Adaptive On-The-Go Scheduling for End-to-End Delay Control in TDMA-Based Wireless Mesh Networks

Yung-Cheng Tu, Meng Chang Chen, Yeali S. Sun

Adaptive On-The-Go Scheduling for End-to-End Delay Control in TDMA-based Wireless Mesh Networks*

Yung-Cheng Tu[1], Meng Chang Chen[2] and Yeali S. Sun[3]

[1]Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
[2]Institute of Information Science, Academia Sinica, Taipei, Taiwan
[3]Department of Information Management, National Taiwan University, Taipei, Taiwan

Abstract. Providing end-to-end delay bound for real-time applications is a major challenge in wireless mesh networks (WMNs) because the bandwidth requirements of flows are time-varied and the channel condition is unstable due to the wireless interference between the links. In this paper, we present a two-stage slot allocation mechanism in TDMA-based WMNs. First, we assume that the bandwidth requirement of each flow is given in the form of a range and use a distributed algorithm to pre-allocate time slots to each link. Then, we implement an On-The-Go scheduling scheme, which enables each link to schedule its transmission time promptly without coordinating with others. In contrast to traditional approaches, our method allows a degree of control over the collision probability, but it only requires a few control messages and the computational overhead is lower. The simulation results show that our mechanism performs efficiently and flexibly on supporting real-time applications in WMNs.

Keywords: TDMA-based scheduling; wireless mesh networks; slot allocation

1 Introduction

For high affordability and easy deployment, wireless mesh networks (WMNs) have been developed as cost-efficient networking platforms to support ubiquitous broadband Internet access in several cities [1][2][3]. A communication flow from source to destination in a WMN usually requires multiple hops. Due to the available bandwidth for each hop is time-varied depending on the current load and interference between the wireless links, supporting end-to-end quality of service (QoS) guarantees for communication flows in WMNs is a challenging task.

In this paper, we focus on QoS provisioning for real-time applications, such as VoIP, video conferencing and multi-media streaming. Real-time applications usually generate variable-bit-rate (VBR) traffic and require an end-to-end QoS guarantee. It is known that contention-based protocols, like CSMA/CA used in IEEE 802.11 [4], cannot provide strict QoS guarantees because they will result in service unfairness

* The project was partly supported by NSC Taiwan under contract NSC98-2221-E-001-005-MY3.

between wireless links [5] in multi-hop wireless networks. In contrast, time division multiple access (TDMA) based MAC protocols, such as the 802.16 mesh protocol [6] and the 802.11s mesh coordination function (MCF) coordinated channel access protocol [7], provide collision-free communications and allow fine control of the throughput and delay of network traffic.

Several TDMA-based scheduling algorithms with different objective functions have been proposed for WMNs, e.g., maximizing system throughput [8][9][10], fairness [10][11], and flow utility [12][13], or minimizing end-to-end transmission delays [14]. These algorithms can be classified into two types: centralized algorithms and distributed algorithms. For centralized algorithms, the central controller requires a substantial amount of time to collect the bandwidth requirements of all links, apply the scheduling algorithm and deliver the scheduling results to all the mesh nodes. In contrast, a distributed scheduling algorithm is applied by each mesh node without the information about the whole network. However, coordination between the mesh nodes is necessary to ensure that the transmissions of different links will not conflict with each other. As a result, the coordination mechanism incurs extra overheads and scheduling waiting time [8][11][13][15].

One of the major challenges in providing end-to-end QoS guarantees for real-time application flows in TDMA-based WMNs is how to rapidly adjust the slot allocation of each link when the traffic load of flows and network condition vary frequently. As mentioned above, for both centralized and distributed scheduling algorithms, there is a certain amount of latency between the time a link requests a scheduler and the time it gets the scheduling result. Therefore, all existing TDMA-based scheduling algorithms assume that the state of each link is fixed. Mostly when a link's state changes, it is usually necessary to re-schedule the transmission time of all the links in the network. To address the above problem, we propose an adaptive *on-the-go* scheduling scheme that allocates time slots for each link dynamically. The contributions of this work are as follows:

1. We propose a **two-stage slot allocation mechanism** for TDMA-based WMNs. The first stage provides minimum end-to-end QoS guarantees to all real-time application flows; and the second stage allows each link to adjust its transmission time dynamically to maximally satisfy the QoS requirements of all flows on the link and prevent transmission collisions with other links.
1. In contrast to traditional conflict-free slot allocation schemes, we allocate **conflict-free slots** and **multi-access slots** to each link**.** The transmissions in the conflict-free slots of each link will not be interfered by those of other links. For the multi-access slots, we can control the number of interference nodes and maximize network utilization by selecting appropriate transmission time slots for each link.
2. We present an adaptive **on-the-go scheduling** scheme that each link can schedule its transmission times without coordinating with other links. The mechanism performs more flexibly on real-time application flows than traditional one-stage conflict-free slot allocation schemes.

