EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

Link Scheduling for Wireless Mesh Networks Considering Gateway Feature

Chun-Cheng Lin[1], Shu-Huai Chang[1], and Chien-Liang Chen[2,*]

[1]Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu, Taiwan
[2]Department of Innovative Living Design, Overseas Chinese University, Taichung, Taiwan

Abstract

Based on different objectives, a variety of mathematical models for the wireless mesh network (WMNs) exist. Among them, the link scheduling model for WMNs aims at finding a data transmission schedule based on network links so that some objectives on packet transmission are optimized under certain transmission constraints. In this paper, a new WMN link scheduling model with additional information of node locations and gateway designs is orchestrated. An approximate dynamic programming algorithm is utilized for scheduling the new model. Experiment results show that, in addition to maintaining many wireless network characteristics, the proposed scheduling algorithm effectively simulates the result of dynamic programming, and has performances superior to genetic algorithm.

Keywords: Wireless mesh network, link scheduling, gateway, genetic algorithm, dynamic programming, approximate dynamic programming

1 Introduction

The technology of wireless mesh network (WMN) has lots of advances, e.g., its specification in IEEE 802.11s has been released recently. It is well-known that the IEEE 802.11 networks have been studied broadly, e.g., IEEE 802.11 DCF priority access scheme [5], IEEE 802.11 DCF access control with optimal contention window [7], QoS issue in IEEE 802.11 WLANs [6], [8], and so on. With the development of WMNs, how maximum data packets may be transmitted in a given time to satisfy client requirements is related to the routing of network nodes and scheduling of paths between nodes. Among the various WMN model, the link scheduling model programs the links that transmit data packets between nodes to achieve certain objectives. The works in [10], [16], [20] divided time and multiplexed by routers in the WMNs model. The work in [9] described the features of WMN signals, and it may be known that the mutually interfering transmission signals between router nodes play an important role in the scheduling model. In addition, the works in [11], [19] discussed the data capacity of the router, and such nodes that may receive and transmit data packets are known to have packing constraints. Under the constraints of the programming time and node transmission mentioned above, many scheduling models with varying emphasized features have constructed their basic concepts and developed various objectives. Among these, the most common objective is to maximize data packet transmission [11], [17], [18]. Such models perform scheduling by some novel algorithms, such as dynamic programming. However, they usually did not include any gateway design and only focused on the reduction of data packet packing at nodes. Furthermore, the design of programming information placed on the links differs from actual situations in these models. Different from the setting in [13] that considered WMN consisting of only mesh routers and mesh clients, a few WMN models have considered gateway designs [1], [12], and achieved the objective of maximizing the data packets received by the gateway by scheduling with genetic algorithm. Although fair performance is obtained, their nodes do not have the function of receiving randomly arriving data packets.

Among the numerous literature of various link models, the scheduling algorithms used by designers are not quite the same, and performances to be evaluated and objectives to be achieved are also different, which includes genetic algorithm [12], dynamic programming using optimization principles, and other heuristic algorithms [17]. It is worthy to note that, among these heuristic algorithms, approximate dynamic programming (ADP) has saved much more time as compared with dynamic programming and achieved better performances as compared with other heuristic algorithms by predicting the calculation results of dynamic programming by simulation. In summary, the design of a new model conforming to features of actual WMNs and allowing different algorithms originally used in different scheduling models to compare performances is worthy of discussion, and the research emphasizes on the effect of the added features on the execution and comparison of each algorithm, the simulation effect of approximate dynamic programming, and whether the algorithm maintains the scheduling performances after expanding the extent of the model.

In this work, we have explored various different WSNs and discovered their distinct highlighted features such as the link transmission direction, node coordinates design, and various constraints; the model objectives also often differ, such as focusing on reducing the node data packing, focusing on maximizing the received packets by the gateway, or maximizing the open times of transmission links between nodes. Due to the variety of types, a model of a single type may lack certain characteristics. This work seeks to include as many WSN features as possible in the new model, such as the time slot concept of dividing programming time, the numerous constraints and

---

[*] Corresponding author. E-mail: clchen@ocu.edu.tw

randomly distributed node coordinates encountered during link transmission, and the design of the gateway location scarcely considered in conventional models. The new model adds gateway designs in the objective and achieves the objective of transmitting large amounts of packets to the gateway by maximizing the node transmission packet amount. In the discussion of the scheduling algorithm, the algorithms used for scheduling the same wireless network model are examined to ensure that the algorithms are compared under the same performance index and model designs. After selecting a superior algorithm, it is used for the comparison of scheduling performances of the designed WSN model.

Note that although the work in [14] has jointly studied routing, scheduling, and variable-width channel allocation in WMNs, their model is much different from ours.

The following sections are arranged as follows: Section 2 introduces the WSN link scheduling model and the genetic algorithm, the concept of conventional dynamic programming and approximate dynamic programming; Section 3 presents the new model and explains how the conventional dynamic programming and approximate dynamic programming are used for solutions; Section 4 gives the experiment results and analysis; Section 5 is the conclusion.

2 Related Work

Wireless mesh network link models proposed in the literature are discussed in this section.

2.1 Scheduling model of Spatial Time Division Multiple Access

A network structure usually is constructed by a group of sets of nodes that may connect to each other, where the nodes communicate by transmission links to transmit data packets. Long-term cautious programming is required to program the link behavior to achieve the objective of configuring the network, and thus it is not surprising that numerous research papers related to such topics have been published [3], [4], [12]. Differences in optimization objectives exist among such researches due to different network node data transmission characteristics and topology structure; however, common features of the various topics may still be deduced. In addition to solving issues of data packet transmission between nodes and their neighboring nodes, the distribution of transmission of data packets also needs attention; since each link has a fixed packing amount of data transmission and only a fixed maximum amount of data packets may be transmitted in a fixed time, the packing of other links have to be decreased to increase the packing of a certain transmission link which is anticipated to result in the reduction in the overall performances [11], [19]. Therefore, Randolph proposed the Spatial Time Division Multiple Access (STDMA) [20], and the nodes of which are fixed geographically and the programming time are divided into multiple time slots, so that data packet transmissions in the same time slot may be deemed as proceeding simultaneously. Each time slot in STDMA is assigned to a group of sets of data packet transmission behavior.

