Research Article

A Dynamical Slot Assignment Method for Wireless Sensor Networks Based on Hopfield Network

Qi Yang, Xiao Lin, Yuxiang Zhuang, and Xuemin Hong

School of Information Science and Technology, Xiamen University, Xiamen, Fujian 361005, China

Correspondence should be addressed to Qi Yang; yangqi@xmu.edu.cn

Received 19 March 2013; Revised 23 January 2014; Accepted 18 March 2014; Published 5 May 2014

Academic Editor: Jose Ramiro Martinez-de Dios

Copyright © 2014 Qi Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes an improved Hopfield neural network (I-HNN) algorithm to optimize the slot assignment scheme in wireless sensor networks. The key advantage of the proposed algorithm is to increase the convergence probability under different traffic loads. To achieve this, nodes can adjust their slot demands according to the traffic load, slots number, and demand history. Various aspects of the network performances with the proposed I-HNN algorithm are evaluated via simulation. The results indicate that I-HNN is suitable for wireless sensor networks with dynamically varying traffic. In particular, it can increase the convergence probability and slot utilization under the heavy traffic load.

1. Introduction

A wireless sensor network is a special wireless network without the support of any fixed infrastructure. Due to the lack of central schedulers, the media access control (MAC) protocol is one of the key design challenges for wireless sensor networks. According to the assignment strategy, the MAC protocols can be classified into three categories: random access mechanism such as IEEE 802.11 [1], allocation based mechanism such as FPRP [2], and hybrid mechanism such as TDMA/CSMA [3]. In this paper, we focus on the allocation based mechanism, which has two assignment strategies: static and dynamic. Compared with the static strategy, the dynamic assignment strategy can adapt to the dynamic network topology to achieve collision-free packet transmission and reduce time delay. However, dynamic assignment protocols require nodes to run in parallel and cooperate mutually. Such a lack of centralized control brings further challenges to the design of slot assignment algorithms.

The Hopfield neural network [1] was originally proposed in the neural network literature. It is composed of several interconnected simple neurons and has been widely used in solving nonlinear optimization problems. The Hopfield neural network algorithm has the following four features: (1) it runs on multiple nodes in parallel, (2) it allows multiple nodes to cooperate in a distributed manner, (3) it approaches an optimization goal with a nonlinear method, and (4) it computes rapidly. Consequently, The Hopfield neural network algorithm is naturally suitable for resource assignment (including TDMA assignment) in decentralized systems such as sensor networks.

In the literature, the neural network algorithm has been proposed for resource allocation in cellular radio networks [4–15]. The continuous Hopfield network [6], firstly used in channel assignment, requires very large iteration numbers to converge and is unsuitable for decentralized wireless sensor networks. Funabiki and Takefuji proposed a modified neural network model called the hysteresis McCulloch-Pitts neuron model [5]. This model uses two thresholds to limit the neuron's input. When the input exceeds the upper threshold or is below the lower threshold, the output then changes to 1 or 0, respectively. Otherwise, the output remains unchanged. All neurons update their outputs in parallel, and four heuristics are introduced to increase the probability of convergence to the global minimum. In [4], another modified Hopfield network algorithm was proposed. To accelerate convergence, the algorithm uses different random initialization methods for the neuron's input and then compares the different initialization methods by simulation. References [7, 8] proposed a TDMA slot assignment scheme based on
the neural model in [5] and achieved a collision-free TDMA broadcast mechanism for every node.

The papers mentioned above cannot solve the local-minimization problem. Therefore, they are not applicable to a distributed dynamical network. As a result, several improvements upon the above algorithms were proposed [6–12]. Reference [9] added a negative self-feedback in the neuron model of [5] and proposed a new parallel algorithm for slot assignment in wireless packet radio network. In [10], a dynamic slot assignment algorithm based on the Hopfield network is proposed. Once a node needs to apply for a slot, it randomly chooses a slot number and runs the Hopfield network to judge whether the slot is available according to local contentions. If the slot is not available, the node will subsequently try other slots and update the entire network. Another algorithm called binary recursion neural network [11] proposed a new activation function. To solve the local minimum problem, it unifies the Gaussian distribution to determine the termination threshold. Reference [12] applied a nonlinear function to modify the neuron’s connection weight and increase the convergence rate. However, this algorithm does not achieve the best performance in terms of convergence probability.

