Title:

A Heuristic Algorithm for Optimum Transmission Schedule in Broadcast Packet Radio Networks

Authors' names and affiliations:

Goutam Chakraborty
Department of Software and Information Science,
Iwate Prefectural University, Takizawamura, Iwate, Japan - 020-0193.
E-mail: goutam@soft.iwate-pu.ac.jp Tel: +81-19-694-2584 Fax: +81-19-694-2691

Debasish Chakraborty
Research Institute of Electrical Communication
Tohoku university, 2-1-1 Katahira, AobaKu, Sendai, Japan - 980-8577.
E-mail: deba@shiratori.riec.tohoku.ac.jp Tel/Fax: +81-22-726-1667

Norio Shiratori
Research Institute of Electrical Communication
Tohoku university, 2-1-1 Katahira, AobaKu, Sendai, Japan - 980-8577.
E-mail: norio@shiratori.riec.tohoku.ac.jp Tel/Fax: +81-22-217-5453

Contact Author:

Goutam Chakraborty
Department of Software and Information Science,
Iwate Prefectural University, Takizawamura, Iwate, Japan - 020-0193.
E-mail: goutam@soft.iwate-pu.ac.jp Tel: +81-19-694-2584 Fax: +81-19-694-2691

Abstract
Packet Radio (PR) networks are to provide data communications among a set of nodes distributed over a wide region. Transmission from nodes are broadcast in nature. Where direct communication between two nodes is not possible, connection is established in multiple hops. A time division multiple access (TDMA) protocol is adopted for conflict free communication among different nodes. The goal is to find a *conflict free transmission schedule* for different nodes at different time-slots of a fixed length time frame, called TDMA cycle. The optimization criterion is primarily to (1) minimize the TDMA cycle length, and then to (2) maximize the number of transmissions. The problem is proved to be NP-complete. A randomized algorithm is proposed, which is very efficient to achieve the first optimization criterion. The results are shown to be superior compared to other recently reported competitive algorithms.

Keywords:
packet radio network, time division multiple access, optimum transmission schedule

A Heuristic Algorithm for Optimum Transmission Schedule in Broadcast Packet Radio Networks

Abstract

Packet Radio (PR) networks are to provide data communications among a set of nodes distributed over a wide region. Transmission from nodes are broadcast in nature. Where direct communication between two nodes is not possible, connection is established in multiple hops. A time division multiple access (TDMA) protocol is adopted for conflict free communication among different nodes. The goal is to find a *conflict free transmission schedule* for different nodes at different time-slots of a fixed length time frame, called TDMA cycle. The optimization criterion is primarily to (1) minimize the TDMA cycle length, and then to (2) maximize the number of transmissions. The problem is proved to be NP-complete. A randomized algorithm is proposed, which is very efficient to achieve the first optimization criterion. The results are shown to be superior compared to other recently reported competitive algorithms.

1 Introduction

Packet Radio networks (PRNET) are widely used for wireless communication over a wide geographical area, where direct radio or cable connection is impractical. A host of papers, published during '84 to '87, contain nice survey about this problem and its variations [1, 2, 3, 4, 5, 6].

The radio frequency band, available for use to a network system, can be allocated to the different nodes in different ways: frequency division [5, 7], time division [8], code division, and spatial reuse. When a single radio channel is used, the communication can be facilitated either by broadcasting, or by activating a subset of network links in proper sequence [9, 10, 11]. In general, when nodes transmit packets in broadcast mode using omnidirectional antennas, network management is simple if all nodes are tuned to the same channel frequency, and use time division and spatial reuse [12, 13, 14]. By spatial reuse we mean, a set of nodes can transmit the same channel at the same time, if that does not cause any interference, due to their far away geographical locations. For example, using the UHF band for ground mobile operation, where the radio range is inherently short, rendering spatial reuse is a natural outcome. Sometimes directional antenna or low transmission power is used to facilitate spatial reuse of channel [15, 16]. Since all nodes can not directly communicate, due to distance or other obstructions, nodes act as store-and-forward repeaters facilitating multi-hop connection. The packet radio network (PRNET) model we assume here uses time division multiplex with spatial reuse.

A packet radio network can be modeled by an undirected graph, where the nodes represent the transreceiver stations. A link between two nodes is present when they can transmit and receive packets directly. An example of a network with six nodes is shown in Fig. 1(a). Here, node-1 can communicate directly with node-2 and node-3, but not with the remaining nodes. When node-1 has to transmit a packet to node-5 say, it has to use node-3 and node-4 as intermediate repeaters.

In the PR network, each station can transmit or receive, which is controlled by its control unit. When a node transmits, all its neighbors connected by direct link in the graph, can receive. The neighboring node/s could absorb the packet, if it is so designated. Else, it may store it to transmit it later, in which case it acts as an intermediate repeater.

We consider a fixed topology PRNET, where a single wide band Radio channel is used by all nodes. A time division multiple access (TDMA) protocol is used [4]. The transmissions of packets are controlled by a single clock. The time is divided into distinct frames consisting of fixed length time-slots. A time-slot equals to the total transmission time required for a single packet to be transmitted and received by a pair of neighboring nodes. Many nodes may transmit simultaneously at the same time-slot without conflict, if they are far apart. In one TDMA frame, all the nodes must be able to transmit at least once. This is termed as *no-transmission* constraint.

The basic optimization objective is to get the smallest length TDMA frame, where many nodes are allowed to transmit simultaneously in a single time-slot in a conflict free manner. The secondary objective is to maximize the number of such transmissions for maximum utilization of the channel. Depending on the traffic distribution over the network, it is possible that some nodes may need more transmissions than others in a single TDMA frame. At first, we will not consider that situation and assign single transmission for each node in a TDMA frame, and minimize the TDMA frame length. In section 3.2, we will modify the algorithm to accommodate situation, where the traffic demand at different nodes are different.

When multiple transmissions per node are scheduled in a TDMA frame, the number of transmissions for individual nodes should conform to their traffic demand [9]. Sometimes, even though traffic demands from different nodes are considered to be the same, the number of transmissions in the TDMA frame is maximized without considering uniform transmission allocation. In that case, one objective could be the fairness of allocation, where divergence of transmission allocations for different nodes should be as small as possible. In addition, one should ensure that the waiting time between transmissions for each node should be concentrated towards the average waiting time - another measure of fairness. These fairness objectives are not considered in any algorithm reported so far.

In addition to *no-transmission* constraint, there are other constraints, namely the *primary conflict* and the *secondary conflict. Primary conflict* says that a particular node can not transmit and receive in the same time-slot. In other words, two neighboring nodes can not transmit simultaneously. A *secondary conflict* occurs when two or more packets arrive at a node in a single time-slot. This will occur when two nodes at a distance of two hops are allowed to transmit simultaneously. Then the intermediate node will receive two different packets from two directly connected nodes, at the same time slot. The transmission schedule in the TDMA cycle should be such that the *primary* and the *secondary conflicts* are avoided.

