

# Adaptive Resource Control in 2-hop Ad-Hoc Networks

Yimeng Yang, Geert Heijenk  
University of Twente  
Enschede, The Netherlands  
{y.yang, geert.heijenk}@utwente.nl

Boudewijn R. Haverkort  
University of Twente  
Enschede, The Netherlands, and  
Embedded Systems Institute  
Eindhoven, The Netherlands  
b.r.h.m.haverkort@utwente.nl

**Abstract**— This paper presents a simple resource control mechanism with traffic scheduling for 2-hop ad-hoc networks, in which the Request-To-Send (RTS) packet is utilized to deliver feedback information. With this feedback information, the Transmission Opportunity (TXOP) limit of the sources can be controlled to balance the traffic. Furthermore, a bottleneck transmission scheduling scheme is introduced to provide fairness between local and forwarding flows. The proposed mechanism is modeled and evaluated using the well-known 20-sim dynamic system simulator. Experimental results show that a fairer and more efficient bandwidth utilization can be achieved than without the feedback mechanism. The use of the structured and formalized control-theoretical modeling framework has as advantage that results can be obtained in a fast and efficient way.

## I. INTRODUCTION

The IEEE 802.11 [1] Medium Access Control (MAC) has become a widely accepted standard for Wireless Local Area Networks (WLANs) in many different environments; it is also regarded as a viable candidate for wireless ad-hoc networks. The Distributed Coordination Function (DCF) (based on CSMA/CA) tends to equally share the bandwidth capacity among contending nodes. Although fair, the DCF will lead to undesirable situations in case certain nodes happen to function as bridges or bottlenecks toward other destinations, e.g., to the fixed internet. Thus packets may get lost at the bottleneck nodes even after having been gone through the initial hops already. In order to improve the efficiency of resource use, and meanwhile ensuring fair access for all flows, regardless of the number of hops, the transmission rate of the sources should be controlled according to the current workload of the bottleneck. Furthermore, the limited capacity of the bottleneck has to be fairly shared by local as well as forwarding traffic.

In this paper, we present a mechanism for resource control in 2-hop ad-hoc networks with bottleneck transmission scheduling. The Request-To-Send (RTS) packets sent by the bottleneck to the next hop are used as feedback to the previous hop as these RTS packets are received by all neighbors. The feedback information is used by the sources to control their TXOP limits, i.e., the number of packets they can send once they get access to medium. The adaptation of the TXOP values is based on the Additive Increase and Multiplicative Decrease (AIMD) algorithm known from TCP. The proposed resource control mechanism is modeled using a control-theoretic method: the models are evaluated with the 20-sim analysis tool [11].

The contribution of this paper lies in the following:

- The new resource control mechanism requires no modifications to the 802.11 frames, no system monitoring, and does not need a centralized controller. Furthermore, no overhead data needs to be exchanged whatsoever.
- Compared to the network system without the new control mechanism, more efficient bandwidth utilization is achieved, and the bottleneck can divide its capacity fairly between local and forwarding traffic.
- The use of the control-theoretical framework, as opposed to an ns-2 simulation, has as advantage that results can be obtained much more quickly, and that the model remains much more structured and formalized.

The rest of the paper is organized as follows. Section II introduces background information and related work on IEEE 802.11. Section III describes the new TXOP control strategy as well as the transmission scheduling scheme for the bottleneck. Section IV presents the control theoretic model of the proposed mechanism. Section V shows experimental results, and in Section VI the paper is concluded.

## II. BACKGROUND AND RELATED WORK

The 802.11 DCF works as a ‘listen before talk’ scheme based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). To mitigate series of data frame collisions and to reduce the hidden terminal problem inherent to CSMA, an optional mechanism called Request-To-Send/Clear-To-Send (RTS/CTS) has been defined as part of DCF. When a station wins a channel access, an RTS packet will be transmitted comprising the destination address and the expected duration of the data transmission. This packet can be received by all other stations in reach of the source station, and their so-called Network Allocation Vector (NAV) will be set accordingly to refrain from accessing the medium. The NAV of a station specifies the earliest time when the station is permitted to attempt transmission. Upon receipt of an RTS, and after waiting for a Short Inter-Frame Space (SIFS), the intended destination station replies with a CTS packet to indicate that it is ready to receive the data packet. All stations hearing the CTS will update their NAVs based on the duration information enclosed. Note that the set of stations receiving the CTS packet may be different from the set of stations that received the RTS packet; therefore, the successful RTS/CTS handshake guarantees undisturbed medium for the intended transmission within the range of the source and the destination station. The

data packet can then be transmitted after another SIFS, and the complete packet reception will be notified by successfully receiving an acknowledgement (ACK) from the destination.

To counteract the inherent Quality of Service (QoS) limitations in the basic 802.11 MAC, the IEEE 802.11e [2] standard has been specified for service differentiation in WLAN systems; this standard introduces a new mechanism called Hybrid Coordination Function (HCF). The HCF operates with two access methods: the contention-based Enhanced Distributed Coordination Access (EDCA) and the contention-free HCF Controlled Channel Access (HCCA) by applying a centralized controller, known as the Hybrid Coordinator (HC) generally located at the access point.

To provide QoS differentiation in 802.11e WLANs, the concept of Transmission Opportunity (TXOP) is proposed; a TXOP limit parameter is defined as the time interval during which multiple packets are allowed to be transmitted by the station that gains medium access. This concept has also been proposed for 802.11 ad-hoc networks to improve performance [3]; however, the choice and possible adjustment of the TXOP parameter, based on the current network situation is not discussed. A stochastic model has been introduced in [4] to evaluate the QoS enhancements using other differentiation parameters defined in 802.11e, e.g.,  $CW_{\min}$  and AIFS, in an ad-hoc network scenario with a bottleneck. However those parameters can only be statically assigned. The work [5] presented a dynamic TXOP configuration scheme for efficiency improvement in 802.11e WLANs, which relies on information broadcasted by the HC. Therefore, it can not be applied in ad-hoc networks without centralized control. A congestion control scheme avoiding traffic aggregation at the bottleneck has been introduced in [6]. In that scheme the bottleneck will only responds to RTS frames when its receiving and forwarding rates are balanced. However, the fairness among different flows can not be ensured.