Recently, with the development of advanced optimization algorithms such as genetic algorithm and chaotic algorithm, new types of neural networks combined with these new algorithms are sought to increase the convergence probability. One mixed assignment algorithm that combines the genetic algorithm and neural network was proposed in [13]. This algorithm adopts the Hopfield network concept and is able to achieve the maximum throughput. Moreover, a fuzzy neural network algorithm was proposed to solve the channel assignment problem in [14, 15]. This algorithm uses the fuzzy C-means clustering to classify nodes in a network, adds the correlation degree between a node and a cluster to the neuron’s weight, and adopts the Hopfield neural network to update the neurons. The modified algorithms can solve the local-minimization problem at the cost of increased iteration number.

The algorithms proposed in the above literature do not consider the dynamic change of traffics in wireless sensor networks, which, in practice, are likely to have varying traffics. This paper proposes an improved HNN (I-HNN) algorithm to solve the channel assignment problem in wireless sensor networks with dynamic traffic demands. Compared with existing algorithms, the proposed algorithm can increase the convergence probability under different traffic loads.

The remainder of this paper is structured as follows. In Section 2, the neural network model and the slots assignment algorithm are described. The proposed algorithm I-HNN is presented in Section 3. Section 4 shows the simulation results and Section 5 concludes the paper.

2. Hopfield Based Channel Assignment Algorithm

2.1. Slot Assignment in TDMA-Based WSN. We consider a TDMA-based wireless sensor network, in which time is divided into frames and a frame is further divided into slots. TDMA-based communication system has the ability to provide collision-free packet transmission regardless of the traffic load.

Figure 1 illustrates a simple wireless sensor network with 7 nodes operating in the TDMA fashion. To avoid transmission collision, nearby nodes cannot use the same time slot. However, a slot can be spatially reused by nodes separated by more than two hops away. Due to the decentralized and multihop characteristic of wireless sensor networks, the slot assignment algorithm should run in a distributed manner. Hopfield neural network is composed of several interconnected neurons which can run in a distributed manner. Therefore, it is naturally suitable for slot allocation in wireless sensor networks.

2.2. Construction of Hopfield Neural Network in Slot Assignment. In a wireless sensor network with \( n \) nodes and \( m \) slots per frame, the slot assignment problem can be considered as a combinatorial optimization problem consisting of a Hopfield neural network with \( n \times m \) neurons, where each slot per node is treated as a neuron. Every neuron connects with other neurons and updates its output periodically according to others’ output until all neuron’s outputs are stable. In this case, each neuron \((i, j)\) is regarded as a processing unit. If the output of neuron \((i, j)\) equals 1, it means that slot \( j \) is assigned to node \( i \); otherwise, slot \( j \) is unassigned. The construction of the neural network is shown in Figure 2.

The Hopfield neural network above consists of an array of simply connected neurons with binary output. Each neuron has several inputs from other connected neurons and has only one output. The neuron model is shown in Figure 3.

A neuron in the Hopfield neural network can be regarded as a simple processing unit, which is composed of an input unit and an output unit. The input unit \( U_{ij}(t) \) is updated by (1) in each iteration. Consider

\[
U_{ij}(t) = U_{ij}(t-1) + \Delta U_{ij}(t-1).
\] (1)

The update of input \( \Delta U_{ij}(t) \) is a function of other connected neurons’ outputs, namely,

\[
\Delta U_{ij}(t) = g \left[ V_{pq}(t), V_{p'q'}(t), V_{p''q''}(t), \ldots \right].
\] (2)

