Optimal Broadcast Scheduling in Packet Radio Networks Using Mean Field Annealing

Gangsheng Wang and Nirwan Ansari, *Senior Member, IEEE*

***Abstract*— Packet radio (PR) is a technology that applies the packet switching technique to the broadcast radio environment. In a PR network, a single high-speed wideband channel is shared by all PR stations. When a time-division multi-access protocol is used, the access to the channel by stations' transmissions must be properly scheduled in both time and space domains in order to avoid collisions or interferences. It is proven in this paper that such a scheduling problem is NP-complete. Therefore, an efficient polynomial algorithm rarely exists, and a mean field annealing-based algorithm is proposed to schedule the stations' transmissions in a frame consisting of certain number of time slots. Numerical examples and comparisons with some existing scheduling algorithms have shown that the proposed scheme can find near-optimal solutions with reasonable computational complexity. Both time delay and channel utilization are calculated based on the found schedules.**

I. Introduction

PACKET RADIO (PR) [1], which can handle bursty-type traffic efficiently, is an option for high-speed wireless data communications, especially over a broad geographic region. In a PR network, each station is equipped with a transmitter/receiver and a control unit. Stations communicate with each other via a shared high-speed broadcast radio channel. The control unit performs the packet switching functions. When a station broadcasts through its antenna, each neighboring station receives the transmission. The neighboring station will absorb the packets to which the transmission designates. Otherwise, the station will store the packets in its buffer and send them out later. Therefore, for any two distant stations where direct connectivity does not exist, the intermediate stations act as repeaters and perform store-and-forward functions. Fig. 1 shows a packet radio network in the Internet. The communication between geographically separated hosts is established through the packet radio network.

In a multihop PR network, since a single channel (usually wideband) is shared by all users, the transmission for each station must be scheduled to avoid any collision or interference. Based on the characteristics of a multihop network, the single radio channel can be shared by all stations in both time and space domains. A multiaccess protocol, namely spatial time-division multiple-access (TDMA), can be used to schedule conflict-free transmission [2]. In the spatial TDMA network, time is divided into frames which consists of fixed-length time slots. When certain stations transmit simultaneously, collision or interference will occur. Therefore, any two stations that may result in collision or interference must be scheduled to transmit at different time slots, while stations some distance away may be arranged to transmit at the same time slot without causing interference. Since the primary objective of the PR network is to provide high throughput with low delay, a scheme must provide a schedule which can achieve maximum channel utilization as well as lower delay. For a fixed-topology PR network in which locations of stations are fixed, the problem is to schedule a frame in which each station transmits at least once. Additional transmissions can be added into the frame if the addition does not cause any collision. The optimal schedule is the one that has the minimum length (the number of time slots in a frame) and provides the maximum number of interference-free transmissions. Such a scheduling problem is proved NP-complete in the Appendix, implying that a good algorithm rarely exists in finding the global optima among a large number of local optima. In this paper, an approximation algorithm based on mean field annealing (MFA) is presented to solve the scheduling problem. In Section II, the scheduling problem is stated. The MFA theory is reviewed in Section III. In Section IV, MFA is applied to solving the optimal scheduling problem. Numerical examples are given, and the resulting performance is evaluated in Section V. Conclusions are made in the last section.

II. Problem Formulation

A PR network can be represented by a graph $G = (V, E)$ where the vertices in V are network stations, and E is a set of edges. The total traffic passing through station i consists of packets received from other stations which will be routed through station i and the packets from the terminals attached to it. Time is divided into unit-length slots. Each frame consists of a fixed number of time slots. Packets can be transmitted in successive frames. The transmission time of stations in a frame is scheduled to avoid any collision. We are concerned with the fixed assignment of transmission for stations in a frame. Thus, once the optimal transmission patterns (the arrangement of transmissions) are determined, the frame is repeated in the time axis. Without loss of generality, we assume that a time slot equals in length to the amount of time for a station to transmit one packet over the channel. We also assume that all stations have the same transmission range R and they are synchronized. Zero-capture is assumed, i.e., when some stations receive two or more overlapping packets, regardless of the difference of received signal power between the stations, collision occurs and all of the packets are destroyed. For any

Manuscript received February 1996; revised July 1, 1996. This work was supported in part by AT&T/Lucent Technologies under a special purpose grant.

G. Wang was with the Center for Communications and Signal Processing, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, University Heights, Newark, NJ 07102 USA. He is now with Sharp Laboratories, Camas, WA 98607 USA.

N. Ansari is with the Center for Communications and Signal Processing, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, University Heights, Newark, NJ 07102 USA.

Publisher Item Identifier S 0733-8716(97)00471-X.

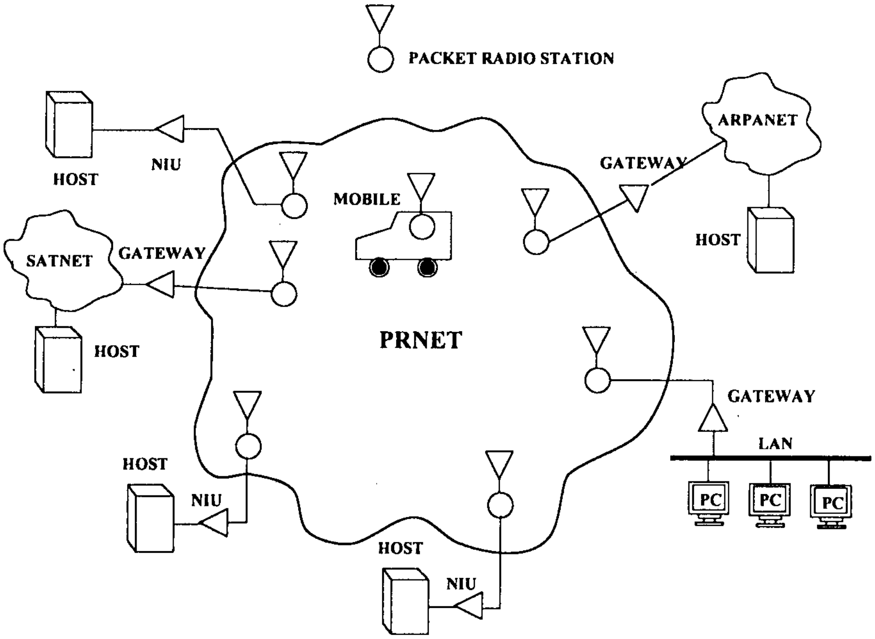


Fig. 1. A PRNET in the Internet.

two stations $i, i' \in V$, if the distance between them is less than R, they can receive the packets transmitted from each other. Therefore, there exists an undirected edge $e = (i, i') \in E$ incident to station i and i', and the two stations are *one-hop* apart. If $(i, i') \notin E$, but there is such an intermediate station j as $(i, j) \in E$ and $(i', j) \in E$, then station i and i' are *two-hop* apart. The topology of a PR network can be described by an $(N \times N)$ symmetric binary matrix C, where $N = |V|$ is the number of stations in the network. The matrix, $C = (c_{ij})(i, j = 1, \cdots N)$, also known as the connectivity matrix, is defined by

$$c_{ij} = \begin{cases} 1, & \text{if } (i, j) \in E \text{ and } i \neq j \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

To ensure that a packet is correctly received in a station, the following constraints must be satisfied.

1) A station cannot have transmission and reception status simultaneously, i.e., if $(i, i') \in E$, station i and i' must be scheduled to transmit in different time slots.
2) A station is not allowed to receive two or more transmissions simultaneously, i.e., if $(i, j) \in E$, $(j, k) \in E$, but $(i, k) \notin E$, station i and k must transmit in different time slots in order to avoid collision in station j.