We propose a simple and fast randomized algorithm to find a pool of valid solutions of the problem. Though the optimization criterion is not considered while generating solutions, the best in the pool is the optimum solution for all the problems tried. Simulations were done with problems of various complexities and the results compared with recent works including one using neural network [17, 18] and the other using mean field annealing [19].

The remainder of the paper is organized as follows. A formal introduction of this scheduling problem and the terms used in the paper are explained in section 2. In section 3, we describe the algorithm. In section 3.2, an extension of the algorithm to consider non-uniform traffic demand at different nodes, is explained. The pseudocode and the complexity analysis of the algorithm are available in the appendix. The details of simulation and results, and comparison with earlier works are in section 4. Section 5 is the conclusion including discussions of possible extensions of this work.

2 The Problem

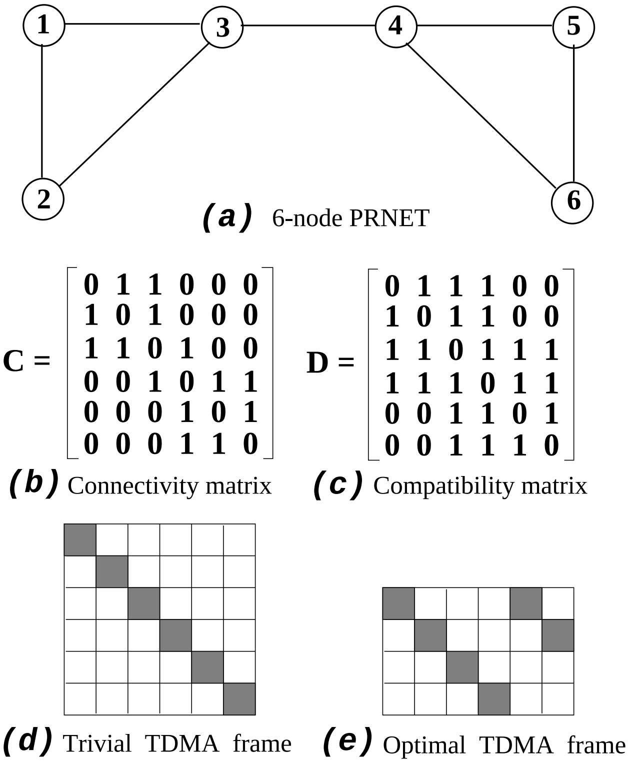
PR network can be represented by a graph $G = (V, E)$, where $V = \{v_1, v_2, ..v_i, ..v_N\}$ is the set of nodes and $E = \{e_1, e_2, ..e_L\}$ is the set of undirected edges. The existence of an edge between two nodes means that both can directly receive packets transmitted from the other. The neighboring information, i.e. the connectivity among network nodes are described by a $N \times N$ symmetric connectivity matrix $\mathbf{C}$, where the element

$$c_{ij} = \begin{cases} 1, & \text{if } v_i \text{ and } v_j \text{ are connected} \\ 0, & \text{otherwise} \end{cases}$$

When two nodes are directly connected, we say that they are one *hop* apart. We also assume slotted time and constant packet length i.e., the time duration to transmit or receive *one packet = one time-slot.* A TDMA frame consists of a fixed number of such time-slots. Packets can be transmitted at the same time-slot from different stations, if there is no interference. Once the optimum transmission pattern for the TDMA frame is decided, the same frame is repeated over time. Let us denote such a TDMA frame by a $M \times N$ matrix $\mathbf{T}$, where its element

$$t_{mj} = \begin{cases} 1, & \text{if } v_j \text{ transmits in time-slot } m \\ 0, & \text{otherwise} \end{cases}$$

When node v_i transmits a packet, none of its neighbors i.e., nodes which are one *hop* away, are allowed to transmit simultaneously, as this would give rise to *primary conflict.* All nodes two *hops* away from v_i should also be disabled to transmit simultaneously with v_i, as this would result in *secondary conflict* of multiple reception at intermediate nodes. All these nodes which are one *hop* and two *hops* away from v_i form, what is called the *broadcasting zone* [20] of v_i. The set of these nodes we denote by B_i. It is clear that non-interference requires that none of the nodes in B_i should be allowed to transmit simultaneously with v_i. From this concept we can generate a $N \times N$ compatibility matrix $\mathbf{D}$, where its element



(a) 6-node PRNET

$$\mathbf{C} = \begin{bmatrix} 0&1&1&0&0&0\\ 1&0&1&0&0&0\\ 1&1&0&1&0&0\\ 0&0&1&0&1&1\\ 0&0&0&1&0&1\\ 0&0&0&1&1&0 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 0&1&1&1&0&0\\ 1&0&1&1&0&0\\ 1&1&0&1&1&1\\ 1&1&1&0&1&1\\ 0&0&1&1&0&1\\ 0&0&1&1&1&0 \end{bmatrix}$$

(b) Connectivity matrix **(c)** Compatibility matrix

(d) Trivial TDMA frame **(e)** Optimal TDMA frame

Figure 1: Example of a (a) 6-node network, corresponding (b) **C** matrix and (c) **D** matrix, with (d) trivial TDMA schedule and (e) optimal TDMA schedule

$$d_{ij} = \begin{cases} 1, & \text{if } v_j \in B_i \\ 0, & \text{otherwise} \end{cases}$$

An example of a 6-node PRNET and the corresponding **C** matrix and **D** matrix are shown in Fig. 1(a), (b), and (c) respectively. Here, $B_1 = \{v_2, v_3, v_4\}$, $B_2 = \{v_1, v_3, v_4\}$, $B_3 = \{v_1, v_2, v_4, v_5, v_6\}$ etc..

The problem is to find shortest TDMA cycle, such that the following constraints are satisfied.

- Every stations should be scheduled to transmit at least once i.e. *no-transmission* constraint is satisfied.

$$\sum_{m=1}^{M} t_{mi} \geq 1 \qquad \forall i \tag{1}$$

- A station can not transmit and receive packets in the same time-slot to avoid *primary conflicts.*
- A station can not receive two or more transmissions simultaneously i.e., *secondary conflicts* are to be avoided. Formally,

$$if \;\; t_{mi} = 1, \; then \;\; \sum_{j=1}^{N} t_{mj} d_{ij} = 0 \;\; \forall m, i$$

i.e.,

$$\sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{j=1}^{N} t_{mi} t_{mj} d_{ij} = 0 \tag{2}$$

The last two conflicts are avoided if the TDMA frame $\mathbf{T}$ is scheduled satisfying Eq. (2). A trivial solution satisfying all the three constraints is a N-slot TDMA frame, where N different stations transmit in N different time-slots, as shown in Fig. 1(d).

