

# Low-Complexity Channel-Aware Scheduling for Multichannel Wireless Local Area Networks

<sup>†</sup>Yihan Li, <sup>†</sup>Shiwen Mao, <sup>†</sup>Prathima Agrawal, and <sup>‡</sup>Scott F. Midkiff

**Abstract**—In multi-channel wireless networks, effective scheduling algorithms are indispensable for fully harvesting the benefit of multiple channels. Since scheduling decisions need to be made quickly under time-varying wireless channels and traffic dynamics, low-complexity algorithms are especially appealing. In this paper, we study the problem of scheduling in multichannel wireless local area networks. We present a new class of polling service-based channel-aware scheduling algorithms, termed Channel-Aware Scheduling with Exhaustive Service and Hamiltonian Walk (CESH). The three proposed scheduling schemes are shown to have significantly reduced computational complexity, but achieve comparable delay performance as Maximum Weight Matching scheduling, the well-known optimal scheme, under various traffic patterns via simulation studies. The proposed schemes provide practical and effective solutions to the scheduling in multi-channel wireless networks.

## I. INTRODUCTION

There has been considerable interest in multi-channel wireless networks recently [1]. In such networks, concurrent transmissions can be carried out in a neighborhood on different channels without interfering with each other, thus effectively enhancing the network throughput performance. In order to fully harvest the benefit of multiple channels, effective scheduling is indispensable to decide which nodes can transmit and on which channel they transmit. An effective scheduling algorithm should maximize the number of concurrent transmissions. In addition, low-complexity schemes, in terms of control overhead and computational complexity, are highly appealing since scheduling decisions must be made quickly in response to the time-varying wireless channels and traffic dynamics.

Several researchers have studied the problem of effective scheduling in multichannel wireless networks [1], [2], [3], [4], [5], [6], [7], [8], [9]. In [3], reservation-based medium access control scheduling was considered in multichannel wireless local area networks (WLANs). In [5], scheduling algorithms in multichannel access point networks under infrequent channel and queue length measurements were presented. In [4], link-layer protocols for utilizing multichannel in mesh networks were proposed. In [6], [7], [8], [9], joint design of multi-channel control and multi-path routing were examined.

This work was supported in part by the National Science Foundation under Grant ECCS-0802113, and through the Wireless Internet Center for Advanced Technology (WICAT) at Auburn University.

Y. Li, S. Mao and P. Agrawal are with Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA. Email: yli@auburn.edu, smao@ieee.org, pagrawal@eng.auburn.edu.

S. F. Midkiff is with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA. Email: midkiff@vt.edu.

In this paper, we study the problem of improving the efficiency of scheduling in multichannel wireless networks. We assume a WLAN with multiple mobile nodes and one access point. There are multiple orthogonal channels. Each mobile node has one transceiver that can tune to any of the channels, but each channel can be accessed by only one node at any time. The access point has a transceiver for each channel, and can receive/transmit at all channels at the same time. We model the multichannel scheduling problem as a *matching problem* in a bipartite graph. Since the condition of a wireless channel varies from time to time, it has to be taken into account when assigning channels to mobile nodes. We therefore propose a new class of polling service-based channel-aware scheduling schemes for such multichannel WLANs.

Polling is a general way of multiplexing service requests for a single server from multiple stations [10]. Once a station wins the server, there are three types of service policies to serve its packets: (i) *exhaustive policy*, where the server serves a station until its buffer is emptied; (ii) *gated policy*, where the server serves for a station only those requests which are already buffered in the station when this service period begins; and (iii) *limited-k policy*, within which a station is served until either the buffer is emptied or the first  $k$  buffered requests are served, whichever occurs first. One special case of the limited- $k$  service is the limited-1 service, where at most one request is served during each service period (as in most existing MAC protocols). It has been shown that both exhaustive service and gated service are more efficient than the limited- $k$  service, and they can guarantee bounded delay as long as the offered load is strictly less than 100% [10].

Most of existing scheduling schemes for medium access control use the limited-1 service, so that only one data frame is transmitted when a mobile node accesses the channel. Motivated by insights from polling system theory, we propose to use exhaustive service to improve the scheduling efficiency. The main idea is to serve multiple data frames after a successful contention resolution, thus amortizing the high control overhead over multiple data frames and making the protocols more efficient. We present three such scheduling schemes and study their delay performance via simulations. We show that all the three proposed schemes achieve low average delay with considerable reduced computational complexity, as compared with an existing approach [5].