In [7], an analysis model has been derived for multi-hop IEEE 802.11 networks to calculate the bandwidth that can be utilized along a path without violating QoS requirements of existing traffic. In [8] the throughput of IEEE 802.11 multi-hop ad hoc networks has been analyzed and a method to estimate the optimal offered load has been formulated. However, all their works considered networks with fixed topology, whereas we propose a scheme that dynamically adapts allocation of capacity, based on topology and offered load.

### III. TXOP CONTROL STRATEGY AND BOTTLENECK TRANSMISSION SCHEDULING

In this section, the TXOP control strategy and bottleneck transmission scheduling scheme will be discussed by considering a simple 2-hop network scenario, as depicted in Fig. 1. There are  $n$  source nodes sending data to some destinations via one bottleneck  $B$ , which also has data to transmit for itself. Sources and the bottleneck are all within reach of each other, hence, share the same radio bandwidth and compete for channel access in a distributed manner without any central coordination function.

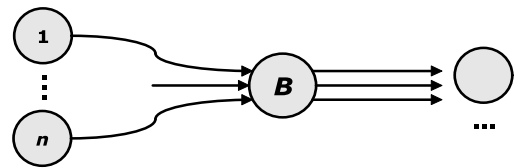


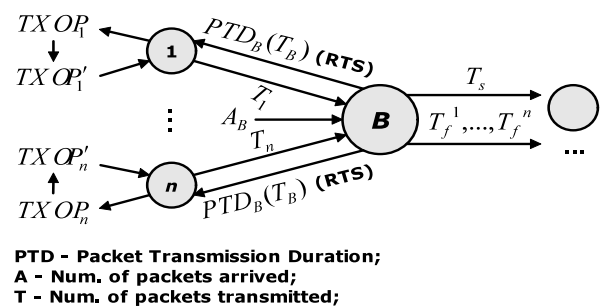
Figure 1. Bottleneck in a 2-hop ad-hoc network

#### A. TXOP control strategy

The key idea behind our TXOP control is that source nodes that are transmitting through the same bottleneck node adjust their TXOP limits based on feedback information obtained from the bottleneck. We will express the value of TXOP limits in terms of numbers of packets that can be transmitted within the chosen TXOP (as alternative notion of time).

The detailed TXOP control procedure is presented in Fig. 2. The bottleneck  $B$  is allowed to send as many packets as possible from its buffer, up to its TXOP limit,  $TXOP_{max}$ . As soon as  $B$  wins a channel competition, an RTS packet will be broadcast including its current expected Packet Transmission Duration information,  $PTD_B$ . This RTS message is not only received by the intended destination, but also by the source nodes. Therefore, each source  $i$  ( $i = 1, 2, \dots, n$ ) that has seen this  $PTD_B$  is informed about the number of packets that will be transmitted by  $B$  in its current channel access,  $T_B$ , and compare it with a general TXOP target,  $TXOP_{tar}$ . Note that the value of the  $TXOP_{tar}$  can be chosen according to the delay jitter requirements of different network systems. If  $T_B$  is smaller than the target value, the bottleneck  $B$  will be perceived to be lightly loaded, otherwise (e.g.,  $T_B = TXOP_{max}$ ), it indicates that  $B$  has many packets to transmit, i.e., is heavily loaded. Based on this feedback information, the TXOP limit of each source  $i$ ,  $TXOP_i$ , will be controlled to keep the  $T_B$  at around the  $TXOP_{tar}$  to avoid overload of the bottleneck. Note that  $TXOP_{tar}$  is also adopted by all sources as their maximum TXOP value. Increasing  $TXOP_i$  will result in more packets sent to  $B$ , which leads to a larger  $T_B$ . After an over-target  $T_B$  being fed back to the sources, the  $TXOP_i$  will be decreased, and then fewer packets can arrive at  $B$ , thus, the  $T_B$  will get smaller. As a result, with this feedback TXOP control strategy applied, fairness between the first and second hop can be achieved. The use of the RTS message for feedback realizes this control strategy at no additional cost in terms of bits to be transmitted.

In order to provide fairness among sources as a further requirement, we use the Additive Increase Multiplicative



**PTD** - Packet Transmission Duration;  
**A** - Num. of packets arrived;  
**T** - Num. of packets transmitted;

Figure 2. TXOP control procedure

Decrease (AIMD) mechanism known from TCP congestion control to decide the  $TXOP_i$  value. AIMD is known to provide fairness between sources [9]. The  $TXOP_i(\alpha, \beta)$  ( $\alpha \geq 1$ ,  $0 < \beta < 1$ ) control algorithm is expressed as follows:

$$TXOP_i' = \begin{cases} TXOP_i + \alpha, & PTD_B \leq TXOP_{tar}, \\ TXOP_i(1 - \beta), & PTD_B > TXOP_{tar}, \end{cases} \quad (1)$$

where  $\alpha$  and  $\beta$  are control gains used to decide the increase rate and decrease ratio of the  $TXOP_i$ , respectively, and  $TXOP_i'$  denotes the value of  $i$ 's TXOP limit *after* control. The influence on the system due to different  $\alpha$  and  $\beta$  settings will be discussed in Section V.