The primary optimization criteria is to minimize the length of TDMA cycle, i.e., M should be as small as possible. Sometimes a secondary optimization objective, to maximize the total number of transmissions, is considered. This is represented by channel utilization index ρ, where

$$\rho = \frac{1}{M \times N} \sum_{m=1}^{M} \sum_{j=1}^{N} t_{mj} \qquad (3)$$

is to be maximized. Our algorithm optimizes the first criterion, and leaves room for improving transmissions for a subset of nodes. The details are in Section 3.2.

It is trivial that if the maximum degree of a node in the net is G, then the tight lower bound for M will be,

$$M \geq (G+1) \qquad (4)$$

As there is no algorithm to find the optimum solution, we will use this tight lower bound $(G+1)$ as an empirical measure to judge the quality of solution. When $M = G+1$, we know that the solution is optimal.

For the 6-node PRNET, the trivial schedule, shown in Fig. 1(d), satisfies all the constraints but does not optimize any of the optimization criteria. Fig. 1(e) is an optimal solution.

2.1 Previous works

When the optimization criterion is only minimizing the length of the TDMA frame, and only *primary conflicts* are considered, the scheduling problem translates to simple graph-coloring problem. To include secondary conflicts, one need to consider the compatibility matrix $\mathbf{D}$ as the connectivity matrix, instead of $\mathbf{C}$, in the graph coloring problem. As the graph coloring problem is NP-complete, so is this TDMA scheduling. Formal proof of NP-completeness is available in [19, 20].

During the last two decades, several algorithms were proposed to solve this problem. Instead of broadcast scheduling, some considered a similar problem of optimal schedule of activating different links [9, 11]. For broadcast scheduling, the different algorithms could be classified depending on their objectives and approaches. Most of the earlier works were either centralized [8, 13, 21, 22] or distributed [20, 23] heuristic algorithms. Their optimization objective was to maximize transmission [20]. Those algorithms started with the trivial initial schedule, as shown in Fig. 1(d), and added transmissions to it to the maximum possibility without violating constraints. The length of the TDMA frame remained same, equal to the number of nodes, and is quite long. While ρ was improved, neither the traffic demands from the different nodes nor the fairness, as discussed in the introduction, were considered.

In recent years, random algorithms using neural network [11, 17, 18, 24] and simulated annealing [19] were proposed. In these works, the main optimization objective was to minimize the length of the TDMA frame. These algorithms too can not reduce M from the initial setting. As there is no clue of what would be the optimum length, they usually started with $M = G+1$, the tight lower bound in Eq. 4. When no solution could be found, M is increased in steps of 1. Every time the algorithm has to from the beginning trying to find a valid solution. Depending on the problem complexity, many trials may be necessary. As simulated annealing and artificial neural network, implemented in conventional computers, are computationally heavy, these approaches could be quite slow.

3 The Algorithm

In our approach, a pool of valid random solutions are created, assigning single transmission per node. While creating different solutions no heuristic is used to optimize the solutions, and therefore it is very fast. Finally, from the pool, the solution with lowest TDMA length is selected. For all the problems tried, it could reach known minimum TDMA length or less. The algorithm is simple and fast, $O(n^2)$ where n is the number of network nodes. Simulating various problems for a large number of times, we show that the probability of getting the optimum solution is nearly 1. The format of the solution allows further addition of transmission for certain nodes, keeping the TDMA length optimal. It is also easy to extend the algorithm for situations, where traffic is non-uniform at different nodes. The basic algorithm is described in Section 3.1. The variations and extensions are discussed in Section 3.2. In section 4, several experiments, and statistical analysis of the results are discussed.

3.1 Description of the Algorithm

We give the informal description of the algorithm with an example. The pseudocode and analysis of the complexity of the algorithm is given in the appendix.

For a N-node PRNET, we start with a TDMA frame $\mathbf{T}_{N\times N}$. Let us represent the elements of the TDMA frame $\mathbf{T}$ as t_{mj}. The algorithm creates a pool, P number of such frames. Let us name them $\mathbf{T}^1, \mathbf{T}^2, \ldots \mathbf{T}^p, \ldots \mathbf{T}^P$. All the elements of the frames are initialized with 0s. After the algorithm is executed, the elements of $\mathbf{T}$ could either be a "0", a "+1", a "-1" or a "9", where "0" stands for *assignable*, "+1" for *assigned*, "-1" for *unassignable* and "9" for *unused*. A "0" at t_{mj} indicates that the j^{th} node, represented by the column number of the TDMA frame, is not using the m^{th} time-slot for transmission, represented by the row number. In addition, it also means that if a transmission is assigned to the j^{th} node in the m^{th} time-slot, there will be no conflict with other existing transmission assignments. A "+1" at t_{mj} indicates that the j^{th} node is using the m^{th} time-slot for transmission. A "-1" at t_{mj} indicates that the j^{th} node is not using the m^{th} time-slot for transmission. Moreover, it is unassignable, because then it would cause interference with some other existing allocations, according to the compatibility matrix $\mathbf{D}$. Indicator "9" at t_{i1}, i.e. at the first column, indicates that the corresponding time-slot is not allotted to any cell. They appear at the end of the schedule, marking the tail part of $\mathbf{T}_{N\times N}$ for those time-slots that remain unused.

The 6-node network of Fig. 1(a) is used to explain the algorithm. Considering uniform traffic at different nodes, the task is to assign a single transmission for each node in a TDMA frame. After initialization, all the entries in $\mathbf{T}$ matrices are 0 s.

Now let us convert one of the initial TDMA frame, say $\mathbf{T}^1$, to a valid solution of the problem. First an arbitrary permutation of the digits $1,\ 2,\ 3, \ldots N$ is done. We get an arrangement whose elements in order are $\sigma_1, \sigma_2, \ldots \sigma_N$, denoting the nodes in a particular sequence. Our example is a 6-node network. Let the result of random permutation be 3, 4, 1, 5, 2, 6. So, $\sigma_1 = 3$, $\sigma_2 = 4$, $\sigma_3 = 1$, $\sigma_4 = 5$, $\sigma_5 = 2$, $\sigma_6 = 6$.

Allotment of transmissions at different time-slots will start with node σ_1, then σ_2 and so on until σ_N. For this particular permutation, allotment will start with node 3, i.e., column 3 of the $\mathbf{T}^1$ matrix. Transmission at time-slot t_1 is assigned to node 3, and t_{13} is changed $0 \rightarrow 1$. Now the algorithm checks which are the nodes, where if a transmission is allowed at time-slot t_1, would result in interference according to the compatibility matrix $\mathbf{D}$. All columns of row 3 in $\mathbf{D}$, where the entries are 1, are conflicting nodes. In this particular example, they are nodes 1, 2, 4, 5, 6. Those entries for time-slot t_1 are changed from 0 to -1. Thus entries for time-slot 1, row 1 of $\mathbf{T}^1$, are changed from the initial setting of [0 0 0 0 0 0] to [-1 -1 +1 -1 -1 -1].