The remainder of this paper is organized as follows. We first define the polling service-based channel-aware scheduling approach in section II, and then present its three varia-

tions in Section III. The delay performance of the proposed schemes are studied via simulation and compared to existing schemes in IV. Section V concludes this paper.

## II. CHANNEL-AWARE SCHEDULING WITH EXHAUSTIVE SERVICE AND HAMILTONIAN WALK

We consider the uplink of a WLAN with  $N$  nodes,  $M$  channels, and an access point (termed an  $N \times M$  system throughout this paper). Scheduling is used to decide which nodes transmit data to the access point and which channel is used by each scheduled node. For simplicity, we assume that each node has only one transceiver,<sup>1</sup> and the access point has a separate transceiver for each channel. Therefore, a node can transmit on at most one channel, while the access point can receive data from up to  $M$  nodes at a time. Only one node can transmit in a channel at a time. We assume that time is slotted, and packet arrivals and departures only occur at the beginning and the end of a time slot, respectively. The length of a time slot equals the transmission time of a packet. Each node maintains  $M$  virtual queues (VQs), one for each channel [5]. The arrival rate of  $VQ_{ij}$ , the  $j$ th virtual queue of node  $i$ , is denoted as  $\lambda_{ij}$ .

Scheduling in such an  $N \times M$  system can be modeled as a matching problem in a bipartite graph shown in Fig. 1, where a node and a channel are connected by an edge if the weight of the corresponding VQ (also defined as the weight of the edge) is larger than 0. If a node is matched to a channel in a time slot, a packet will be transferred from the node to the access point over that channel. As each node has only one transceiver, a node can be matched to at most one channel, and a channel can be matched to at most one node.

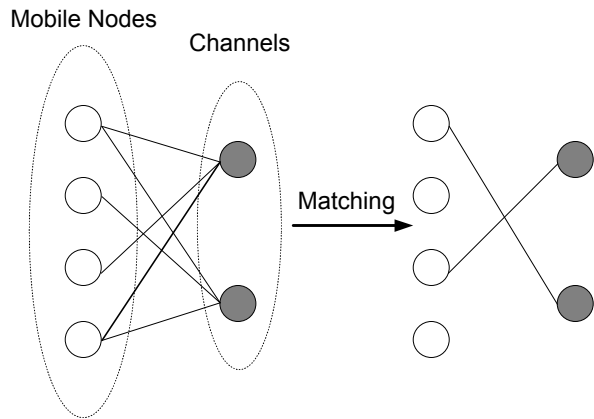


Fig. 1. Assigning two channels to four mobile nodes: modeled as a matching problem in the bipartite graph.

At time  $t$ , let  $Q(t) = [q_{ij}]_{N \times M}$ , where  $q_{ij}$  is the queue length of  $VQ_{ij}$ , for  $i = 1, 2, \dots, N$  and  $j = 1, 2, \dots, M$ . We define the weight of the edge for  $VQ_{ij}$  as

$$q'_{ij} = \beta_{ij} q_{ij}, \quad (1)$$

where  $\beta_{ij} \in [0, 1]$  is determined by the status of the  $j$ th channel measured or estimated at node  $i$ . For the two-state

<sup>1</sup>The proposed schemes can be easily extended to the case of multiple transceivers.

on-off channel considered in this paper,  $\beta_{ij} = 1$  when the  $j$ th channel at node  $i$  is on, and  $\beta_{ij} = 0$  when the  $j$ th channel at node  $i$  is off. We denote  $Q'(t) = [q'_{ij}]_{N \times M}$ , the weight matrix. The channel assignment at time  $t$  is termed *schedule* and denoted by  $S(t)$ . The weight of a schedule  $S(t)$ , which is the sum of the weight of all matched VQs, is denoted by  $W(t) = \langle S(t), Q'(t) \rangle$ .

In a *Channel-aware Exhaustive Service Matching Algorithm* (CES), if a node is matched to a channel in time slot  $t - 1$  and the weight of the corresponding VQ is larger than 0 at the beginning of time slot  $t$ , the same match will be used in time slot  $t$ . A matching algorithm will be used to seek matches among all unmatched nodes and channels in time slot  $t$ .

