

## Research Article

# An Efficient Algorithm to Find Broadcast Schedule in Ad Hoc TDMA Networks

Imtiaz Ahmad, Buthaina Al-Kazemi, and A. Shoba Das

*Department of Computer Engineering, Kuwait University, P.O. Box 5969, Safat 13060, Kuwait*

Correspondence should be addressed to Imtiaz Ahmad, imtiaz@eng.kuniv.edu.kw

Received 16 September 2008; Accepted 3 December 2008

Recommended by Nidal Nasser

The broadcast scheduling is of fundamental importance and practical concern for ad hoc network performance measures such as the communication delay and the throughput. The scheduling problem on hand involves determination of a collision-free broadcast schedule with the minimum length TDMA frame and the maximum slot utilization by efficient distribution of slots among stations. The problem is widely known as NP-complete, and diverse heuristic algorithms were reported to solve this problem recently. The intractable nature of the broadcast scheduling problem and its importance in ad hoc TDMA networks necessitates development of more efficient heuristic algorithms. In this paper, we developed a new heuristic approach which employs a tight lower bound derived from the maximal incompatibles and generates a search space from the set of maximal compatibles. The developed algorithm is very efficient and effective in conquering the intractable nature of the broadcast scheduling problem in the sense that it explores complex solution space in smaller CPU time. A comparison with existing techniques for the test examples reported in the literature shows that our algorithm achieves a collision-free broadcast with minimum frame length and the maximum slot utilization in relatively shorter time.

Copyright © 2008 Imtiaz Ahmad et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. INTRODUCTION

Ad hoc networks have attracted a lot of attention in the recent years because of their flexible structure and useful applications particularly in areas such as mobile commerce, combat search, and rescue, to name a few. These networks apply a packet switching technique over a shared radio channel to provide flexible high-speed communications between a large numbers of potentially mobile stations which may be geographically disbursed. Communication over the shared radio channel is established by broadcasting [1, 2]. Mobile stations are assumed to use unidirectional antennas. The wireless channel is assumed to be noise-free, and an unsuccessful reception is only due to collisions. Mobile stations operate in half-duplex mode; that is, a mobile station can transmit or receive but cannot do both at the same time. Thus, it is often necessary to use intermediate stations as relays to forward messages over the network to the intended recipients. Since every station in the network shares the same channel, it is a fundamental requirement that precautions are taken when messages are scheduled to be transmitted.

That is, stations should be scheduled in such a way that there is no destructive interference, or message collision. Collisions in ad hoc networks may occur in two ways: direct or hidden. Direct collision is a result of two adjacent stations broadcasting at the same time. Hidden collision occurs when two nonneighboring stations transmit simultaneously to a station that can receive messages from both senders [3].

Time division multiple access (TDMA) technology is envisioned to be widely used to provide collision-free packet transmission with quality-of-service (QoS) support for ad hoc networks [4, 5]. In a TDMA-based ad hoc network, time is divided into equal-length frames, and each frame is composed of a fixed number of unit-length transmission slots. The duration of each slot is equal to one maximum-length packet transmission time plus the maximum propagation time between two neighboring mobile stations in the network. Once the optimum transmission pattern for the TDMA frame is decided, the same frame is repeated over time. It should be ensured that all the stations must be allocated at least one time slot in every TDMA frame. This is one

of the constraints of scheduling problem and the other being the collision-free transmission. The broadcast scheduling problem in ad hoc networks involves the determination of a collision-free broadcast schedule with the minimum length TDMA frame and the way to distribute the slots among stations for maximum slot utilization. The frame structure is directly related to the main network performance measures such as the communication delay and the throughput. Frame length essentially determines the packet average delay and for a fixed frame length the channel utilization, which is the number of simultaneous transmissions of noninterfering stations, determines the throughput in ad hoc network [6, 7]. Therefore, broadcast scheduling problem is of crucial importance and practical concern to effectively harness the shared radio channel bandwidth in ad hoc networks.

A number of approaches with varying degree of success have been applied for solving scheduling problem for delay and throughput in ad hoc networks [1–17]. These approaches can be classified as graph theoretic [1–3, 13], graph coloring [6], and probabilistic approaches such as mean field annealing [5], tabu search [10], genetic algorithms [9, 11, 16], neural networks algorithms [14, 15, 17], mixed neural-genetic [7], and greedy randomized adaptive search procedure (GRASP) [12]. A good overview of the techniques applied for the broadcast scheduling problem can be found in [18]. Most of these algorithms are based on two phases: phase one minimizes the frame length without considering the slot usage, and phase two attempts to maximize the slot utilization within the frame achieved during the first phase. Optimizing the two objectives separately does not lead to a good solution with respect to both criteria. A better approach is to consider both of these criteria in an integrated fashion to solve the broadcast scheduling problem. Probabilistic approaches have considered both of these criteria at the same time and produced better results as compared with other techniques.

In this paper, a new heuristic algorithm based on concepts from the field of finite state machine synthesis [19, 20] is presented for the broadcast scheduling problem which determines the minimum frame length with the maximum slot utilization. The stations which can broadcast without collisions among themselves are grouped as maximal compatibles. A tight lower bound derived from set of maximal incompatibles forms the basis for deriving minimum frame length. The proposed algorithm applies set of rules on the maximal compatibles in order to maximize utilization of slots. To our knowledge, this approach based on maximal compatibles and maximal incompatibles for the broadcast scheduling problem has not been reported so far. A comparison with existing techniques for the test examples reported in the literature shows that our algorithm explores a complex solution-space in smaller CPU time.

