A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks

by Chenxi Zhu, M. Scott Corson

CSHCN T.R. 97-27
(ISR T.R. 97-67)

A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks

Chenxi Zhu and M. Scott Corson
Institute for Systems Research
A.V. Williams Building (115)
University of Maryland
College Park, Maryland 20742
(czhu,corson@isr.umd.edu)
(ph) 301-405-6630
(fax) 301-314-8586

Abstract

A new single channel, TDMA-based broadcast scheduling protocol, termed the Five-Phase Reservation Protocol (FPRP), is presented for mobile ad hoc networks. The protocol jointly and simultaneously performs the tasks of channel access and node broadcast scheduling. The protocol allows nodes to make reservations within TDMA broadcast schedules. It employs a contention-based mechanism with which nodes compete with each other to acquire the TDMA slots. The FPRP is free of the "hidden terminal" problem, and is designed such that reservations can be made quickly and efficiently with minimal probability of conflict. It is fully distributed and parallel (a reservation is made through a *localized* conversation between nodes in a 2-hop neighborhood), and is thus arbitrarily scalable. A "multihop ALOHA" policy is developed to support the FPRP. This policy uses a multihop, pseudo-Baysian algorithm to calculate contention probabilities and enable faster convergence of the reservation procedure. The performance of the protocol is studied via simulation, and the node coloring process is seen to be as effective as an existing centralized approach. Some future work and applications are also discussed.

Keywords: TDMA broadcast scheduling, graph coloring, channel access, pseudo-Baysian, scalability, distributed algorithm, mobile ad hoc network, multihop wireless network

1 Introduction

We consider the scheduling of broadcasts in a mobile ad hoc network. Existing standards for ad hoc networking [1, 2] do not address the problem of broadcast scheduling. An mobile ad hoc network is a mobile, multihop wireless network with no fixed infrastructure. The multihop nature of an ad hoc network allows spacial reuse of the TDMA slots [3]. Different nodes can use the same time slot if they do not interfere with each other. The problem of how to assign these slots to nodes is commonly referred to as scheduling. Here, we consider the problem of scheduling broadcast transmissions in a single channel radio network where nodes employ omnidirectional antennas. By broadcast, we mean that when a node transmits, every one-hop (i.e.

adjacent) neighbor of the node receives the packet. A conflict-free broadcast schedule requires that any two simultaneously transmitting nodes be at least three hops apart.

Many algorithms have been developed to schedule broadcasts in multihop radio networks [4, 5, 6, 7, 8, 9, 10, 11]. Some are centralized algorithms and depend on the existence of a central controller [8, 11]. Some are distributed, but often use fixed TDMA schedules in the assignment [5, 7, 9]. In these approaches, the length of the scheduling process is proportional to the size of the network. Hence, these protocols are inapplicable for use in large networks.

We have developed a distributed protocol which is arbitrary scalable, i.e. its performance is not affected by the size of the network. Simulations have shown that the node coloring procedure can produce a schedule as good as a simple, greedy centralized algorithm, yet maintain its distributed and parallel nature. It requires no *a prior* knowledge. The amount of overhead is minimal. Along with the reservation protocol, we have also developed a multihop, pseudo-Baysian policy. This policy speeds the convergence of the reservation procedure.

2 The Five-Phase Reservation Protocol

2.1 Overview

The FPRP is a contention-based protocol which uses a five-phase reservation process to establish TDMA slot assignments that are non-conflicting with high probability. The channel is divided into two segments: a *control* segment where TDMA slots are scheduled among the nodes (also referred to as a reservation frame), and an *information* segment where information is carried according to the TDMA schedule. The FPRP is a fully distributed protocol which executes in parallel over the entire network. By parallel, we mean that the FPRP permits multiple reservations to be made at various parts of the network simultaneously. The reservation process for a given node only involves nodes within a two-hop radius, and is thus a local process. No coordination is necessary with more distant nodes. By keeping the reservation process localized (and running simultaneously over the entire network), the FPRP is insensitive to the network size, i.e. it is arbitrary scalable. This makes the protocol suitable for large networks as well as small networks. It also works efficiently when the network becomes partitioned. A node needs no *a priori* information about the network, i.e. it does not need knowledge regarding network membership, its neighbor set or network size. This makes the FPRP robust in a rapidly-changing topology. The FPRP does not need the support of additional protocols for medium access control or network exploration. *The protocol jointly and simultaneously performs the tasks of channel access and node broadcast scheduling.* A node uses the FPRP to explore its neighborhood and to make nearly conflict-free reservations. The FPRP has no restriction on the topology of the network, but it requires that every link used be bidirectional. It is assumed that nodes keep timing synchronization sufficient to permit global slot synchronization.