In the permutation, the next σ_2 is node 4. We start from top row of $\mathbf{T}^1$ along column 4 to find a "0", i.e. a time-slot which could be allotted to node 4 for transmission. As t_{24} is a "0", transmission for node 4 can be assigned at time-slot 2 without any constraint violation. Following same rule as in the case of node σ_1 (i.e., node 3), "-1"s are packed at relevant locations of $\mathbf{T}^1$, i.e., entries "1" in row 4 of the compatibility matrix $\mathbf{D}$. This will put "-1"s at $t_{21}, t_{22}, t_{23}, t_{25}, t_{26}$. This procedure is repeated until the last node i.e., node σ_N (here node 6) gets its transmission time-slot.

Finally, solution $\mathbf{T}^1$, as shown in Table. 1, is created by the algorithm, when the random permutation is the sequence of nodes:

$$3,\ 4,\ 1,\ 5,\ 2,\ 6$$

The maximum degree G is 3 for nodes 3 and 4. This schedule of length 4 is therefore optimum. As only four time-slots were sufficient, t_{51} and t_{61} are changed to "9", indicating that those time-slots are unused.

Some other permutations would also lead to the same solution. Those permutations are:

$$3,\ 4,\ 5,\ 1,\ 2,\ 6; \quad 3,\ 4,\ 5,\ 1,\ 6,\ 2;$$

With different permutations the schedule will be different, though for this small problem, with all possible permutations, the length of the TDMA cycle will always be 4, and therefore optimal.

	node 1	node 2	node 3	node 4	node 5	node 6
t_1	-1	-1	+1	-1	-1	-1
t_2	-1	-1	-1	+1	-1	-1
t_3	+1	-1	-1	-1	+1	-1
t_4	-1	+1	-1	-1	-1	+1
t_5	9	0	0	0	0	0
t_4	9	0	0	0	0	0

Table 1: An example of a valid TDMA frame $\mathbf{T}^1$

In the TDMA schedule of length 4 time-slots (row 1 to row 4 in Table. 1), all "0"s of the initial setting are either changed to "+1" or "-1". If there remains a "0" unchanged in the TDMA schedule, it means a transmission could be allocated at that location without any constraint violation. Thus, transmission could be added at those locations.

With complex problems, due to increased size and degree of connectivity of the nodes, different permutations would generate TDMA schedules of different lengths. In all experiments we found that, out of the pool of solutions generated, those with minimum TDMA length are optimum. Moreover, even with the minimum length TDMA schedule, there could be some "0" entries left unchanged in the TDMA schedule where transmissions could be added. For larger problems this is true as shown in Fig. 3 to Fig. 6, and is further discussed in section 3.2.

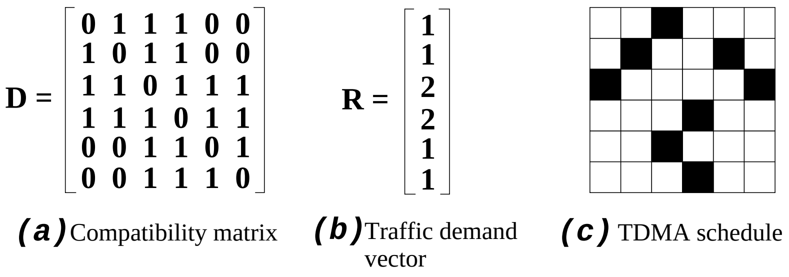
3.2 Extensions of the algorithm

We will explain simple yet important possible (1) extension of the algorithm to improve transmission, and (2) modification of the algorithm when the traffic demand for different nodes are different.

We have seen in the previous section that if an element t_{mj} of the TDMA frame is "0", a transmission for node "j" at time-slot "m" would cause no interference. Such a transmission could be easily added to node "j" in the TDMA frame as and when necessary. This added transmission will give rise to other interferences, and the TDMA schedule is to be updated by adding "-1" at relevant positions. For details see ADD_TRANSMISSION(ν), lines 21 to 26, in the pseudocode.

Non-uniform traffic demand at different nodes could be expressed by a traffic demand vector $\mathbf{R} = \langle r_1, r_2, \ldots, r_N \rangle$, where the traffic requirement at i^{th} node is expressed by r_i. We solve this nonuniform traffic demand problem in the same way as is done for uniform traffic in section 3.1. Only, in this case, i^{th} cell will appear r_i times in the permutation. Thus, instead of $N!$ the total number of possible permutations will be

$$\frac{(\sum_{i=1}^{N} r_i)!}{\prod_{i=1}^{N} r_i!}$$



$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

(a) Compatibility matrix

$$\mathbf{R} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 1 \\ 1 \end{bmatrix}$$

(b) Traffic demand vector

		■			
	■			■	
■					■
			■		
		■			
			■		

(c) TDMA schedule

Figure 2: The 6-node network (a) compatibility matrix, (b) traffic demand vector, and (c) TDMA schedule

To explain this modification of the algorithm, we use the same 6-node network of Fig. 1(a). The traffic demands for different nodes is described by **R**, as shown in Fig. 2(b). Here, as node-3 and node-4 have traffic demand of 2 each, we start with permutations of [1 2 3 3 4 4 5 6]. Let one such permutation be [3 2 5 1 6 4 3 4]. We first allot for node-3 at time-slot 1. From **D**, node-3 conflicts with all other nodes. So we can not schedule the next node in the permutation, i.e., node-2 in time-slot 1. We assign time-slot 2 for node-2. Here the conflicting nodes are node-1, node-3, and node-4. So, t_{21}, t_{23}, t_{24} are marked with "-1". The next node in the permutation is node-5, for which time-slot 2 could be assigned without conflict. Node-5 conflicts with node-6. So t_{26} is made "-1". Progressing in this way the final schedule will be as shown in Fig. 2(c).

For different permutations, different schedules are obtained, and the shortest of them is selected as the solution.

4 Experiments: Set up, Results and Analysis

To test the performance of the algorithm we did exhaustive simulations with problems of various sizes and varied degrees of connection. For most of the problems, we could get optimum solutions with the lowest bound $(G + 1)$. In all the cases, our solutions are equally good or better than those obtained by other approaches. The algorithm is fast, as we can see from the execution time included in Table. 2 later in section 4.2.

In section 4.1, we describe how the different networks were created for simulation. In section 4.2, we give the simulation results for matrix networks, which are planar graphs. In section 4.2.1, statistical analysis of the results is done, and the probability of finding optimum solution, depending on the size of the solution pool, is calculated. In section 4.3, we give simulation results for networks with random connections.