The remaining paper is organized in the following manner. In Section 2, we discuss the broadcast scheduling problem based on finite state machine approach. Section 3 explains the heuristic scheduling algorithm proposed in this paper. Results are discussed in Section 4, and Section 5 concludes the paper.

## 2. BROADCAST SCHEDULING PROBLEM: FINITE STATE MACHINE APPROACH

In this section, we discuss the broadcast scheduling problem modeling the network as a finite state machine. An ad hoc TDMA network can be modeled by an undirected graph  $G = (V, E)$ , where the node set  $V$  represents the mobile stations and edge set  $E$  represents the set of transmission links between adjacent mobile stations in the network [8, 9]. Two stations  $i$  and  $j$  connected by an undirected edge  $(i, j) \in E$  means that both can directly receive packets transmitted from the other, but both cannot transmit simultaneously in the same time slot. It represents direct collision and we say stations  $i$  and  $j$  are one-hop neighboring stations. If an undirected edge  $(i, j) \notin E$  but there exists an intermediate node  $k \in V$  such that  $(i, k) \in E$  and  $(k, j) \in E$ , then stations  $i$  and  $j$  are referred as two-hop neighboring stations. A hidden collision is a result of two-hop neighbors transmitting in the same time slot. Two mobile stations can transmit in the same time slot without mutual interference if they are located more than two hops apart. Therefore, it is highly desirable for multiple stations to transmit during the same time slot provided that they do not cause any collision, either direct or hidden.

Some basic definitions related to finite state machine [20] as applicable to our scheduling problem on hand are presented here.

### *Finite state machine*

a triplet  $M = (I, \varepsilon, \delta)$  defines a finite state machine  $M$ , where  $I$  is a unit value function representing the transmission path between two adjacent stations,  $\varepsilon$  is the set of states of the machine,  $\delta : I^* \varepsilon \rightarrow \varepsilon$  is the next state function.

### *State transition graph*

a state transition graph (STG) describes an ad hoc TDMA network modeled as an FSM.

### *Compatible and incompatible*

two states  $s_i$  and  $s_j$  are compatible if and only if there is no collision (direct or hidden) between them. Otherwise,  $s_i$  and  $s_j$  are incompatible. A set of states is compatible (incompatible) if and only if each pair of states in the set is compatible (incompatible). A set of compatible states  $C_i$  ( $IC_i$ ) is referred to as a compatible (incompatible), for short.

### *Compatible (incompatible) covering*

a compatible  $C_i$  (incompatible  $IC_i$ ) covers another compatible  $C_j$  (incompatible  $IC_j$ ) if and only if every state contained in  $C_j$  ( $IC_j$ ) is contained in  $C_i$  ( $IC_i$ ).

### *Maximal compatible (incompatible)*

a compatible (incompatible) becomes a maximal compatible (maximal incompatible) if it is not covered by any other

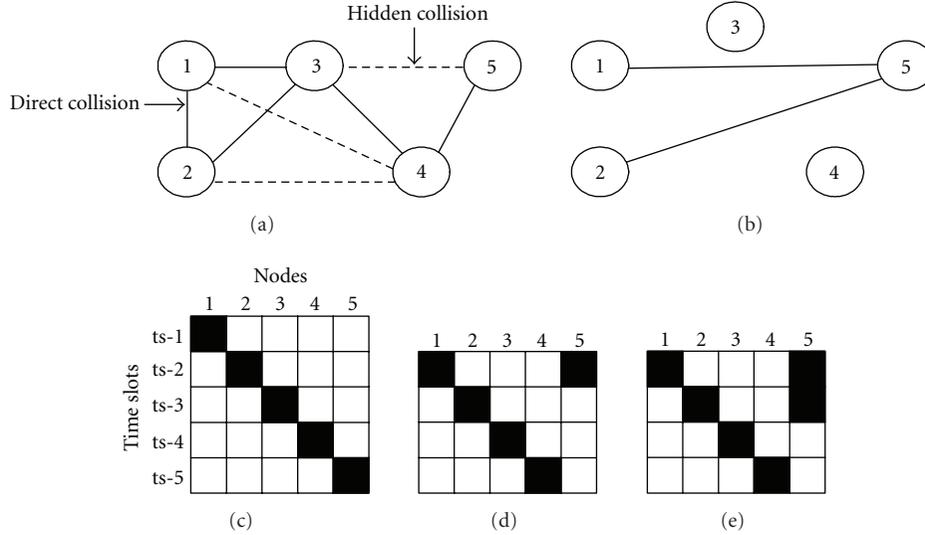


FIGURE 1: (a) Graph of a five-node ad hoc network, (b) complement graph of the five-node network, (c) trivial TDMA broadcast schedule, (d) and (e) optimal TDMA schedule.

compatible (incompatible). The set of all maximal compatibles (incompatibles) is denoted by  $\Omega_c$  ( $\Omega_i$ ). In the context of the broadcast scheduling problem, maximal compatibles are set of nodes, which can be assigned to the same time slot. These sets of maximal compatibles are maximal cliques in the complement graph of a given ad hoc network. The maximal incompatibles are the ones which cannot be assigned to the same time slot.