A *Hamiltonian Walk* is a walk that visits every vertex of a graph exactly once. For an  $N \times M$  system, the total number of possible matches is  $\binom{N}{M} M!$ . If these matches are mapped to a graph with  $\binom{N}{M} M!$  vertices so that each vertex corresponds to a match, a Hamiltonian Walk on the graph visits each vertex exactly once every  $\binom{N}{M} M!$  time slots. The vertex which is visited at time  $t$  is denoted by  $H(t)$ . Since the complexity of generating  $H(t+1)$  from  $H(t)$  is  $O(1)$  [11], computing schedule using Hamiltonian Walk does not increase the implementation complexity in a CES scheme.

Let  $S(t)$  be the schedule used at time  $t$ . The *Channel-aware Exhaustive Service with Hamiltonian Walk* (CESH) scheduling is defined as follows:

- Step 1: At time  $t + 1$ , use CES to generate match  $Z(t + 1)$  based on the previous schedule  $S(t)$ .
- Step 2: Generate  $H(t + 1)$  from  $H(t)$  from the Hamiltonian Walk.
- Step 3: The schedule for time  $t + 1$  is computed as:

$$S(t + 1) = \operatorname{argmax}_{S \in \{Z(t+1), H(t+1)\}} \langle S, Q'(t + 1) \rangle.$$

Note that Exhaustive Service Matching with Hamiltonian Walk algorithm, a special case of CESH with perfect channel condition, have been used in high-speed fixed length packet switches [15], [16].

## III. LOW-COMPLEXITY CESH SCHEMES

It is well-known that Maximum Weight Matching (MWM), which finds the match with the maximum weight in any time slot, usually achieves the best performance [12], [13]. However, the complexity of MWM for an  $N \times M$  bipartite graph is  $O(MN^2)$ . Such high computational complexity makes it unsuited for practical implementation. In this section, we present three variations of CESH executed at the access point, which achieve comparable delay performance but with greatly reduced complexity as compared with MWM.

### A. CESH-Maximal Weight Matching

A *Maximal Matching* algorithm leads to a *maximal match* by adding connections incrementally, without removing connections made earlier in the matching process. In a maximal match, if a non-empty input is not matched to any output,

all destination outputs of the cells waiting in this input must be matched to other inputs. Note that Maximal Matching algorithm belongs to the class of greedy algorithms. It achieves suboptimal matchings but with low-complexity, as compared with MWM.

In CESH-Maximal Weight Matching (CESH-MLWM),  $Z(t+1)$  is generated by a Channel-aware Exhaustive Service Maximal Weight Matching (CES-MLWM). CES-MLWM works as follows at time  $t+1$ :

- Step 1: If a node  $i$  was matched to a channel  $j$  in time slot  $t$ , and  $q'_{ij} > 0$ , keep this pair of node/channel matched. All other nodes and channels are considered as unmatched.
- Step 2: for each unmatched channel, among all VQs in unmatched nodes destined for this channel, pick the one with the largest weight.
- Step 3: Among all picked VQs, select the one with the largest weight and match the corresponding node and channel. Ties are broken randomly.
- step 4: If there is any unmatched channel and at least one VQ for unmatched channels has a weight larger than 0, go to Step 2; otherwise, stop.

It can be shown that the complexity of CES-MLWM (and also for CESH-MLWM) is  $O(M \log N)$ , which is a great reduction from that of MWM.

### B. CESH-Weighted Multiple Iteration Matching

Multiple iterations are used to converge to a maximal matching in multiple iteration matching (MIM), and to converge to a maximal weighted matching in weighted multiple iteration matching (WMIM). The MIM and WMIM used in our scheme are motivated by iSLIP in an input queued switch [14]. In CES-Weighted Multiple Iteration Matching (CES-WMIM),  $Z(t+1)$  is generated at time  $t+1$  as follows.