4.1 Problem Set up

As there is no standard benchmark problem set, we will perform simulations on networks similar to what were used in recently reported publications [18, 19, 20] and [23]. In practice, for PRNET, the degree of connectivity of nodes are usually low. Here, we will set up two types of networks:

Type I. Networks which are described by planar graphs. Here, only neighboring nodes are directly connected. Nodes in large networks are placed in the form of matrices, as was done in [18, 19].

Type II. Networks that have random interconnections between nodes. Here, the probability of connection is decreased as the distance between the nodes increases.

To create large problems of type I, a matrix of dots in X-Y plane, as shown in Fig. 3 to Fig. 6, are arranged. Now, a node is randomly connected to a subset of its neighboring eight nodes (for nodes on the boundary it is less than 8). It is ensured that finally it is a connected network, so that any node can communicate with any other node. The average degree of connection is set to different values.

For random networks, a connected network of N nodes and average degree of connection d, are created [25]. First a square area of $\sqrt{N} \times \sqrt{N}$ is considered, and N points are put on it randomly. Here too, all the nodes are connected, and the degree of each node is at least 2. The average node degree could be set to different values. The probability of connection between two nodes decreases exponentially with the distance, and is given by the following equation.

$$Prob = exp\left(-\frac{distance\ \ between\ \ nodes}{L_{max} \times \alpha}\right)$$

$L_{max} = \sqrt{2\,N}$ is the length of the diagonal of the square area. When the value of the constant α is high, even nodes at longer distances are connected with high probability. We used $\alpha = 2$. Links are added until the number reaches the value $N \times d/2$, i.e., the average degree of connection per node becomes d.

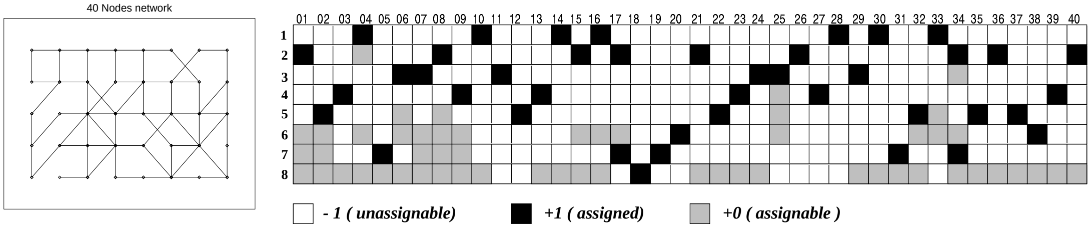


Figure 3: 40-node network problem #3 and its solution

4.2 Simulation and results with Type I networks

Results from Type I networks are summarized in Table. 2. In the column "Problem specifications", we mention the number of nodes and their arrangement, the total number of links, the average and highest degree (G) of nodes in the network. The lowest bound of the optimum TDMA frame length is ($G+1$), which is achieved for most of the problems, as shown in fifth column of Table. 2. As the algorithm is random, its behavior is probabilistic. We repeated all the experiments in Table. 2 for 1000 times. Every time a pool of 1000 TDMA frames are created. Number of solutions with minimum TDMA frame length were different in different pools. Their average is reported in column 6 of Table. 2. Computation time in a PentiumIV machine, for generating a pool of 1000 solutions, are shown in the last column.

Problem #1 was used in [20] and [18]. In [20] the optimization criterion was not the TDMA cycle length. The initial TDMA frame has the trivial length of 14, equal to the number of nodes. The number of transmissions was improved. In [18], the same problem was tried, where every nodes transmit only once in a TDMA frame and the optimization criterion was minimizing the frame length. The optimum TDMA frame length of 6 is achieved by our algorithm. For problem #2, first proposed in [23] and used in [18] too, the minimum TDMA frame length of 5 could be achieved by our algorithm very fast. Problem #3 was experimented in [19], where a minimum TDMA cycle length of 9 could be achieved. Our algorithm could find a TDMA frame of length 8, which is optimal. Three 100-nodes networks, problems #4 to #6, with different degrees of connections, were used for experiments. The problem becomes more complex, as the degree of interconnections increase. This is evident in Table. 2 from the optimum TDMA frame length, and the average number of best solutions in 1000 frames. With large problems, though the solutions does not achieve the lowest bound ($G + 1$), it differs only by unity. Compared to [18], the results obtained are better. Problem #7(100-nodes, 189-links), #8(100-nodes, 282-links), #10(196-nodes, 370-links), and #13(400-nodes, 805-links) of [18] are similar to our problems #4(100-nodes, 200-links), #6(100-nodes, 300-links), #7(200-nodes, 400-links), and #9(400-nodes, 800-links) respectively. The first three problems are slightly more complex in this set up, with more connections. For problems #8, which is similar to problem #10 of [18], our TDMA schedule is better by one time slot. For problems #3 to #6, the network as well as the solutions are shown in Fig. 3 to Fig. 6. Here we can see that lots of assignable slots are available. Thus, further transmissions could be easily added to specific nodes if necessary.

4.3 Statistical analysis of the results

We have shown in column 6 of Table. 2 that only a percentage of solutions in the pool could *hit* the target of minimum length TDMA. Here we report on the frequency distribution of the solutions of different lengths for different problems. The results for different 100-nodes networks are shown in Fig. 7. The results are average of 1000 trials. It is evident that, as the degree of connection increases, the length of the minimum TDMA frame increases, and the percentage of the best solutions in the pool decreases.

These frequency distribution of solutions for all problems are summarized in Table. 3. We call it a *hit* when the solution is optimum length TDMA. The probability of at least one *hit* in a population of 1000 solutions is included in column 7. The method of calculation of this probability is discussed in the next paragraph. Also included, in column 8, is the number of solutions needed for having at least one *hit* with a probability of 99%.

Suppose, out of τ number of trial solutions, the total number of *hit*s is ϕ. When τ is sufficiently large,