2.2 Signal interference types in the scheduling model

When actual data packet transmission occurs, it is inevitably interfered with by other channels transmitting data at the same time. The most suitable model to describe such a mechanism among the numerous literature is the signal interference model [9], [12], [17], which is mostly adopted by the references mentioned in this paper. When routers have transmission abilities between each other, there exist quantifiable transmission energy, and an index for evaluating the level of interference is the ratio of the transmission energy to the total transmission energy of routers transmitting data in the network at the same time; usually, such ratio has to exceed a default signal interference ratio threshold for proper transmission to occur.

2.3 Objectives, constraints, and decision variables of the link scheduling model

Common scheduling objectives of wireless mesh networks may be divided into two types. The first type maximizes the open times of transmission links [12]. Under the above STDMA structure, each link may open at any time slot, and the more times it opens in the programming time, the higher the efficiency of the transmission of the data packets at the node. However, many transmission constraints result from the opening of the links, and thus the scheduling model of this type is the process of opening the link as much as possible under transmission constraints to effectively transmit packets.

The second type of objection emphasizes maximizing the data packet transmission amount [9], [17], [18]. In order to observe the status of packet transmission amount, cost information is added to the nodes [17], [18] where the cost is configured so that higher packet transmission amounts correspond to network statuses with lower total cost. Under such a situation, the model has the following formula:

$$Min(\sum_{t=0}^{T} H_t)$$

where H_t represents the total network cost at programming time t; some models with gateway designs may require these nodes to transmit large amounts to the gateway.

Common restraints in scheduling models may be categorized into node packing restraints [11], [19] and signal interference restraints [1]; the former refers to the inability of a node to carry out too many scheduling operations at one time, such as receiving and transmitting packets simultaneously, transmitting data simultaneously to more than two nodes, or receiving data packets simultaneously from more than two nodes; the latter refers to some transmissions with weaker signal intensities being affected or unable to proceed due to the mutual interference of signals when links transmit data packets in the overall network model. Decision variables in wireless mesh network models usually are the open statuses of the links at each stage and are thus usually designed as Boolean variables, including parameters such as time or node or link number; the value is 1 when open and 0 when close. Link scheduling of wireless mesh networks is the process of the determination of decision variable values of each stage by scheduling algorithms in a model comprising various transmission constraints to achieve the objective required by the designed model.

Algorithms applied in various wireless mesh networks are discussed in the following. The purposes of such algorithms are all to effectively transmit packets by network link scheduling, which includes optimal dynamic, ADP optimal, ADP heuristic, packing heuristic, and genetic algorithm [9], [12], [17], [18].

2.4 Dynamic programming scheduling

Papadaki has used dynamic programming to solve wireless mesh network scheduling problems [17], [18]. The logic of the programming is to collect packet holding costs from network nodes and transmit data packets as many as possible from the nodes by minimizing the holding cost. The dynamic programming equation mainly constitutes of two parts, wherein the first part is the optimization formula $V_t(S_t)$ in below:

$$V_t(S_t) = \min_X[C_t(S_t, X_t) + V_{t+1}(S_{t+1})]$$

where t is the programming time slot, S_t is the status of each node of the network at programming time slot t, $C_t(S_t, X_t)$ represents the cost resulting from decision variable X_t made at programming time slot t, and $V_t(S_t)$ represents the minimum total cost achievable from the programming time slot t to the end of the programming time under the network status S_t. The calculation method is to minimize $C_t(S_t, X_t)$ and the optimization formula of the next stage V_{t+1} (S_{t+1}) via the decision variable X_t, and the value of the formula when t is 1 is the optimal value of the objective of the link model. The second part is the formula of the transfer of the network node status of each stage; when the optimization formula $V_t(S_t)$ determines the optimal decision variable of that stage, such decision variables will change the network status of the next stage via this formula. The formula additionally includes randomly arriving data packets as below:

$$s_{t+1} = (s_t - r\, x_t + a_t)^+$$

where r is the number of data packets transmitted by the transmission link per unit time, x_t is a Boolean variable and indicates that data packet transmission occurred in the link in the programming time if 1, a_t is the number of data packets arriving at the link in the programming time, and s_t is the number of the queuing data packets. It is worthy to note that, in this recursive formula, the location of saved data packets and location of randomly arrived data packets are placed on transmission links and thus it is impossible to determine which node is the source of the data packets of the link or which node is the destination (in fact, such data packets are not transmitted to nodes but to other links in the author's mathematical model). Although the decision variables x_t indicates the start and end nodes of the link, yet since the source of the data packets received by other links are collectively represented by the random variable a_t, the relationship between the generation of the random variable and the data packets received by the link becomes obscured. In other words, it is only known that the number of data packets transmitted by the link is rx_t, yet the number of packets received by a node from the link is not certain to be rx_t.

2.5 Scheduling by ADP or other heuristic algorithms

The works in [17], [18] pointed out that under requirements of conventional dynamic programming algorithms, the extent of each stage shall exponentially increase with the increase in the number of nodes and result in difficulties in the actual execution of conventional dynamic programming. One way to handle such issues is to design approximate mathematical models to estimate the original function in a function parameter environment the same as the original problem. In conventional dynamic programming, the problem includes a cost-oriented function f which usually is the optimization function of the conventional dynamic programming and must include all possible extents of the problem and provide recursive functions according to which each stage status will transfer; an approximate formula is a function complying with $\breve{f}(r) \approx f$, and since the formula should correspond to the characteristics of the original formula to yield an approximate result, the designer must be experienced with the problem in practice and able to provide coarse information of the original formula. After providing sufficient information of the original formula, an effective equation to calculate suitable parameters is required as below:

$$\breve{f}(r) = \sum_{k=1}^{K} r_k \theta_k$$

where k represents the parameter subscript in the problem space, r_k is the basic function under the parameter subscript k, and θ_k is the weighting of the basic function k. It was mentioned in the previous subsection that Papadaki [17], [18] used dynamic programming to solve the scheduling problem of wireless network models and heuristic algorithms such as ADP optimal, ADP heuristic, and packing heuristic was adopted to seek to compare with each other for the reduction of computing complexity. The results showed that the performances of ADP optimal are second only to dynamic programming algorithm and largely saves computing time; therefore, such algorithm shall be used in this research and performances will be compared with dynamic programming to observe if effective simulation may be performed as in conventional scheduling models.