2.2 Detailed Description

The protocol's frame structure (shown in Fig. 1) is as follows. There is a Reservation Frame (RF) followed by a sequence of Information Frames (IF). There are N Information Slots (IS) in an IF. There are also N Reservation Slots (RS) in an RF. Each RS is dedicated to the reservation of a corresponding IS. If a node wants to reserve an IS, it contends in the corresponding RS. A TDMA schedule is generated in the RF, and is used in each of the subsequent IFs until the next RF where the scheduled is regenerated. The structure is most similar to that of D-TDMA [12]. It can be viewed as an extension of D-TDMA in a multi-hop environment.

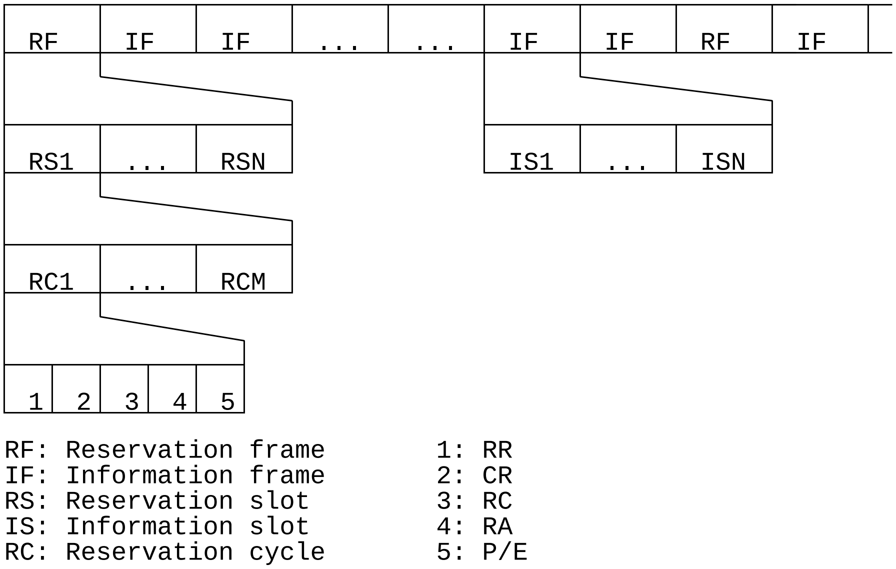

Figure 1: Frame structure of the FPRP

A RS is composed of M Reservation Cycles (RC) (the value of the parameter M must be determined heuristically for a given network). Each RC consists of a five-phase dialogue from which the protocol receives its name. Within a RS, a reservation is made through a sequence five-phase dialogues between a contending node and its neighbors.

Loosely stated, a node that wishes to make the reservation first sends out a request, and feedback is given to the node from its neighbors regarding the request. If the request is successful (i.e. it does not collide with other requests), the node reserves the slot. This reservation is passed to every node within two hops.

These nodes will honor this reservation and will not contend further for the slot. If not successful, the node will contend in subsequent RC for this RS with some probability, until itself, or another node one or two hops away, succeeds. As a result, the node will either transmit (T), receive (R) or be blocked (B) in the information slot. A five-phase dialogue ensures: 1) if two requests collide, neither makes the reservation; 2) once a node makes a reservation, it will have sole use of the slot in its neighborhood with high probability. As will be seen, the design of the protocol allows a slot be reused efficiently throughout.

2.2.1 The five-phase dialogue in detail

A node keeps global time, and knows when a five-phase cycle starts. A node can transmit or receive, but cannot do both at the same time. We assume every node participates in the reservation process.

A reservation cycle has five phases. They are:

1. Reservation Request phase (RR), where nodes make their requests for reservations;
2. Collision Report phase (CR), where nodes report collisions that just occurred in phase 1;
3. Reservation Confirmation phase (RC), where nodes make confirmations of their requests. A reservation is established in this phase;
4. Reservation Acknowledgement phase (RA), where nodes that heard a RC in phase 3 acknowledge with a RA packet. This RA also serves to inform those nodes that are two hops away of the recent reservation;
5. Packing and Elimination phase (P/E). In this phase, two kinds of packets are transmitted. A packing packet serves to make the broadcasting pattern denser in a given slot. An elimination packet is used to remove possible deadlocks (DL) between adjacent broadcast nodes.

The first three phases are analogous to the distributed protocol in [9].

The details of each phase are given below:

1. *Reservation Request Phase*

   In this phase, a node which wants to make a reservation sends a Reservation Request packet (RR) with a probability p. A node sending a RR is referred to as a Requesting Node (RN). The calculation of the probability p will be discussed later. The only information necessary in a RR packet is the sender's

ID. A node which does not transmit a RR listens in this phase. It may receive zero, one or more RR's from its neighbors. In the last case all the RR's are destroyed and the node senses a collision.