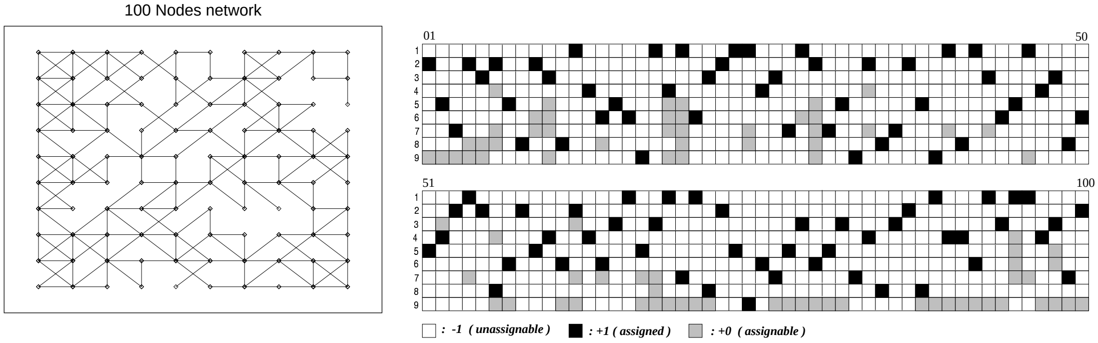


Figure 4: 100-node network problem #4 and its solution

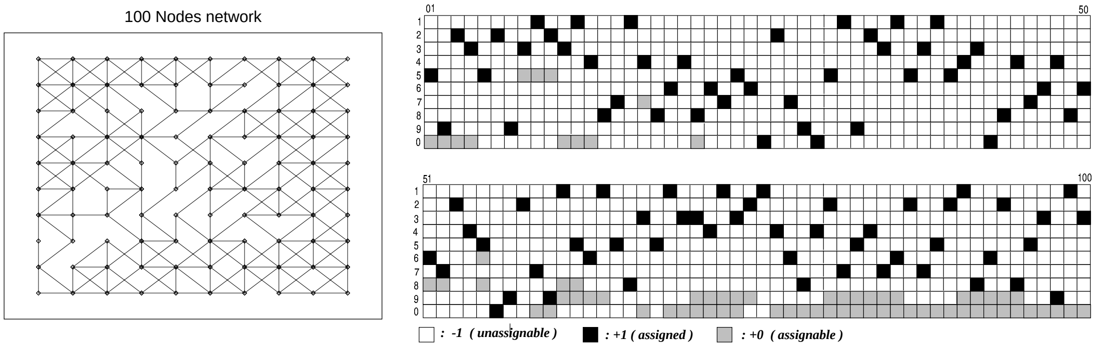


Figure 5: 100-node network problem #5 and its solution

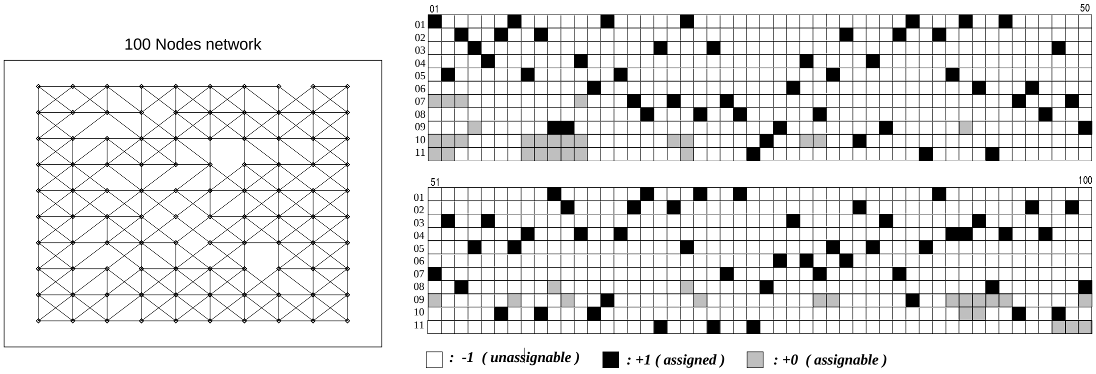


Figure 6: 100-node network problem #6 and its solution

Problem number	Problem specifications			minimum TDMA frame length	Av. no. of best sols. in pool of 1000	Computation time for 1000 TDMA frames
	No. of nodes row × col.	*No. of links*	*av. Degree and (G+1)*			
1	14	23	3.3, 6	6	103.3	0.005 Sec.
2	16	22	2.75, 5	5	64.3	0.006 Sec.
3	40 (5 × 8)	66	3.3, 8	8	153.1	0.032 Sec.
4	100 (10 × 10)	200	4.0, 9	9	15.591	0.118 Sec.
5	100 (10 × 10)	250	5.0, 9	10	1.507	0.124 Sec.
6	100 (10 × 10)	300	6.0, 9	11	0.06	0.129 Sec.
7	200 (10 × 20)	400	4.0, 9	9	0.324	0.316 Sec.
8	300 (15 × 20)	600	4.0, 9	10	73.250	0.712 Sec.
9	400 (20 × 20)	800	4.0, 9	10	5.67	1.172 Sec.

Table 2: Simulation Results with Type I networks

we can say that in a single trial the probability of *hit* is ϕ/τ. The probability of *no-hit*, i.e. solution longer than optimum TDMA frame, in a trial is $(1 - \phi/\tau)$. Thus, the probability of *no-hit* in μ trials is $(1 - \phi/\tau)^{\mu}$, whence the probability of at least one hit in μ trials is $(1 - (1 - \phi/\tau)^{\mu})$. Therefore the probability of at least one hit in 1000 trials is $(1 - (1 - \phi/\tau)^{1000})$. To calculate the required number of trials to have at least one *hit* with 99% probability, the probability of at least one *hit* in μ trials, i.e. $(1 - (1 - \phi/\tau)^{\mu})$ is set to 0.99, and the corresponding μ is calculated.

Problem number	Problem Spec.			shortest TDMA	Freq. of *hit*s in 1000 **T**s	Prob. of *hit* in 1000 **T**s	No. of **T**s required for 99% *hit* probability
	no. of nodes	no. of links	$(G+1)$				
1	14	23	6	6	103.3	≈100%	42
2	16	22	5	5	64.3	≈100%	69
3	40	66	8	8	153.1	≈100%	28
4	100	200	9	9	15.59	≈100%	293
5	100	250	9	10	1.507	77.9%	3053
6	100	300	9	11	0.06	05.8%	7670
7	200	400	9	9	.324	27.7%	1420
8	300	600	9	10	73.25	≈100%	60
9	400	800	9	10	5.67	99.7%	810

Table 3: Simulation Results with *hit* probabilities

All the experiments were done several times. Assuming normal distribution of the results, the 99% *confidence interval* [26] (chap. 17) of the number of *hit*s was estimated for the different experiments. The number of optimum length TDMA frames were noted for 50 trials, each time creating a pool of one million solutions. From the average (μ) and variance (σ) of the results from 50 trials, the 99% confidence interval was calculated. For the largest 400-node problem #9, $\mu = 5668.68$ and $\sigma = 224.6$. Thus 99% confidence interval was only 0.014 fraction of the average value. For the rest of the problems, this 99% confidence intervals were even lower fraction of the average. In other words, statistically the results in Table. 3 are highly consistent and reliable.

4.4 Simulation results with Type II networks

Here two non-planar networks of 30 and 100 nodes were experimented with. The 30-nodes network problem is same as used in [19]. Maximum degree was 8. The algorithm could find a TDMA frame of length 10, better than 12 as reported in [19]. The 100 nodes network had a total of 492 links, and highest degree (G) was 10. Optimum TDMA frame of length 11 could be found. The frequency distribution of solutions of different TDMA lengths are shown in Fig. 8.