2.6 Link scheduling for wireless network by genetic algorithm

Badia et al. [2] proposed using genetic algorithms to solve the programming problem of wireless mesh networks. When the programming begins, a population is generated randomly or heuristically, and the algorithm operates in repeated processes on generations in practice; a new population $P(t+1)$ is generated from the old population $P(t)$ in every new generation, and the population improves its internal values by genetic operators imitating the evolution mechanism of nature. The evolution mechanism selects superior individuals from the old population to crossover or mutation by a lower probability, and the selection operator handles the work of the selection of superior individuals; the elites in the individuals of the old population usually survive and are preserved during evolution, while other selected individuals are reassembled to generate offspring by the crossover operator and the offspring is preserved to the next generation. After the optimization objective is decided, the crossover and mutation mechanisms are usually designed to seek optimal solutions under confinements, even though it is not guaranteed that the optimal solution may be sought. Despite this, the genetic algorithm is often treated as the method for seeking "sufficiently good" solutions in practice due to the fast execution; in fact, as long as the given evolution generations are large enough, a genetic algorithm usually yields superior solutions. In work in [2], genetic algorithm effectively schedules wireless mesh network models; however, the solved scheduling model design differs from that of Papadaki, and the performance indices are not the same, and thus may not be compared with dynamic programming as to their scheduling schemes. Therefore, the performances of genetic algorithm, dynamic programming, and ADP are all tested in the new wireless mesh network model designed in this research.

3 Our Method

This section describes how the wireless mesh network model version is redesigned in this research and how the scheduling algorithms are used. The link relationship formed by every node with neighboring nodes is designed according to the mesh topology network structure, and data packets are gradually transmitted from the start to the end as the programming schedule proceeds; each programming unit carries out a dynamic programming algorithm under the prerequisite of meeting all constraints.

3.1 Symbol definition

All variables are defined as follows:

- N: Total number of topology nodes
- E: Total number of links in network
- T: Total programming time slots
- n: nth node in network
- e: the eth link in network
- t: the tth programming time slot
- γ: Signal interference ratio threshold
- $r_{(i,j)}$: Set of nodes that may transmit data packets to node (i,j)
- $s_{(i,j)}$: Set of nodes that may receive data packets from node (i,j)
- $W_{(i,j)}$: Number of transmission links on node (i,j)
- $Q_{t,(i,j)}$: Number of queuing data packets at slot t at node (i,j)
- $Z_{t,(i,j),w} = \begin{cases} 1, & \text{if the } w\text{th transmission link has not} \\ & \text{transmitted before slot } t \text{ at node } (i,j); \\ 0, & \text{otherwise.} \end{cases}$
- $Z_{t,(i,j)} = \begin{cases} 1, & \text{if all the transmission links have not} \\ & \text{transmitted before slot } t \text{ at node } (i,j); \\ 0, & \text{otherwise.} \end{cases}$
- $S_{t\,(i,j)}$: Status of node (i,j) at slot $t = \{Q_{t,(i,j)}, Z_{t(i,j)}\}$.
- $G_{t,(i,j),w}$: Random signal intensity of wth transmission link at slot t at node (i,j).
- $R_{t,(i,j),w}$: Random transmission rate of wth transmission link at slot t at node (i,j).
- $h_{(i,j)}$: Data packet holding cost of node (i,j).
- $p_{(i,j)}$: Transmission link penalty cost on node (i,j).
- $q^k_{t,(i,j)}$: Approximate holding cost slope of the kth simulation of node (i,j) at time slot t.
- $z^k_{t,(i,j)}$: Approximate penalty cost slope of the kth simulation of node (i,j) at time slot t.
- $X_{t,(i,j),w} = \begin{cases} 1, & \text{if the } w\text{th transmission link transmits} \\ & \text{at slot } t \text{ at node } (i,j); \\ 0, & \text{otherwise.} \end{cases}$

3.1.1 Improved wireless mesh network link scheduling model

In the wireless mesh network link model, common objectives are maximizing the data packet transmission amount and maximizing the open times of transmission links; the common objective of such objectives is to effectively transmit maximum data packets in a given time limit. The transmission link scheduling should be considered to maximize the efficiency of data packet transmission when handling such problems and a wireless mesh network link model that conforms to the topology and signal interference characteristics of the wireless mesh network needs to be designed first. In the model of this research, the time flow mechanism is in accordance with time division multiple access (TDMA), and the approach is to divide programming steps according to time into different time slots; after determining the total programming time slot T, t represents a certain time slot.

3.1.2 Programming information on nodes

Conventional wireless network scheduling model places programming information on links, which is different from reality and easily confuses the affiliation of the data packets. To solve such issues, the present research moves the programming information to the nodes. First, the data packet number information at router nodes is defined, and (i,j) represents the coordinates of the node on a two-dimensional map, and $Q_{t,(i,j)}$ is the number of queuing data packets to be transmitted at slot t at node (i,j) and $W_{(i,j)}$ is the number of transmission links on node (i,j). Subsequently, random signal intensity is defined for the links through which the node transmits data to neighboring nodes, denoted as $G_{t,w,(i,j)}$, and the transmission link also has a changing data packet transmission rate following the signal intensity, denoted as $R_{t,w,(i,j)}$. Further, $r_{(i,j)}$ denotes the set of nodes that may transmit data packets to node (i,j) and $s_{(i,j)}$ denotes the set of nodes that may receive data packets from node (i,j); each node may be the basic programming unit.