The remainder of this paper is organized as follows. In Section 2, we define the network and system model of our work. In Section 3, we present our two-stage slot allocation mechanism. In Section 4, we introduce the adaptive on-the-go scheduling scheme for our two-stage slot allocation mechanism. In Section 5, we evaluate the scheme's performance via simulations. Finally, we give a conclusion in section 6.

2 Network and System Models

We model a WMN by a directed network graph NG=(N,V), where N={n_a,n_b,n_c,…} is the set of nodes and V={v_1,v_2,v_3,…} is the set of directed links. In a WMN, a node n_a can transmit data to another node n_b if n_b is in the transmission range D_{TR} of n_a. A link from node n_a to node n_b means the node n_a has the ability to transmit data to node n_b. Two links in a WMN will interfere with one another if they can not transmit packets simultaneously. In this paper, we adopt the protocol model proposed in [16] as our interference model. Under the model, a transmission from node n_a to node n_b is successful if and only if there is no other node within the interference range D_{IR} of n_b transmitting data at the same time. With the protocol model, we can construct a undirected contention graph CG=(V,E), where V is the same as above and an edge (v_k,v_l) is in E if links v_k and v_l can not transmit data simultaneously. Then, we define link v_k's neighbor set as NB(v_k)={v_l | (v_k,v_l) ∈E}.

A real-time application flow f_i in a WMN consists of a routing path, defined as $Path_i$={v_k| v_k∈V, f_i passes through the link v_k}, and the range of the flow's demand rate ($r_{i,k}^{\min}$,$r_{i,k}^{\max}$) at all v_k∈$Path_i$, where $r_{i,k}^{\min}$ and $r_{i,k}^{\max}$ are, respectively, the minimum and maximum demand rates of f_i at link v_k. Then, the bandwidth requirement of link v_k is bound by $(R_k^{\min}, R_k^{\max})$, where $R_k^{\min}$ is the summation of $r_{i,k}^{\min}$ of all flows that pass through v_k; $R_k^{\max}$ can be obtained in a similar manner.

Like most TDMA-based protocols, we divide the timeline into recurrent frames, each comprised of M fixed-length time slots. Suppose the demand rate of flow f_i at the link v_k in the frame t is $r_{i,k}(t)$ and C_k is the channel capacity of link v_k., we can estimate the total bandwidth requirement $R_k(t)$ and slot requirement $T_k(t)$ of the link v_k in the frame t as follows:

$$R_k(t) = \sum_{i|v_k \in Path_i} r_{i,k}(t) \quad (1)$$

$$T_k(t) = \lceil M \cdot R_k(t)/C_k \rceil \quad (2)$$

Similarly, we can derive the minimum and maximum slot requirements, denoted as $T_k^{\min}$ and $T_k^{\max}$, of v_k from $R_k^{\min}$ and $R_k^{\max}$. We assume that the demand rate $r_{i,k}(t)$ of each flow i at link v_k is between $r_{i,k}^{\min}$ and $r_{i,k}^{\min}$, so the slot requirements of link v_k is bounded by $(T_k^{\min}, T_k^{\max})$.

3 The Two-Stage Slot Allocation Mechanism

The system framework of our two-stage slot allocation mechanism with on-the-go scheduling is shown in Fig. 1. The demand rate of a flow f_i is estimated based on the traffic load, network condition and the end-to-end QoS requirement of the flow. Any bandwidth estimation algorithm that supports end-to-end delay control, such as the per-node delay assignment scheme [17] or the bulk scheduling scheme [18], can be used in our mechanism.

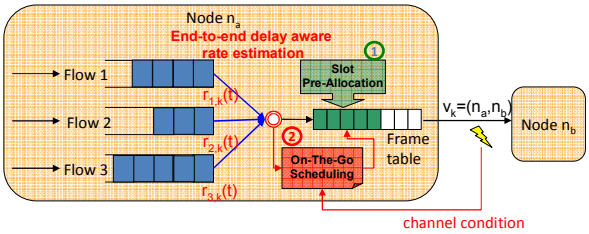

Fig. 1. The framework of the two-stage slot allocation mechanism with On-The-Go scheduling.