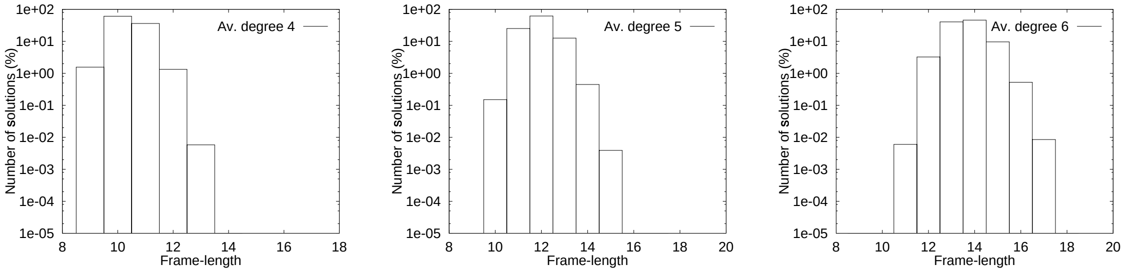


Figure 7: Distribution of solutions of different TDMA length for 100-node network problems #4, #5 and #6

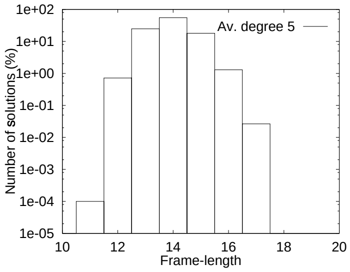


Figure 8: Distribution of solutions of different TDMA length for 100-node random network

4.5 Comparison with recent works

In [19] mean field annealing method was used, for networks with number of nodes up to 40. For the same 40-node network problem, we could improve the TDMA frame length by one time slot. Large networks in our experiments were similar in structure and connectivity, as were used in [18]. Problem #7(100-nodes, 189-links), #8(100-nodes, 282-links), #10(196-nodes, 370-links), and #13(400-nodes, 805-links) of [18] are similar to our problems #4(100-nodes, 200-links), #6(100-nodes, 300-links), #7(200-nodes, 400-links), and #9(400-nodes, 800-links) respectively. In fact, the first three problems are slightly more complex in our set up, as the number of connections are more. For problems #8 and #10 of [18], our TDMA schedule is better by one time slot. Comparison of results on similar networks is summarized in Table. 4.

Our solution			solutions reported in [18],[19]		
N	links, G+1	M	N	links, G+1	M
40	66, 8	8	40	66, 8	9
100	200, 9	9	100	198, 9	9
100	300, 9	11	100	282, 9	12
200	400, 9	9	196	370, 9	10
400	800, 9	10	400	805, 9	10

Table 4: Comparison of results obtained by the proposed algorithm and other competitive algorithms. We put problems of comparable size and complexity in the same row. The first row is comparison with work reported in [19]. The lower 4-rows are comparison with work reported in [18]

5 Discussion and Conclusion

We have proposed a very fast algorithm to solve the problem of optimum transmission schedule in broadcast packet radio networks. The idea is to generate a number of valid solutions of the problem, during which the optimization criterion of minimizing the TDMA frame length is not given any attention. Finally if the best solution is selected, we have shown that, with a very high probability the optimum or very near to optimum solution is obtained. From Table. 2, it is clear that for all the problems experimented in earlier researches, we could find optimum solutions even when a small pool of solutions is created. The probability is much more when near optimal results are acceptable too, as is evident from frequency distribution plots in Fig. 7 and Fig. 8.

With the quadnary representation [0, +1, -1, +9] of time-slot assignments, the format of the TDMA solution frame is such that, it is simple to extend its use for dynamic allocation of additional transmission. In the solution frame a 0 at m^{th}- row and j^{th}- column indicates that a transmission could be allocated to the j^{th} node at the m^{th} time-slot without causing any conflict to other existing assignments. Thus, it is easy to add transmissions for some specific nodes in the TDMA schedule as and when required (line 21 to 26 of the pseudocode in the appendix). In section 3.2, it is also shown that, with slight modification of the algorithm one could handle situation when the traffic demand at different nodes are different.

References

[1] F. Tobagi, R. Binder, and B. M. Leiner, "Packet radio and satellite networks," *IEEE Comm. Mag.*, November 1984.

[2] J. Jubin, "Current packet radio protocols," in *INFOCO85 Proceedings*, March 1985.

[3] B. M. Leiner, D. L. Nielson, and F. A. Tobagi, "Issues in packet radio network design," *Proceedings IEEE, vol. 75, no. 1*, pp. 6-20, January 1987.

[4] L. Kleinrock and J. Silvester, "On the behavior of packet radio networks,"*Proceedings IEEE, vol. 75, No. 1*, pp. 156-167, January 1987.

[5] F. Tobagi, "Modeling and performance analysis of multihop packet radio networks,"*Proceedings IEEE, vol. 75, No. 1*, pp. 135-155, January 1987.

[6] N. Shacham and J. Westcott, "Future directions in packet radio architectures and protocols,"*Proceedings IEEE, vol. 75, No. 1*, pp. 83-99, January 1987.

[7] M. B. Pursley, "The role of spread spectrum in packet radio networks," *Proceedings IEEE, vol. 75, No. 1*, pp. 116-134, January 1987.

[8] F. A. Tobagi, "Multi-access protocols in packet communication systems," *IEEE Trans. Comm., vol. COM-28*, pp. 46 488, April 1980.

[9] I. Chlamtac, "Fair algorithm for maximal link activation in multihop radio networks," *IEEE Trans. Comm., vol. COM-35, no. 7*, pp. 739-746, July 1987.

[10] N. Funabiki, and S. Nishikawa, "A binary neural network approach for link activation problems in multihop radio networks," *IEICE Trans. on comm, vol. E-79B, no. 8*, pp. 1086-1093, August 1996.

[11] C. M. Barnhart, J. E. Wieselthier, and A. Ephremides, "A neural network approach to solving the link activation problem in multihop radio networks," *IEEE Trans. Comm., vol. COM-43, no.2/3/4*, pp. 1277-1283, Feb/Mar/April 1995.

[12] R. Kahn, S. Gronemeyer, J. Burchfiel, and R. Kunzeman, "Advances in packet radio technology," *Proc. IEEE, vol. 66, no. 11*, pp. 1468-1496, Nov. 1978.

[13] R. Nelson and L. Kleinrock, "Spatial TDMA: a collision-free multihop channel access protocol," *IEEE Trans. Comm, vol. COM-33*, pp. 934-944, September 1985.

[14] B. H. Davies and T. R. Davies, "The application of packet switching techniques to combat net radio," *Proceedings IEEE vol. 75, No. 1*, pp. 43-55, January 1987.