If the first constraint is violated, the *primary interference* (collision-type) is said to have occurred. The *secondary interference* occurs if the second constraint is violated. In short, a station and its one-hop or two-hop neighboring stations must be scheduled to transmit in different time slots.

We can form a new $(N \times N)$ matrix called the compatibility matrix $F = (f_{ij})$ from matrix C, where

$$f_{ij} = \begin{cases} 1, & \text{if stations } i \text{ and } j \text{ are one-hop} \\ & \text{or two-hop apart} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Note that $f_{ii} = 0 \; \forall i$, and F is symmetric, i.e., $f_{ij} = f_{ji}$. Therefore, for any two stations i and j, if $f_{ij} = 0$, both stations can transmit in the same slot with no interference. We assume that each frame consists of M time slots. In a frame, each station must be scheduled to transmit at least once (one time slot). Additional transmissions can be arranged provided that the addition does not cause interference. We use an $(M \times N)$ binary matrix $S = (s_{ij})$ to express a transmission schedule, where

$$s_{ij} = \begin{cases} 1, & \text{if station } j \text{ transmits at} \\ & \text{the } i\text{th slot in a frame} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Let ρ_k be the channel utilization for station k, then

$$\rho_k = \frac{\text{number of transmission slots assigned to station } k}{\text{frame length}} = \frac{\sum_{i=1}^{M} s_{ik}}{M}. \tag{4}$$

The channel utilization for the whole network, ρ, is given by

$$\rho = \frac{1}{N} \sum_{j=1}^{N} \rho_j = \frac{1}{NM} \sum_{i=1}^{M} \sum_{j=1}^{N} s_{ij}. \tag{5}$$

Denote S' as a set of interference-free schedules where $S' = \{S^1, S^2, \cdots, \}$, and each feasible schedule S^i is an $M \times N$ binary matrix defined by (3). Define ρ_{S^i} as the channel utilization achieved by schedule S^i. Therefore, the optimal scheduling problem is described as follows. Find the optimal schedule $S^{opt} \in S'$ so that it adheres to the following constraints.

1) It has the shortest frame length M.
2) It satisfies the constraints

$$\sum_{i=1}^{M} s_{ij}^{opt} \geq 1 \quad (j = 1, 2, \cdots, N) \tag{6}$$

and

$$\sum_{k=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij}\, s_{ki}^{opt}\, s_{kj}^{opt} = 0. \tag{7}$$

3) It yields the maximum channel utilization, i.e.,

$$\rho_{S^{opt}} = \max_{S^i \in \mathcal{S}'} \rho_{S^i}. \tag{8}$$

Equation (6) reflects the constraint that each station must be able to transmit at least once in a frame. Equation (7) eliminates simultaneous interference transmissions between one station and all of its one-hop or two-hop neighboring stations.

For a given M and N, there are 2^{MN} schedule configurations. An exhaustive search for the optimal schedules is prohibitive when M and N get larger. For a given PR network, the minimum frame length depends on the topology of the network and is generally unknown. However, a tight lower bound for a frame length can be found, thus allowing one to estimate the minimum required frame length. By defining the degree of a vertex i as the number of edges incident to it and denoting the degree as deg (i), we have the following lemma.

Lemma 1: The frame length M satisfies

$$M \geq X(G) + 1 \tag{9}$$

where

$$X(G) = \max_{i \in V} \deg(i). \tag{10}$$

Proof: It is obvious that deg (i) equals the number of one-hop neighbors of station i. Denote $B(i)$ as the set of one-hop neighbors of station i. For any two stations $j,\, j' \in B(i)$, since $c_{ij} = 1$ and $c_{ij'} = 1$, station j and j' are one-hop neighbors if $c_{jj'} = 1$ and two-hop neighbors if $c_{jj'} = 0$. According to the constraints mentioned above, station i and all its deg (i) one-hop neighbors must be arranged to transmit in different distinct time slots in order to obtain interference-free transmissions. Any two stations $j,\, j' \in B(i)$ cannot transmit in the same slot. Therefore, the required number of time slots for transmission for station i and its one-hop neighbors is $\deg(i) + 1$, and the least required number of time slots of a frame for the network, $\Delta(G)$, is given by

$$\Delta(G) = X(G) + 1 = \max_{\forall i \in V} \deg(i) + 1. \tag{11}$$

Equation (11) only provides a lower bound for the frame length. For a given network, the frame length for any of the interference-free schedules is always greater than or equal to $\Delta(G)$, i.e., the inequality $M \geq \Delta(G)$ holds. The real frame length for an optimal schedule depends on the topology of a network. For certain networks, a feasible schedule with exact frame length $\Delta(G)$ may not exist. Therefore, a longer frame length is required. Equation (9) provides useful information when interference-free transmissions are scheduled. We can start to search for the optimal schedules with a frame length equal to the lower bound $\Delta(G)$. If no feasible schedules with this length can be found, we will increase the frame length, and then search further for the feasible solutions. In this way, the scheduled frame length would be minimized. Once the frame length is determined, the optimal scheduling procedure will continue until a interference-free schedule with maximum channel utilization, defined in (5), is found. It is shown that such a scheduling problem is NP-complete, which means that an efficient polynomial algorithm rarely exists. In Section IV, we will discuss how to use neural networks to solve such an optimal scheduling problem.

III. Mean Field Annealing

A. Statistical Mechanics

In statistical mechanics, a physical process called *annealing* is often performed in order to relax the system to the state with the minimum energy. In the annealing process, a solid in a *heat bath* is heated up by increasing the temperature of the bath until the solid is melted into liquid, then the temperature is lowered down slowly. At each temperature, all particles randomly arrange themselves until thermal equilibrium is reached. If the cooling is slow enough to allow the solid to reach thermal equilibrium at each temperature, the low energy crystalline solid would be formed when the system is frozen ($T \rightarrow 0$). However, if the annealing is too fast, the solid may become glass with noncrystalline structure or the defected crystal with meta-stable amorphous structures. If a state is defined by the set of particle positions, then, at thermal equilibrium, the probability of the system being in state i is represented by the *Gibbs distribution* [3], [4]

$$\begin{aligned}\pi_i &= \Pr\{\mathbf{s} = i\}\\ &= \frac{\exp\left[-\dfrac{E(i)}{k_b T}\right]}{\mathcal{Z}}\end{aligned} \tag{12}$$

where $\mathcal{Z} = \sum_{i \in S} \exp\left[-E(i)/k_b T\right]$ is called the *partition function*, k_b is the *Boltzmann constant*, T is the temperature, and $E(i)$ is the energy of state i, $\mathcal{S}$ is the state space, and $k_b,\, T,\, E(i) \in \mathbf{R}^+$, where $\mathbf{R}^+$ is the positive real space. It is easy to find that [3]

$$\begin{aligned}\lim_{T \rightarrow 0} \pi_i &= \lim_{T \rightarrow 0} \frac{\exp\left[-\dfrac{E(i) - E_{\min}}{k_b T}\right]}{\displaystyle\sum_{j \in \mathcal{S}} \exp\left[-\dfrac{E(j) - E_{\min}}{k_b T}\right]}\\ &= \begin{cases} \dfrac{1}{|\mathcal{S}_{\min}|}, & \text{if } i \in \mathcal{S}_{\min}\\ 0, & \text{otherwise}\end{cases}\end{aligned} \tag{13}$$

where $\mathcal{S}_{\min} = \{i\colon E(i) = E_{\min}\}$ and $E_{\min} = \min_{j \in \mathcal{S}} E(j)$. From this equation, we can see that as the temperature approaches zero, the system will converge to the states with the minimum energy, i.e., the states with the minimum energy are reached at lower temperatures.