#### Lower and upper bounds

the lower bound is equal to the maximum degree of the graph modeling the network +1 (direct collision states) and upper bound is the number of nodes in the network. Optimum frame length is between the lower and upper bounds. The lower bound represents cardinality of a set which consists of incompatible state pairs with one state identical in all the pairs +1 and upper bound is the number of states in STG.

#### Tight lower bound

minimum slot length required for collision-free (direct and hidden) broadcast schedule accommodating all the stations in the network is the tight lower bound. In the FSM model, tight lower bound is the cardinality of an incompatible with maximum number of states.

#### Cover

a set that includes all the states of a machine is referred to as a cover of a machine  $M$ . The set of states of the machine, set of maximal compatibles, and set of incompatibles are all covers of the machine [20]. In the context of broadcast schedule, a cover refers to the condition that every node is allowed to transmit at least once in a TDMA cycle.

#### Optimal cover

a cover is said to be optimal  $\mathfrak{R}_m$  if number of compatibles in the cover is equal to the tight lower bound and utility factor

$\mu(s_i)$  of every state is maximum, that is, its frame length is equal to tight lower bound and slot utilization is maximum.

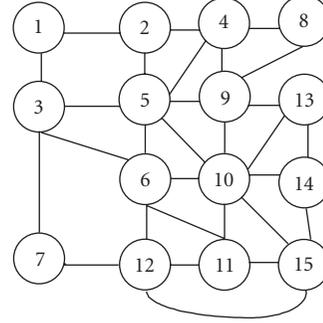
As an example, a 5-node network is shown in Figure 1(a) [9], where the graph has been augmented with two-hop neighbors to represent hidden collision. Nodes 1 and 3 are one-hop neighbors, while nodes 3 and 5 are two-hop neighbors. A trivial TDMA broadcast schedule satisfying all constraints is shown in Figure 1(c). However, it is not optimized with respect to both the frame length and the slot utilization.

An optimal TDMA broadcast schedule with respect to frame length is shown in Figure 1(d). Figure 1(b) shows the complement graph and the maximal compatibles are  $\{1, 5\}, \{2, 5\}, \{3\}, \{4\}$ . This set of maximal compatibles is a cover of the machine. The incompatible pairs are  $\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{3, 5\}, \{4, 5\}$ . For the graph of Figure 1(a), the maximal incompatible is  $\{1, 2, 3, 4\}$  as all the states in this are incompatible to each other representing direct and indirect collisions. The largest cardinality maximal incompatible will provide the tight lower bound for the frame length [7]. The optimal solution with the minimum frame length and the maximum slot utilization are shown in Figure 1(e), where the maximal compatibles  $\{1, 5\}, \{2, 5\}, \{3\}, \{4\}$  are assigned to different slots.

### 3. SCHEDULING ALGORITHM

This section discusses the scheduling algorithm for a 15-node network [12] shown in Figure 2 and modeled as an FSM ( $M_1$ ). The goal of the algorithm is to find an equivalent FSM which is a minimized cover of the given FSM. The basic steps of the algorithm are summarized in Algorithm 1 and individual steps are explained in detail in subsections. From the STG of  $M_1$ , all the compatible (incompatible) pairs which are more than 2 hops away to each other are generated. If there are no compatible pairs, then no two states can be allotted to the same slot, hence, all the states of the machine

Source node	Destination nodes							
1	2	3						
2	4	5	1					
3	5	6	7	1				
4	5	8	9	2				
5	6	9	10	2	3	4		
6	10	11	12	3	5			
7	12	3						
8	9	4						
9	10	13	4	5	8			
10	11	13	14	15	5	6	9	
11	12	15	6	10				
12	15	6	7	11				
13	14	9	10					
14	15	10	13					
15	10	11	12	14				

(a) FSM table representation ( $M_1$ ) of the network

(b) Undirected graph of the network (STG)

FIGURE 2: Network with 15 stations.

```

Input:   FSM table of network
Output:   $\mathfrak{R}_m$  - Optimum cover
(1) From FSM table of network, generate all compatible (incompatible) pairs;
    IF (no compatibles) {
         $\mathfrak{R}_m \leftarrow$  set of states  $\varepsilon$ ;
        stop; }
    else Find maximal compatibles  $\Omega_m$  and incompatibles  $\Omega_i$ ;
(2) Find the tight lower bound L and the bound incompatible  $IC_b$ ;
(3) Form compatible groups CG with respect to  $IC_b$ ;
(4) Let limit =  $\min(|CG_1|, 100)$ ;  $j = 1$ ; Let  $s_1$  be the first state in  $IC_b$ ;
(5) while ( $j <$  limit) {
    Include in schedule  $\mathfrak{R}(j)$ , compatible  $C_j \in CG(s_1)$ ;
    for (every state  $s_i \in IC_b$  where  $s_i \neq s_1$ ) {
        (a) Apply selection criteria and include a compatible  $C_n \in CG(s_i)$  in  $\mathfrak{R}(j)$ ;
        (b) for (every state  $s_k \in C_n$ ) Increment utility factor  $\mu(s_k)$ ; }
    for (every state  $s_i \mid s_i \in \varepsilon, s_i \notin \mathfrak{R}(j)$ )
        Include in  $\mathfrak{R}(j)$  largest cardinality compatible covering the state  $s_i$ ;
     $j = j + 1$ ; }
(6) for (every  $\mathfrak{R}(j)$ )
    compute the performance indices and choose  $\mathfrak{R}_m$ .