How to design the function \( \Delta U_{ij}(t) = g[ ] \) will be discussed in Section 2.3. The output of a neuron is given by [5]

\[
V_{ij}(t) = f \left[ U_{ij}(t) \right]
\]

\[
= \begin{cases} 
1 & \text{if } U_{ij} > \text{UTP}, \\
0 & \text{if } U_{ij} < \text{LTP}, \\
\text{not changed} & \text{other.}
\end{cases}
\] (3)

Here, UTP (upper trip point) and LTP (lower trip point) denotes maximum and minimum thresholds, respectively, and the values of UTP and LTP are given in [5]. When \( V_{ij}(t) \) equals one, it means that slot \( j \) is assigned to node \( i \). The task of channel assignment is to ensure that nearby nodes in wireless networks should not be assigned with the same channel.
2.3. The Energy Function. The slot assignment problem in wireless networks must meet the following two constraints: one is that the same slot must not be assigned to nodes within two hops; the other is that the slots assigned to a node should meet the node's demand of slots. Therefore, the slots assignment problem can be transformed to an energy-minimization problem, as formulated in (4). Consider

\[
E_i(t) = \frac{A}{2} \left( \sum_{j=1}^{m} V_{ij}(t) - d_i \right)^2 + B \sum_{j=1}^{m} \sum_{p=1, p \neq i}^{n} \sum_{j=1}^{m} V_{pq}(t) \cdot V_{ij}(t) .
\]  

(4)

Here, \( E_i(t) \) denotes the total neuron energy of node \( i \); \( A \) and \( B \) are constant coefficients. Parameter \( d_i \) denotes the slot demand of node \( i \).

The slot assignment algorithm is to minimize the energy function \( E_i(t) \). When the first term equals zero, it indicates that slots assigned to node \( i \) in a frame equals \( d_i \) and hence satisfies the demand. When the second term is zero, it means that slot \( j \) and slot \( q \) assigned to nodes \( i \) and \( p \), respectively, are not in conflict. The energy function reflects the two above constraints of slot assignment. When the slot assignment is correct, the energy function will output 0. Otherwise, the algorithm will continually iterate until the energy function outputs 0 or the maximum iteration number is reached.

In symmetric neural networks, we have \( w_{ij,pq} = w_{pq,ij} \), where \( w_{ij,pq} \) denotes the connection weight between neuron \( (i,j) \) and neuron \( (p,q) \) [4]. It follows that (4) can be rewritten as follows:

\[
E_i(t) = \frac{A}{2} \left( \sum_{j=1}^{m} V_{ij}(t) - d_i \right)^2 + B \sum_{j=1}^{m} \sum_{p=1, p \neq i}^{n} \sum_{j=1}^{m} w_{ij,pq} \cdot V_{pq}(t) \cdot V_{ij}(t) .
\]  

(5)

It has been proved by [5] that when \( dU_{ij}(t)/dt = -(\partial E_i(t)/\partial V_{ij}(t)) \) and \( V_{ij}(t) = f[U_{ij}(t)] \), then \( (dE_i(t)/dt) < 0 \), which means that the energy function will reduce with increasing iteration. Hence, the function \( \Delta U_{ij}(t) = g[ ] \) is given by \( \Delta U_{ij}(t) = -(\partial E_i(t)/\partial V_{ij}(t)) \). Because the energy is bounded, the algorithm can guarantee that the energy value will eventually approach zero. In a wireless network, the slot assignment problem can be formulated as minimizing the energy function in a corresponding neural network as shown in Figure 2.
As described above, the Hopfield neural network can solve the slot assignment problem in a parallel and nonlinear way. In a network with \( n \) nodes and \( m \) slots to be assigned per frame, there are \( n \times m \) neurons to implement the slot assignment tasks. These neurons can be divided into \( n \) groups, each group contains \( m \) neurons. All the neurons can operate in parallel on \( n \) nodes. The connections among the neurons running in a node can be arbitrarily adjusted according to the slot assignment algorithm. On the contrary, the connections among the neurons running in different nodes should reflect the topology of the network. Therefore, it can be naturally adopted for time-slot allocation in TDMA-based wireless sensor networks. In the next section, we will propose a new slot assignment algorithm based on Hopfield neural network.