2. *Collision Report Phase*

   If a node receives multiple RR's in phase 1, it transmits a Collision Report packet (CR) to indicate the collision. Otherwise it is silent. By listening for any CRs in this phase, a RN determines whether its RR has collided with others. On receiving no CR, it assumes that its RR reached every neighbor safely. Such a node becomes a transmission node (TN). It will go ahead and make a reservation in phase 3 and transmit in the subsequent information slots unless disabled in phase 4 or 5.

   It should be clear that the RR/CR exchange eliminates the "hidden terminal" problem [13].

3. *Reservation Confirmation Phase*

   A TN sends a Reservation Confirmation packet (RC) in this phase. Every node which is one hop away receives the RC and understands the slot has been reserved. They will receive from the TN in the information slots. They will not contend further for this slot.

4. *Reservation Acknowledgement Phase*

   In phase 4, a node acknowledges a RC it just received by sending a Reservation Acknowledgement packet (RA). This tells a TN that its reservation has been established. If the TN is not connected with any other nodes, it does not receive any RA and thus becomes aware of its isolation. This prevents isolated nodes from transmitting–it can save power by not transmitting in this slot.. Without this phase, an isolated RN would never receive a CR and would then always become an TN.

   We define transmitter deadlock (DL) to be the situation where two or more TNs are adjacent–these nodes are referred to as a *deadlocked set*. A deadlocked set can be of one of two types: (i) an *isolated* deadlock (when no node of the set is connected to any non-deadlocked nodes) and (ii) a *non-isolated* deadlock (when some node in the deadlocked set is connected to an adjacent, non-deadlocked node).

   Deadlocks begin to form during phase 1. Because nodes cannot receive while transmitting in phase 1, they cannot sense a collision directly. To avoid deadlock, the transmitting nodes must rely on the existence of a *common* neighbor to send a CR in phase 2. If no such neighbor exists, in the absence of a CR, they will each claim success, become TNs during phase 2 and a deadlock is formed.

   Phase 4 serves to resolve isolated deadlocks. In this case, since none of the nodes will transmit a RA, none of the nodes will hear an RA and, hence, all will abort their transmissions, thus resolving the deadlock.

A RA transmission also serves to inform the nodes which are two hops away from the TN of its success. These nodes also label this slot as reserved and cease contention. They become blocked (B) in this slot.

5. *Packing/Elimination Phase*

   In this phase every node that is two hops from a TN which has made its reservation since the last P/E phase sends a Packing Packet (PP). A node receiving a PP therefore learns there is a recent success three hops away. As a consequence, some of its neighbors cannot contend further for this slot. It can take advantage of this and adjust its contention probability accordingly. This can speed up the convergence. This also increases the probability of success of the nodes three hops away. Hence, two TN's are more likely to be only three hops apart rather than further. This is preferable, because, when TN's are only three hops apart, more nodes are allowed to transmit and less nodes are blocked. This is often referred to as "maximal packing". Through the encouragement of maximal packing, the FPRP uses a slot more efficiently.

   In the same phase, any TN sends an Elimination Packet (EP) with a probability of 0.5. This is intended for another TN, which could be potentially adjacent, in an attempt to resolve a non-isolated deadlock. If a TN does not transmit, but receives an EP in this phase, it learns there is a deadlock. In this case it will relabel the slot as reserved by the other TN (the one that sent the EP) and will receive, rather than transmit, in it. It will contend further in other slots. There is no need to inform its neighbors about this.

   The EP can be sent more frequently in order to further reduce the deadlock probability. This can be achieved if a TN, after acquiring a reservation, transmits an EP in the phase 1 of every cycle in the same reservation slot. This EP will not interfere with any RR's. (After a reservation is made, every node within 2 hops will not contend in the same slot. So the EP from the TN cannot collide with a RR.) An EP in phase 1 works in the same manner as an EP in phase 5. The elimination process is thus execuated more often and the DL probability is reduced essentially to zero.

   The fifth phase helps only after a successful reservation is made. Since the throughput of contention-based protocols (such as ALOHA) is much lower than one packet per slot, it is more economical to place a fifth phase in every few reservation cycles. Thus, a typical sequence would be a sequence of one, two or three four-phase cycles followed by a fifth phase. How often a fifth phase is used can be determined heuristically.

The five-phase scheme minimizes the probability of collision in a way that is efficient and robust. The meaning of a packet is implicitly conveyed simply by *when* (i.e. in which phase) the packet is sent. Thus, a packet need only consist of a single, logical bit. A packet may collide with another packet, but the correct semantic is always inferred in the context of the protocol. The decision is made on the basis of the absence/presence/collision of various packets. The packets can be made extremely small and a reservation cycle is very compact. The FPRP uses the fact that a collision always occurs one hop away from the sender. A collision is detected at the node where it occurs (unlike the CSMA/CA protocol, where the sender detects the collision). The sender functions as a local hub. It collects collision information and makes the final decision. Before a reservation is deemed successful, no information has to be collected from or dissipated to nodes more than one hop away. This greatly simplifies the reservation process.