3.1.3 Objective considering gateway feature

Conventional wireless network scheduling models usually do not include gateway designs, and thus designed objectives in such link models only focus on how to largely transmit data packets regardless of whether these packets were transmitted to the gateway; common forms are maximizing the data packet transmission amount and maximizing the open times of transmission link. The link model in this research combines the features of the objectives of the above models and adds a feature that transmits packets to gateways; by minimizing the holding cost on nodes, the nodes may transmit packets extensively. As the nodes in the model of this research transmit packets unilaterally towards the gateway (the actual construction method is given in detail in the experiments in Section 4), the effect of extensive packets arriving at the gateway may be achieved by maximizing the packet transmission amount. The objective is given as the following:

$$Min(\sum_{t=0}^{T}\sum_{(i,j)=1}^{N}Q_{t,(i,j)}h_{(i,j)})$$

where $h_{(i,j)}$ is every possible kind of cost of the node (i, j), and the point of the objective is to minimize the total cost of all nodes after executing all time slots. X_t is the decision variable and is given in detail as:

$$X_{t,(i,j),w}=\begin{cases}1, & \text{if the } w\text{th transmission link transmits at} \\ & \text{slot } t \text{ at node } (i,j); \\ 0, & \text{otherwise.}\end{cases}$$

3.1.4 Constraints in link model

After considering the characteristics of the wireless mesh network under actual situations, the following constraints appear:

1. Flow Constraints

$$Q_{t+1,(i,j)}=\sum_{(p,q)\in s_{(i,j)}}[Min(Q_{t,(p,q)},\sum_{w=1}^{W_{(p,q)}}X_{t,(p,q),w}X_{t,(p,q),w)}] \\ +Max(0,Q_{t,(i,j)}-\sum_{k=1}^{K_{(i,j)}}X_{t,(i,j),w}R_{t,(i,j),w}), \\ \forall(i,j)\in N,\forall t=0,\ldots,T-1 \tag{1}$$

This constraint ensures that when the number of queuing data packets of node (i,j) is smaller than the number of packets that may be transmitted by the link per unit time, only the number of presently queuing data packets may be transmitted at maximum, and when the number of queuing data packets of node (i,j) is larger than the number of packets that may be transmitted by the link per unit time, only the number of packets that may be transmitted by the link per unit time may be transmitted at maximum.

$$\sum_{(i,j)\in N}Q_{0,(i,j)}=\sum_{(i,j)\in Y}Q_{T,(i,j)} \tag{2}$$

This constraint ensures that every data packet at the start node may be transmitted to the end node; however, the following experiments will not adopt this constraint, but the ratio of data packets arriving at the gateway is used as the performance index.

$$Q_{0,(i,j)}=0,\forall(i,j)\in Y \tag{3}$$

This constraint ensures that the gateway does not have any packet at the beginning; i.e., it is not treated as the start of data transmission.

$$X_{t,(i,j),w}\le 0,\forall(i,j)\in Y,\forall t=0,\ldots,T-1 \tag{4}$$

This constraint ensures that the gateway does not transmit any data packet, i.e., a data packet does not leave once it enters.

2. Direct Compatibility Constraints

$$\sum_{(p,q)\in s_{(i,j)}}\sum_{k=1}^{K_{(p,q)}}X_{t,(p,q),w}\le 1,\forall(i,j)\in N,\forall t=0,\ldots,T-1 \tag{5}$$

In a certain programming time slot, no node may simultaneously receive data packets from more than two nodes.

$$\sum_{w=1}^{W_{(i,j)}}X_{t,(i,j),w}R_{t,(i,j),w}\le 1,\forall(i,j)\in N,\forall t=0,\ldots,T-1 \tag{6}$$

In a certain programming time slot, no node may simultaneously transmit data packets to more than two nodes.

$$\sum_{(p,q)\in s_{(i,j)}}\sum_{w=1}^{W_{(p,q)}}X_{t,(p,q),w}+\sum_{w=1}^{W_{(i,j)}}X_{t,(i,j),w}R_{t,(i,j),w}\le 1, \\ \forall(i,j)\in N,\forall t=0,\ldots,T-1 \tag{7}$$

In a certain programming time slot, no node may simultaneously receive and transmit data packets.

3. Interference Compatibility Constraints

$$G_{(i,j)}\ge\gamma\sum_{k\in s_{(i,j)}/(i,j)}G_{kj}(\sum_{l\in r_k}X_{t,(k,l)}+X_{t,(i,j)}-1),\forall(i,j)\in N, \\ \forall t=0,\ldots,T-1 \tag{8}$$

3.1.5 Use ADP for scheduling in the improved model

After introducing the proposed improved model, an algorithm that effectively executes link scheduling is required. From literature, it is known that the performances of ADP is superior to other heuristic algorithms used in WMN link scheduling, and it largely reduces computing amount and achieves performances similar to the optimal algorithm of dynamic programming by simulation methods; therefore, this research adopts this algorithm. Since the wireless mesh network link model is improved, the programming information location and received packet sources in this research are different from the literature.

3.1.6 Dynamic programming operation logic

Before introducing the algorithm logic of ADP, how dynamic programming logic may be used to solve the problems of the redesigned WMN model should be understood first. In the model of the previous subsection, the focus of programming has already been placed on network nodes. In order to facilitate the execution of dynamic programming and enhance link transmission efficiency, more transmission information is added to transmission links on the nodes. First let Z_t represent the transmission status of the mesh network at slot t, where $Z_{t,(i,j),w}$ is a binary variable defined as:

$$Z_{t,(i,j),w}=\begin{cases}1, & \text{if the } w\text{th transmission link has not} \\ & \text{transmitted before slot } t \text{ at node } (i,j); \\ 0, & \text{otherwise.}\end{cases}$$

then $Z_{t,(i,j)}$ may be defined as:

$$Z_{t,(i,j)}=\begin{cases}1, & \text{if } \sum_{w=1}^{W}Z_{t,(i,j),w}=W_{(i,j)}; \\ 0, & \text{otherwise.}\end{cases}$$

When not all transmission links of a node have transmitted data before the programming ends, the node receives a penalty because since link transmission does not create cost in this model, all links should transmit as much as possible without violating constraints. Subsequently, let S_t be the status of the wireless network at slot t, which equals $\{Q_t, Z_t\}$, and includes queuing data packet number and penalty information of the node. The status of each stage of the WMN may be understood via S_t, which may be referred to in the dynamic programming optimization process and the status transfer of each stage.