In the first stage, we pre-allocate slots to each link v_k according to the link's minimum and maximum slot requirements. Since T_k^{min} and T_k^{max} are determined by the minimum and maximum demand rates of all flows in v_k, slot pre-allocation is only required when a flow joins or leaves this link. It is not necessary to dynamically adjust the pre-allocated slots with the variable traffic load; therefore coordination between the links is allowed in this stage. The second stage of our slot allocation mechanism implements the on-the-go scheduling scheme. The scheme tries to select slots dynamically from the pre-allocated slots in the first stage to maximally satisfy the immediate bandwidth requirements of all flows on the link and prevent transmission collisions with other links.

3.1 Conflict-free and multi-access slots

The objective of the slot pre-allocation scheme is to allocate slots for each link v_k such that T_k^{min} can be guaranteed and T_k^{max} can be satisfied as much as possible in the second stage. Therefore, our pre-allocation scheme assigns two types of slots: *conflict-free slots* and *multi-access slots* for each link v_k. The former is not allocated to any neighbor of v_k while the latter can be allocated to some of its neighbors. Let m_k and m'_k be, respectively, the number of conflict-free slots and all pre-allocated slots. In our pre-allocation scheme, the m_k is equal to T_k^{min} and m'_k must be greater than T_k^{min} but cannot exceed T_k^{max}. Our scheme has the same constraint as [14], which requires that a link's pre-allocated slots are continuous in order to reduce the overhead of coordination between links. In addition, we assume that each active link contains at least one conflict-free slot to prevent flow starvation. Then the conflict-free slots of each link must also be continuous because the pre-allocated slots between any two conflict-free slots of a link v_k cannot be pre-allocated to any neighbor of v_k or it will result in the fragmentation of the pre-allocated slots of the link v_k. Thus the pre-allocated slots of each link can be divided into three periods as in Fig. 2.

Suppose the starting positions of all pre-allocated slots and conflict-free slots of link v_k in the frame table are the slot s'_k and slot s_k respectively. We define three periods of the pre-allocated slots of v_k, namely, the head period Z_k^{head}, the body period Z_k^{body} and the tail period Z_k^{tail}, as follows:

$$Z_k^{head} = \{i \mid \Delta(s'_k, i) < \Delta(s'_k, s_k)\} \quad (3)$$

$$Z_k^{body} = \{i \mid \Delta(s'_k, s_k) \le \Delta(s'_k, i) < \Delta(s'_k, s_k + m_k)\} \quad (4)$$

$$Z_k^{tail} = \{i \mid \Delta(s'_k, s_k + m_k) \le \Delta(s'_k, i) < m'_k\} \quad (5)$$

where $\Delta(i,j)=(j-i) \bmod M$ represents the distance from slot i to slot j in recurrent frames. The relationships between the three periods are shown in Fig. 2.

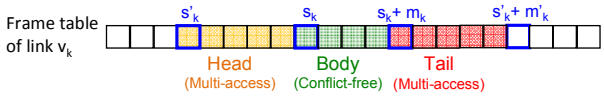

Fig. 2. The head, body and tail periods of a link.

The transmission of link v_k in a slot i during its head period will only compete for transmission opportunities with link v_l, where $v_l \in NB(v_k)$ and v_l's tail period Z_l^{tail} contains the slot i. Moreover, a transmission of v_k in a slot $j \in Z_k^{tail}$ will only compete with the link v_m, where $v_m \in NB(v_k,)$ and $j \in Z_m^{head}$. Thus, the contention degree of a link v_k in our two-stage slot allocation mechanism can be bounded. Let $CG(v_k)$ be a subgraph of CG that contains only the neighbors of v_k and the edges between the neighbors. Under our pre-allocation scheme, the number of links that compete for transmission opportunities with v_k, in any time slot is not greater than the size of the maximum independent set of all links in $CG(v_k)$ because the links that can compete for transmission opportunities with v_k at any slot cannot be neighbors with each other.