3 Validation of the protocol

3.1 Correctness of the protocol

A broadcast is successful only if every neighboring node receives the packet successfully. A node cannot receive packets from more than one sources, neither can it receive and transmit simultaneously. A node receives a packet successfully only if the packet is the only one it receives, and the node itself is not transmitting at the same time. We call the collision of packets at a node which is not transmitting a type I collision, and the collision of packets at a node which is transmitting a type II collision. A type II collision is the same as a deadlock. We claim that a type I collision cannot happen, and that a type II collision can only happen with minimal probability.

When more than one RR's reach a node at the same time, if this node is not transmitting, it senses the collision and transmits a CR. All the RN nodes receive the CR and none of them succeeds. If a TR is the first one to make a successful reservation, every other node within two hops is informed (in phase 3 for one-hop neighbors and in phase 4 for two-hop neighbors). These neighboring nodes will honor the reservation and will not contend further in the same slot. So once a reservation is made, it will be the only one in its neighborhood. It can be concluded that no two transmissions would collide at a third node, i.e. a collision of type I cannot happen.

There is a minimal probability for a type II collision to occur in the FPRP. A node cannot transmit and receive at the same time. The collision detection mechanism depends on another node which hears both RR's. In many cases, when two RN's are adjacent and collide with each other (type II collision), their RR's also collide at some other nodes which hear the collision (type I collision). Both RN's will receive the CR and fail. In a mesh type network, it is often the case that two adjacent nodes share many neighbors. A DL cannot happen in this situation. Unfortunately, these common neighbors do not always exist.

If two neighboring nodes request at the same time and they do not have a common neighbor, neither will discover the collision and both of them will make reservations of the slot. A deadlock is formed. Deadlock sets containing more than two nodes are very rare. A deadlock is most likely to form at a "bridge"[1]. If it does occur, the adjacent TN's use elimination packets in an attempt to eliminate each other. After every elimination phase, the probability of a deadlock is reduced by half. A deadlock is likely to be resolved during the elimination process, especially if this is imbedded in phase 1 as mentioned previously. This is seen in simulation results to be discussed. Based on these results, we conclude that the probability of a type II collision is minimal, and it does not significantly affect the performance of the FPRP.

3.2 An example of the FPRP

We illustrate the execution of a five-phase cycle with a tandom network of 10 nodes (see Fig. 2).

No reservations have been made before this cycle. A cycle of five phases is shown, with the transmission of every node in each phase. In phase 1, nodes 1, 3 and 7 transmit RR's. The RR's from nodes 1 and 3 collide at node 2 while the RR from node 7 reaches its neighbors (nodes 6 and 8) ungarbled. In phase 2, node 2 reports the collision. On hearing the CR from node 2, nodes 1 and 3 become aware of the collision and do not proceed further. Node 4, which receives a RR in phase 1 but nothing in phase 3, learns that the RR from node 2 collided with another RR somewhere else. Node 7 does not receive any CR from its neighbors and assumes there is no collision. In phase 3, it sends a RC telling nodes 6 and 8 of its confirmation of its reservation. In phase 4, nodes 6 and 8 acknowledge with a RA. Their RA's also inform nodes 5 and 9, which are two hops away from 7, that a successful reservation was just made and they are blocked from contending further in the following cycles in the same slot. In phase 5, node 7 transmits an EP. Note that there is no deadlock in the example and this EP eliminates nobody. (In reality, DL's are most likely to occur in a tandom network because every link is a bridge. Elimination is most important in a network like this). Simultaneously, in phase 5, nodes 5 and 9 transmit a PP announcing the recent success of node 7 and encouraging nodes 4 and 10 to contend. By adjusting their contention probability (to be discussed in Section 5), nodes 4 and 10 become more likely to succeed in the following cycles. As a result, more nodes will transmit and less nodes will be blocked, and the slot is used more efficiently.

The operation of the FPRP can be viewed as follows. The first four phases are used to establish reservations and eliminate the hidden terminal problem. The fifth phase performs color packing, and attempts to eliminate

---

[1] A bridge is a link between two larger groups of nodes that are otherwise not locally connected. The nodes at either end of the bridge do not share any common neighbors.

Node	1 ___	2 ___	3 ___	4 ___	5 ___	6 ___	7 ___	8 ___	9 ___	10
Packet(phase)										
RR(1)	t	r,r	t	r		r	t	r		
CR(2)	r	t	r							
RC(3)						r	t	r		
RA(4)					r	t	r,r	t	r	
PP(5)				r	t	r		r	t	r
EP(5)						r	t	r		
Result:	I	I	I	I	B	R	T	R	B	I

I: idle
B: blocked
R: receive
T: transmit
(in the corrosponding slot)

t: transmit a packet
r: receive a packet

Figure 2: A five-phase reservation cycle in a tandom network

any non-isolated deadlocks that may exist between adjacent nodes. As mentioned previously, the deadlock elimination of the fifth phase can and should be imbedded in phase 1 for most efficient operation (this was done for the simulation results we present).