Subsequently, the cost is collected from the information in S_t, and the purpose of maximizing the data packet transmission amount is achieved by minimizing these costs. Let $h_{(i,j)}$ represent the data packet holding cost of node (i,j), and $p_{(i,j)}$ represent the penalty cost on node (i, j) for any unused transmission link. Up to now, every single node contains four kinds of different information (not including link information on the node), wherein the number of queuing data packets Q_t is basic information in the link model, and holding cost h, penalty cost p_z, and variable Z for judgment of whether to collect penalty are new information added to the original link model for the convenience of executing dynamic programming. Various costs are denoted as $C_t(S_t, X_t)$, and the collection method is in below:

$$C_t(S_t, X_t) = C_t(Q_t, G_t, X_t) + C_T(Z_T)$$

$$C_t(Q_t, G_t, X_t) = Q_t h$$

$$C_T(Z_T) = Z_T p_z$$

Subsequently, the recursive formula for the status transfer of each stage is defined; the number of queuing data packets and penalty of transmission links of nodes are included in each of the previously defined stage statuses, and the recursive formula for the number of queuing data packets may be given as:

$$Q_{t+1,(i,j)} = Q_{t,(i,j)} + \sum_{(p,q)\in s_{(i,j)}} \sum_{w=1}^{W_{(p,q)}} X_{t,(p,q),w} - \sum_{w=1}^{W_{(i,j)}} X_{t,(i,j),w} R_{t,(i,j),w} + A_{t,(i,j)}, \quad \forall t < T$$

From the equation it can be observed that the number of queuing data packets of the node in the next stage equals the number of queuing data packets in this stage plus the number of received data packets from neighboring nodes and the number of randomly arriving data packets minus the number of data packets transmitted from neighboring nodes close to the gateway. The recursive formula for penalty may be given as:

$$Z_{t+1,(i,j)} = (1 - X_{t,(i,j)}) Z_{t,(i,j)}, \forall t < T$$

Next, the optimization formula $V_t(S_t)$ is designed for each time slot to determine the decision variable of each time slot, and is given as:

$$V_t(S_t) = V_t(Q_t) + V_T(Z_T)$$

This formula takes into account both the holding cost and penalty cost, which includes recursive logic with reversed time and represents all generated costs from programming time t to programming end time T and is given in detail in below:

$$V_t(S_t) = Min_{X_t} C_t(Q_t, G_t, X_t) + V_{t+1}(S_{t+1}), \forall t < T$$

$$V_T(Z_T) = C_T(Z_T) = Z_T p_z$$

$$V_{T+1}(S_{T+1}) = 0$$

Due to the dynamic programming using reverse-time computing logic, the optimization function is set to 0 for programming time T+1, and the optimization function is recalculated for each stage in decreasing time slot order; the calculation is complete when the time slot decreases to the lowest value.

To gain insight into reverse-time programming of dynamic programming, all the node information S_t that might be generated in all stages are listed first, and then the optimal decision variable is sought by the optimization function starting from time T; since the optimization function $V_t(S_t)$ in a new stage includes the optimization function V_{t+1} in the previous stage, the decision of the new stage is made by taking into account the optimization function of the previous stage. In other words, the decision variable of this stage will generate a certain node status S_{t+1}, and the corresponding optimization function $V_{t+1}(S_{t+1})$. When the time slot reduces to the lowest value 1, all possible node statuses will correspond to a function value, including slot 1 to T, and the minimum value is the desired minimum cost value. All stage node statuses at the end of programming may be traced back via decision variables at each stage, as shown with the red arrows. However, since all possible node statuses of all stages must be listed first, the computing amount may rapidly increase if the network extent is overly large; thusly, an algorithm that reduces computing amount needs to be adopted for handling larger computing amount.

3.1.7 Reducing computing amount with ADP

ADP uses simulation to approximate dynamic programming results and is a heuristic algorithm; as long as the parameters are appropriately designed and the simulation time is sufficient, performances similar to dynamic programming may be achieved. Improved link models were used in dynamic programming units in above, and there already are four kinds of information at each node, and two other kinds of information require to be added in order to use ADP algorithm, which is the approximate holding cost slope to simulate holding cost $q_{t,(i,j)}^k$ and the approximate penalty cost slope to simulate penalty cost $z^k_{t,(i,j)}$. The definitions are below:

$q_{t,(i,j)}^k$ = Approximate holding cost slope of the kth simulation of node (i,j) at time slot t.

$z_{t,(i,j)}^k$ = Approximate penalty cost slope of the kth simulation of node (i,j) at time slot t.

Up to now, there are six kinds of information included in the nodes of the model, and except for the number of queuing data packets that belong to information of the conventional linear WMN link model, all other information is added for the scheduling algorithm of the model. The approximate optimization function $\tilde{V}_t$ with approximate cost features are redesigned in detail below:

$$\tilde{V}_t(Q_t, Z_t) = Min_X [C(Q_t, G_t, X_t) + \hat{V}_{t+1}(Q_{t+1}, Z_{t+1})]$$

$$\hat{V}_t(Q_t, Z_t) = q^T Q_t + z^T Z_t$$

where $\widehat{V}_t$ is the prediction of the optimization function V_t which stands for the lowest predicted cost from time slot t to time slot T, and by using the approximate holding cost slope $q_{t,(i,j)}^k$ and the approximate penalty cost slope $z_{t,(i,j)}^k$, $\widehat{V}_t$ may be used to predict future penalty cost and holding cost. The calculation of $q_{t,(i,j)}^k$ and $z_{t,(i,j)}^k$ are as below:

$$\overline{q}_{t,(i,j)}^k = \tilde{V}_t^K(Q_t^k + e_{(i,j)}, Z_t^k) - \tilde{V}_t^k(Q_t^k, Z_t^k)$$

$$\bar{z}^k_{t,(i,j)} = \begin{cases} \tilde{V}^k_t(Q^k_t, Z^k_t + e_{(i,j)}) - \tilde{V}^k_t(Q^k_t, Z^k_t), & \text{if } Z^k_{t,(i,j)} = 0; \\ \tilde{V}^k_t(Q^k_t, Z^k_t) - \tilde{V}^k_t(Q^k_t, Z^k_t - e_{(i,j)}), & \text{if } Z^k_{t,(i,j)} = 1. \end{cases}$$