B. Simulated Annealing

Based on the annealing process in statistical mechanics, Kirkpatrick *et al.* [5] proposed an algorithm, namely, *simulated annealing* (SA), for solving complicated combinatorial optimization problems. In the SA algorithm, a simulation of the annealing process is performed. The cost function and configuration in optimization correspond to the energy function and state in statistical physics, respectively. The temperature is introduced as a control parameter.

A cost function $f\colon \mathcal{S} \rightarrow \mathbf{R}^+$, $s \in \mathcal{S}$, to be minimized is usually defined on some finite set $\mathcal{S}$. For each configuration

$s \in \mathcal{S}$, there is a neighboring set $\mathcal{N}(s) \subseteq \mathcal{S}$, which is generated by a small perturbation of s.

In [3], it shows that the SA algorithm asymptotically converges to the configurations with the minimum cost, i.e., if the temperature is slowly lowered and at each temperature the system performs a sufficient number of transitions, the configurations (solutions) with the global minimum cost can be found with probability one.

C. Mean Field Annealing

Even though SA is proved to be able to reach the global optima asymptotically, it is time consuming to reach thermal equilibrium at each temperature. Finite number of transitions at each temperature cannot guarantee convergence to the global optima. In statistical physics, *mean field* approximation is often used. Mean field annealing (MFA) uses a set of deterministic equations to replace the stochastic process in SA. It uses saddle point approximation in the calculation of the stationary probability distribution at equilibrium, and reaches equilibrium at each temperature much faster than SA. Even though this approximation method may not guarantee convergence to global minima, it does provide a good approximation in finding near-optimal solutions with much less computing effort.

The MFA theory and the related derivations can be found in [7]. For a binary system, a configuration $\mathbf{s} = [s_1, s_2, \cdots, s_n]^T$ is represented by a sequence of binary values, i.e., $\mathbf{s} \in \{0, 1\}^n$. We have the following MFA equations:

$$v_i = \overline{s}_i = \frac{1}{2}\left[1 + \tanh\left(\frac{h_i}{2T}\right)\right] \tag{14}$$

$$h_i = -\frac{\partial f(\mathbf{v})}{\partial v_i} \tag{15}$$

where $\overline{(\cdot)}$ is the average operator. h_i in (15) is called the *mean field*. In MFA, the iterative procedure to reach thermal equilibrium at each temperature is called *relaxation*, in which the mean field is updated by

$$h_i(t + \Delta t) = h_i(t) + \Delta t\left[-\frac{\partial f_h(\mathbf{v})}{\partial v_i} - h_i(t)\right]. \tag{16}$$

IV. The Optimal Scheduling Algorithm Based on MFA

A. Energy Function

To solve the optimal scheduling problem by using MFA, we first need to map the channel utilization to be maximized and the constraints into an energy function. We assume that the frame length is M and there are N stations in a PR network. $M \times N$ neurons are required to represent a schedule. Each neuron s_{ij} $(i = 1, \cdots, M;\ j = 1, \cdots, N)$ is defined in (3). The following energy function is derived:

$$\begin{aligned} E = &-\frac{w_1}{2}\left(\frac{1}{NM}\sum_{i=1}^{M}\sum_{j=1}^{N} s_{ij}^2\right) \\ &+ \frac{w_2}{2}\sum_{k=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij}\, s_{ki}\, s_{kj} \\ &+ w_3 \sum_{i=1}^{M}\sum_{j=1}^{N} s_{ij}(1 - s_{ij}) \end{aligned} \tag{17}$$

where weights $w_i > 0$, $i = 1, 2, 3$. The first term in (17) is the negatively weighed channel utilization. The second term is a penalty function for constraint violations. When the constraint is satisfied, it becomes zero. The third term is used to force neurons to converge to either 0 or 1 (if all $s_{ij} = 0$ or 1, the third term equals zero). The mean field of neuron ij is

$$\begin{aligned} h_{ij} &= -\frac{\partial E}{\partial v_{ij}} \\ &= \frac{w_1}{NM}\cdot v_{ij} - w_2 \sum_{k=1}^{N} f_{jk} \\ &\quad \cdot v_{ik} - w_3(1 - 2v_{ij}), \qquad \text{where } v_{ij} = \overline{s}_{ij}. \end{aligned} \tag{18}$$

B. The MFA Scheduling Algorithm

The updating of the neuron average is given by

$$\begin{aligned} v_{ij}(t + \Delta t) = \frac{1}{2}\Bigg\{1 + \tanh\Bigg[\frac{1}{2T}\Bigg(&\frac{w_1}{NM}\, v_{ij} - w_2 \\ &\cdot \sum_{k=1}^{N} f_{jk}\cdot v_{ik} - w_3(1 - 2v_{ij})\Bigg)\Bigg]\Bigg\}. \end{aligned} \tag{19}$$

The MFA iteration proceeds until freezing occurs. Since the exact frame length is unknown, we can start to schedule the frame with length $\Delta(G)$, the lower bound of the frame length shown in (11). The proposed scheduling algorithm includes three steps.

Step 1) *Presetting Neurons:* Find the station p which has the maximum degree $X(G)$, then set the initial frame length $M = \Delta(G)$ as defined in (11), and assign station p and its one-hop neighboring stations $k \in B(p) = \{k\colon c_{pk} = 1\}$ to the different distinct time slots. For example, set $v_{1p} = 1$ and $v_{ij} = 1 \quad \forall j \in B(p)\ [i = 2, \cdots, \Delta(G)]$. For the ith slot, since $v_{ij} = 1$, the kth neuron with $f_{jk} = 1$ must be set to $v_{ik} = 0$ to resolve interference [see (7)]. The preassigned neurons no longer need to be updated, and their values will be used to update the other neurons.

Step 2) *Performing the MFA Iterations Based on (19):* The iteration continues until freezing occurs and the freezing state should provide the maximum channel utilization within the frame length M.

Step 3) *Applying the Heuristic Algorithm for Unassigned Stations:* After completing the above two steps, some stations might remain unassigned for transmission due to the interference-free constraint. The number of unassigned stations depends on the topology of the network. Usually, after the first two steps, only a few stations are unassigned. Extra time slots are needed to arrange the remaining transmissions. We use the following heuristic algorithm to schedule the transmissions of the unassigned stations. Denote the unassigned stations as $U = \{U_1, \cdots, U_q\}$.

a) Sort the stations in U in a descending order of station degree such that $\deg(U_i) \geq \deg(U_{i+1})$.

b) Add a time slot to the frame, and assign the stations in U to transmit in the slot. The priority of assigning a station's transmission is based on the order of U, i.e., the priority of U_i is greater than that of U_{i+1}. The stations arranged in the slot must be interference-free. Repeat the above procedure until U is empty.

c) The actual frame length M equals $\Delta(G) +$ the number of added time slots.

d) Check the stations which have been assigned to transmit in the first $\Delta(G)$ time slots. If any of the stations can transmit in the added time slots without conflict, assign the transmissions of the stations in the corresponding time slots.

After the three steps are completed, the optimal schedule represented by $v_{ij} \forall i, j$ is translated into the actual transmission assignment (i.e., $v_{ij} = 1$ means that station j can transmit in slot i).