```

ALGORITHM 1: Outline of broadcast scheduling algorithm.

which is the upper bound are the solution and every state has to be in a separate slot. The algorithm terminates at this point. If the algorithm proceeds, solution is obtained at Step (6).

The complexity of our algorithm is dictated by the number of maximal compatibles generated in Step (1) of the algorithm. Modeling an ad hoc TDMA network as an undirected graph, where the nodes  $n$  represent the stations and edges  $m$  represent the transmission path between two stations, the maximal compatibles represent the cliques present in the graph. The complexity for finding cliques in a graph is  $O(n*m*p(n))$  [21], where the polynomial  $p(n)$  represents the number of maximal compatibles in the graph. Step (2) basically sorts the incompatibles and sorting has a complexity  $O(p(n)\log(p(n)))$ . As the rest of

the steps basically deal with fewer than  $p(n)$  compatibles, the complexity of Step (3) to Step (6) is much smaller than the other two steps. For the graphs, where  $n*m \gg p(n)$ , complexity will be  $O(n*m*p(n))$ .

### 3.1. Generation of maximal compatibles (incompatibles)

The set of maximal compatibles (incompatibles) is generated from the set of compatibles (incompatibles) by excluding compatibles (incompatibles) which are covered by other compatibles (incompatibles). For generating compatible pairs, maximal compatibles and incompatibles, the part of state minimization algorithm of [20] is used with the modified definition of the FSM. For FSM of Figure 2, the

TABLE 1: Maximal compatibles and incompatibles of  $M_1$ .

Maximal compatibles ( $\Omega_m$ )				Maximal incompatibles ( $\Omega_i$ )	
(12,13,2)	(13,7,2)	(12,4)	(15,2)	(13,10,9,8,5,4)	(12,11,10,6,5,3)
(12,9,1)	(10,7)	(11,4)	(12,13,1)	(14,15,12,11,10,6,5)	(10,9,6,5,4,3,2)
(15,8,3)	(9,7)	(14,8,3)	(15,8,1)	(14,15,13,11,10,9,6,5)	(12,11,7,6,5,3)
(14,8,7)	(8,6)	(13,3)	(10,1)	(15,12,11,7,6,5)	(10,9,8,5,4,2)
(12,8,1)	(14,7,4)	(11,2)	(14,8,1)	(13,10,9,6,5,4)	(6,5,4,3,2,1)
(11,8,1)	(15,4)	(14,7,2)	(11,10,9,6,5,3)	(7,6,5,3,1)	

TABLE 2: Compatible groups of  $M_1$ .

Incompatible state $s_i \in IC_b$	Compatible groups $CG(s_i)$	Weight of compatibles in order $\{W(C_j)\}$
14	(14,7,4) (14,8,3) (14,7,2) (14,8,7) (14,8,1)	15, 16, 16, 20, 20
15	(15,4) (15,2) (15,8,3) (15,8,1)	8, 9, 15, 19
13	(13,3) (12,13,2) (13,7,2) (12,13,1)	7, 14, 15, 16
11	(11,4) (11,2) (11,8,1)	7, 8, 18
10	(10,7) (10,1)	8, 9
9	(9,7) (12,9,1)	8, 14
6	(8,6)	9
5	(5)	1

number of states  $n = 15$  and the set of states,  $\varepsilon = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$ . The FSM has 23 maximal compatibles and 12 maximal incompatibles as listed in Table 1.

### 3.2. Tight lower bound

Lower bound takes into account only direct collision states. A tight lower bound is hence derived by taking into consideration the hidden collision states. In the FSM model, an incompatible represents the states which have direct or indirect collision. A maximal incompatible with the highest cardinality represents all the states which have either a direct or indirect collision and hence cannot be in the same time slot. Hence, the tight lower bound for the frame length is imposed by a maximal incompatible with the highest cardinality [19]. In addition, the optimum solution should cover all the internal states of the FSM which is the upper bound.

Finding the highest cardinality maximal incompatible from  $\Omega_i$ , tight lower bound =  $|\{14, 15, 13, 11, 10, 9, 6, 5\}| = 8$  and let this incompatible be  $IC_b = \{14, 15, 13, 11, 10, 9, 6, 5\}$ . Every state in  $IC_b$  has to be covered by a different maximal compatible in the solution set. If there exist more than one maximal incompatible with the maximum cardinality, choose one at random.

### 3.3. Compatible group formation

The broadcast schedule is built by including compatibles which cover every state in  $IC_b$ . As several compatibles cover a state in  $IC_b$ , a weight factor is associated with maximal compatibles to aid in selecting a compatible. Compatibles with smaller weights are more essential than a bigger weight

compatible and they get a preference if there is a tie in selection in the solution schedule.

The compatible group formation is done in three steps:

- (1) group the compatibles which cover one state of  $IC_b$ ;
- (2) find the weight of maximal compatibles;
- (3) sort the compatibles in CG according to their weights in ascending order.