From the above equation it may be seen that $q^k_{t,(i,j)}$ represents the optimization function value increase in time slots $t \sim T$ when one unit of the data packet is increased on node (*i*, *j*); $z^k_{t,(i,j)}$ represents the optimization function value difference between links that have never transmitted data packets and links that have transmitted data packets on node (*i*, *j*); $e_{(i,j)}$ is the decision variable operator only operating on node (*i*,*j*); $\bar{q}$ and $\bar{z}$ are the approximation operators not yet smoothed with the previous experiment. Subsequently, a smoothing process is applied to ensure convergence, and the process is described as:

$$q^k_t = \delta^k \bar{q}^k_t + (1 - \delta^k) q^{k-1}_t$$

$$z^k_t = \delta^k \bar{z}^k_t + (1 - \delta^k) z^{k-1}_t$$

where δ^k is the smoothing coefficient of the *k*th experiment, and the smoothing coefficients from the first to the *k*th experiment are required to satisfy the three following conditions:

1: $\sum_{k=0}^{\infty} \delta^k = \infty$

2: $\sum_{k=0}^{\infty} (\delta^k)^2 < \infty$

3: $0 < \delta^k < 1$

It may be observed that the estimation method of approximate cost is to try to increase the number of data packets or to open links to see how much change is done on the optimization value, and the changed value is the updated approximate cost after smoothing. In this research, the initial value of the approximate cost in the first simulation is 0, and the updated approximate cost after the calculation is used in the next simulation; after multiple simulations, the predicted value of the approximate cost shall come closer to the actual situation (i.e., dynamic programming results). Therefore, the ADP uses normal time algorithm and does not list all the possible node statuses S_t but predicts the best decision in the present stage by using approximate slope with the progress of the time slots, and the best decision in the present stage will transfer the node status of the present stage to the node status of the next stage until the time slot ends and then proceed with the next simulation.

3.1.8 Executing ADP

The following describes how the ADP algorithm may be executed.

Step 1. Initialize the approximate cost that affects the slope variation of the optimization function and let q^0_t and z^0_t be 0.

Step 2. Initialize programming time and testing times and let $k = 1$, $t = 0$.

Step 3. Generate random variables on all nodes G^k_t.

Step 4. Determine decision variable $X_{t,(i,j)}$ by the approximate optimization function $\tilde{V}_t(Q_t, Z_t)$:

If $t < T - 1$:

$$Min_X C(Q^k_t, G^k_t, X_t) + (q^{k-1}_t)^T Q^k_{t+1} + (z^{k-1}_t)^T Z^k_{t+1}$$

If $t = T - 1$:

$$Min_X C(Q^k_t, G^k_t, X_t) + (p_z)^T Z^k_{t+1}$$

Step 5. Determine all node statuses of the next stage by the recursive formula

Step 6. Update q^k_t and z^k_t as q^{k+1}_t and z^{k+1}_t by the optimization function $\tilde{V}_t(Q_t, Z_t)$.

Step 7. If $t < T - 1$ then let $t = t + 1$ and return to Step 3; otherwise go to Step 8

Step 8. If $k < K$, then let $k = k + 1$ and return to Step 3; otherwise, stop the algorithm.

4. Experimental results and analysis

In this section, different algorithms are used in the scheduling of the new network model, and the same performance index is used for comparison. In previous literature, the comparison results of dynamic programming, ADP algorithm, ADP optimal algorithm, dynamic programming heuristic algorithm, and packing heuristic algorithm show that dynamic programming has the best performance but is not beneficial for large scale computing, while ADP optimal algorithm has second-best performances but largely saves computing time. Therefore, this research uses these two algorithms along with genetic algorithm to perform scheduling on the new model to compare the results for the purposes of observing whether ADP still effectively reduces computing amount and simulates dynamic programming results in the improved link model and comparing the performances of genetic algorithm and ADP. In the following experiments, link scheduling performances of the three algorithms in a small scale WMN model are compared; as for large scale network link scheduling experiments, since possible statuses of hundreds of nodes at each stage cannot be completely listed, only performances of ADP and genetic algorithm will be compared.

4.1 Performances of the algorithms in a small scale WMN model

Figure 1 shows the network topology generated in the present experiment having 8 nodes and 11 links.

Parameter settings in the experiment are as below:

- Total number of time slots: 9
- Arriving amount and their probability of random packets at single node: 0: p = 0.4792, 1: p = 0.25, 2: p = 0.125, 3: p = 0.0833, 4: p = 0.0625
- Sample number: 50
- ADP simulation times: 20
- ADP smoothing coefficient: 0.3
- Performance index: Ratio of number of data packets arriving gateway to the total number of data packets

where the arriving amount and their probability of random packets at a single node refer to the corresponding probability of randomly arriving packet amount at each node, which has an expected value of 1 in a time slot. Therefore, it may be inferred that when the programming time with a total of 9 slots ends, the total data packet expectation value of this network topology with 8 nodes is 8 × 9 × 1=72. The ADP simulation times and smoothing coefficient are suitable values selected after multiple tests. To compare the three different algorithms, the ratio of the number of data packets arriving gateway to the total number of data packets is uniformly selected to be the performance index; Figure 2 and 3 in below shows the results of experiments.