3.2 The slot pre-allocation scheme

The objective of our slot pre-allocation scheme is to maximally satisfy the slot requirements of all links. The pre-allocation scheme can be centralized or distributed with some coordination. However, the scheduler needs information about all the links in the WMN to get the global optimal solution, which incurs a large communication overhead and requires a great deal of computation time. Moreover, because all links in a WMN must adjust their pre-allocated slots once a link changes its pre-allocation, each link's pre-allocation will be changed frequently. This results in instability in the short-term throughput of each link and higher delay jitters between links. Therefore, we consider the following local optimization problem for each link which can be solved by a distributed pre-allocation algorithm:

$$\textbf{Given}: M,\ T_k^{\min},\ T_k^{\max},\ NB(v_k),$$
$$\text{and } (s'_l, s_l, m_l, m'_l) \text{ of all } v_l \in NB(v_k)$$
$$\textbf{Find}: s'_k, s_k, m_k, m'_k$$
$$\textbf{Maximize}: m'_k$$
$$\textbf{s.t.}\quad 0 < m_k = T_k^{\min} \le m'_k \le T_k^{\max}$$
$$0 \le \Delta(s'_k, s_k) \le m'_k - m_k$$
$$\Delta(s'_k, s_k) \le \Delta(s_l + m_l, s_k) \quad ,\forall v_l \in NB(v_k)$$
$$\Delta(s_k + m_k, s'_k + m'_k) \le \Delta(s_k + m_k, s_l) \quad ,\forall v_l \in NB(v_k)$$

To solve this problem, each node only needs to gather the pre-allocated slots, which consist of the head, body and tail periods, from all neighbors. This problem can be easily solved by a linear search. Since more than one region would meet the above

requirements, choosing an appropriate one as the pre-allocation is also a problem in the pre-allocation scheme. In this paper, we choose pre-allocate slots in the longest available period for v_k to achieve the objective of maximizing the m'_k.

Any existing TDMA-based distributed slot allocation protocol can be applied to our pre-allocation scheme with a little modification where the allocated slots are represented by four parameters (s'_k, s_k, m_k, m'_k). For example, the slot allocations in IEEE 802.16 are performed by a three-way handshake: request, grant and confirm. To apply our mechanism in IEEE 802.16, a node sends a request message containing the minimum and maximum slot requirements $(T_k^{\min}, T_k^{\max})$ to its receiver only when the bandwidth requirement $(R_k^{\min}, R_k^{\max})$ of the link is changed. The receiver schedules a pre-allocation as mentioned above, broadcasts the result as a grant in the (s'_k, s_k, m_k, m'_k) format and waits the confirm from the sender. Thus, each link in the WMN knows the pre-allocation of its neighbors and keeps a list of available conflict-free and multi-access slots as in Fig. 3. Our algorithm only needs to retrieve all available slots and find the one with maximum m'_k from feasible allocations. The complexity of the algorithm is O(M) because the maximum number of available slots is M.

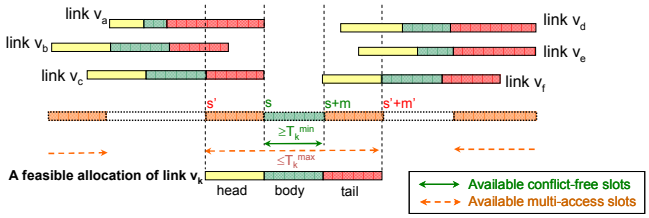

Fig. 3. The illustration of retrieving the frame to find the feasible allocations.

4 On-The-Go Scheduling Scheme

In previous section, we introduced the pre-allocation scheme for each link v_k. However, we still need an efficient scheduling scheme to select slots from the pre-allocated slots to transmit data in each frame. Therefore, we propose the *on-the-go* scheduling scheme that 1) chooses slots dynamically for transmissions without coordinating with other links or waiting for scheduling from a central server; and 2) prevents collisions between links and maximize the network throughput. Specifically, the on-the-go scheduling scheme assigns an index value to each slot and chooses slots according to the immediate slot requirement and index values.

4.1 Index values and transmission slots selection

In the following, we use $\mathbf{x_k}$ to denote the vector of all index values of link v_k, where $x_k(i)$ is the i-th element in $\mathbf{x_k}$ and represents the index value of slot i of link v_k. With our design, the index value of a slot in the body period and the idle period must be

zero and infinite respectively. Each multi-access slot has a unique integer index value between 1 and $m'_k - m_k$. Let $x_k^{max}(t)$ denote the maximum index value of slots that v_k can transmit data in the frame t.