4 Performance of the FPRP

4.1 Application of the FPRP to the graph coloring problem

So far no rules have been given as to why the nodes contend for the slots. This depends on the nature of the network and the higher layer protocols. The FPRP only provides a means for the nodes to make TDMA broadcast slot reservations. Nodes can make their reservations depending on their traffic load. The TDMA schedules produced thereof can be used to transmit data packets. The FPRP can also be used to make reservations for network control frames. A node can reserve a TDMA slot and participate in the organization/control phase. It is ideal for distributed network control protocols. A node requires no more than a slot and the problem is equivalent to coloring a graph.

The graph coloring problem corresponding to TDMA broadcast slot assignment is to assign colors to nodes of a network such that no two nodes within two hops from each other have the same color. This can be transformed to the standard graph coloring problem. For a given graph $G(V, E)$, if we connect every pair of nodes that are two hops apart, we get a new graph G'. The problem becomes how to color G' so that the same color is not given to two adjacent nodes. To color a graph with minimal number of colors is NP-complete and is often intractable for a network of reasonable size [14]. Various heuristics have been developed. Many of them are centralized algorithms and require global knowledge of the network. Recently it was shown that global sorting of some kind produces good results [11]. It is not clear how this can be done by a distributed algorithm. Among these centralized protocols is the RAND protocol, where nodes are colored in a random ordering in a greedy fashion. It is a centralized protocol with minimum coordination. We now evaluate the performance of the FPRP when used as a pure graph coloring protocol (assign one slot or color to every node). We also compare the performance with the RAND protocol and a degree lower bound. This degree lower bound is the maximal degree of the graph plus one. This lower bound is found to be very tight and is used to approximate the optimal coloring solution.

4.2 Simulation results

Networks of random topology are generated. For a network of size N, N nodes are generated in an area of $\sqrt{N}$ by $\sqrt{N}$ units. The location of a node is generated randomly, using a uniform distribution for its X and Y coordinates. Thus the average density of the network is 1 node per square unit. The transmission range R of a node is chosen typically to be 1.5 units. The purpose of generating a network this way is that the size of the network and the transmission range R (relative to the node density) can be varied independently. R is the same for every node, making every link bidirectional. The average degree of a node is approximately 7. The generated network is converted into a undirected graph $G(V, E)$. The FPRP and RAND protocols are used to color the graph. In the FPRP, every node stops contention after it acquires a color. During each cycle, some nodes acquire the corresponding color. The reservation cycles are repeated until the FPRP converges, e.g. the color can not be assigned to any other nodes. The next color is assigned with the same fashion. The FPRP terminates after every node has acquired a color. The number of colors required is the measure of coloring efficiency.

Networks of various sizes ranging from $N = 100$ to $N = 500$ are tested. The transmission range of 1.5 is used for all of them. The results are given in Table 1. DLB is the degree lower bound. The effect of increasing connectivity (R) on a given network is also investigated. A network of 100 nodes is produced and

Size	DLB	RAND	FPRP
100	15	16	16
200	16	19	17
300	15	17	17
400	15	18	19
500	19	21	22

Table 1: Coloring of networks of different size.

R	DLB	RAND	FPRP
1.0	9	9	9
1.5	15	16	16
2.0	20	23	24
2.5	29	32	33
3.0	33	39	38

Table 2: Coloring of networks of different transmission range.

the transmission range R varies from 1.0 to 3.0. As the number of neighbors increases, so does the number of colors used. The results are shown in Table 2.

The overall performances of the FPRP and the RAND are comparable. The performance of FPRP appears insensitive to network size. Essentially, both approaches are randomized coloring processes and they are expected to perform similarly. Algorithms with global coordination often produce better results, but it is not clear how it can be done by a distributed algorithm with localized knowledge. It is worth noting that while the RAND algorithm is a centralized solution and requires global knowledge as to which nodes have been given what colors; the FPRP, on the other hand, is totally distributed and fully parallel with no *a prior* knowledge. This makes the FPRP more practical and more implementable on a large, mobile ad hoc network.

5 Contention Probability and a Pseudo-Baysian Approach

5.1 Rivest's pseudo-Baysian Algorithm

The FPRP requires a suitable contention policy. Theoretically any slotted ALOHA policy can be used, since every node has only one packet in a reservation frame and the arrival rate is zero–the contention process would always be stable. However, a good ALOHA protocol would make the reservation process converge quickly. Most ALOHA protocols are developed for networks with a central basestation [3]. The situation here differs in that it is a multihop environment and there is no basestation. Every node is a potential source or destination of a packet. We are not aware of a protocol that perfectly meets this requirement. Therefore, we chose to modify Rivest's pseudo-Baysian Broadcasting Algorithm [15] to fit into this role.