C. Critical Temperature

Each neuron is updated according to

$$v_{ij}^{(n)} = \frac{1}{2} + \frac{1}{2} \tanh \left(-\frac{1}{2T} \frac{\partial E[v^{(n-1)}]}{\partial v_{ij}^{(n-1)}} \right) \qquad \forall i, j \quad (20)$$

where $v_{ij} = \bar{s}_{ij}$, $v = (v_{ij})$ is the neuron matrix of dimension $M \times N$, and n stands for the nth iteration.

From (20), it is seen that iterations starting at too high a temperature result in trivial solutions. Starting at too low a temperature, on the other hand, might force the system into a poor or invalid solution. The critical temperature is defined as the temperature at which fast state transitions begin.

For the scheduling problem

$$v_{ij}^{(n)} = \frac{1}{2} + \frac{1}{2} \tanh \left(-\frac{1}{2T} \left\{ -\frac{w_1}{MN} v_{ij}^{(n-1)} + w_2 \sum_{k=1}^{N} f_{jk} \cdot v_{ik}^{(n-1)}) + w_3[1 - 2v_{ij}^{(n-1)}] \right\} \right). \quad (21)$$

It is seen from (21) that the state of each neuron remains relatively unchanged at high temperatures. The iteration procedure in (21) should start at a temperature (the critical temperature) at which fast transition begins as illustrated in Fig. 2 until steady state is reached. Thus

$$\left| \frac{1}{2T_c} \frac{\partial E}{\partial v_{ij}^{(0)}} \right| = \alpha$$

where $0.1 \leq \alpha \leq 0.2$ is chosen, at which state transition becomes rapid. Since all neurons v_{ij} are initialized to $0.5 \pm \delta 1$ ($\delta 1$ is a small random number), we chose

$$T_c = \frac{1}{2\alpha} \left| \frac{\partial E}{\partial v_{ij}^{(0)}} \right|$$
$$\approx \frac{1}{2\alpha} \left| -\frac{0.5W_1}{MN} + W_2 \sum_{k=1}^{N} 0.5 f_{jk} + W_3[1 - 2v_{ij}^{(0)}] \right|$$

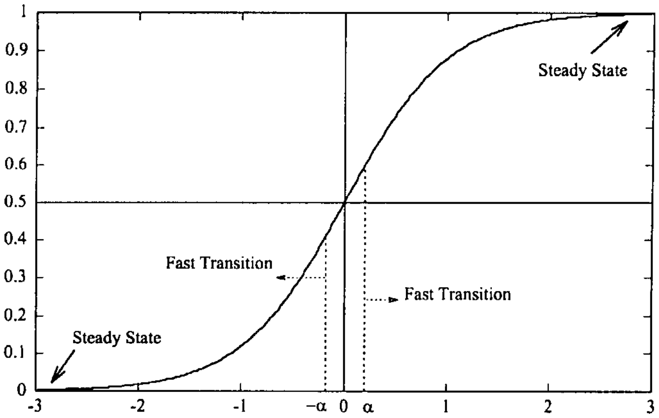


Fig. 2. Illustration of state transition.

$$\approx \frac{1}{2\alpha} \left| -\frac{0.5W_1}{MN} + W_2 \sum_{k=1}^{N} 0.5 f_{jk} \right|. \quad (22)$$

Thus

$$T_c \geq \frac{1}{4\alpha} \left(-\frac{W_1}{MN} + W_2 \cdot \min_j \sum_{k=1}^{N} f_{jk} \right)$$

or

$$T_c \leq \frac{1}{4\alpha} \left(-\frac{W_1}{MN} + W_2 \cdot \max_j \sum_{k=1}^{N} f_{jk} \right).$$

The lower bound for the temperature is taken as

$$T_c = \frac{1}{4\alpha} \left(-\frac{W_1}{MN} + W_2 \cdot \min_j \sum_{k} f_{jk} \right). \quad (23)$$

The derived critical temperature is suitable only for the synchronous mode, in which all $v_{ij}^{(n)}$ are updated simultaneously using the previous $v^{(n-1)}$. For the asynchronous mode, each neuron is sequentially updated. The critical temperature for asynchronous iteration can be estimated by a *trial-and-error* method, i.e., the iteration starts at a very high temperature and the temperature is gradually lowered. At each temperature, each neuron is sequentially updated once. At the end of each iteration, the absolute average value

$$\epsilon_1 = \frac{1}{MN} \sum_i \sum_j |v_{ij}^{(n)} - v_{ij}^{(n-1)}| \quad (24)$$

is checked. At high temperatures, $\epsilon_1 \ll \alpha$ for some small constant α. When $\epsilon_1 \geq \alpha$, significant state transitions begin. Therefore, when $\epsilon_1 \geq \alpha$, the trial process ends, at which point the corresponding temperature is critical.

D. Annealing Schedule

The annealing schedule reflects the way the temperature is reduced, and the following empirical annealing schedule

$$T_{n+1} = 0.9 \cdot T_n \quad (25)$$

is adopted in our simulations.

E. Stopping Criterion

At a very low temperature, all neurons converge to either zero or one. Let

$$\epsilon_2 = \frac{1}{MN} \sum_i \sum_j v_{ij}^{(n)} [1 - v_{ij}^{(n)}]. \quad (26)$$

Thus, the iterative process may stop either when the error is approaching zero (in our case, $\epsilon_2 \leq 0.01$) or when the temperature reaches zero. The final values of neurons represent the schedule.

V. Numerical Examples and Performance Analysis

A. Channel Utilization and Average Time Delay

The performance of the resulting schedules can be evaluated by two criteria: channel utilization and average time delay. The channel utilization is defined in (5). Before we derive the average time delay, the following assumptions are made.

1) Packets have a fixed length, and the length of a time slot equals the time required to transmit a packet.
2) The interarrival time for each station i is statistically independent from other stations, and packets arrive according to a *Poisson* process with a rate of λ_i (packets/slot). The total traffic in stations i consists of the traffic incoming from other stations and the data from terminals attached to it. Packets are stored in buffers in each station and the buffer size is infinite.
3) The probability distribution of the service time of station i is deterministic and statically independent from other stations. The average service rate is μ_i (packets/slot).
4) Packets can be transmitted only at the beginning of each time slot.

Under the above assumptions, a network can be modeled as N M/D/1 queues, where N is the number of stations. According to the *Pollaczek–Khinchin formula* [9], the average delay for each queue i is given by

$$D_i = \overline{X}_i + \frac{\lambda_i \overline{X_i^2}}{2(1-\rho_i)} \tag{27}$$

where

$\overline{X_i} = 1/\mu_i$ average service time for station i;
$\rho_i = \lambda_i/\mu_i$ utilization factor for station i;
$\overline{X_i^2}$ second moment of service time for station i.