### 3. Dynamical Slot Assignment Based on Hopfield Neural Network

In TDMA-based wireless sensor works, the slots in a frame have two types: a control slot at the beginning of a frame, followed by several traffic slots to be allocated. The control slot is used to exchange control information for scheduling the traffic slots. The traffic slots are used to carry the actual payload. To establish a neural network with \( n \times m \) neurons running in \( n \) nodes to carry out the slot assignment task, the control slot is used to exchange information among different nodes.

#### 3.1. Modeling Assumptions and Discussions.

Three key assumptions are made in our system model. Assumption 1 is that all nodes keep perfect timing synchronization. This assumption is the premise of all TDMA-based communication systems and is widely applied in the studies of TDMA systems [2, 3]. Assumption 2 is that the network topology will not change during one iteration interval. In practice, because the iteration time is typically very short, the change of network topology during one iteration time is negligible. Therefore, the network topology can be treated as static within one iteration time. This is a widely used assumption for ad hoc network studies [2]. Assumption 3 is that there is no packet error in the channel. Although communication errors are inevitable in practice, various practical techniques such as forward error coding (FEC) and ARQ can be used to manage the error rate at a desired level. In this paper, we consider the exchange of channel allocation information between nodes. In practice, such sensitive control information is typically well protected with FEC and ARQ schemes to have very low bit error rates. In addition, preliminary simulations show that the main conclusions of this paper remain robust for an error rate below \( 10^{-6} \). The above assumptions enable us to make a tractable analysis in this paper. However, the results should be understood as the theoretical performance upper bounds.

#### 3.2. Energy Function and Motivation Equation.

In a wireless sensor network, nodes can only receive local information within two hops. If the network guarantees that different time slots are assigned to nodes within two hops, it will realize conflict-free transmission.

As shown in (4), the information from other nodes is mainly used to calculate the second term, which guarantees that the slots assigned to different nodes do not conflict. Because nodes more than two hops away can reuse the slots, the calculation of the second term in each node only needs the information of nodes within its neighboring two hops. It follows that the energy function can be modified as follows:

\[
E(t) = \sum_{i=1}^{n} \left[ A \left( \sum_{j=1}^{m} V_{ij}(t) - d_i(u) \right)^2 + B \sum_{j=1}^{m} \sum_{p \in N(i)} w_{ij,pq} \cdot V_{pj}(t) \cdot V_{ij}(t) \right],
\]

where \( N(i) \) denotes the set of node \( i \)'s two hop neighbor nodes and \( d_i(u) \) is the acquired demands of slots. The first term of (6) guarantees that the slots assigned to node \( i \) meet the slot requirements of node \( i \). The second term guarantees that the same slot is not assigned to nodes within two hops. To compute this equation, each node should receive \( V_{pj}(t) \) from other nodes within two hops away. This process can be achieved with an additional control slot at the head of a frame.

Since the connection between two neurons is restricted to the topology of the network, the weight of \( w_{ij,pq} \) is set as follows:

\[
w_{ij,pq} = \begin{cases} 
1, & \text{if } j = q \text{ and } p \in N_i, \\
0, & \text{others}.
\end{cases}
\]

According to \( dU_{ij}(t)/dt = -(\partial E_i(t))/\partial V_{ij}(t) \) proved by [5], the change of the input is

\[
dU_{ij}(t) = -\frac{\partial E_i(t)}{\partial V_{ij}} = -A \left( \sum_{j=1}^{m} V_{ij}(t) - d_i(u) \right) - B \sum_{p \in N(i)} w_{ij,pq} \cdot V_{pj}(t).
\]

Therefore, the motivation equation can be written as follows:

\[
\Delta U_{ij}(t) = -A \left( \sum_{j=1}^{m} V_{ij}(t) - d_i(u) \right) - B \sum_{p \in N(i)} w_{ij,pq} \cdot V_{pj}(t).
\]