### B. Bottleneck transmission scheduling scheme

The presented TXOP control strategy aims at a fair capacity distribution between sources and the bottleneck  $B$ . However,  $B$  can have its own packets to transmit as well, as shown in Fig. 3, and thus the capacity should be fairly divided between  $B$ 's own and forwarded packets, while in total maintaining fairness with the sources. Therefore, the goal is to make all flows passing through the bottleneck get equal capacity from the network, regardless of the number of hops. A bottleneck transmission scheduling scheme is investigated for that, using multiple queues.

We see in Fig. 3 that there are two queues associated to the bottleneck  $B$ , one for  $B$ 's own packets and the other for  $B$ 's forwarding packets, the queue lengths of which are denoted by  $q_s$  and  $q_f$ , respectively. We assume that the number of flows sharing  $B$ 's link,  $N_B$ , can be detected, e.g., from the addresses used in the packet headers. Note that  $N_B$  includes the bottleneck itself if it also has packets to transmit.  $B$ 's total number of packet transmission per successful channel acquisition after scheduling,  $T_B$ , is equal to the sum of the packets allowed to be transmitted from both queues,  $T_s + T_f$ .

The proposed scheduling scheme is described in Fig. 4. Given the target for  $T_B$  to be  $TXOP_{tar}$  (see Section III-A), to achieve fairness,  $B$ 's own packets can take up the capacity  $TXOP_{tar} / N_B$ , and the rest (up to  $TXOP_{max}$ ) can be used for forwarding packets. We give higher priority to forwarding flows to avoid packet loss for packets that already consumed capacity in their first hop. However, we set capacity aside for local traffic to avoid starvation.

After checking  $q_s$ , if the capacity  $TXOP_{tar} / N_B$  is not filled, i.e.,  $q_s < TXOP_{tar} / N_B$ , the left capacity  $TXOP_{max} - q_s$  can be used for forwarding packet transmission. Otherwise, if there are

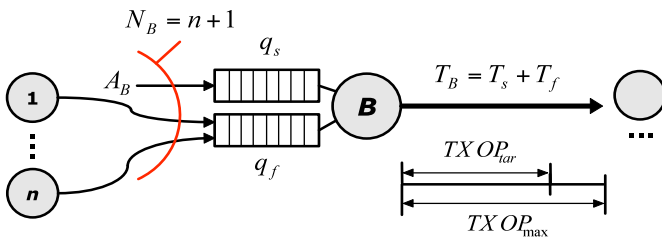


Figure 3. Bottleneck with transmission scheduling

---

**Input:**  $q_s, q_f, N_B,$

**Output:**  $T_s, T_f,$

---

**if**  $q_s \leq TXOP_{tar} / N_B$  **then**

$T_s = q_s;$

$T_f = \min(q_f, TXOP_{max} - q_s);$

**else**

**if**  $q_f < (N_B - 1) \times TXOP_{tar} / N_B$  **then**

$T_f = q_f;$

$T_s = \min(q_s, TXOP_{tar} - q_f);$

**else**

$T_s = TXOP_{tar} / N_B;$

$T_f = \min(q_f, TXOP_{max} - TXOP_{tar} / N_B);$

**end;**

**end;**

---

Figure 4. Bottleneck transmission scheduling scheme

only a small number of packets in the forwarding queue, the  $TXOP_{tar} - q_f$  may be allocated for transmitting  $B$ 's own packets. Note that in this case, the scheduled total packet transmission  $T_B$  is still limited by  $TXOP_{tar}$ . Thus, on the source nodes' side, after receiving the RTS generated for transmitting the current  $T_B$  packets, they will perceive the bottleneck  $B$  as lightly loaded all the same. As a result, their TXOP limits will be increased accordingly without being influenced by  $B$ 's own packet transmission.

## IV. RESOURCE CONTROL SYSTEM MODELING

In order to study the resource control mechanism, we first abstract from the random access method, and construct a basic control-theoretic model without considering the bottleneck's own packet transmission. Based on this model, we include random access behavior, leading to a control model with bottleneck transmission scheduling.

### A. Basic control model

We consider a scenario in which the bottleneck does not have packets to transmit itself. It will only forward packets received from the sources. The sources and bottleneck get the chance to transmit in turn, and a constant time duration  $R$  is assigned for one round transmission of all the nodes within the network (including one bottleneck and  $n$  homogeneous source nodes sharing the bottleneck link).

The expected change in the bottleneck queue length  $q_B(t)$  is determined by  $B$ 's current packet transmission, as well as the packet arrivals from all the  $n$  sources. For each source node  $i$  ( $i = 1, 2, \dots, n$ ), a pair of parameters  $\alpha$  and  $\beta$  is adopted to control the increase rate and decrease ratio of its TXOP limit  $TXOP_i(t)$ , respectively. Describing the TXOP control in the domain continuous,  $q_B(t)$  and  $TXOP_i(t)$  can be expressed by the following nonlinear differential equations (adapted from the stochastic model of TCP [10]) under the assumption that the bottleneck may empty its queue after each transmission epoch (i.e.,  $q_B(t) = T_B(t)$ ), and the TXOP limits of all source nodes