4.2 The Near-Body-First (NBF) policy

In our mechanism, the slot with smaller index value has higher priority to be selected to transmit data. The $\mathbf{x_k}$ is used to prevent transmission collisions and maximize the network throughput. This is achieved by the **Near-Body-First** policy. An index value assignment $\mathbf{x_k}$ of v_k is a **Near-Body-First (NBF)** index assignment if the index values of any two adjacent slots i and j, j=(i+1) mod M, of v_k satisfy the constraints:

$$\begin{cases} x_k(i) > x_k(j) & \text{,if } i, j \in Z_k^{head} \\ x_k(i) < x_k(j) & \text{,if } i, j \in Z_k^{tail} \end{cases} \quad (6)$$

Theorem 1. Given the head, body, and tail periods of each link and an index value assignment $\mathbf{x_k}$ whose expected throughput for each link v_k is θ_k, we can always find an NBF index assignment $\mathbf{x'_k}$ such that the expected throughput of each link v_k with $\mathbf{x'_k}$ is higher than or equal to θ_k.

The Theorem 1 can be proved by iteratively exchanging the index values of any two adjacent slots that violate the NBF policy and ensuring that each iteration will not diminish the expected throughput of each link. The detail of the proof is omitted here due to the limitation of paper length.

From Theorem 1, we know that the NBF index assignments can provide the highest expected throughput for all links. However, given the head, body and tail periods of a link v_k, there are many index assignments belonging to NBF index assignments. To dynamically calculate the index assignment for each link according to the links channel conditions to avoid collisions is a major concern of the on-the-go scheduling scheme. In this paper, we define the parameters $w_k^{h2t}(t)$, called the weight of the head over the tail, for the index assignment of v_k in frame t as follows:

$$w_k^{h2t}(t) = \frac{g_k^{head}(t-1) + \delta}{g_k^{tail}(t-1) + \delta} \quad (7)$$

where $g_k^{head}(t\text{-}1)$ and $g_k^{tail}(t\text{-}1)$ are the numbers of packets transmitted successfully during the head and the tail periods of v_k in the previous frame. The parameter δ is a very small value to avoid the illegal division by zero and let $w_k^{h2t}(t)$ be 1 when both $g_k^{head}(t\text{-}1)$ and $g_k^{tail}(t\text{-}1)$ are equal to 0. Since a link v_k will compete for transmission opportunities with different neighbors in its head and tail periods, it will experience different interferences in the two periods.

The parameter $w_k^{h2t}(t)$ represents the relation of the interferences in head period and that in tail period. A higher value of $w_k^{h2t}(t)$ indicates that the link v_k has fewer collisions during its head period. Thus, we calculate the index value of each multi-access slot in frame t as follows:

$$x_k(i)=\begin{cases}\Delta(i,s_k)+\min\left(\left|Z_k^{tail}\right|,\left\lfloor(\Delta(i,s_k)-1)/w_k^{h2t}(t)\right\rfloor\right) & ,\ \text{if i}\in Z_k^{head}\\ \Delta(s_k+m_k,i)+\min\left(\left|Z_k^{head}\right|,w_k^{h2t}(t)\cdot\Delta(s_k+m_k,i)\right) & ,\ \text{if i}\in Z_k^{tail}\end{cases} \quad (8)$$

where $|Z_k^{tail}|$ and $|Z_k^{head}|$ are the lengths of the tail and head periods of v_k. Fig. 4 shows three examples of index assignments in which $w_k^{h2t}(t)$ is infinite, 0 and 2 respectively.

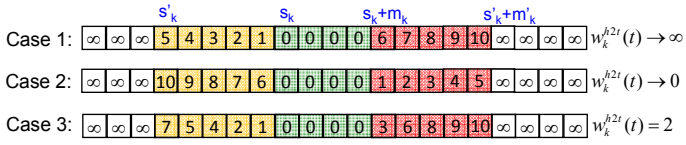

Fig. 4. Three examples of NBF index assignment.

4.3 Congestion control

Although the design of $w_k^{h2t}(t)$ reflects the difference between interferences in the head and tail periods, simply adjusting the index values of each slot can not avoid collisions when the slot requirement is large because the link will tend to use all the multi-access slots to transmit data. Therefore, we incorporate a congestion control mechanism into our on-the-go scheduling scheme. The congestion control is achieved by introducing the parameter $x_k^{max}(t)$ and limiting the link v_k can only transmit data in slots with index values smaller than $x_k^{max}(t)$. If collisions occurred in the transmissions of v_k in the previous frame, the value of $x_k^{max}(t)$ is reduced to slow down the transmission rate of v_k as follows:

$$x_k^{\max}(t)=g_k^{head}(t-1)+g_k^{tail}(t-1)-1 \quad (9)$$

If there were no collisions in the transmissions of v_k in the previous frame, we gradually raise the transmission rate by increasing the $x_k^{max}(t)$:

$$x_k^{\max}(t)=x_k^{\max}(t-1)+1 \quad (10)$$

4.4 Drop tail policy

The drop tail policy is another collision avoidance mechanism. Its function is to "stop all the transmissions of v_k after a collision of v_k has occurred during its tail period until the next head period begins." The concept of the drop tail policy is based on the property that the transmissions of v_k during its tail period will only be affected by the transmissions of the neighbors of v_k in their head periods. Moreover, according to the NBF policy, once a slot i is selected to transmit data by v_k, the slots following slot i in the head period of v_k will also be selected to transmit data. Therefore, if a collision occurs in the tail period of v_k, the residual slots in the same tail period will fail. We can discard the transmissions in these slots that the v_k and its neighbors will have lower collision probabilities.

5 Performance Evaluation

In this section, we evaluate the performance of the proposed two-stage slot allocation mechanism and on-the-go scheduling scheme by simulations. All of the simulations were performed using the ns-2 network simulator [19] with TDMA enhancements in the MAC layer of IEEE 802.11. We show the benefit of the proposed mechanism by comparing with other two types of transmission schemes: the IEEE 802.11, and the traditional conflict-free TDMA-based schemes with average-rate and peak-rate slot allocations, denoted as TDMA-avg and TDMA-peak, respectively. The system configuration of our simulations is listed in Table I. All flows use the UDP as their transport protocol. The period of each simulation is 3000 seconds. The distance between the source and destination of each link is 200 meters in all scenarios.

Table 1. The system Configuration of our simulations.

Parameter	Value
Channel Bandwidth	11Mbps
Interface Queue Size	100 packets
Packet Size	1500 bytes
Flow type	Exponentially distributed ON/OFF model Average ON/OFF period : 1000/1000 ms.
Transmission/Interference Range	250m/420m
TDMA slot/frame duration	1.2 ms/60 ms (50slots per frame)

5.1 Chain topology

In this scenario, we examine the cases of one flow and multi-flows on a 6-hop chain topology as shown in Fig. 5(a). We first consider the wireless network with a 6-hop flow. We evaluate the performance of our mechanism by increasing the average sending rate of the 6-hop flow and Fig. 5(b), 5(c) and 5(d) show the simulation results. We can see that our on-the-go (OTG) scheduling provide higher end-to-end throughput than IEEE 802.11 and TDMA-avg as shown in Fig. 5(c). Although TDMA-peak has shorter end-to-end delay than OTG, but it can not admit the flow with average sending rate more than 1200 Kbps while OTG can allow the sending rate of the 6-hop flow up to 1600 Kbps. Note that in the cases of TDMA-peak and OTG with sending rate more than their schedulable limits, we still admit the flow but only the maximum schedulable slots are allocated. The TDMA-avg has shorter average end-to-end delay when traffic load is high because most packets are dropped by the interface queue. We also show the performance of our enhancements, congestion control and drop tail, for on-the-go scheduling in Fig. 5(d). The results show that the two enhancements can reduce the collision probability of the on-the-go scheduling when traffic load is high.

In the second part of this scenario, we consider multiple flows with different path lengths. There are one 6-hop, two 3-hop and six 1-hop flows, as in Fig. 5(a). We fix the sending rates of 1-hop and 3-hop flows to 512Kbps while the sending rate of the 6-hop flow varies from 200Kbps to 1000Kbps in this simulation. We show the

average end-to-end delay of each flow and collision probability under IEEE 802.11 and OTG in Fig 5(e) and 5(f). The results of TDMA-peak and TDMA-avg are not presented because the comparisons of their performances with OTG are similar as in the scenario of single flow shown in Fig. 5(b). We can see that OTG provides constant end-to-end delay for each flow when the requests are schedulable. The average scheduling delay [14] of each hop is about 30ms in our simulation, which results in about 150ms and 60ms end-to-end scheduling delay to 6-hop and 3-hop flows. If we only consider the queueing delay, the on-the-go scheduling performs fairly on flows with different hops.