In Rivest's pseudo-Baysian algorithm, every node estimates the number of contenders (n) and adjusts its contention probability $p := 1/n$. After every contention slot, a node updates its estimate n on the basis of the feedback:

success or idle

$$n := n - 1;$$

collision

$$n := n + (e - 2)^{-1}.$$

It is designed to support stable throughput with minimal amount of delay. The original algorithm works for a single-hop ALOHA network fairly well. The situation here differs in that: 1) a node only cares for the contenders which are within two hops of itself; 2) the network typically has a random shape and every node has different neighbors; 3) every node has only one packet to send; 4) in the contention for a particular slot, if a node succeeds, every other node will not contend further in this slot, but will resume contention in other slots. We modify Rivest's algorithm into a multihop, pseudo-Baysian algorithm to adapt to these characteristics.

5.2 Multihop pseudo-Baysian Algorithm

A node estimates the number of contenders within two hops and calculates its contention probability $p := 1/n$. From a node's point of view, n is the number of contenders within two hops of itself. They are called neighboring contenders. A node updates its estimate on the basis of what it hears:

success A node always learns of a success within two hops, for it is either informed in phase 3 (nodes 6, 8 in Fig. 2) if the success is one hop away, or in phase 4 (nodes 5, 9) if the success is two hops away. In the Packing phase, a node learns of a recent success three hops away.

idle An idle is always detected (if there is no node contending within its two hop range, a node hears nothing and thus assumes the slot is idle).

collision A collision is a more complicated. A node knows of a failed contender which is one hop away. If it receives more than one RR (node 2), it senses the collision directly. If it receives a RR in phase 1 but no RC in phase 3 (node 4), it reasons that there is a node contending one hop away and its RR has collided. If a node receives no RR in phase 1, but receives a CR in phase 2, it knows that two nodes which are two hops are contending and their RR's collided at one of its immediate neighbors. A collision two hops away cannot always be detected. In the example, node 5 does not know that node 3 contended and collided with node 1. This occurs when one of the contenders is two hops away, while the other is three or four hops away. In the current protocol, a node has no way to detect a collision like this and we conjecture that the overhead required to detect such collisions is not worth the cost. We opt to ignore these cases at this time.

If there is a success within two hops, a node will stop contention in the same slot but will contend in other slots. This results in an oscillation of the number of contenders in a neighborhood. A node needs to keep two estimates: one for the number of nodes that contend within two hops, n_c; the other for the number of nodes within two hops which need reservations, but cannot contend in the current slot due to a nearby success, n_b. Some heuristic constants are used to estimate the effect of a success on the number of contenders nearby. The effect of a success on its neighbors is modelled as follows: for a node one hop away from the success, a portion (R_1) of its neighboring contenders cease to contend in the current slot; for a node two hops away, this ratio is R_2; and for three hops away, R_3. The pseudo-Baysian algorithm becomes:

1. At the beginning of a reservation slot, a node resets its n_c and n_b as follows:

$$n_c := n_b;$$

$$n_b := 0.$$

   (for the very *first* reservation slot, $n_c := n_{c0}$, where n_{c0} is a predefined constant)
2. After every reservation cycle, on hearing an:

   idle

$$n_c := n_c - 1;$$

   collision

$$n_c := n_c + (e-2)^{-1};$$

   success if the success is some x hops away, where x is:

   zero

   done;

   one (it does not contend in the same slot anymore);

$$n_c := n_c * (1 - R_1) - 1;$$

$$n_b := n_b + n_c * R_1;$$

   two (it does not contend in the same slot anymore);

$$n_c := n_c * (1 - R_2) - 1;$$

$$n_b := n_b + n_c * R_2;$$

three

$$n_c := n_c * (1 - R_3);$$

$$n_b := n_b + n_c * R_3.$$

3. It then calculates the contention probability $p := 1/n_c$; if it is able to contend in the next cycle, it contends with probability p.