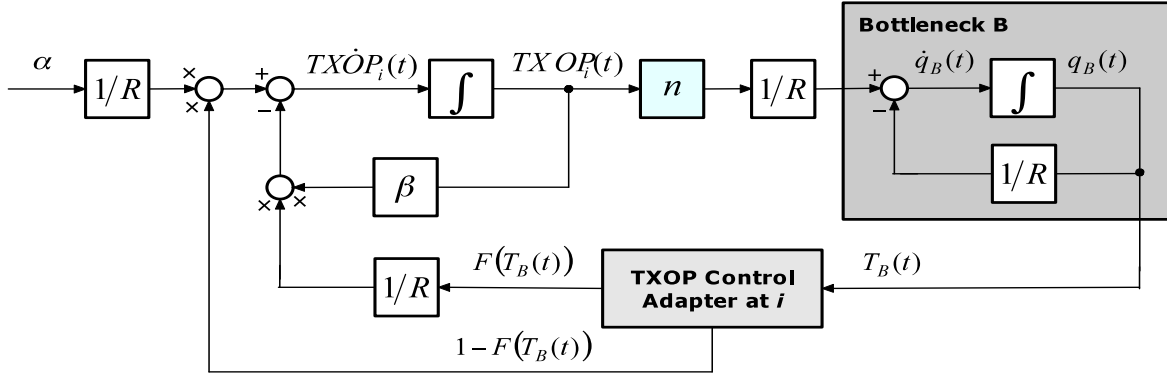


Figure 5. The guaranteed access based TXOP control model

will always be fully utilized, i.e., the sources are always saturated:

$$\begin{aligned} \dot{q}_B(t) &= -\frac{q_B(t)}{R} + \frac{n \cdot TXOP_i'(t)}{R}, \quad i = 1, \dots, n, \\ TXOP_i'(t) &= \frac{\alpha}{R} (1 - F(q_B(t))) - \frac{\beta \cdot TXOP_i'(t)}{R} F(q_B(t)), \end{aligned} \quad (2)$$

where

$$F(q_B(t)) = F(T_B(t)) = \mathbf{1}(T_B(t) > TXOP_{tar}), \quad (3)$$

is an indicator function used to switch between the additive increase and multiplicative decrease processes by comparing the bottleneck  $B$ 's current number of packet transmission  $T_B(t)$  with the target value  $TXOP_{tar}$ .

The block diagram presenting the differential equations is illustrated in Fig. 5, which highlights the TXOP control and queue dynamics of the bottleneck  $B$ . It is clear that  $T_B(t)$  announced through the currently received RTS is used as a feedback from  $B$  to control the TXOP of the  $n$  sources, which will further impact  $B$ 's following packet transmission (see multiplication block “ $n$ ” in Fig. 5).

### B. Control model with random access

In ad-hoc networks, all the stations with packets to transmit will contend for the channel. To model the behavior of the random channel access mechanism, we adopt a discrete time scale as presented in Fig. 6. Note that, in fact,  $t$  represents an epoch in discrete time and implicitly, the time interval following it. We ignore collisions among the nodes within the network for simplification, so that in each  $t$ , there will always be a successful Channel Access (CA) randomly won by a node  $j$ , either the bottleneck or a source node ( $j = B, 1, 2, \dots, n$ ).

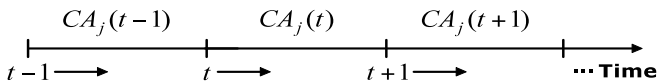


Figure 6. Timing scheme for random access ad-hoc networks

According to the timing scheme described above, a control-theoretic model for TXOP control in random access based ad-hoc networks has been designed and is illustrated in Fig. 7. We see that the integrator adopted in Fig. 5 is represented in a discrete manner by using the delay  $z^{-1}$ . Note that the TXOP adaptation behavior of the sources is the same, and is identically controlled by the bottleneck  $B$ 's current packet transmission  $T_B(t)$ . Therefore, we only model it once as shown in Fig. 7. The adapted TXOP limit of each source  $i$  for the following time interval,  $ATxop_i(t+1)$ , is equal to its TXOP limit at the beginning of  $t$ , plus the amount that has to be changed after the TXOP control in the current interval  $t$ ,  $CTxop_i(t)$ , that is,

$$ATxop_i(t+1) = TXOP_i(t) + CTxop_i(t). \quad (4)$$

In addition, the TXOP limit of each  $i$  should be no smaller than one, i.e., at least allowing one packet to be transmitted in case of a successful channel acquisition. Moreover, the general TXOP upper limit,  $TXOP_{tar}$ , still has to be applied to restrict its capacity occupancy in time interval  $t+1$ :

$$\begin{aligned} TXOP_i(t+1) &= \begin{cases} 1, & ATxop_i(t+1) < 1, \\ ATxop_i(t+1), & 1 \leq ATxop_i(t+1) \leq TXOP_{tar}, \\ TXOP_{tar}, & ATxop_i(t+1) > TXOP_{tar}. \end{cases} \end{aligned} \quad (5)$$

We also assume the saturated network condition that there are always enough packets at each  $i$  to fill its TXOP limit, thus, the current packet transmission  $T_i(t)$  will be determined by  $TXOP_i(t)$ .

In order to model the random channel access behavior of all the nodes within the network, a Random Access Controller (RAC) module has been included in this model as well. By judging the current Channel Access Request (CAR) of the bottleneck  $B$ ,  $CAR_B(t)$ , and the specified total number of active nodes  $n+1$  (including  $n$  sources and one bottleneck), the RAC provides the channel arbitration, and activates one of the nodes to access the medium by setting the corresponding Channel Access Opportunity (CAO) output,  $CAO_j(t)$ , to 1. Subsequently,



sharing  $B$ 's link in time interval  $t$  ( $N_B(t)$ ) is computed using the  $NF_B$  function block:

$$N_B(t) = \begin{cases} n, & q_s(t) = 0, \\ n+1, & q_s(t) > 0. \end{cases} \quad (7)$$

With  $N_B(t)$  combined with the current queue lengths  $q_s(t)$  and  $q_f(t)$ , the Transmission Scheduler function block determines the number of packet transmission from both queues in epoch  $t$  (denoted  $T_s(t)$  and  $T_f(t)$ ), in which the bottleneck transmission scheduling scheme described in Fig. 4 is implemented. Furthermore, the actual number of own and forwarding packets allowed to be transmitted by  $B$  in  $t$  (denoted  $AT_s(t)$  and  $AT_f(t)$ ) will be controlled by the  $CAO_B(t)$  input:

$$\begin{cases} AT_s(t) = CAO_B(t) \cdot T_s(t), \\ AT_f(t) = CAO_B(t) \cdot T_f(t). \end{cases} \quad (8)$$