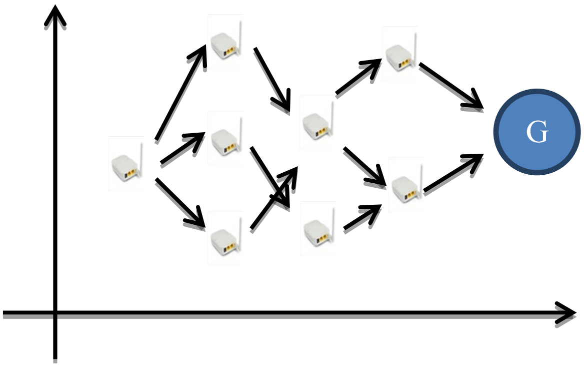

Figure 1. Topology of small scale WMN experiment

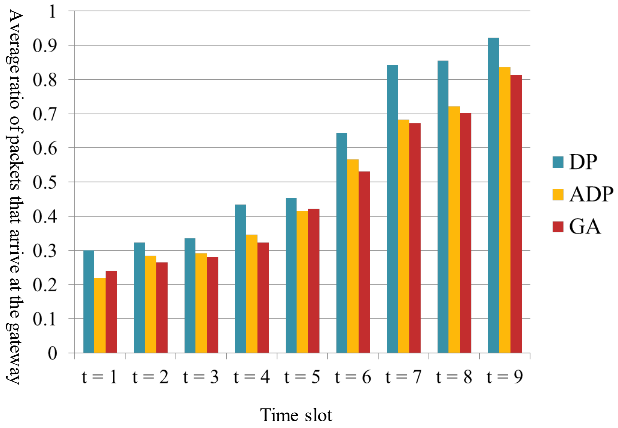

Figure 2. Schematic of scheduling performance of the algorithms

Figure 2 in above shows the ratio of the three different algorithms transmitting data packets to the gateway under time; it may be seen that dynamic programming always has the lowest-cost solution under the optimization principle, and although the performance of ADP is slightly inferior to that of the former, the performance is still superior to a genetic algorithm as long as the parameters are designed properly and the simulation time is sufficient. Figure 35 shows the comparison of all costs of dynamic programming and ADP after the programming ends, where the total cost is the sum of the penalty cost and the holding cost; the genetic algorithm is not included since it does not have a cost calculation mechanism like the costs of dynamic programming. The performances of dynamic programming are still slightly superior to ADP. However, since the performances of the latter are all only slightly inferior to those of the former, it is worthy of continuing using in the following large scale network experiments.

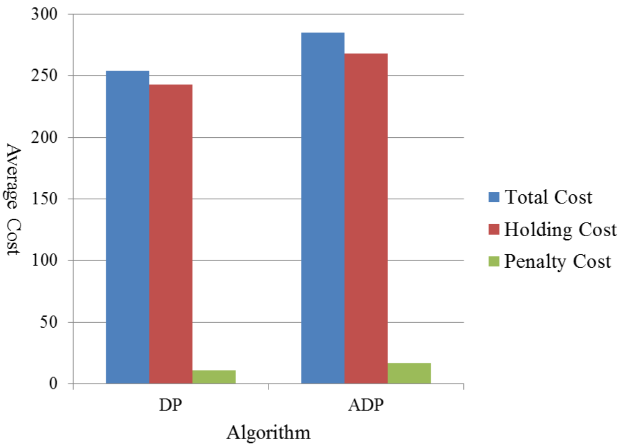

Figure 3. Cost diagram of DP and ADP

4.2 Performances of the algorithms in a large scale WMN model

Network topology of 256 nodes, 473 links, and 30 gateways is generated in the experiment. Parameter settings in the experiment are as below:

- Total number of time slots: 120
- Arriving amount and their probability of random packets at single node: 0: p = 0.9479, 1: p = 0.025, 2: p = 0.0125, 3: p = 0.00833, 4: p = 0.00625
- Sample number: 50
- ADP simulation times: 20
- ADP smoothing coefficient: 0.3
- Performance index: Ratio of the number of data packets arriving gateway to the total number of data packets.

The expectation value of randomly arriving packets in a time slot of each node in this topology is 0.1. Therefore, when the programming time with a total of 120 slots ends, the total data packet expectation value of this network topology with 256 nodes is $256 \times 120 \times 0.1 = 3072$. The ADP simulation times and smoothing coefficient are set as in the previous experiment, and the ratio of the number of data packets arriving gateway to the total number of data packets is similarly selected to be the performance index to observe the results of every slot. The experiment results in Figure 4 below show that as the time slots increase, the performance of transmitting data packets to gateways of ADP is gradually superior to that of a genetic algorithm.

To observe when all the data packets in the network may be transmitted to the gateways, randomly arriving data packets are not further added after the programming time ends ($T = 120$), yet the scheduling activity is allowed to continue operating; at this moment the nodes still transmit data packets to the gateways, but it should be noted that the data packets received by any node now is either from itself or received by other nodes before $T < 120$. According to Figure 5, all experiment samples of the WMN using ADP scheduling have transmitted all data packets to gateways at $t = 127$, while the model using a genetic algorithm still cannot achieve 100% gateway transmission rate in all experiment samples up to $t = 130$.

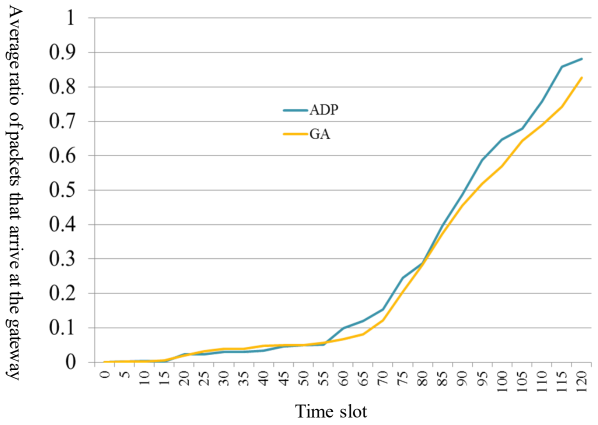

Figure 4. Scheduling performance comparison of ADP and genetic programming (1)