*Step 1.* for every state  $s_i \in IC_b$ , form a cover group such that it consists of all maximal compatibles which covers  $s_i$ , that is,  $CG(s_i) = \{C_j \mid s_i \in IC_b, s_i \in C_j\}$

*Step 2.* for every state  $s_i \in \varepsilon$ , associate a weight  $\omega(s_i)$  which is the number of maximal compatibles covering the state  $s_i$ . Adding  $\omega(s_i)$  of every state  $s_i \in C_j$ , find the weight of maximal compatible,  $W(C_j)$ . The compatible  $C_j = (14, 7, 4)$  has three states 14, 7, and 4. Find the number of maximal compatibles covering every state in this compatible from Table 1,  $\omega(14) = 5$ ,  $\omega(7) = 6$ , and  $\omega(4) = 4$ . Add the weight of individual states in  $C_j$ , and find weight  $W(C_j)$ , which is 15.

*Step 3.* sort the compatibles in CG according to their weights  $W(C_j)$  in ascending order.

In Table 2, compatible groups for every state in  $IC_b = (14, 15, 13, 11, 10, 9, 6, 5)$  formed using the above three listed steps are shown. The maximal compatibles are grouped to cover the respective states in  $IC_b$ .

### 3.4. Iteration limit

Multiple feasible schedules are possible for the broadcast schedule problem. The performance parameters aid in selecting the optimum schedule. The states in  $IC_b$ , weight of

TABLE 3: Feasible schedules of  $M_1$ .

Schedule index $j$	Schedule $\mathfrak{R}(j)$
1	(14,7,4) (15,8,3) (12,13,2) (11,8,1) (10,7) (12,9,1) (8,6) (5)
2	(14,8,3) (15,8,1) (12,13,2) (11,4) (10,7) (12,9,1) (8,6) (5)
3	(14,7,2) (15,8,3) (12,13,1) (11,4) (10,7) (12,9,1) (8,6) (5)
4	(14,8,1) (15,8,3) (12,13,2) (11,4) (10,7) (12,9,1) (8,6) (5)
5*	(14,8,7) (15,8,3) (12,13,2) (11,8,1) (10,7) (12,9,1) (8,6) (5) (11,4)

TABLE 4: Performance characteristics of  $M_1$ .

Schedule index $j$	Average time delay $\tau$	Throughput $\sigma$	Frame length ( $L$ )
1	6.8	20	8
2	7.1	19	8
3	6.9	19	8
4	7.1	19	8
5*	6.45	22	9

compatibles, and utility factor of states in a compatible guide the selection of compatibles to be included in the solution. Among the several compatibles in  $CG(s_i)$  covering the first state  $s_i \in IC_b$ , one of the compatibles has to be selected as a starting point. The states in this compatible decide the formation of rest of the schedule. Hence, the compatibles  $CG(s_i)$  covering the first state  $s_i$  are used in order as a starting point and several schedules are derived. An upper limit of 100 is set for the number of schedules generated. The number of schedules for  $M_1$  is 5 as there are five compatibles in  $CG(14)$ , where  $s_i = 14$  is the first state in  $IC_b$ . One of the compatibles in the first group is included by default in every schedule and the selection criteria are applied only for subsequent CGs. For  $M_1$ , in the first schedule  $\mathfrak{R}(1)$ , the first compatible in  $CG(14)$ , namely, (14,7,4) is included. At this stage,  $\mathfrak{R}(1) = \{(14,7,4)\}$ .

### 3.5. Compatible selection criteria

The feasible schedule should basically cover all the states of the machine without any collision and with the minimum frame length and the maximum slot utilization. A compatible  $C_j \in CG(s_i)$  is included in a feasible schedule  $\mathfrak{R}(k)$  based on two criteria applied in succession: (i) *uncovered State factor*, that is, number of uncovered states in  $C_j$ , and (ii) cardinality of the compatible  $C_j$ .

Initially, for every  $C_j \in CG(s_i)$  compute *uncovered state factor*  $\gamma(C_j) = |\{s_i \mid s_i \in C_j \wedge s_i \notin \mathfrak{R}(k)\}|$ . Choose the compatible with maximum  $\gamma(C_j)$ . If there is a tie, then apply the second criterion, that is, choose the compatible with higher cardinality as this represents a compatible which if included will improve the channel utilization. If there is still a tie, select the first compatible in the list.

In  $M_1$ ,  $\mathfrak{R}(1) = \{(14, 7, 4)\}$  and the uncovered states are  $\{1, 2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 15\}$ . In the group of next state 15 in  $IC_b$ , that is,  $CG(15)$ ,  $\gamma(C_j)$  of the compatibles  $\{(15,4) (15,2) (15,8,3) (15,8,1)\}$ , respectively, are (1, 2, 3, 3).

Out of these, the compatibles  $\{(15,8,3) (15,8,1)\}$  have a tie as none of the states in them are covered by  $\mathfrak{R}(1)$  and both of them have  $\gamma(C_j) = 3$ , in this compatible group. Hence, the first one in the list is included.  $\mathfrak{R}(1) = \{(14,7,4) (15,8,3)\}$ . Proceeding to next three states,  $\mathfrak{R}(1) = \{(14,7,4) (15,8,3) (12,13,2) (11,8,1) (10,7)\}$ . At this point, the states that are not covered by  $\mathfrak{R}(1)$  are (5,6,9). Taking the group of next state 9,  $CG(9) = \{(9,7) (12,9,1)\}$ . Both cover only one uncovered state 9, and the tie is broken by choosing the second one with greater cardinality as there are two covered states, thereby improving the channel utilization. Including the next two compatibles in order to cover all the states in the network, the schedule  $\mathfrak{R}(1) = \{(14,7,4) (15,8,3) (12,13,2) (11,8,1) (10,7) (12,9,1) (8,6) (5)\}$ .