For each queue of the bottleneck  $B$ , the queue length at the beginning of the following time interval  $t+1$  is equal to that at  $t$ , plus the number of packets that arrived and minus those transmitted within  $t$ :

$$\begin{cases} q_s(t+1) = q_s(t) + A_B(t) - AT_s(t), \\ q_f(t+1) = q_f(t) + T_i(t) - AT_f(t). \end{cases} \quad (9)$$

Besides, the current queue length of  $B$ ,  $q_s(t)$  and  $q_f(t)$  are used by the  $CAR_B$  function block to decide the  $CAR_B(t)$  output, that is,

$$CAR_B(t) = \begin{cases} 1, & q_s(t) + q_f(t) > 0, \\ 0, & q_s(t) + q_f(t) = 0. \end{cases} \quad (10)$$

## V. EXPERIMENTAL RESULTS

The modeled resource control systems are analyzed using 20-sim simulations [11]. All (mean) values of the simulation results and the corresponding 95% confidence intervals are obtained by performing 10 simulation runs of 10000 time intervals. Unless otherwise specified, the system parameters were used as specified in Table 1.

Note that all the simulations can be completed within a few seconds by using about 50% of the CPU time (Intel P4 CPU 3.0 GHz). The memory usage (no more than 5MB based on current parameter specifications) will be moderately increased when more nodes and number of time intervals are simulated.

### A. Results without bottleneck transmission scheduling

In this section, the control model with random access, as explained in Section IV-B, is studied without the bottleneck  $B$ 's transmission scheduling. We consider a network scenario in which one bottleneck  $B$  (with  $A_B(t) = 0$ ) contends for access to the medium with  $n$  source nodes in order to forward packets

TABLE I. SYSTEM PARAMETER SPECIFICATION

Parameter	Value
$n$	2
$\alpha$	1
$\beta$	0.5
$TXOP_{tar}$	12
$TXOP_{max}$	14

received from them towards other nodes. It is assumed that all the  $n$  sources are saturated, i.e., the TXOP limit of each source will always be reached per channel access. We also assume that the buffer size of the bottleneck  $B$  is big enough so that there will be no packet loss at  $B$ . The system response under this scenario is depicted in Fig. 10, where the performance shows  $B$ 's queue length and the TXOP limit of each source node  $i$  at each time epoch  $t$ . It is observed that  $q_B$  can be stabilized based on the dynamic control of the  $TXOP_i$  (confined to the interval  $[1, 12]$ ). The fluctuation of  $q_B$  is caused by the  $TXOP_i$  adaptation as well as the random access behavior, which can be compensated by adjusting the control parameters  $\alpha$  and  $\beta$ . Note that to evaluate the TXOP control, an infinite buffer size is adopted by  $B$ , so that the system performance will not be influenced by packet loss at  $B$ .

The expected queue length of the bottleneck  $B$ ,  $E[q_B]$ , and the expected TXOP limit of each  $i$ ,  $E[TXOP_i]$ , are studied by varying the number of source nodes  $n$ . The simulation results are shown in Fig. 11.

We see in Fig. 11(a) that the  $E[q_B]$  grows with increasing  $n$ . This is due to the fact that as more source nodes become active in the network, the number of opportunities for  $B$  to access the medium will decrease, which leads to a lower frequency of packet transmission as well as to TXOP adaptation since less RTS packets can be sent to control the  $TXOP_i$ . When  $n$  grows, the variance of  $q_B$  increases as seen from the larger confidence

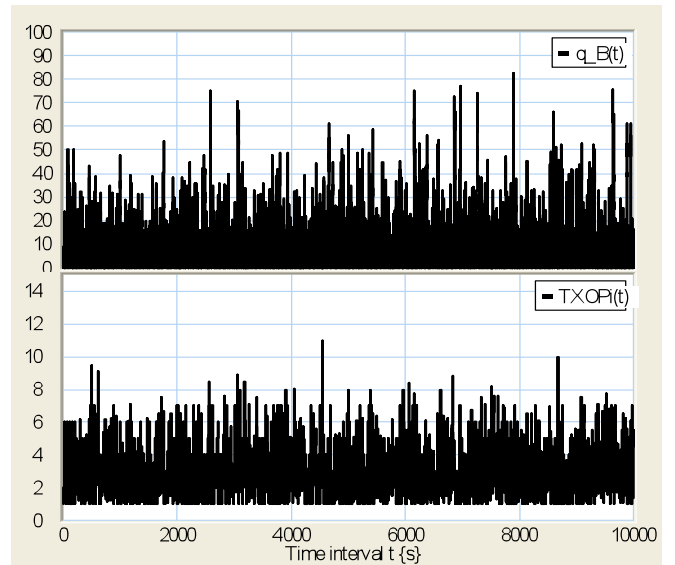


Figure 10.  $q_B(t)$  and  $TXOP_i(t)$  obtained with  $n = 2$ ,  $\alpha = 1$ ,  $\beta = 0.5$