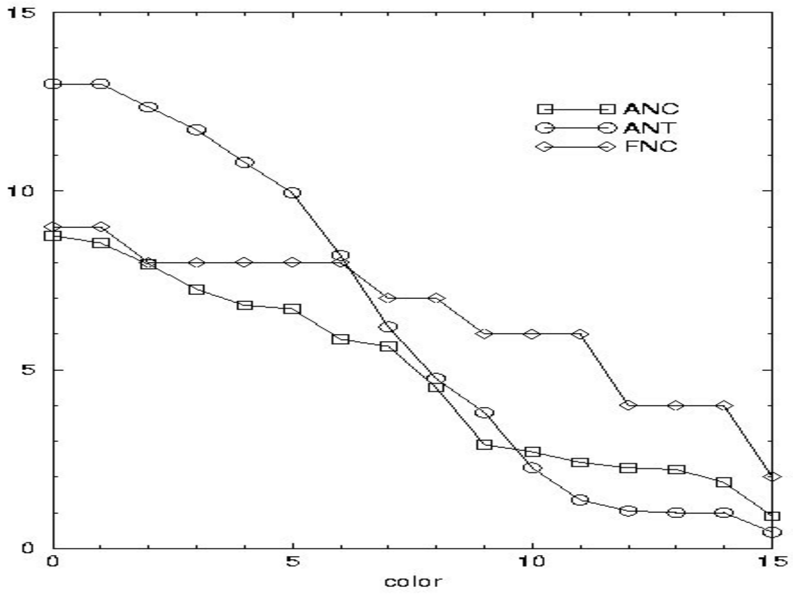

Figure 3: Simulation results of the FPRP using the multihop, pseudo-Baysian algorithm in a network with $N = 100$ and $R = 1.5$. 16 colors were necessary, numbered 0-15. In the figure, ANC is the Average Number of Cycles needed to assign each color, ANT is the Average Number of Transmission nodes assigned each color, and FNC is the Fixed Number of Cycles for each color when the FPRP runs without a central coordinator. The FNC permits a node to be colored with a probability higher than 99%.

The multihop pseudo-Baysian algorithm described above is implemented and tested in the graph coloring process as described in Section 4. The parameters, R_1, R_2 and R_3, are estimated with a separate program. In the simulations here, they are fixed at $R_1 = 0.80$, $R_2 = 0.60$, $R_3 = 0.33$. The number of cycles required for the protocol to converge for each color is used to study the speed with which the reservations are being made. For a network of $N = 100$ and $R = 1.5$, the simulation results are shown in Fig. 3.

Inspection of the simulation runs also showed that cases of non-isolated deadlock almost never occur. When the FPRP was applied 30 times to the above network, a deadlock occurred only 11 times, most of them involving only 2 nodes, and all of them were resolved by the elimination procedure. It is reasonable to conclude that the collision probability of the FPRP is very small and has no significant effect on the performance of the protocol.

The multihop pseudo-Baysian algorithm converges steadily and fast. In the simulation, a coordinator is used to globally monitor the coloring process to determine when all the nodes are colored. The Average Number of Cycles (ANC) required per color is shown in Fig. 3. However, use of such a coordinator is infeasible in a real network. It is possible based on experimentation to closely predict how many cycles are reasonable for the reservation of each slot, when the typical topology (node density, transmission range) of the network is given. The ANC values provide a useful reference when the parameters of the protocol are to be chosen. Because of the random nature of the FPRP, there is no coordination mechanism to tell the nodes when to stop. The number of cycles required for each color should be higher than the ANC in order to assign a color fully. For a given network, we refer to this set of fixed values as the Fixed Number of Cycles (FNC). The FNC values (shown in Fig. 3) are chosen as a reasonable upper bound on the number of cycles required. When the FNC values are used in the FPRP, it was empirically confirmed that a node can acquire a color (i.e. a slot) with a probability higher than 99%. A further increase in the number of cycles would drive this probability very close to 100%, but the gain is not likely not worth the cost in scheduling delay.

Once known, the FNC values can be built into the protocol. This permits protocol execution which needs no coordination at all. From a node's point of view, it knows how many colors are available, and which cycle is for which color. It simply uses the FPRP to acquire a color.

6 Future work and applications

TDMA systems require a mechanism for maintaining slot synchronization, and for providing sufficient inter-slot guard time to absorb differences in message propagation delays due to relative transmitter/receiver positions. Since there is no central base station in a mobile ad hoc network, nodes are unable to synchronize to a shared, pilot signal as is commonly done in cellular systems. However, it is still possible to implement a slot timing scheme for some low-to-medium bit rate wireless networks.

Using the time signal available via the Global Positioning System (GPS), military-grade GPS provides timing synchronization to less than 100 ns RMS [16] and commercial-grade GPS is only several time that figure. Even if we double or triple the GPS uncertaintly, and assume that inter-node distances are less than one kilometer, then for a low-rate system operating at 9.6 kbps, the required inter-slot guard time (measured in bits) is much less than a single bit time. Thus, considering only guard time, it is quite feasible to implement the FPRP protocol efficiently (recall that each phase of the five-phase dialogue only requires a logical bit to be transmitted). If the bit rate is increased to 20 Mbps (as in proposed Wireless ATM systems and future military systems), then the guard times increase to roughly 60-70 bits. Here, the transmission of a single, logical bit in a slot would be more expensive.

In both cases, however, the dominant term is not the GPS uncertainty, but rather the term that must account for the potential difference in signal propagation delays. Also, these guard times are small relative to the time typically required for current radios to switch between transmission and reception modes. For example, the IEEE 802.11 specification [2] calls for a switching time of 19 microseconds which, at a transmission rate of 20 Mbps, is 380 bits–roughly five times the required guard time. In the future, it is expected that improvements in hardware will lower this figure to that comparable to the guard time or less. What is clear is that, at present, the uncertainty introduced by the GPS time signal is small and not an impediment to implementation.