The termination condition of the iteration of neural network is that the energy value of the whole network equals zero. This implies that all nodes can acquire the global information about the network's energy by use piggybacking techniques to relay this information while iterating. To
increase the convergence speed, [5] adopts some heuristics. It follows that $\Delta U_{ij}(t)$ can be changed to

$$
\Delta U_{ij}(t) = \begin{cases}
-f \left( \sum_{q=1}^{m} V_{qj}(t) - d_{ij}(u) \right) - \sum_{p \in N(i)} \omega_{ij,pq} \cdot V_{pj}(t) \\
-V_{ij}(t) + C \cdot h \left( \sum_{q=1}^{m} V_{qj}(t) - d_{ij}(u) \right) (1 - V_{ij}(t)) \\
-V_{pj}(t) + C \cdot h \left( \sum_{q=1}^{m} V_{qj}(t) - d_{ij}(u) \right) (1 - V_{ij}(t)) \\
-0 & \text{other,}
\end{cases}
$$

where

$$
f(x) = \begin{cases}
A_{\max} & x \geq A_{\max}, A_{\max} = 5 \\
A_{\min} & x \leq A_{\min}, A_{\min} = -5 \\
x & \text{other,}
\end{cases}
$$

$$
h(x) = \begin{cases}
1 & x < 0 \\
0 & x \geq 0.
\end{cases}
$$

3.3. Slot Demand Model. The slot demand of a node in the $u$th frame is the packet arrival number within a frame. The demand is a random variable and can be expressed as follows:

$$
d_{\text{actual}}(u) = \xi,
$$

Suppose that the packet arrival rate is a Poisson's distribution with mean $\lambda T$, where $T$ is the duration of a frame. We have

$$
P(\xi = k) = \frac{\lambda^k e^{-\lambda T}}{k!}.
$$

The expectation of packet arrival number in a frame of time is $\lambda T$.

The Hopfield slots assignment algorithm introduced above does not consider dynamic changes of the traffic load in the wireless sensor network. When a node has a heavy traffic, it needs more slots. The high slot demands in some nodes may lead to a failed convergence in Hopfield slot assignment algorithm. If these nodes can automatically adjust and reduce their slot demands, the convergence probability can be increased under different traffic conditions.

An appropriate slot demand model should include many factors, such as packet arrival rate $\lambda$, total slots number $M$, and history slot demand of other contention nodes within two hops $d_{ij}(u - v)$, $j \in N_i$ and $v = 1, 2, \ldots, M$, and the slot demands of node $i$ in $u$th assignment should be

$$
d_i(n) = F(\lambda, M, d_j(u - v)).
$$

Here, $F(\cdot)$ is a function. Nodes can confirm the suitable slots demands due to the function. In this paper, we propose a simple function as follows:

$$
d_i(u) = \begin{cases}
d_{\text{actual}}(u) & d_{\text{actual}}(u) < \text{TH}, \text{ aver } < M, \\
\text{TH} & \text{other},
\end{cases}
$$

where $d_i(u)$ is the acquired demands of slots, $d_{\text{actual}}(u)$ is the slots number that nodes actually need, TH is a maximum threshold, and aver is the average history demand of other contention nodes within two hops. TH and aver are represented by the following two formulas:

$$
\text{TH} = \frac{M}{\max(A_j)} \cdot \frac{\sum_{j \in N(i)} A_j}{A_i^2},
$$

$$
\text{aver} = \frac{1}{A_j} \sum_{j \in N(i)} d_j(n - 1),
$$

where $A_j$ is the two-hop neighbor number of node $i$ and $N(i)$ is the set of node $i$’s neighbor nodes.

Threshold TH considers the maximum neighbor number, which is max$(A_j)$, and to let the nodes with lower neighbor, formula (15) also considers the rate of average neighbor number and its own neighbor number.