For nodes and channels matched in time slot  $t$ , if the weight of the corresponding VQ is larger than 0 at the beginning of time slot  $t+1$ , we say that node and channel are matched in time slot  $t+1$ ; otherwise, set the node and channel as unmatched at the beginning of time slot  $t+1$ . For all the nodes and channels which are unmatched at the beginning of time slot  $t+1$ , CES-WMIM tries to match them in multiple iterations, with three steps in each iteration:

- Step 1: Request. Each unmatched node sends a request to every channel along with the weight of the corresponding VQ if it is larger than 0.
- Step 2: Grant. If an unmatched channel receives any requests, it chooses the one with the largest weight to grant. Ties are broken randomly.
- Step 3: Accept. If a node receives any grant, it accepts the one with the largest weight. The node and channel are therefore matched. Ties are broken randomly.

Note that the request, grant and accept actually occur in the access point based on the queue status and channel status it collects. There is no need to transmit such control messages between the access point and mobile stations. The implementation complexity of one iteration is  $O(\log N)$ . Our

simulation result shows that  $\log M$  iterations are sufficient for converging to a Maximal Weight Match. Therefore, the complexity for CESH-WMIM to achieve Maximal Weight Matching is  $O(\log N \log M)$ , a further reduction from CESH-MLWM.

### C. CESH-Multiple Iteration Matching

In CESH-Multiple Iteration Matching (CESH-MIM),  $Z(t+1)$  is generated using CES-MIM as follows. For nodes and channels matched in time slot  $t$ , if the weight of the corresponding VOQ is larger than 0 at the beginning of time slot  $t+1$ , we say that node and channel are matched in time slot  $t+1$ ; otherwise, set the node and channel as unmatched at the beginning of time slot  $t+1$ . For all the nodes and channels which are unmatched at the beginning of time slot  $t+1$ , CES-MIM tries to match them in multiple iterations, with three steps in each iteration:

- Step 1: Request. Each unmatched node sends a request to all unmatched channels if the weight of the corresponding VQ is larger than 0.
- Step 2: Grant. If an unmatched channel receives any requests, it chooses one of them in a fixed round-robin order starting from the current position of the pointer. Note that the output pointer points to the granted input if the grant is accepted in Step 3.
- Step 3: Accept. If a node receives any grant, it accepts one of the multiple grants in a fixed round-robin order starting from the current position of the pointer. The node and the channel are therefore matched, and the node pointer then points to the matched channel.

As in CESH-WMIM, the request, grant and accept operations actually take place in the access point based on the VQ status and channel status it collects from the mobile nodes. The implementation complexity of one iteration is  $O(\log N)$ , and  $\log M$  iterations are good enough to converge to a Maximal Match. Therefore, the complexity for HE-MIM to achieve maximal Matching is  $O(\log N \log M)$ , the same as CESH-WMIM. However, CESH-MIM is the simplest among the three proposed schemes, since it does not compare weights.

## IV. DELAY PERFORMANCE

In this section, the simulated performance of average delay is presented under uniform and nonuniform traffic and compared to the MWM scheduling schemes with infrequent queue measurement [5]. We show that our new schemes achieve comparable delay performance as MWM with much lower complexity.

### A. Network Setting

Since all data packet are destined to the access point, they can be transmitted on any channel. As in [5], when a packet arrivals, it joins the shortest queue among all queues in the node. A two-state on-off channel model from [17] is used in our simulation as shown in Fig. 2.

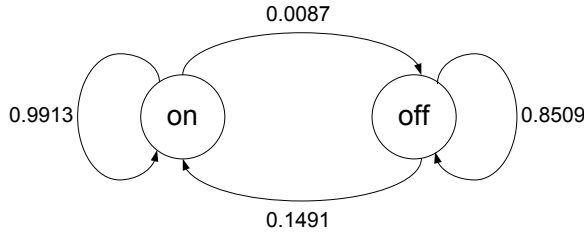


Fig. 2. Channel model state transition diagram.

In [5], time slots are grouped into intervals of  $T$  slots, and queue lengths are measured only at the beginning of each  $T$ -slot interval. This is a reasonable assumption since the overhead on delivering the queue length information for all the  $N \times M$  queues to the access point in every time slot is too high and will significantly reduce the throughput of data packets.

