the project.

Although we had limited success with TSMA in the context of MMWN, there can be other scenarios (not limited to packet radio networks) where the guarantee provided with TSMA is desirable. With this in mind, we suggest the following future work:

- Simulate TSMA in large networks of low degree. This would require using coarse-grained simulations in contrast to the high fidelity packet level MMWN simulations, which were limited to a few tens of nodes.
- Simulate TSMA and compare with other protocols under different loads.
- Modify the acknowledgment scheme to support broadcast and multicast traffic.
- Investigate modifications to TSMA by which different capacities can be assigned to different nodes.
- The MMWN testbed does not currently support slotted protocols and their implementation was beyond the scope of this project. We recommend that TSMA be studied in detail by porting TSMA to a system that natively supports slotted protocols.
- Implement a fragmentation and reassembly scheme or a framing scheme to reduce slot overhead in the case of small packets.
- Study effects of high mobility rates including vulnerabilities of TSMA to degradation or denial-of-service attacks.
- We recommend a more rigorous study of the impact of topology control on TSMA.
- Study effects of periods of temporary loss of synchronization due to loss of connectivity.
- Study methods to reduce the need for large guard slots when a mix of long (e.g. LEOs and UAVs) and short delay terrestrial radio links are used.

Acknowledgments

We wish to thank all the people who have contributed to the development of the ideas presented in this document and/or the MMWN system software. They are listed here in no particular order: Prof. Imrich Chlamtac, Prof. András Faragó, Prof.
References