3.4. Algorithm Description. Consider a wireless sensor network with $n$ nodes and $m$ slots per frame, the slot assignment task can be achieved by constructing a Hopfield neural network with $n \times m$ neurons. Each slot per node is treated as a neuron. In each iteration, the input unit $U_{ij}(t)$ is updated by $U_{ij}(t) = U_{ij}(t - 1) + \Delta U_{ij}(t - 1)$. The function of $\Delta U_{ij}(t)$ is given by formula (10), whose input is mainly determined by other neurons’ output $V_{ij}(t)$ and its slot demand $d_{ij}(u)$. The output of a neuron $V_{ij}(t) = f[U_{ij}(t)]$ is given by formula (3). Every neuron connects with other neurons and updates its output periodically according to others’ output until all neurons’ outputs become stable. If the output of neuron $(i, j)$ equals 1, it means that slot $j$ has been assigned to node $i$; otherwise, slot $j$ is unassigned.

The above slot allocation algorithm runs in the control slot at the beginning of each frame. All the $n \times m$ neurons compute the iteration parallel in $n$ nodes. To realize this process, all the neurons are divided into $n$ groups, each group has $m$ neurons. Neurons $(1, 1)$ to $(1, m)$ are running serially in node 1, and, similarly, neurons $(n, 1)$ to $(n, m)$ are running serially in node $n$. All $n$ groups of neurons run in parallel in $n$ nodes. Each node sends the output $V_{ij}(t)$ of those neurons running in it to its neighbor nodes. Each node also gets the output $V_{pq}(t)$ of other neurons which is running in its two hop neighbor nodes.

The main process of the proposed I-HNN algorithm is as shown in Algorithm 1.

When the Hopfield neural network converges successfully, channel scheduling with no conflict and low time delay can be achieved. The performance of the proposed I-HNN algorithm depends on the frame size, node density, and...
traffic load. The next section of the paper will evaluate the performance of I-HNN.

3.5. Performance Evaluation. The main performance metrics are as follows.

(1) Average convergence probability: the convergence probability is defined as the probability of successful convergence.

(2) Average iteration number: it is the required iteration number when the Hopfield neural network successfully converges. When the network fails to converge, the iteration number is equal to the maximum iteration number.

(3) Slot utilization: the slots utilization is defined as the acquired slot number in one assignment for each node.

(4) Average time delay: the time delay is defined as the average service time for one packet. To derive the time delay, we assume the $N$ nodes in the network are $NM/D/1$ queues, and the time delay for each node is denoted as $D_i$ and can be achieved by Pollaczek-Khinchine [16] formula as follow:

$$D_i = \frac{1}{\mu_i} + \frac{1}{\lambda_i} \frac{(\lambda_i/\mu_i)^2}{2 (1 - \lambda_i/\mu_i)},$$

where the service rate of the station $i$ is $\mu_i = \sum_{j=1}^{M} V_{ij} / M$.

The total time delay is given by

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

4. Simulation Analysis

4.1. Simulation Scenarios. Our simulation aims to analyze the average convergence probability, average iteration number, average acquire slots number for each node, and average time delay, under different traffics, frame sizes, and nodes densities. And we assume that

(1) the packets have a fix length, which is also equal to the length of one slot;

(2) the traffic demands of nodes are independent. Packets of each node arrive according to a Poisson's process with rate $\lambda_i$. Packets are stored in buffers in each station and the buffer size is infinite.

In our simulation setting, the nodes are distributed in a $10 \times 10$ square unit area. In each unit area, a node is randomly generated, as shown in Figure 4. The communication range of each node is 1.5 units. There are 100 nodes and 30 slots in the network, the demands of nodes follow a Poisson's distribution with parameter $\lambda$. The maximum iteration number in each simulation is 100. The simulation results are averaged over 100 simulation runs.

4.2. Simulation Results and Analysis. This paper compares the Hopfield neural network based slots assignment algorithm
in [7] (HNN) and our improved Hopfield neural network slots assignment (I-HNN) from four aspects: average convergence rate, average iteration number, slots utilization, and average time delay.

We first take an example to illustrate how the I-HNN algorithm works. It is assumed that the network traffic is under saturated traffic condition and all nodes require at least one slot. We use the sign "T" to represent that this node is assigned with a current slot. Figure 5 illustrates a simulated network topology with 16 nodes. Table 1 shows that the proposed I-HNN algorithm can assign slots well to achieve better slot utilization.