Let  $N$  be the number of nodes and let  $L$  be the number of slots in a TDMA schedule. The feasible transmission schedule will be an  $N * L$  binary matrix  $S = (s_{ij})$  such that

$$s_{ij} = \begin{cases} 1, & \text{if time slot } i \text{ in a frame is assigned to a node } j, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

In  $M_1$ , number of nodes  $N = 15$  and  $L = 8$ , and all the schedules generated are shown in Table 3. Even though 5 schedules were generated, one of them did not satisfy the constraint that all states should be covered with in tight lower bound and the solution slot length has to be increased above the tight upper bound by 1 to accommodate a compatible covering the uncovered state. Hence, this solution had a slot length of 9 and this schedule is marked with (\*) in Table 3.

### 3.6. Optimum schedule selection

The following performance indices are computed for every feasible schedule  $\mathfrak{R}(j)$ .

- (i) Average time delay for each node to broadcast  $\tau$  [14].
- (ii) Throughput  $\sigma$ .

Average time delay

$$\tau = \frac{L}{N} \sum_{i=1}^N \left( \frac{1}{\sum_{j=1}^L s_{ij}} \right). \quad (2)$$

Throughput

$$\sigma = \sum_{i=1}^N \sum_{j=1}^L s_{ij}. \quad (3)$$

Nodes \ Slots	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1				■			■								■
2			■					■							■
3		■									■	■			
4	■							■			■				
5							■			■					
6	■								■			■			
7						■		■							
8					■										

■ Allotted slots

FIGURE 3: Optimal schedule of 15-station network.

Nodes \ Slots	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1				■	■									■		
2						■						■				
3	■	■								■						■
4									■		■		■			
5		■	■				■	■								

■ Allotted slots

FIGURE 4: Optimal schedule of 16-station network.

TABLE 5: Specifications of test cases.

Problem no.	No. of nodes	No. of edges	Maximum degree
1 [2]	7	6	7
2 [12]	15	29	7
3 [2]	16	22	4
4 [12]	30	70	8
5 [12]	40	66	7

TABLE 6: Performance measurements.

Problem no.	Network nodes ( $N$ )	Average time delay ( $\tau$ )	Throughput ( $\sigma$ )
1	7	3	7
2	15	6.8	20
3	16	4.8	17
4	30	9.2	35
5	40	6	64

The performance indices of the above schedules for  $M_1$  are shown in Table 4 along with the frame length.

The schedule within the tight lower bound which yields the maximum throughput and the minimum average time delay is selected as the optimum schedule  $\mathfrak{R}_m$ . The optimum

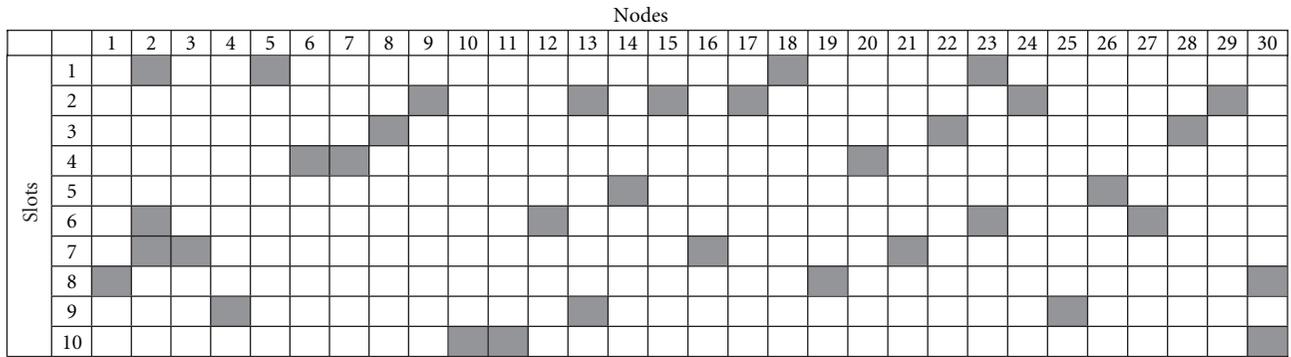
schedule for  $M_1$  is  $\mathfrak{R}(1) = \{(14,7,4) (15,8,3) (12,13,2) (11,8,1) (10,7) (12,9,1) (8,6) (5)\}$ .

#### 4. EXPERIMENTAL RESULTS

The experiments were conducted on a Pentium 4/1.7 GHz PC and tested on network examples available in the literature. The specifications of the three network cases, namely, 15, 30, and 40 station networks introduced by Wang and Ansari [5] which have become the benchmark test cases for broadcast scheduling problem and the other networks found in literature are shown in Table 5. The broadcast networks are modeled as FSMs modifying the standard definitions to suit the network parameters. The maximal compatibles and incompatibles were generated by modifying an algorithm for state minimization of FSMs [20].