Since the service time is deterministic, the variance equals zero, and thus

$$\overline{X_i^2} = \overline{X_i}^2 = \frac{1}{\mu_i^2} \tag{28}$$

and

$$\mu_i = \frac{\sum_{j=1}^{M} v_{ji}}{M} \text{ (packets/slot).} \tag{29}$$

The total time delay is given by

$$D = \frac{\sum_{i=1}^{N} \lambda_i D_i}{\sum_{i=1}^{N} \lambda_i}. \tag{30}$$

B. Numerical Results

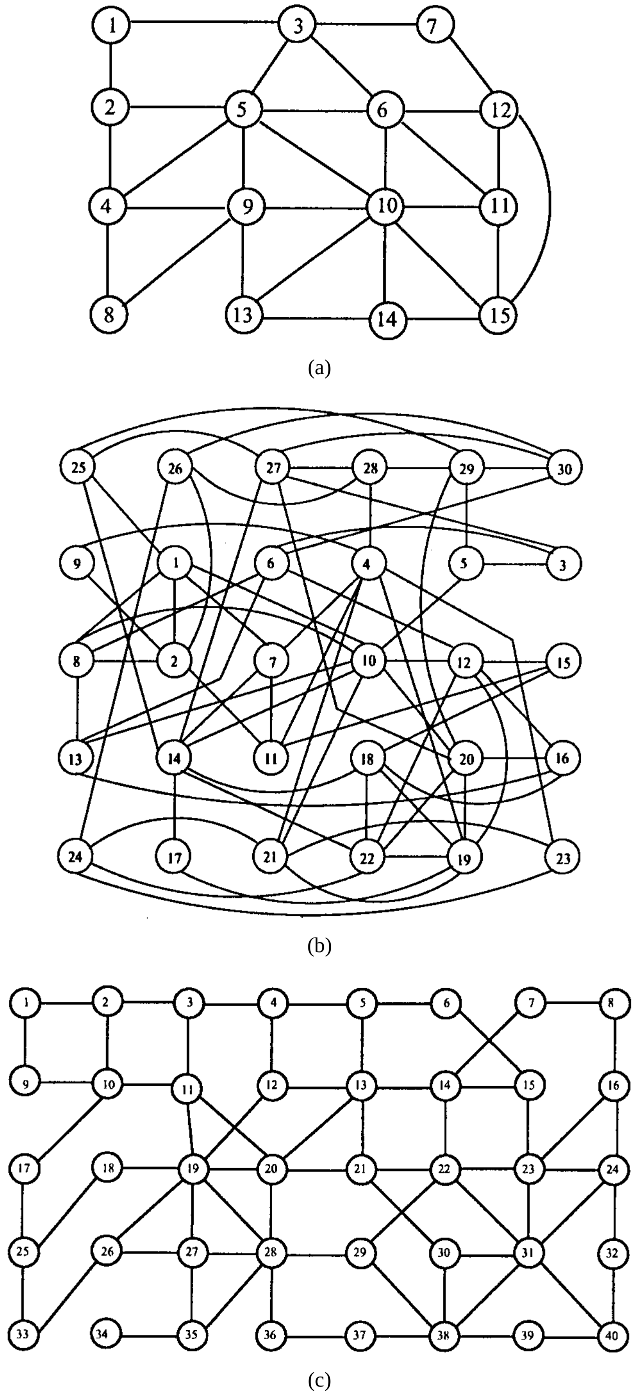
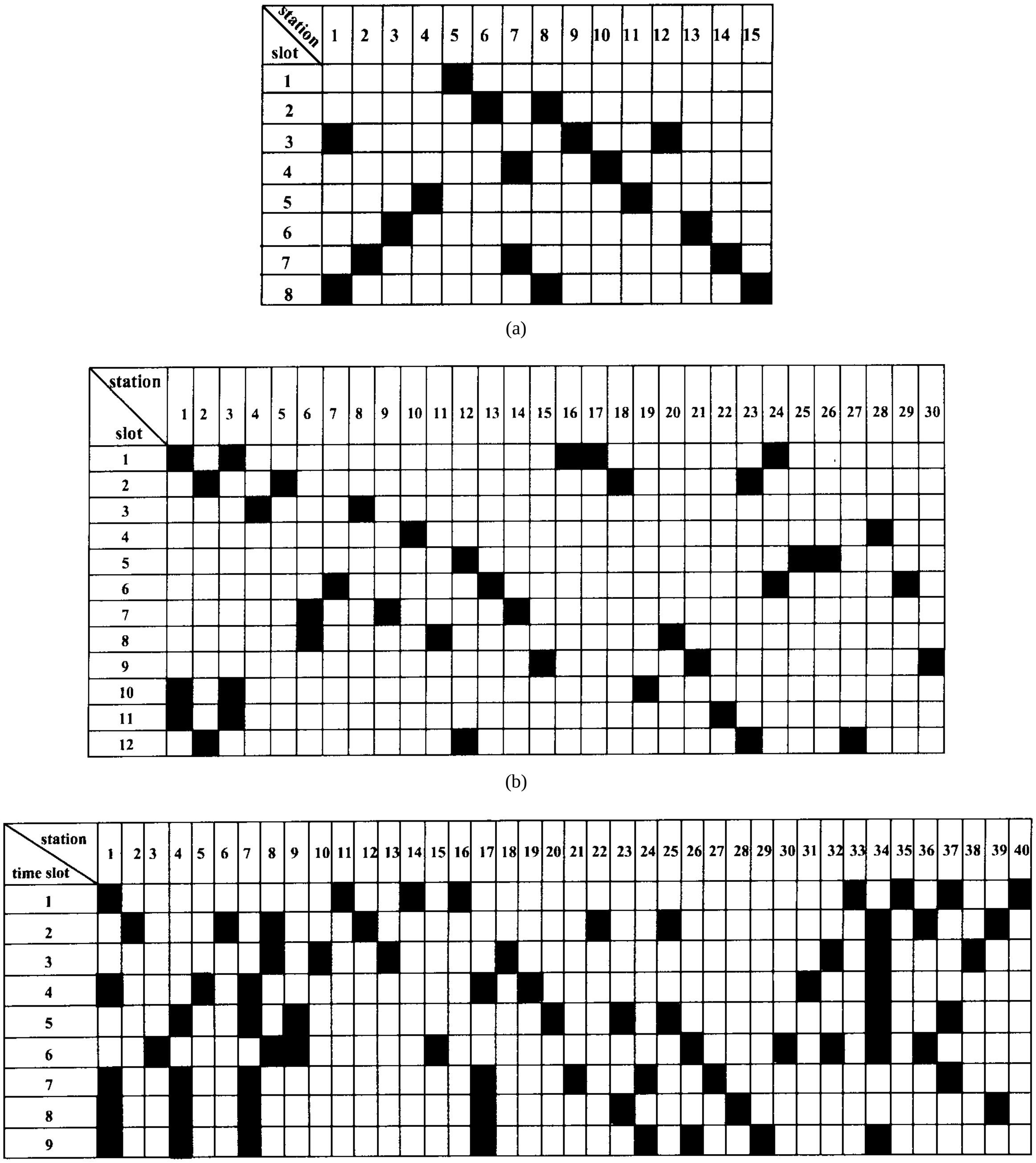


Fig. 3. The radio networks used in the simulation: (a) the 15-station network, (b) the 30-station network, and (c) the 40-station network.

Three PR networks shown in Fig. 3 with 15, 30, and 40 stations are scheduled. The resulting schedules are shown in Fig. 4, where a black box represents a transmission time slot. Note that $\Delta(G) = 9$, 10, 9, and the scheduled frame length is 9, 11, 10, respectively, which are close to the lower bound described by (9).