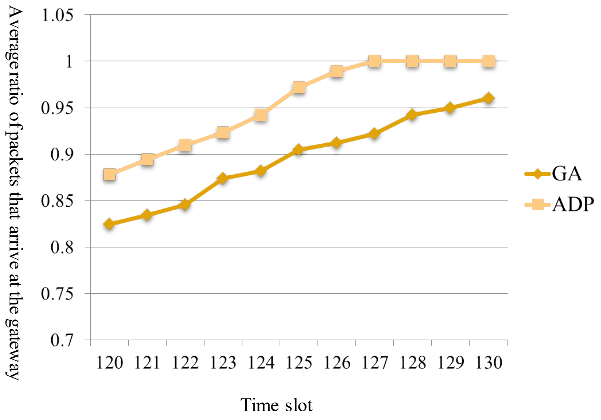

Figure 5. Scheduling performance comparison of ADP and genetic programming (2)

5. Conclusion

The characteristics of WMN are nothing but the rapid transmission of data packets by using a mesh structure and the constraints of capacity and interference encountered during transmission. However, due to differences in the emphasized purposes of the designers, many scheduling models contain variations and imperfections, such as the lack of randomly arriving data packets, the lack of gateway designs, or the unrealistic packet information saving location. The present research seeks to add as many as possible various characteristics of WMN into the model and make the models from literature reflect closer to reality. In this paper, the WMN model is improved, and programming information is moved to the nodes. In addition to conforming to reality, all algorithms may still be well applied to perform scheduling, and the effect of the simulation of dynamic programming by ADP is still apparent. The new WMN model comprises gateways to which the packets move, and the queuing of data packets at the nodes in each stage and the status of flow to the next stage may be observed via STDMA mechanism and decision variables of each stage. Finally, the experiments indicate that the performance of using ADP in the new model is superior to that of the genetic algorithm.

References

[1] B. Aoun, G. Kenward, R. Boutaba, and Y. Iraqi, Gateway Placement Optimization in wireless Mesh Networks with Qos Constraint, *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 11, pp. 2127-2136, 2006.

[2] L. Badia, A. Botta, and L. Lenzini, Joint routing and link scheduling for wireless mesh networks through genetic algorithms, in *Proceedings of 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Nerworks and Workshops*, IEEE Press, pp. 1-10, 2007.

[3] A. Capone, G. Carello, I. Filippini, S. Gualandi, and F. Malucelli, Routing, scheduling and channel assignment in wireless mesh networks: Optimization models and algorithms, *Ad Hoc Networks*, vol. 8, no. 6, pp.545-563, 2010.

[4] J. H. Chang and L. Tassiulas, Maximum lifetime routing in wireless sensor networks, *IEEE/ACM Trans Networks*, vol. 12, no. 4, pp. 609-619, 2004.

[5] D.-J. Deng and R.-S. Chang, A priority scheme for IEEE 802.11 DCF access method, *IEICE Transactions on Communications*, vol. E82-B, no. 1, pp. 96-102, 1999.

[6] J.-W. Ding, D.-J. Deng, T.-Y. Wu, and H.-H. Chen, Quality-aware bandwidth allocation for scalable on-demand streaming in wireless networks, *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 3, pp. 366-376, 2010.

[7] D. J. Deng, C.-H. Ke, H.-H. Chen, and Y.-M. Huang, Contention window optimization for IEEE 802.11 DCF access control, *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 5129-5135, 2008.

[8] D.-J. Deng and H.-C. Yen, Quality-of-service provision system for multimedia transmission in IEEE 802.11 wireless LANs, *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 6, pp. 1240-1252, 2005.

[9] D. P. De Farias and B. Van Roy, The linear programming approach to approximate dynamic programming, *Operations Research*, vol. 51, no. 6, pp. 850-865, 2003.

[10] J. Gronkvist, Traffic controlled spatial reuse TDMA in multi-hop networks, in *Proc. of PIMRC 1998*, vol. 142, no. 1, pp. 1203-1207, 1998.

[11] P. Gupta and P. R. Kumar, The capacity of wireless mesh networks, *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388-404, 2003.

[12] B. Leonardo, B. Alessio, and L. Luciano, A genetic approach to joint routing and link scheduling for wireless mesh networks, *Ad Hoc Networks*, vol. 7, no. 3, pp. 654-664, 2009.

[13] C.-C. Lin, S. Lei, and D.-J. Deng, Router node placement with service priority in wireless mesh networks using simulated annealing with momentum terms," to appear in *IEEE Systems Journal*, 2014.

[14] C.-C. Lin, W.-Y. Liu, C.-H. Chou, and D.-J. Deng, A novel joint problem of routing, scheduling, and variable-width channel allocation in WMNs, *The Scientific World Journal*, vol. 2014, Article ID 754749, 15 pages, 2014. doi:10.1155/2014/754749.

[15] O. E. Muogilim, K.-K. Loo, and R. Comley, Wireless mesh networks security: a traffic engineering management approach, *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 478-491, 2011.

[16] R. Nelson and L. Kleinrock, A collision-free multi-hop channel access protocol, IEEE transactions on Communications, vol. 33, no. 9, pp. 934-944, 1987.

[17] K. Papadaki and V. Friderikos, Approximate dynamic programming for link scheduling in wireless mesh networks, *Computer & Operational Research*, vol. 35, no. 3, pp. 3848-3859, 2008.

[18] K. Papadaki and W. B. Powell., A monotone adaptive dynamic programming algorithm for a stochastic batch service problem, *European Journal of Operational Research*, vol. 142, no. 1, pp. 108-127, 1997.

[19] V. Ranamurthi, A. Reaz, S. G. Dixit, and B. Mukherjee, Channel, capacity, and flow assignment in wireless mesh networks, *Journal of Computer Networks*, vol. 55, no. 9, pp. 2241-2258, 2011.

[20] N. Randolph, Spatial TDMA: A collision-free multiple channel access protocol, *IEEE Transaction on Communications*, vol. 33, no. 9, pp. 934-944, 1985.