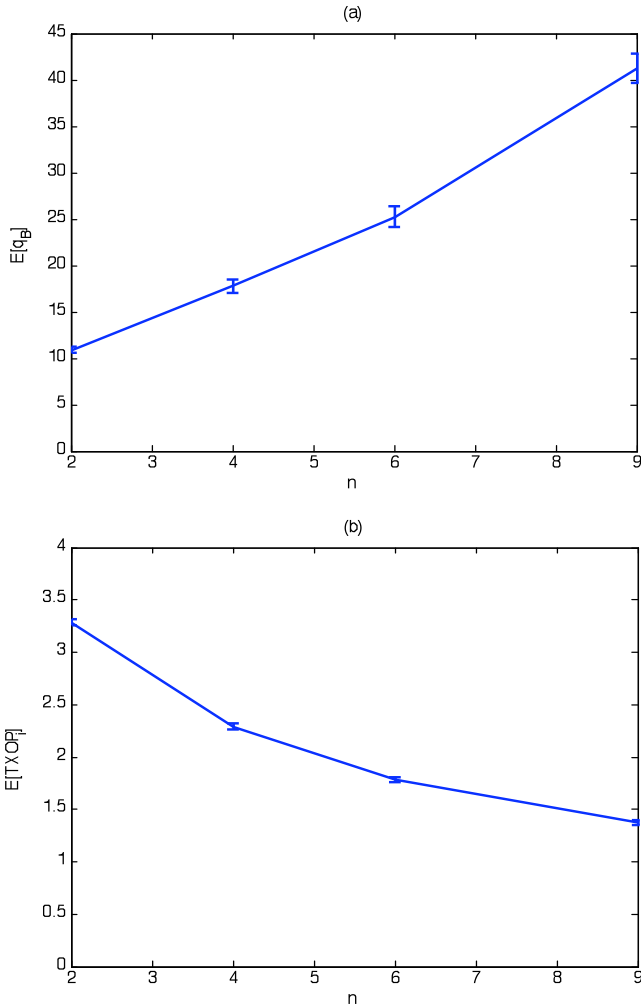


Figure 11. (a) Expected queue length of  $B$ ;  
(b) Expected TXOP limit of  $i$

intervals.

In Fig. 11(b), for smaller  $n$  each source  $i$  is allowed to transmit with a larger  $E[TXOP_i]$ . When  $n$  grows, the growing  $E[q_B]$  in Fig. 11(a) indicates that the workload of  $B$  is increased gradually; as a result, the  $TXOP_i$  will be controlled to be smaller in order to decrease the transmission rate of each source. This ensures that  $B$  will not become overloaded in case of more sources joining in.

To evaluate the efficiency, experiments are designed to compare the simulation results derived from the random access based model with and without the TXOP control strategy implemented. For the scenario without TXOP control, the bottleneck uses a TXOP limit of  $TXOP_{max}$ , whereas the sources are assumed to have their TXOP limits pre-configured to  $TXOP_{i\_ini}$ .

Based on the parameter setting  $n = 2$ , Fig. 12 shows the average number of packets transmitted by the bottleneck  $B$  per successful channel access with and without feedback control for TXOP. It is observed in Fig. 12 that without the TXOP control, the small  $TXOP_{i\_ini}$  restricts the packet transmission of

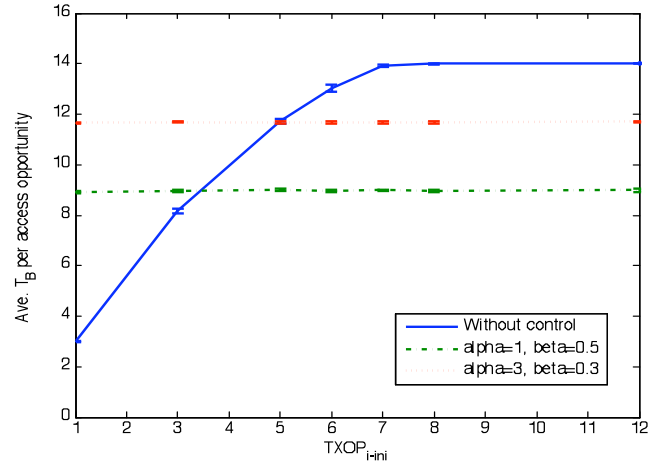


Figure 12. Average packet transmission of  $B$  per access opportunity

the sources, so that fewer packets can be transmitted by each source  $i$  as well as the bottleneck  $B$  per successful channel acquisition. Hence, comparing to the situation under larger  $TXOP_{i\_ini}$  settings, the network throughput is lower due to the fact that the channel competition occurs more frequently. As the  $TXOP_{i\_ini}$  is set to be larger, the bottleneck  $B$ 's TXOP limit,  $TXOP_{max} = 14$ , will be filled in the end. However, in this case  $B$  becomes overloaded with its queue length building up, which indicates that packets received from the sources can not be all forwarded by  $B$ , as shown in Fig. 13. Fig. 13 shows the fraction of packets forwarded, which is defined as the ratio of  $B$ 's total number of packet transmissions to its total number of packet arrivals within the simulation duration. Note that the  $TXOP_{i\_ini}$  is assigned as the initial value of the TXOP limit of each source node  $i$ . In the non-TXOP control case, it is a static value that can not be adapted.

Correspondingly, with our proposed TXOP control strategy, a relatively high average  $T_B$  can be achieved, which will not be influenced by varying the initial values of the  $TXOP_i$ , since the  $TXOP_{i\_ini}$  will be further adapted based on the current transmission of the bottleneck  $B$ . Enlarging the TXOP increase

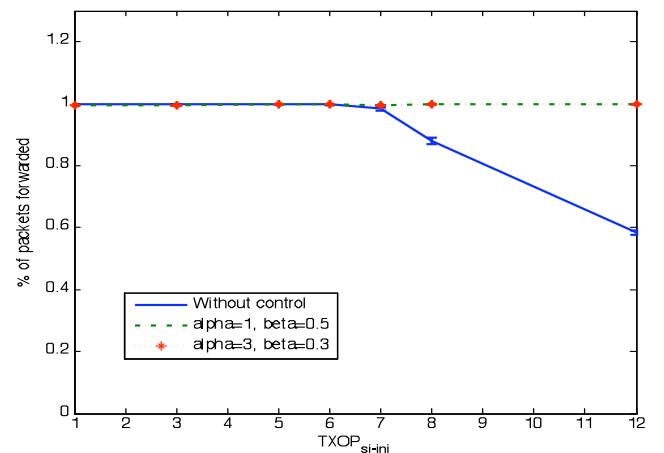


Figure 13. The fraction of packets forwarded

rate  $\alpha$  and decreasing the TXOP decrease ratio  $\beta$ , an even higher average  $T_B$  can be obtained. We see in Fig. 12 that under the control gain settings  $\alpha = 3$  and  $\beta = 0.3$ , the average  $T_B$  almost reaches the desired TXOP control target  $TXOP_{tar} = 12$ . Note that the change of the control gain may lead to a higher  $E[q_B]$  as a tradeoff, whereas, there is no influence on the fraction of packets forwarded, as shown in Fig. 13, since  $q_B$  may always be controlled and stabilized around a certain value.