We compare the performance achieved by the MFA scheduling algorithm with the other two scheduling algorithms [10], [11] in which the objective of scheduling is to achieve the maximum channel utilization. The time delay and channel utilization are plotted in Fig. 5. From this figure, it is seen that the time delay experienced by the MFA schedule is much less than that of the other two scheduling algorithms which have the same time delay, but the channel utilization achieved by MFA is a little bit less than the other two.



station slot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1					■										
2						■		■							
3	■								■			■			
4							■			■					
5				■							■				
6			■										■		
7		■					■							■	
8	■							■							■

(a)

station slot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	■		■													■	■							■						
2		■			■													■					■							
3				■				■																						
4										■																		■		
5												■													■	■				
6							■						■											■					■	
7						■			■					■																
8						■					■									■										
9															■						■									■
10	■		■																■											
11	■		■																			■								
12		■										■											■				■			

(b)

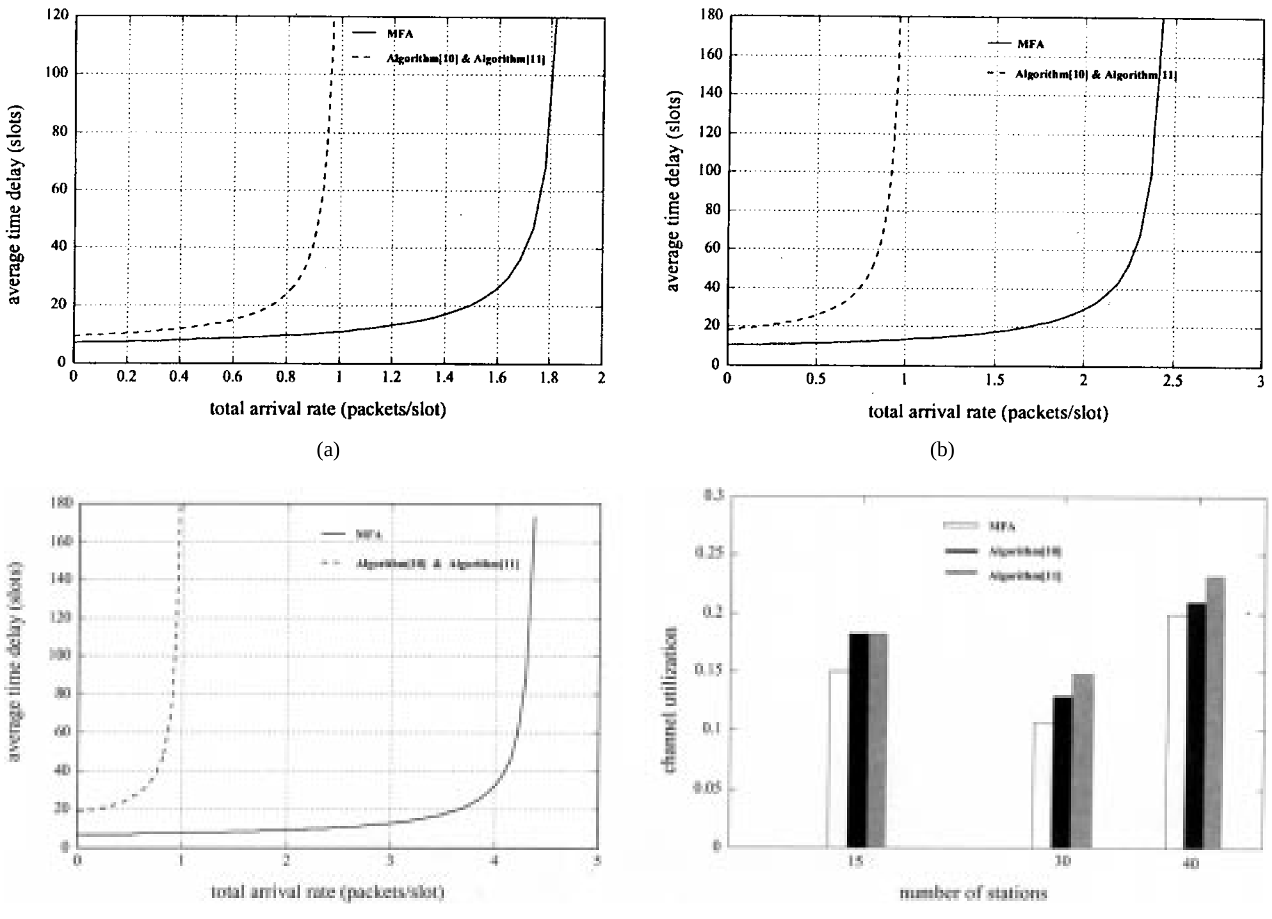
station time slot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	■										■			■		■																	■		■		■			■
2		■				■		■				■										■			■									■		■			■	
3								■		■			■					■														■		■				■		
4	■				■		■										■		■												■			■						
5				■			■		■											■			■		■									■			■			
6			■					■	■						■											■				■		■		■		■				
7	■			■			■										■				■			■			■										■			
8	■			■			■										■						■					■											■	
9	■			■			■										■							■		■			■					■						

(c)

Fig. 4. The MFA schedules: (a) the 15-station network, (b) the 30-station network, and (c) the 40-station network.

VI. Conclusion

In this paper, we presented an efficient broadcast scheduling algorithm based on MFA neural networks. As shown in the Appendix, the TDMA broadcast scheduling in a PR network is an NP-complete combinatorial optimization problem. We first map the channel utilization to be maximized and the interference-free constraints onto an energy function, and then the MFA procedure is applied to searching for the optimal solutions. To reduce the computational complexity, we divide the algorithm into three steps. In the first step, the solution space is reduced by presetting some neurons according to the topology of the scheduling network. The preset neurons need not be updated further, and their values can be used to update other neurons. In the second step, the MFA procedure is executed to maximize channel utilization. At the end of the execution, a solution with near-maximal channel utilization can be found. After the first two steps, some stations might not be assigned to transmit in any slots. To arrange the unassigned stations, additional time slots are needed. In the last step, a heuristic method is used to arrange the transmissions of the unassigned stations. This step guarantees that the additional number of time slots is minimal. Since neural networks provide a parallel computing strategy, the proposed scheduling algorithm will obtain the optimal solution faster than other heuristic algorithms. Numerical results have shown that the proposed algorithm can find the shortest interference-free frame schedule while providing the maximum channel



(a)

(b)

(c)

(d)

Fig. 5. Time delay for different schemes: (a) 15-station, (b) 30-station, (c) 40-station, and (d) channel utilization.

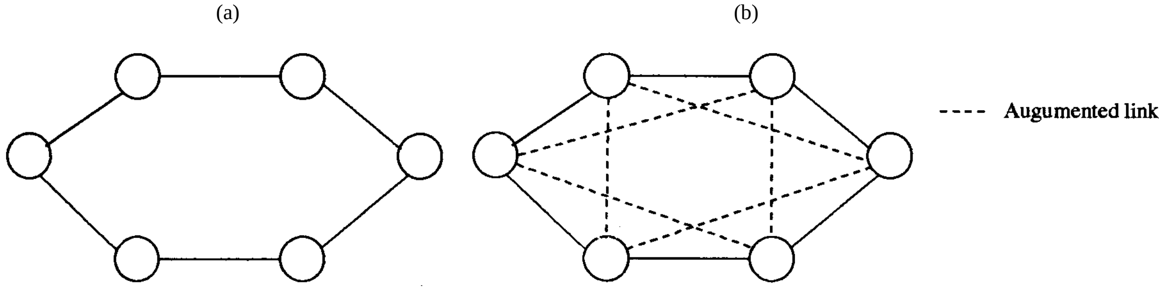


Fig. 6. The derivation of an augmented graph from a graph G: (a) graph G and (b) graph G_a.

utilization. The average time delay is much smaller than that of the other two algorithms.

Appendix
NP-Completeness of the Broadcast Scheduling Problems

We denote the broadcast scheduling problem described in Section II as Π_B. It is defined as finding an interference-free transmission schedule with the minimum frame length. Π_B can be described as a decision problem.