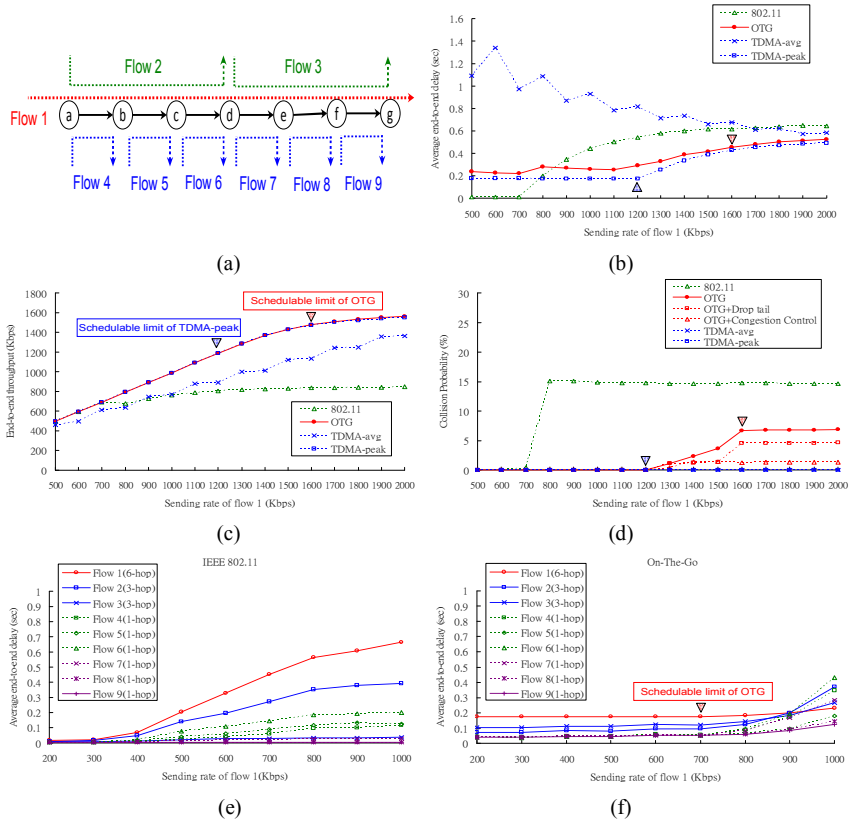

Fig. 5. The topology and simulation results in the 6-hop chain wireless network.

5.2 Mesh topology

In this section, we consider four flows in a wireless mesh network as shown in Fig. 6(a). We fixed the sending rate of flow 1, 2 and 4 to 1000Kbps and varied the sending rate of flow 3 from 100Kbps to 1000Kbps to examine the performance of flows with different channel conditions. The results of average end-to-end delay of each flow and collision probability with different mechanisms are shown in Fig. 6(b), 6(c) and 6(d).

From the results, we can see that OTG preserves the property of isolation between flows that, as shown in Fig 6(b), the end-to-end delays of OTG flows maintain constant when the rate of flow 3 increases. Similarly, the TDMA-peak also provides isolation between flows but it can only admit the flow 3 with traffic load less than 300Kbps. We have not shown the collision probability of TDMA-avg and TDMA-peak in Fig. 6(d) because their collision probabilities all equal to zero. The drop tail policy only has little improvement because almost all collisions are occurred in the last slot in the tail period of each link when traffic load is not heavy.

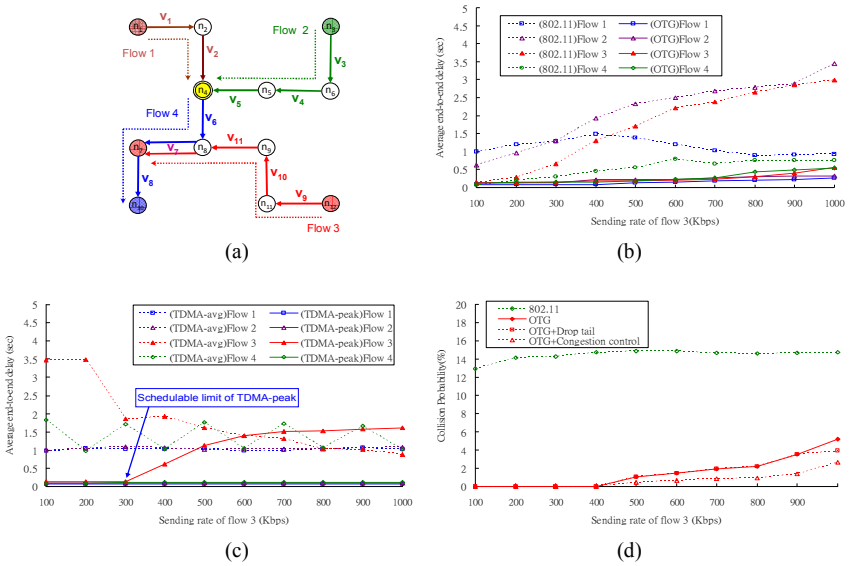

Fig. 6. The topology and simulation results of four flows in a 4x4 grid wireless network.