Based on the simulation results reported in both Fig. 12 and Fig. 13, we see that with an appropriately selected static TXOP limit, e.g.,  $TXOP_{i\_ini} = 6$ , the uncontrolled system performance is still reasonable. However, if the number of active sources  $n$  would change over time,  $TXOP_{i\_ini}$  needs to be adapted, in order to maintain good performance.

### B. Results with bottleneck transmission scheduling

The performance of the TXOP control under the network condition considering  $B$ 's self-packet arrivals is evaluated by comparing the simulation results with and without the transmission scheduling scheme applied. Note that without transmission scheduling, results are obtained for the case that packets in  $B$ 's forwarding queue will be transmitted preferentially compared to its self-packets whenever  $B$  is allowed to use the medium. The parameter specification are as listed in Table 1, and each source  $i$ 's initial TXOP limit,  $TXOP_{i\_inis}$ , is set to be the maximum value,  $TXOP_{tar} = 12$ .

With a relatively high self-packet arrival rate  $A_B(t) = 10$ , the behavior of the TXOP limit of each source node  $i$  is

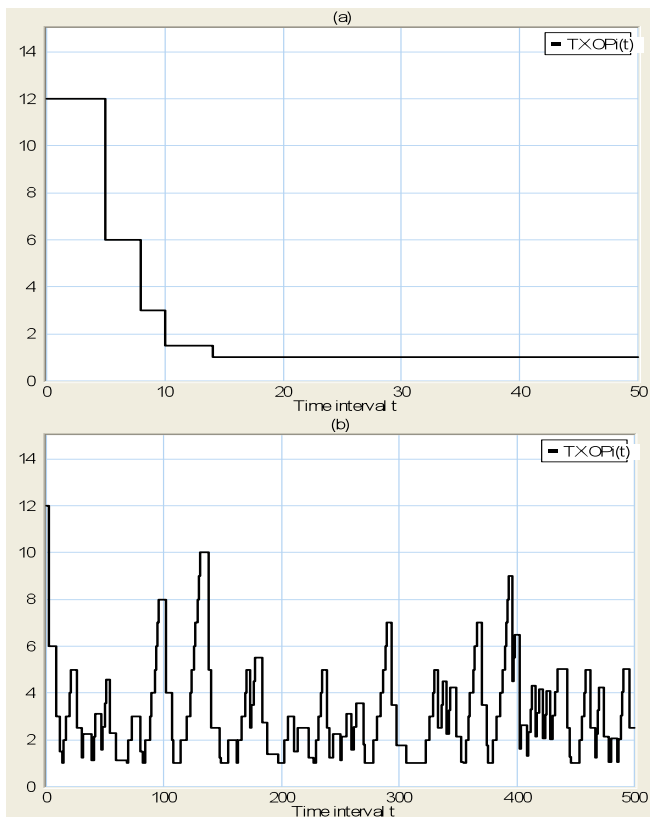


Figure 14. (a)  $TXOP_i(t)$  without scheduling; (b)  $TXOP_i(t)$  with scheduling

investigated at each time epoch  $t$ . The simulation results are shown in Fig. 14. We see in Fig. 14(a) that without  $B$ 's transmission scheduling, the  $TXOP_i$  will be soon cut back to 1 in this overload condition due to the fact that the bottleneck  $B$  is always perceived to be busy. As a result, most capacity of  $B$  will be consumed by  $B$ 's self-packet transmission, which is unfair for all the other flows. This can be improved by implementing the transmission scheduling at  $B$ , so that  $B$ 's own packet transmission may be reduced and thus the TXOP control will not be disturbed by the large number of  $B$ 's own packet arrivals, as presented in Fig. 14(b).

Fig. 15 shows the average number of  $B$ 's own and forwarding packet transmission per successful channel access with and without the transmission scheduling scheme implemented. As the value  $A_B(t)$  is increased, part of  $B$ 's capacity will be used to transmit its own packets, which results in a decrease of the average  $T_f$ . In the non-scheduling case, as presented in Fig. 15(a),  $T_f$  will finally drop to 2 (approximately), since the TXOP limit of each source  $i$  will get restricted to 1 under the overloaded network condition, so on average the bottleneck will only forward 1 packet for each of the two sources per transmission opportunity. Hence, most of the capacity will be consumed by  $B$ 's own packet transmission. With the scheduling scheme applied, the average  $T_f$  can be