Instance: The scheduling problem $\Pi_B = (G, K)$, where $G = (V, E)$ represents a packet radio network, and K is a positive integer ($K \leq |V|$).

Question: Is there a schedule which has the frame length K or less and each station transmits at least once in such a frame?

Using graph theory to assist the proof of NP-completeness of the scheduling problem, we form an augmented graph $G_a = (V_a, E_a)$ in such a way that

$$V_a = V,\ E_a = E \bigcup_{\substack{(i,j)\in E\\ (j,k)\in E\\ (i,k)\notin E.}} (i,k).$$

Fig. 6 illustrates the augmented graph G_a derived from a graph G. For problem Π_B, finding an interference-free transmission schedule in any time slot within a frame is equivalent to finding a set of vertices $V' \subseteq V$ in G_a such that, if any $i, j \in V'$, then $(i, j) \notin E_a$. All stations (vertices) in the set V' can transmit simultaneously with no interference.

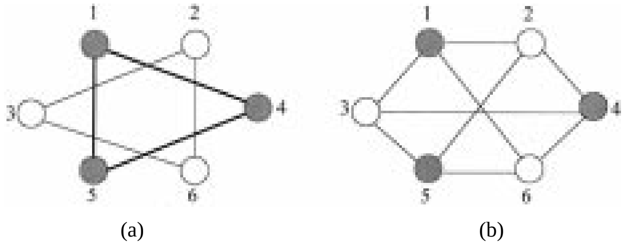


Fig. 7. Illustration of the complement of a graph: (a) graph G and (b) graph G^c.

In graph theory, a subset $V' \subseteq V$ is called an *independent set* in a graph $G = (V, E)$ if, for all $i, j \subseteq V'$, the edge $(i, j) \notin E$. Vertices are said to be *independent* from each other in the subset V'. The problem to find a maximum independent set in a graph is denoted as Π_I. Since we will use Π_I in the NP-completeness proof of problem Π_B, we first need to prove that Π_I is NP-complete.

Theorem A.1:

Proof: Π_I is NP-complete. To prove Π_I is NP-complete, we first need to show $\Pi_I \in NP$, and then find a polynomial transformation from a known NP-complete problem called *clique*, denoted by Π_C.

In an undirected graph $G = (V, E)$, a *clique* is a subset $V' \subseteq V$ of vertices, each pair of which is connected by an edge in E. Therefore, a clique is a complete subgraph of G. The size of a clique is the number of vertices it contains. It has been proved that Π_C is an NP-complete problem [12]. Here, we simply use the result to prove the NP-completeness of the Π_I problem.

1) $\Pi_I \in$ *NP:* "NP" stands for nondeterministic polynomial time. *Class NP* is defined as the class of languages that can be verified by a polynomial-time algorithm. Notice that polynomial time verifiability does not imply polynomial time solvability. *Class P* consists of problems that can be solved in polynomial time, whereas the class NP consists of problems for which a solution can be verified rather than solved in polynomial time.

   For a given graph $G = (V, E)$, it is easy to see that the decision problem verifying whether a selected subset of vertices $V' \in V$ is independent and less than a certain constant $K \leq |V|$ can be completed in polynomial time. Therefore, $\Pi_I \in NP$.

2) *Polynomial Transformation* $\Pi_C \propto_p \Pi_I$*:* We will show that there is a polynomial transformation from Π_C to Π_I, denoted by $\Pi_C \propto_p \Pi_I$. Here, a polynomial transformation from one problem Π_1 to another problem Π_2 is defined as: there is a polynomial-time computable function f such that all instances $x \in \Pi_1$ if and only if $f(x) \in \Pi_2$. The transformation $\Pi_C \propto_p \Pi_I$ is based on the notion of the *complement* of a graph. For an undirected graph $G = (V, E)$, the complement of G is defined as $G^c = (V, E^c)$, where $E^c = \{(i, j) : (i, j) \notin E\}$. Fig. 7 shows a graph and its complement and the transformation from Π_C to Π_I.

   The transformation takes an instance of the clique problem $[G, K]$ as *input*. It computes the complement G^c, which is easily derived in polynomial time $O(|V|^2)$. The *output* of the transformation algorithm is the instance $[G^c, K]$ of the problem Π_I. To show this transformation is valid, we need to prove that the graph G has a clique of size K if and only if the graph G^c has an independent set of size K.

   Suppose that G has a clique $V' \subseteq V$ with $|V'| = K$, we claim that V' is an independent set in G^c. Based on the formation of G^c, if any $i, j \in V'$ and $(i, j) \in E$, we have $(i, j) \notin E^c$, which implies that vertices i, j are independent in G^c. Since (i, j) is chosen arbitrarily from E and every edge $(i, j) \in E$ is not in E^c, all vertices $K = |V'|$ in a clique V' are independent in E^c. Hence, the set V' of size K forms an independent set in G^c. Conversely, suppose that G^c has an independent set $V' \subseteq V$ of size $K = |V'|$, then, for all $i, j \in V'$, we have $(i, j) \notin E^c$, and $(i, j) \in E$ in G, which forms a complete subgraph of $V' \in V$ in E. In other words, V' is a clique with size K in G.

From the above statement, we conclude that a graph G has a clique V' if and only if the graph G^c has an independent set V'. The derivation of G^c from G can be completed in polynomial time, and therefore $\Pi_C \propto_p \Pi_I$.

Since we have shown that $\Pi_I \in NP$ and $\Pi_C \propto_p \Pi_I$, we have proved that Π_I is NP-complete.

Theorem A.2:

Proof: Π_B is NP-complete. We first form an augment graph $G_a = (V, E_a)$ from graph G described above. Thus, scheduling an interference-free transmission in any time slot within a frame is equivalent to finding such a set $V' \subseteq V$ of vertices that, if any $i, j \in V'$, then $(i, j) \notin E_a$. To prove Π_B is NP-complete, we first need to show $\Pi_B \in$ NP, and then find a polynomial transformation from the NP-complete problem Π_I.

The broadcast schedule can be represented by

$$s_{ij} = \begin{cases} 1, & \text{if station } j \text{ transmits at time slot } i \\ 0, & \text{otherwise.} \end{cases}$$

1) $\Pi_B \in$ *NP:* To prove $\Pi_B \in$ NP, we first need to *guess* a frame schedule with an arbitrary frame length $M \leq N$ (N is the number of stations in the packet radio network), then form the augmented graph $G_a = (V, E_a)$ from graph G at each time slot t. Denote $V'_t = \{i: i \in V, s_{ti} = 1\}$, where s_{ti} is defined by the above equation. We need to check whether $(i, j) \notin E_a$ for any $i, j \in V'_t$. At the same time, we need to check if the guessed schedule length $M \leq K$. It is easy to verify that the checking process can be completed in polynomial time $O(N^2)$. Therefore, $\Pi_B \in$ NP.