[15] T-C. Hou and V. O. K. Li, "Transmission range control in multihop packet radio networks," *IEEE Trans. Comm., vol. COM-34, no. 1*, pp. 38-44, Jan. 1986.

[16] H. Takagi and L. Kleinrock "Optimal transmission ranges for randomly distributed packet radio networks," *IEEE Trans. Comm., vol. COM-32, no. 3*, pp. 246-257, March 1984.

[17] G. Wang and N. Ansari, "A neural network approach to broadcast scheduling in multihop radio networks," in *Prod. ICNN'94*, pp. 4699-4704, June 1994.

[18] N. Funabiki, and Y. Takefuji, "A parallel algorithm for broadcast scheduling problems in packet radio networks," *IEEE Trans. on comm, vol. 41, no. 6*, pp. 828-831, June 1993.

[19] G. Wang, and N. Ansari, "Optimal broadcast scheduling in packet radio networks using mean field annealing," *IEEE journal on selected areas in comm, vol. 15, no. 2*, pp. 250-260, February 1997.

[20] A. Ephremides, and T. V. Truong, "Scheduling broadcasts in multihop radio networks," *IEEE Trans. Comm, vol. COM-38, no. 4*, pp. 456-460, April 1990.

[21] J. A. Silvester, "Perfect scheduling in multi-hop broadcast network," in *Proc. 6th Int. Conf. on Computer Communication*, pp. 449-454, London, England, 1982.

[22] W. C. Fifer and F. J. Bruno, "The low-cost packet radio," *Proceedings IEEE, vol. 75, No. 1*, pp. 33-42, January 1987.

[23] I. Cidon and M. Sidi, "Distributed assignment algorithms for multihop packet radio networks," *IEEE Trans. Computers, vol. 38, no. 10*, pp. 1353-1361, October 1989.

[24] J. E. Wieselthier, C. M. Barnhart, and A. Ephremides, "A neural network approach to routing without interference in multihop radio networks," *IEEE Trans. Comm., vol. COM-42, no.1*, pp. 166-177, January 1994.

[25] B. M. Waxman, "Routing of multipoint connection," *IEEE journal on selected areas in comm, vol. 6, no. 9*, pp. 1617-1622, 1988.

[26] G. M. Clarke and D. Cooke, *A Basic Course in Statistics.* Edward Arnold. 3rd. edition. 1992.

APPENDIX

The pseudocode of the algorithm to generate random valid solutions:

Input: N: Number of cells.

$\mathbf{D}[N][N]$: Compatibility matrix.

P: Population size, the number of solutions to be created.

Output: A set of valid solutions of the TDMA schedule. $\mathbf{T}^1, \mathbf{T}^2, \ldots, \mathbf{T}^P$ are the P number of solutions of the broadcast scheduling problem. Each $\mathbf{T}^i$ is a $M \times N$ matrix, where N is the number of nodes and M is the number of time-slots.

```
CREATE_POP(D)      /* Create P number of valid TDMA schedules */
01  for $p \leftarrow 1$ to $P$
02       $\sigma[N] \leftarrow$ SHUFFLE-LIST($N$)
03       do for each node $\nu = \sigma[n]$, $n = 1$ to N
04           for $m = 1$ to M
05               if $\mathbf{T}^p[m][\nu] == 0$
06                   $\mathbf{T}^p[m][\nu] \leftarrow +1$
07                   PUT_MINUS1($\mathbf{T}^p[m][\nu]$)
08                       break      /* the for loop beginning at line 04 */
09               endif
10           endfor
11       enddo
12       MARK_UNUSED_FREQ($F^p$)
13   endfor

SHUFFLE-LIST($N$)
14  Return a random permutation of numbers 1 to N

PUT_MINUS1($\mathbf{T}^p[m][\nu]$)      /* Put "-1" at relevant locations of $\mathbf{T}^p$ */
15  for each node $j = 1$ to N
16       if  $\mathbf{D}[\nu][j] == 1$
17           $\mathbf{T}[m][j] \leftarrow -1$
18       endif
19   endfor

MARK_UNUSED_TIME-SLOTS ($\mathbf{T}^p$)
20  Put '9' at column 1 of unused time-slots of $\mathbf{T}^p$

ADD_TRANSMISSION ($\nu$)      /* Allocate extra transmission to node $\nu$ */
21  for each time-slot $m = 1$ to M
22       if  $\mathbf{T}[m][\nu] == 0$
23           $\mathbf{T}[m][\nu] \leftarrow +1$
24                   PUT_MINUS1($\mathbf{T}[m][\nu]$)
25       endif
26  endfor
```

The pseudocode of the algorithm to generate random valid solutions, when traffic at different nodes are different:

Input: N: Number of cells.
$\mathbf{D}[N][N]$: Compatibility matrix.
$\mathbf{R}[N]$: Demand vector.
P: Population size, the number of solutions to be created.

Output: A set of valid solutions of the transmission schedule, $\mathbf{T}^1$, $\mathbf{T}^2, \ldots, \mathbf{T}^P$.

```
CREATE_POP_NONUNIFORM(D, R)  /* Create TDMA frames for non-uniform transmission requirement */
27  for $p \leftarrow 1$ to $P$
28      $\sigma[N] \leftarrow$ SHUFFLE-LIST($N$)
29      do for each node $\nu = \sigma[n]$, $n = 1$ to N
30          $demand = \mathrm{R}[\nu]$
31          for $m = 1$ to M
32              if $\mathbf{T}^p[m][\nu] == 0$
33                  $\mathbf{T}^p[m][\nu] \leftarrow +1$
34                  $demand \leftarrow (demand - 1)$
35                  PUT_MINUS1($\mathbf{T}^p[m][\nu]$)
36                  if ($demand == 0$)
37                      break     /* the for loop beginning at line 31 */
38                  endif
39              endif
40          endfor
41      enddo
42      MARK_UNUSED_FREQ($\mathbf{T}^p$)
43  endfor
```

Complexity Analysis:

We here examine the complexity of the algorithm for creating a valid solution. In the above pseudocode, a loop on lines from 03 to 11 iterates for N number of times. Within this loop, for a particular node i, once a time-slot is assigned for transmission, i.e., a "0" in $\mathbf{T}^p$ is turned to a "+1", PUT_MINUS1 function is called and executed. PUT_MINUS1 function has a loop on lines 15 to 19 with iterations for all nodes i.e., N times. Thus for assigning a time-slot for all the nodes, the complexity of the algorithm is $O(N^2)$, where N is the number of nodes. The execution time, as shown in Table. 2, conform to this conclusion.

The actual time spent in the loop, between lines 15 to 19 in PUT_MINUS1, depends on the number of 1's in the compatibility matrix, i.e., on the degree of connectivity of the network nodes. For same degree of connection, the time of execution of the algorithm is very close to N^2, as can be verified from the computation time of problems in rows 7, 8, and 9 in Table. 2.