We intend to apply the FPRP to various situations, where TDMA schedules are desirable for carrying both data and control traffic. In particular, we are using the FPRP as the basis of an adaptive network control protocol framework that we are currently developing. In this framework, the nodes of a mobile ad hoc network use the FPRP to obtain one or more slots in the control frames to execute distributed network control protocols, thus allowing the nodes to organize themselves autonomously. The rare cases of non-isolated deadlock that arise are not a problem as they can be treated as transient cases of network unreliability. In a mobile wireless network, the reasons a packet may not be delivered are numerous (node mobility, link variability, packet droppage, etc.) and reliability mechanisms at both the link and/or transport layers are still necessary to guarantee the desired level of reliability.

In the current version of the FPRP, the key parameters of the network, such as the number of slots to be assigned N, the number of contention cycles for each slot M, the frequency with which fifth phase packing is used, and the topology parameters, R_1 through R_3, are all obtained heuristically from experiments with typical networks and fed into the FPRP in advance. A truely self-adaptive protocol would be able to adjust these parameters by itself. We intend to develop policies that permit nodes to estimate and adjust these parameters dynamically.

7 Conclusion

A new TDMA slot assignment protocol, viz. FPRP, has been presented. It allows nodes in a mobile ad hoc network to reserve TDMA broadcast slots and form broadcast schedules. It jointly and simultaneously performs the functions channel access and graph coloring. It does so without any centralized mechanism or constraint on scalability. Thus, it is well-suited for use in large, mobile networks. It requires minimal computation capability in the nodes and can be easily implemented, provided a global time synchronization signal of sufficient accuracy is available. Simulations show that it is as efficient as an existing, albeit simple, centralized protocol.

References

[1] European Telecommunications Standards Institute, 06921 Sophia Antipolis Cedex - France. *"Radio Equipment and Systems (RES): HIgh PErformance Radio Local Area Network (HIPERLAN), Type 1 Functional Specification"*, draft pr ets 300 652 edition, December 1995.

[2] Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ, 08855, USA. *"IEEE P802.11 Draft Standard for Wireless LAN: Medium Access Control (MAC) and Physical Layer (PHY) Specification"*, draft p802.11d5.0 edition, July 1996.

[3] D. Bertsekas and R. Gallager. *"Data Networks (2nd ed.)"*. Prentice-Hall, Englewood Cliffs, NJ, 1992.

[4] M. Post, A. Kershenbaum, and P. Sarachik. "A Distributed Evolutionary Algorithm for Reorganizing Network Communications". In *Proc. MILCOM*, pages 133–139, 1985.

[5] L. C. Pond and V. O. K. Li. A distributed time-slot assignment protocol for mobile multi-hop broadcast packet radio networks. In *Proc. IEEE MILCOM*, 1989.

[6] R. Ramaswami and K. K. Parhi. Distributed scheduling of broadcast in a radio network. In *Proc. IEEE INFOCOM*, 1989.

[7] A. Ephremides and T. Truong. Scheduling broadcasts in multihop radio networks. *IEEE Transactions on Communications*, COM-38(4), April 1990.

[8] S. Ramanathan and E. L. Lloyd. Scheduling algorithms for multihop radio networks. *IEEE/ACM Trans. on Networking*, 1:166–177, 1993.

[9] I. Cidon and M. Sidi. Distributed assignment algorithms for multihop packet radio networks. *IEEE Trans. on Computers*, 38:1353–1361, 1989.

[10] N. Funabiki and Y. Takefuji. A parallel algorithm for broadcast scheduling problem in packet radio networks. *IEEE Trans. on Communications*, 41:828–831, 1993.

[11] S. Ramanathan. "A Unified Framework and Algorithm for (T/F/C)DMA Channel Assignment in Wireless Networks". In *Proc. INFOCOM*, Kobe, Japan, April 1997.

[12] K. Joseph N. D. Wilson, R. Ganesh and D. Raychaudhuri. Packet CDMA versus dynamic TDMA for multiple access in an integrated voice/data pcn. *IEEE JSAC*, 11:870–884, 1993.

[13] F. A. Tobagi and L. Kleinrock. Packet switching in radio channels: part ii-the hidden terminal problem in carrier sense, multiple access modes and the busy tone solution. *IEEE Transactions on Communications*, COM-23(12):1417–1433, December 1975.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completene ss.* Freeman, New York, NY, 1979.

[15] R. L. Rivest. *Network Control by Baysian Broadcast (Report MIT/LCS/TM-287).* MIT, Laboratory for Computer Science, Cambridge, MA, 1985.

[16] Department of Defense Document MZ10298.001. *“NAVSTAR GPS User Equipment Introduction”*, February 1991.