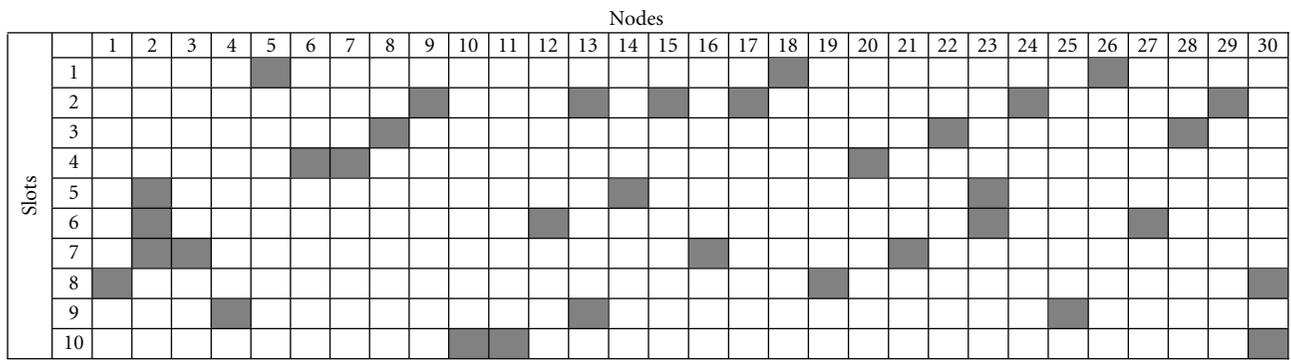
The optimal schedules for the test cases 2 to 5 are shown in Figures 3, 4, 5, and 6, respectively. The rows represent the slots in a TDMA frame and columns represent the nodes in the network. Out of the four feasible schedules in 15 station networks, only one is the optimal solution. Two feasible schedules exist for problem no. 3 and the optimum solution is the one shown in Figure 4.

In 30-node network, there are two feasible schedules, which are shown in Figures 5(a) and 5(b),  $\{(18,23,5,2)$



■ Allotted slots

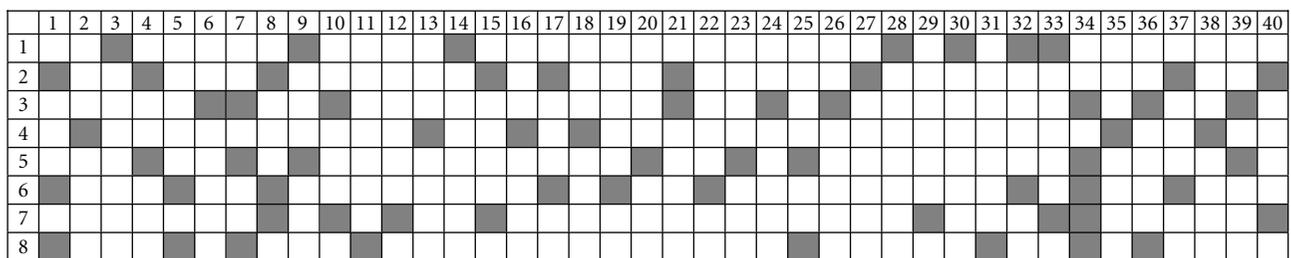
(a)



■ Allotted slots

(b)

FIGURE 5: Optimal schedule of 30-station network.



■ Allotted slots

FIGURE 6: Optimal schedule of 40-station network.

TABLE 7: Comparison of average time delay and number of slots.

Nodes ( $N$ )	Our algorithm		BSC-NCNN [14]		HNN-GA [7]		SVC [6]		MFA [5]		GNN [2]	
	( $L$ )	( $\tau$ )	( $L$ )	( $\tau$ )	( $L$ )	( $\tau$ )	( $L$ )	( $\tau$ )	( $L$ )	( $\tau$ )	( $L$ )	( $\tau$ )
15	8	6.8	8	6.8	8	7.0	8	7.2	8	7.2	8	7.1
30	10	9.2	10	9.2	10	9.3	10	10	12	10.5	10	9.5
40	8	6	8	5.8	8	6.3	8	6.76	9	6.9	8	6.2

(17,24,15,13,29,9) (22,28,8) (20,6,7) (14,26) (12,23,27,2) (16,21,3,2) (30,19,1) (13,4,25) (30,11,10)}, and {(18,5,26) (17,24,15,13,29,9) (22,28,8) (20,6,7) (14,23,2) (12,23,27,2) (16,21,3,2) (30,19,1) (13,4,25) (30,11,10)}. Average time delay and throughput are identical in both the schedules and hence both are optimum schedules. There are 14 feasible schedules in the case of 40-node network but only one of them is an optimum which is shown in Figure 6.

We report two performance parameters, namely, average time delay ( $\tau$ ) and throughput ( $\sigma$ ), and these are listed in Table 6 for the test cases. The average time delay and number of slots in a TDMA frame of our algorithm are compared with respective values of [2], [5], [6], [7], and [14] which is presented in Table 7. We achieved the average delay close to BSC-NCNN [14] and much better than all the other cases listed in this table and maximum slot utilization in negligible time.

For the problem no. 2, our algorithm has an average delay of 4.8 with a throughput of 17 and the solution in [2] has a delay of 5 with a throughput of 16. The comparison tables show that our algorithm could generate optimum schedule for the benchmark cases with minimum average delay and maximum utilization compared to almost all other cases shown in the table.