Our new schemes are compared to two scheduling approaches in [5], termed TMWM-1 and TMWM-2 in this paper. In TMWM-1, a maximum weight matching is selected only at the beginning of each  $T$ -slot interval, and the same matching is used for all the time slots in that interval. The weight of a matching  $S(kT)$  computed at time  $kT$  is given by  $W(kT) = \sum_{(i,j) \in S} \hat{\beta}_{ij}(kT) q_{ij}(kT)$ , where  $\hat{\beta}_{ij}(kT)$  is the average channel rate until the next channel measurement instance, given the current channel rate. In TMWM-2, the channel state measurement and scheduling decisions are made in every time slot, while the queue measurements are made only once every  $T$ -slot interval. The weight of a matching  $S(kT)$  computed at time  $t \in [kT, (k+1)T)$  is given by  $W(kT) = \sum_{(i,j) \in S} \beta_{ij}(t) q_{ij}(kT)$ . Note that due to the high complexity of MWM, TMWM-2 may not be suited for even moderate-sized networks.

In this paper, similar to TMWM-2, the channel state is measured and CESH is executed in every time slot, while the accurate queue length information is updated every  $T$  time slots. If queue length information is needed in the scheduling, rather than using  $q_{ij}(kT)$  to replace the  $q_{ij}(t)$  in (1) for  $t \in [kT, (k+1)T)$ , as in TMWM-2 [5], the proposed schemes use  $w_{ij}(t)$  instead, which will be defined shortly. Since the complexity of the three new CESH schemes is pretty low, there is no problem to run them in every time slots.

We assume that the access point obtains channel status information by exchanging short messages with mobile nodes. In every  $T$  time slots, the queue length of each queue is sent to the access point, and  $w_{ij}(kT) = q_{ij}(kT)$ . In every other time slot, each queue informs the access point if there is an arrival to this queue in the current time slot, which only takes 1 bit in the short message for channel measurement. If there is an arrival to  $VQ_{ij}$  at time  $t$ ,  $t \neq kT$ , and the short message is received by the access point (when the channel is on),  $w_{ij}(t) = w_{ij}(t-1) + 1$ ; otherwise,  $w_{ij}(t) = w_{ij}(t-1)$ .<sup>2</sup> By the end of each time slot, if there is a departure from

<sup>2</sup>If channel  $j$  at node  $i$  is off, the short message won't be received by the access point. However the corresponding weight  $q'_{ij} = \beta_{ij} w_{ij} = 0$  since  $\beta_{ij} = 0$ . The matching decision won't be affected.

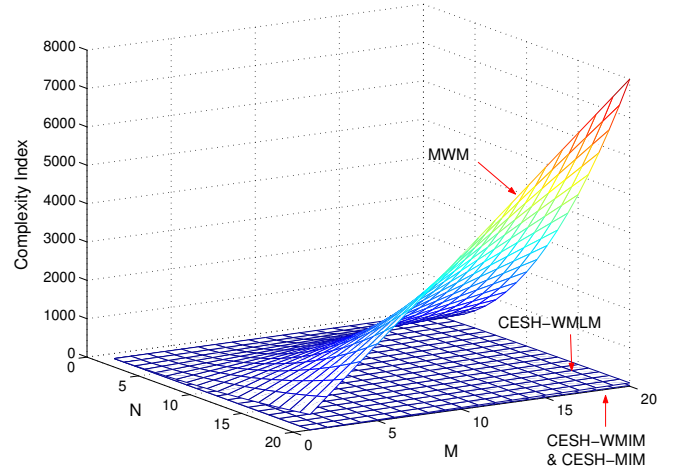


Fig. 3. Complexity of the scheduling algorithms.

$VQ_{ij}$ ,  $w_{ij}(t)$  will be decreased by 1. By using  $w_{ij}(t)$  instead of  $q_{ij}(kT)$ , the queue length information used in scheduling is more accurate, so that a “better” matching can be found and better performance can be achieved.

In our simulation for TMWM-2, the weight of a matching  $S(kT)$  computed at time  $t \in [kT, (k+1)T)$  is given by  $W(kT) = \sum_{(i,j) \in S} \beta_{ij}(t) w_{ij}(kT)$ , which we found achieves better performance than the original scheme in [5].

In CESH-MIM, no queue length information is needed for scheduling decision. The centralized scheduler only needs to know the binary information if the queue is empty or not, which only take one bit for each queue and can be piggybacked in the channel measurement message sending at the beginning of each time slot.