2) $\Pi_I \propto_p \Pi_B$: Finding an interference-free schedule with the minimum frame length is equivalent to coloring vertices in G_a so that, for any pair of vertices $i, j \in V$, if $(i, j) \in E_a$, then i and j must be assigned to different colors. The coloring problem is to find the minimum number of colors that can cover all vertices in a graph. Here, the color corresponds to a time slot and the frame length to the total number of colors n_c. In order to find the minimum n_c, we should assign a color to as many

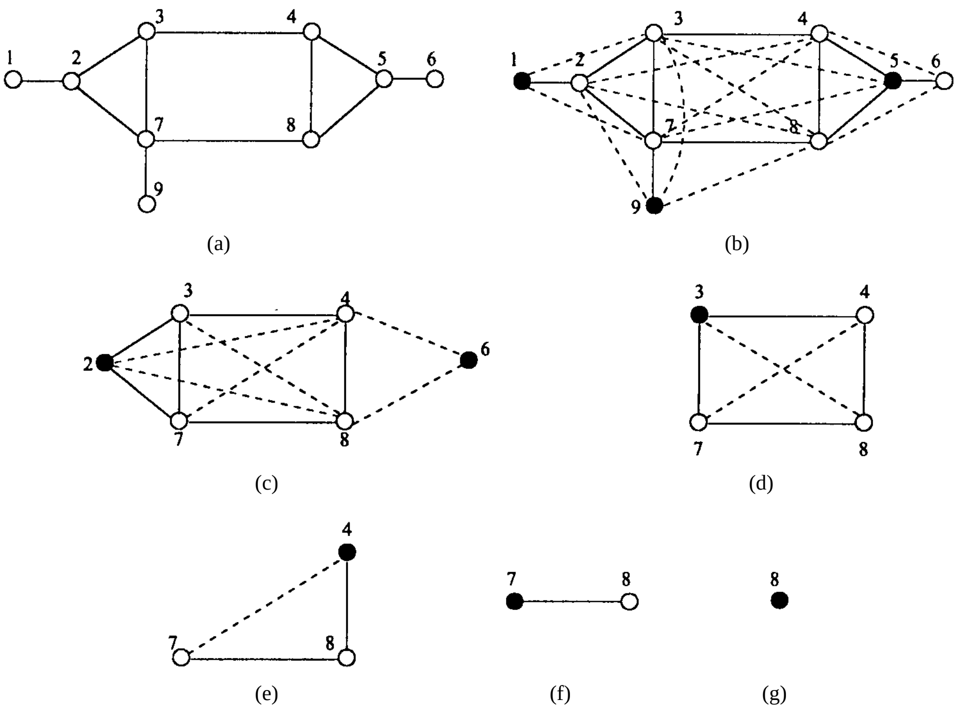
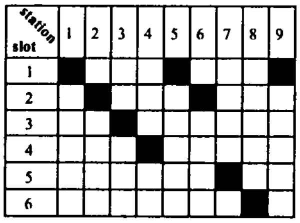


Fig. 8. The decomposition of the scheduling problem.



station / slot	1	2	3	4	5	6	7	8	9
1	■				■				■
2		■				■			
3			■						
4				■					
5							■		
6								■	

Fig. 9. The decoding of the coloring graph.

vertices as possible provided that the coloring does not cause conflict. Therefore, solving the problem Π_B can be performed as follows.

a) Form an augmented graph G_a based on G, $t \leftarrow 1$.

b) Find the maximum independent set V_t' for $G_a = (V, E_a)$ and assign a color c_t to the set V_t'.

c) $V \Leftarrow V - V_t'$, $E_a \Leftarrow E_a - \{(i, j) : i \in V_t', (i, j) \in E\, \forall j\}$.

If G_a is not empty, then $t \leftarrow t+1$ and repeat Step b), else the coloring procedure ends. The results are encoded into a frame schedule by

$$s_{tj} = \begin{cases} 1, & \text{if } j \in V_t' \\ 0, & \text{otherwise.} \end{cases}$$

Fig. 8 illustrates the formulation of the graph G_a and the coloring process. We have $V_1' = \{1, 5, 9\}$, $V_2' = \{2, 6\}$, $V_3' = \{3\}$, $V_4' = \{4\}$, $V_5' = \{7\}$, $V_6' = \{8\}$. Fig. 9 is the decoding of the graph coloring result.

We notice that the coloring problem is equivalent to finding the maximum independent set in G_a. Therefore, the broadcast scheduling problem Π_B can be transformed into the independent set problem Π_I by finding an augmented graph G_a from G, which is easily generated in polynomial time. Thus, we have $\Pi_I \propto_p \Pi_B$.

Having shown $\Pi_B \in NP$ and $\Pi_I \propto_p \Pi_B$, we can conclude that Π_B is NP-complete.

References

[1] B. M. Leiner, D. L. Nielson, and F. A. Tobagi, “Issues in packet radio network design,” *Proc. IEEE*, vol. 75, no. 1, pp. 6–20, Jan. 1987.

[2] L. Kleinrock and J. Silvester, “Spatial reuse in multihop packet radio networks,” *Proc. IEEE*, vol. 75, no. 1, pp. 156–167, Jan. 1987.

[3] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machine—A Stochastic Approach to Combinatorial Optimization and Neural Computing.* New York: Wiley, 1989.

[4] G. Ho. Wannier, *Statistical Physics.* New York: Wiley, 1966, ch. 4.

[5] S. Kirkpatrick, C. Gelatt, and M. Vecchi, “Optimization by simulated annealing,” *Sci. 220*, pp. 671–680, 1983.

[6] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *J. Chem. Phys.*, vol. 2, no. 1, pp. 1087–1092, 1953.

[7] C. Peterson and B. Soderberg, “A new method for mapping optimization problems onto neural network,” *Int. J. Neural Sys.*, vol. 1, no. 1, pp. 3–22, May 1989.

[8] J. J. Hopfield and D. W. Tank, “Neural networks and physical systems with emergent collective computational abilities,”*Proc. Nat. Acad. Sci.*, pp. 2541–2554, 1982.

[9] D. Bertsekas and R. Gallager, *Data Networks.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[10] A. Ephremides and T. V. Truong, “Scheduling broadcasts in multihop radio network,” *IEEE Trans. Commun.*, vol. 38, pp. 456–460, Apr. 1990.

[11] G. Wang and N. Ansari, “A neural network approach to broadcast scheduling in multihop radio networks,” in *Proc. ICNN’94*, June 1994, pp. 4699–4704.

[12] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* New York: W. H. Freeman, 1979.

Gangsheng Wang received the B.S.E.E. degree from Huazhong University of Science and Technology, China, in 1982, the M.S.E.E. degree from The Second Academy, Ministry of Aerospace, China, in 1988, and the Ph.D. degree from New Jersey Institute of Technology, Newark, NJ, in 1996.

From 1988 to 1991, he conducted R&D work in real-time signal processing and FEC coding and decoding in China. From May 1995 to October 1996, he worked at Bell Laboratories, Lucent Technologies, and conducted research in CDMA cellular mobile communications. He has been conducting research in multimedia communications at Sharp Laboratories of America, Inc. since November 1996. His main research interests are in the fields of neural networks, wireless communications, and communication networking.

Nirwan Ansari (S'83–M'88–SM'94) received the B.S.E.E. degree (summa cum laude) from the New Jersey Institute of Technology, Newark, NJ, in 1982, the M.S.E.E. degree from the University of Michigan, Ann Arbor, in 1983, and the Ph.D. degree from Purdue University, in 1988.

In 1988, he joined the Electrical and Computer Engineering Department, the New Jersey Institute of Technology, Newark, NJ, where he is an Associate Professor and the Assistant Chair for Graduate Studies. His current research interests include distributed and adaptive detection in CDMA, ATM networks, data fusion, computational intelligence, and nonlinear signal processing. He co-edited *Neural Networks in Telecommunications* (Norwell, MA: Kluwer, 1994) and co-authored *Computational Intelligence for Optimization* (Norwell, MA: Kluwer, 1997).

Dr. Ansari is the Chair of the North Jersey Chapter of the IEEE Communications Society which received the 1996 Chapter of the Year Award.