(1) Average Convergence Probability. The relation between the convergence probability and slot demand is shown in Figure 6. The slot demand \( d \) depends on the traffic load. The simulation result indicates that the convergence probability of the HNN algorithm reduces quickly with increasing slot demand. Although the increase of frame size can improve the convergence probability to a certain extent, the convergence probability is also low with high slot demands. In the proposed I-HNN, the slot demands can be dynamically adjusted according to the traffic demand. As a result, the convergence probability is increased under different traffic loads. Figure 6 also shows that the I-HNN algorithm can increase the convergence probability of the Hopfield neural network under the condition of high slot demands.

(2) Average Iteration Number. We note that for iterative algorithms, the minimum necessary iteration number can serve as an effective indicator of the computational burden. Therefore, Figure 7 also suggests the advantage of I-HNN over HNN in terms of computation burden.

Figure 7 shows the relation between the iteration number and slots demands. The higher the demand is, the more iteration is needed for the Hopfield slot assignment algorithm to converge. As for the HNN algorithm, the iteration number increases with the growth of slot demands and eventually reaches the maximum iteration number. By dynamically adjusting the threshold \( T \), the I-HNN algorithm can reduce the iteration number and accelerate the convergence rate.

(3) Slot Utilization. Slots utilization can be evaluated as the average number of assigned slots for each node. This simulation shows the slots assignment efficiency with different slot demands and frame size.

In conventional reservation based TDMA protocols, the frame size has a significant effect on the network performance. Because the contention nodes within two hops cannot reuse the same slot, the slot assignment will not succeed when the number of slots is less than the number of contention nodes within two hops.

Figure 8 shows that the slot utilization will change when the average slot demands increases. When the average slot demand is low, the slot utilization of HNN is higher than that of I-HNN; this is because the I-HNN can adjust and reduce its slot demands automatically. However, when the average slot demand is high, the slot utilization of HNN is lower than that of I-HNN; this is because higher slot demand may lead to failed convergence of HNN; hence, the slot utilization

![Figure 4: An instance of network topology in a simulation.](image4.png)

![Figure 5: An illustration of a simulated network topology with 16 nodes.](image5.png)

![Figure 6: The relation between the convergence probability and slot demand.](image6.png)

![Figure 7: The relation between the iteration number and slots demands.](image7.png)

![Figure 8: The slot utilization change.](image8.png)

![Table 1: Slot assignment according to the proposed I-HNN algorithm.](table1.png)
when the slot demands are high, the I-HNN algorithm can obviously increase the slot utilization.

Similarly, the network performance is also related to the contention range of nodes. As the contention range expands, the number of contenting nodes within two hops will also increase, leading to a decrease in slot utilization. As shown in Figure 10, the effect of contention range on slot utilization is similar to the effect of frame size.

(4) Average Time Delay. When the packet arrival rates are lower than the packet transmitting rates, all packets can be sent eventually. However, when the packet arrival rates are higher than the packet transmit rates, some packets cannot be sent, which will lead to an infinite time delay. Figure 11 compares the time delay of HNN with I-HNN.

The packet transmit rate is denoted by $\mu_i$ which represents the average packet number that can be transmitted in a frame. The slot demands are simulated as Poisson’s distributed traffic. It is assumed that the network traffic is saturated and every node requires at least one slot. From Figure 10, we can see that when the slot demand is low, the I-HNN algorithm can achieve slightly lower time delays than HNN; this is because I-HNN can dynamically adjust the slot assignment threshold $T$. However, the time delay in I-HNN increases very fast with growing slot demands because the packet transmit rate is kept low by the I-HNN algorithm to avoid contentions.