## 5. CONCLUSION

A heuristic algorithm for finding an optimal feasible broadcast schedule for ad hoc TDMA networks is presented. By modeling the network as an FSM, the concept of maximal compatibles and incompatibles is used to find a schedule that will minimize the frame length and maximize the slot utilization in an integrated fashion. To our knowledge, this approach of modeling the network as an FSM has not been reported so far. Suboptimal schedules with an increased time slot per frame also were generated, which gives more throughputs. The algorithm was efficient for all the benchmark cases and generated schedules within the tight lower bound in negligible time. Future work will take into account the QoS of the nodes (i.e., bandwidth or delay requirements) when determining the broadcast schedule.

## ACKNOWLEDGMENT

This research is funded by Kuwait University Grant EO 07/06.

## REFERENCES

- [1] A. Ephremides and T. V. Truong, "Scheduling broadcasts in multihop radio networks," *IEEE Transactions on Communications*, vol. 38, no. 4, pp. 456–460, 1990.
- [2] N. Funabiki and Y. Takefuji, "A parallel algorithm for broadcast scheduling problems in packet radio networks," *IEEE Transactions on Communications*, vol. 41, no. 6, pp. 828–831, 1993.
- [3] S. Ramanathan and E. L. Lloyd, "Scheduling algorithms for multihop radio networks," *IEEE/ACM Transactions on Networking*, vol. 1, no. 2, pp. 166–177, 1993.
- [4] H. Fattah and C. Leung, "An overview of scheduling algorithms in wireless multimedia networks," *IEEE Wireless Communications*, vol. 9, no. 5, pp. 76–83, 2002.
- [5] G. Wang and N. Ansari, "Optimal broadcast scheduling in packet radio networks using mean field annealing," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 2, pp. 250–260, 1997.
- [6] J. Yeo, H. Lee, and S. Kim, "An efficient broadcast scheduling algorithm for TDMA ad-hoc networks," *Computers & Operations Research*, vol. 29, no. 13, pp. 1793–1806, 2002.
- [7] S. Salcedo-Sanz, C. Bousoño-Calzón, and A. R. Figueiras-Vidal, "A mixed neural-genetic algorithm for the broadcast scheduling problem," *IEEE Transactions on Wireless Communications*, vol. 2, no. 2, pp. 277–283, 2003.
- [8] Z. Cai, M. Lu, and C. N. Georghiades, "Topology-transparent time division multiple access broadcast scheduling in multihop packet radio networks," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 4, pp. 970–984, 2003.
- [9] G. Chakraborty, "Genetic algorithm to solve optimum TDMA transmission schedule in broadcast packet radio networks," *IEEE Transactions on Communications*, vol. 52, no. 5, pp. 765–777, 2004.
- [10] Y. Peng, B. H. Soong, and L. Wang, "Broadcast scheduling in packet radio networks using mixed tabu-greedy algorithm," *Electronics Letters*, vol. 40, no. 6, pp. 375–376, 2004.
- [11] C. Y. Ngo and V. O. K. Li, "Centralized broadcast scheduling in packet radio networks via genetic-fix algorithms," *IEEE Transactions on Communications*, vol. 51, no. 9, pp. 1439–1441, 2003.
- [12] C. W. Commander, S. I. Butenko, P. M. Pardalos, and C. A. S. Oliveira, "Reactive GRASP with path relinking for the broadcast scheduling problem," in *Proceedings of the 40th Annual International Telemetry Conference (ITC '04)*, pp. 792–800, San Diego, Calif, USA, October 2004.
- [13] J.-C. Chen, Y.-C. Wang, and J.-T. Chen, "A novel broadcast scheduling strategy using factor graphs and the sum-product algorithm," *IEEE Transactions on Wireless Communications*, vol. 5, no. 6, pp. 1241–1249, 2006.
- [14] H. Shi and L. Wang, "A hybrid neural network for optimal TDMA transmission scheduling in packet radio networks," in *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN '05)*, vol. 5, pp. 3210–3213, Montreal, Canada, July-August 2005.
- [15] W. Bi, Z. Tang, J. Wang, and Q. Cao, "An improved neural network algorithm for broadcast scheduling problem in packet radio," *Neural Information Processing-Letters and Reviews*, vol. 9, no. 1, pp. 23–29, 2005.
- [16] X. Wu, B. S. Sharif, O. P. Hinton, and C. C. Tsimenidis, "Solving optimum TDMA broadcast scheduling in mobile ad hoc networks: a competent permutation genetic algorithm approach," *IEE Proceedings: Communications*, vol. 152, no. 6, pp. 780–788, 2005.
- [17] Y.-J. Shen and M.-S. Wang, "Broadcast scheduling in wireless sensor networks using fuzzy Hopfield neural network," *Expert Systems with Applications*, vol. 34, no. 2, pp. 900–907, 2008.
- [18] C. W. Commander, S. I. Butenko, and P. M. Pardalos, "On the performance of heuristics for broadcast scheduling," in *Theory and Algorithms for Cooperative Systems*, chapter 1, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.
- [19] R. Puri and J. Gu, "An efficient algorithm to search for minimal closed covers in sequential machines," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 6, pp. 737–745, 1993.

- [20] I. Ahmad and A. S. Das, "A heuristic algorithm for the minimization of incompletely specified finite state machines," *Computers & Electrical Engineering*, vol. 27, no. 2, pp. 159–172, 2001.
- [21] B. Rosgen and L. Stewart, "Complexity results on graphs with few cliques," *Discrete Mathematics and Theoretical Computer Science*, vol. 9, pp. 127–136, 2007.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