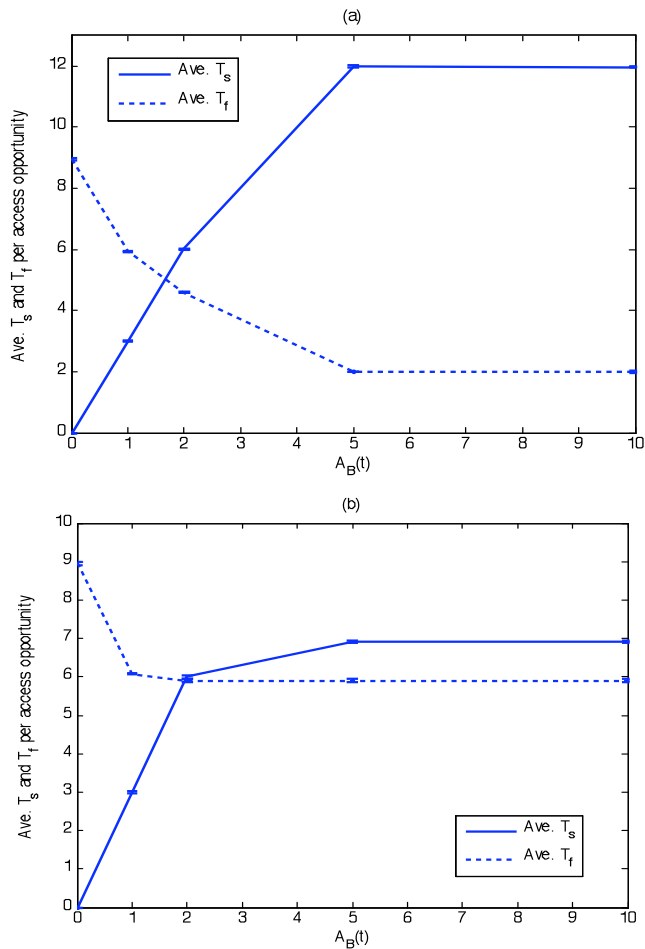


Figure 15. (a) Average  $T_s$  and  $T_f$  per access opportunity without scheduling; (b) Average  $T_s$  and  $T_f$  with scheduling



maintained around 6, as we see in Fig. 15(b), no matter how large  $A_B(t)$  is. Note that as a result of the AIMD saw-tooth behavior, the average  $T_f$  can not reach the desired value 8 (two third of the target value  $TXOP_{tar}$ ; the other one third is for  $B$ 's own packet transmission). Instead, it switches between 8 and 4, resulting in an average value of 6. However, this can be further improved by tuning the parameter settings of the control gains  $\alpha$  and  $\beta$ .

## VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented a feedback resource control mechanism with bottleneck transmission scheduling in 2-hop ad-hoc networks. The TXOP limits of all the sources are dynamically adapted, based on the current RTS packet received from the bottleneck. There are no requirements on 802.11 modifications and there is no control overhead introduced. The proposed approach can be applied in real ad-hoc system without the intervention of a centralized controller. The resource control mechanism is modeled in a control-theoretical way. The model has been implemented and studied using the software package 20-sim. Compared to systems without TXOP control, results show that our new control mechanism is able to provide fair and efficient bandwidth utilization in different traffic load conditions. The mechanism avoids packet loss in a bottleneck node due to a mismatch between traffic load and transmission opportunities. Further, our scheduling mechanism fairly divides capacity between local and forwarded traffic.

As future work, the proposed control model will be validated using the simulation platform OPNET. Next to that, also the adaptation of control parameters  $\alpha$  and  $\beta$  in dynamic network environments as well as QoS differentiation among the sources will be considered. The adaptive control of the other service differentiation parameters defined in IEEE 802.11e, e.g.,  $CW_{min}$ ,  $m$  and  $AIFS$ , will also be investigated. Furthermore, we will investigate *multi*-hop wireless ad-hoc networks (instead of just 2-hop), in which bottlenecks may appear anywhere.

## REFERENCES

- [1] IEEE 802.11 WG, "Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", *IEEE*, 1999.
- [2] IEEE 802.11 WG, "Draft Supplement to Part II: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)", *IEEE 802.11e*, Jan. 2005.
- [3] Y. Yang, B.R. Haverkort, and G. Heijenk, "A Control-theoretic Modeling Approach for Service Differentiation in Multi-hop Ad-hoc Networks", *Proceedings of 5<sup>th</sup> International Conference on the Quantitative Evaluation of Systems*, IEEE CS Press, France, Sep 2008, pp. 7-16.
- [4] A. Remke, B.R. Haverkort, G. Heijenk, and L. Cloth, "Bottleneck Analysis for Two-hop IEEE 802.11e Ad Hoc Networks", *Proceedings of 15<sup>th</sup> International Conference on Analytical and Stochastic Modeling Techniques and Applications, LNCS 5055*, Springer Verlag, Cyprus, Jun 2008, pp. 279-294.
- [5] J. Majkowski and F.C. Palacio, "Dynamic TXOP Configuration for QoS Enhancement in IEEE 802.11e Wireless LAN", *Proceedings of the 2006 International Conference on Software, Telecommunications and Computer Networks*, ACM, Croatia, Sep 2006, pp. 66-70.
- [6] Binh Ngo and Steven Gordon, "Avoiding Bottlenecks Due to Traffic Aggregation at Relay Nodes in Multi-hop Wireless Networks",

*Proceedings of the 11<sup>th</sup> Asia-Pacific Conference on Communications*, IEEE Press, Western Australia, Oct 2005, pp. 769-773.

- [7] K. Wang, F. Yang, Q. Zhang, and Y. Xu, "Modeling Path Capacity in Multi-hop IEEE 802.11 Networks for QoS Services", *IEEE Transactions on Wireless Communications*, 6(2):738-749, 2007.
- [8] P.C. Ng, and S.C. Liew, "Throughput Analysis of IEEE 802.11 Multi-hop Ad Hoc Networks", *IEEE/ACM Transactions on Networking*, 15(2):309-322, 2007.
- [9] J.F. Kurose and K.W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, Addison-Wesley Longman, Third Edition, 0-321-26976-4.
- [10] [V. Misra, W.B. Gong, and D. Towsley, "Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows", *Proceedings of the ACM SIGCOMM 2000*, ACM, Sweden, Aug 2000, pp. 151-160.
- [11] [<http://www.20sim.com/>].