5. Conclusions

An improved HNN algorithm has been proposed in this paper to solve the channel assignment problem in TDMA-based wireless sensor networks. The proposed algorithm is
adopted from the classic HNN algorithm in the neural network literature and features two major modifications. First, a “two-hop” constraint has been imposed on the output of the network to reflect the nature of local channel contention in wireless sensor networks. Second, a simple but effective mechanism is introduced to dynamically adjust the channel assignment policy according to traffic demands. Through simulation, the performance of the proposed algorithm has been evaluated in terms of average convergence rate, average iteration number, slots utilization, and average time delay. It has been observed that the proposed algorithm is effective in improving all the above aspects compared with the conventional HNN algorithm.

Conflict of Interests
The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments
The authors would like to acknowledge the support of the Natural Science Foundation of China (nos. 61201196 and 61201195), the Natural Science Foundation of Fujian (nos. 2012J05127 and 2012J01292), and the PhD Programs Foundation of Ministry of Education of China (no. 20100121110020).

References
[1] ILMS Committee, “Wireless LAN medium access control (MAC) and physical layer (PHY) specifications,” IEEE Standard, vol. 802, p. 999, 1999.
[2] C. Zhu and M. S. Corson, “Five-phase reservation protocol (FPRP) for mobile ad hoc networks,” in Proceedings of the 17th Annual IEEE Conference on Computer Communications (INFOCOM ’98), vol. 1, pp. 322–331, April 1998.
[3] B. A. Sharp, E. A. Grindrod, and D. A. Camm, “Hybrid TDMA/CSMA protocol for self managing packet radio networks,” in Proceedings of the 4th IEEE International Conference
[4] J.-S. Kim, "Cellular radio channel assignment using a modified hopfield network," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 4, pp. 957–967, 1997.

[5] N. Funabiki and Y. Takefuji, "Neural network parallel algorithm for channel assignment problems in cellular radio networks," *IEEE Transactions on Vehicular Technology*, vol. 41, no. 4, pp. 430–437, 1992.

[6] D. Kunz, "Channel assignment for cellular radio using neural networks," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 188–193, 1991.

[7] N. Funabiki and J. Kitamichi, "A gradual neural network algorithm for broadcast scheduling problems in packet radio networks," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 82, no. 5, pp. 815–823, 1999.

[8] N. Funabiki, N. Okutani, and S. Nishikawa, "A three-stage heuristic combined neural-network algorithm for channel assignment in cellular mobile systems," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 2, pp. 397–403, 2000.

[9] B. Weixing, Z. Tang, J. Wang, and Q. Cao, "An improved neural network algorithm for broadcast scheduling problem in packet radio," *Neural Information Processing—Letters and Reviews*, vol. 9, pp. 23–29, 2005.

[10] A. Dang and S. Zhu, "A compact dynamic channel assignment scheme based on Hopfield networks for cellular radio systems," *International Journal of Communication Systems*, vol. 22, no. 1, pp. 25–35, 2009.

[11] T. Gracián, J. Muñoz, and E. Domínguez, "A recurrent neural network for channel assignment problems in mobiles," *Advances in Artificial Intelligence—IBERAMIA*, vol. 6433, pp. 406–412, 2010.

[12] M. Omid, "A hopfield neural network for channel assignment problem in cellular radio networks," *Computer and Information Science*, vol. 4, p. 116, 2011.

[13] S. Salcedo-Sanz, C. Bousoño-Calzón, and A. R. Figueiras-Vidal, "A mixed neural-genetic algorithm for the broadcast scheduling problem," *IEEE Transactions on Wireless Communications*, vol. 2, no. 2, pp. 277–283, 2003.

[14] Y.-J. Shen and M.-S. Wang, "Broadcast scheduling in wireless sensor networks using fuzzy Hopfield neural network," *Expert Systems with Applications*, vol. 34, no. 2, pp. 900–907, 2008.

[15] X.-Z. Zhang, "Efficient broadcast scheduling based on fuzzy clustering and hopfield network for ad hoc networks," in *Proceedings of the 6th International Conference on Machine Learning and Cybernetics (ICMLC '07)*, pp. 3255–3260, August 2007.

[16] F. S. Hillier, J. L. Gerald, and H. Mark, *Introduction to Operations Research*, vol. 6, McGraw-Hill, New York, NY, USA, 1990.