6 Conclusion and Discussion

In this paper, we proposed a two-stage slot allocation mechanism on TDMA-based transmission protocols in WMNs. The mechanism allocates not only conflict-free slots but also multi-access slots to each link compared to traditional one-stage slot allocation algorithms. An adaptive on-the-go scheduling for the two-stage slot allocation mechanism is also introduced to dynamically schedule the transmission time slots within the multi-access slots for each link. The on-the-go scheduling scheme selects slots for transmission and avoid collisions without coordinating with other links that it can afford rapidly adjusting the slot allocation to support real-time applications with variable-bit-rate traffic. The simulation results also show that our on-the-go scheduling scheme achieves higher utilization than IEEE 802.11 MAC protocol and performs more flexibly and efficiently than traditional TDMA-based slot allocations.

References

1. Seattle wireless, (http://www.seattlewireless.net)
2. Southampton Open Wireless Network (http://www.sown.org.uk)
3. Rice TFA Network (http://tfa.rice.edu)
4. "IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999), pp.C1-1184.
5. M. Garetto, T. Salonidis, and E. Knightly, "Modeling Per-flow Throughput And Capturing Starvation in CSMA Multi-hop Wireless Networks," in Proceedings of IEEE INFOCOM 2006, Barcelona, Spain, April 2006.
6. "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems," IEEE Std 802.16-2004 (Revision of IEEE Std 802.16-2001), pp.0_1-857, 2004
7. "IEEE Draft STANDARD for Information Technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking," IEEE Unapproved Draft Std P802.11s/D6.0, June 2010.
8. A. Kabbani, T. Salonidis, and E. W. Knightly, "Distributed Low-Complexity Maximum-Throughput Scheduling for Wireless Backhaul Networks," in Proceedings of IEEE INFOCOM 2007, , Anchorage, AK, May 2007.
9. H. T. Cheng and W. Zhuang, "Pareto Optimal Resource Management for Wireless Mesh Networks with QoS Assurance: Joint Node Clustering and Subcarrier Allocation," in IEEE Transactions on Wireless Communications, vol. 8, no. 3, March 2009.
10. X.-Y. Li, A. Nusairat, Y. Wu, Y. Qi, J. Zhao, X. Chu, and Y. Liu, "Joint Throughput Optimization for Wireless Mesh Networks," in IEEE Transactions on Mobile Computing, vol. 8, no. 7, pp. 895-909, July 2009.
11. C. Cicconetti, I. F. Akyildiz, and L. Lenzini, "FEBA: A Bandwidth Allocation Algorithm for Service Differentiation in IEEE 802.16 Mesh Networks," in IEEE Transactions on Networking, vol. 17, no. 3, pp. 884-897, June 2009.
12. B. Wang, and M. Mutka, "Qos-Aware Fair Rate Allocation in Wireless Mesh Networks," in Elsevier Computer Communications, vol. 31, issue 7, pp. 1276-1289, May 2008.
13. Y. Hou, and K. K. Leung, "A Distributed Scheduling Framework for Multi-User Diversity Gain and Quality of Service in Wireless Mesh Networks," in IEEE Transactions on Wireless Communications, vol. 8, no. 12, Dec.2009.
14. P. Djukic, and S. Valaee, "Delay Aware Link Scheduling for Multi-Hop TDMA Wireless Networks," in IEEE/ACM Transactions on Networking, vol. 17, no. 3, June 2009.
15. I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks," in IEEE Transactions on Mobile Computing, vol. 8, no. 10, Oct. 2009.
16. P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," in IEEE Transactions on Information Theory, vol. 46, no. 2, pp. 388-404, Mar. 2000.
17 A. Vagish, T. Znati and R. Melhem, "Per-Node Delay Assignment Strategies for Real-Time High Speed Network," in Proceedings of IEEE Globecom 1999, pp. 1323-1327, Dec. 1999.
18. Y.-C. Tu, M. C. Chen, Y. S. Sun and W.-K. Shih, "Enhanced Bulk Scheduling for Supporting Delay Sensitive Streaming Applications," in Elsevier Computer Networks, Vol. 52, Issue 5, pp. 971-987, April 2008.
19. "Network Simulator ns-2," http://www.isi.edu/nsnam/ns/.