### B. Complexity

In this section, we consider all schemes in a  $6 \times 4$  system as in [5], and our proposed schemes in a  $20 \times 6$  system. TMWM-1 and TMWM-2 are not considered in the latter WLAN because their complexity is too high and it takes a long time to compute a matching in the latter case (i.e., unlikely to be finished within a time slot). A qualitative comparison of the complexities of all the schemes are presented in Fig. 3. We can see that the complexity of the two MWM schemes, TMWM-1 and TMWM-2, increases significantly when  $N$  and  $M$  increase. Even for moderate-sized WLANs, the complexity may be too high for execution in every time slot. However, this is not the case for the proposed CESH schemes, which have greatly reduced computational complexity.

### C. Delay Under Uniform Traffic

Each point in the following figures is the average of either 10 or 20 simulation runs. We also plot the 95% confidence intervals in the following figures, which are generally very small.

In Fig. 4, we compared CESH-MLWM, CESH-WMIM and CESH-MIM to TMWM-1 and TMWM-2 in terms of

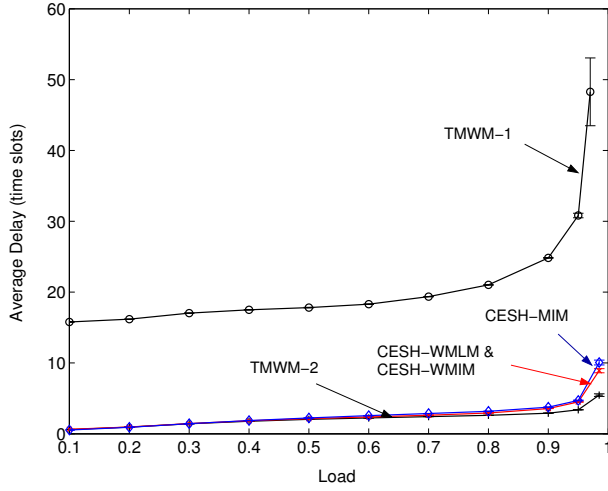


Fig. 4. Average delay under uniform traffic in a 6x4 system.

average delay in a  $6 \times 4$  system under uniform traffic. The load of the system shown in figures is defined as  $\sum_{ij} \lambda_{ij}/M$ , where  $\lambda_{ij}$  is the arrival rate of  $VQ_{ij}$ . In all our simulations, we find that the CESH-MLWM curve largely overlaps with the CESH-WMIM curve. This is because CESH-WMIM converges to a maximal weighted matching with multiple iterations. Therefore, we only plot a single curve for both schemes for clarity in all the figures shown in this section.

In addition, we find that TMWM-2 always has the lowest average delay and TMWM-1 always has the highest average delay. The average delays of CESH-MLWM, CESH-WMIM and CESH-MIM are comparable, while the average delay of CESH-MIM is slightly higher than the other two. All the three CESH curves lie between the two TMWM curves. However, the CESH schemes have very close average delay values as TMWM-2 for all the loads we examined, although with significant reduction in computational complexity.

The average delay of CESH-MLWM, CESH-WMIM and CESH-MIM under uniform traffic in a  $20 \times 6$  system is shown in Fig. 5. The average delay of CESH-MIM is a little bit higher than those of CESH-MLWM and CESH-WMIM. Compared to Fig. 4, the average delay for each scheme becomes larger, but not significantly, when both  $N$  and  $M$  are increased.

#### D. Delay Under Nonuniform Traffic

We also examine the performance of the scheduling algorithms under nonuniform traffic. The nonuniform traffic pattern used in this section is defined as

$$\lambda_i = \begin{cases} 2\lambda, & 1 \leq i \leq \lfloor N/2 \rfloor \\ \lambda, & \lfloor N/2 \rfloor + 1 \leq i \leq N, \end{cases} \quad (2)$$

where  $\lambda_i$  is the arrival rate for node  $i$ . Similar observations are made for other nonuniform traffic pattern but are omitted for brevity.

The average delays of five scheduling algorithms in a  $6 \times 4$  system under nonuniform traffic are shown in Fig. 6. Similar to the case of uniform traffic, TMWM-2 always has the

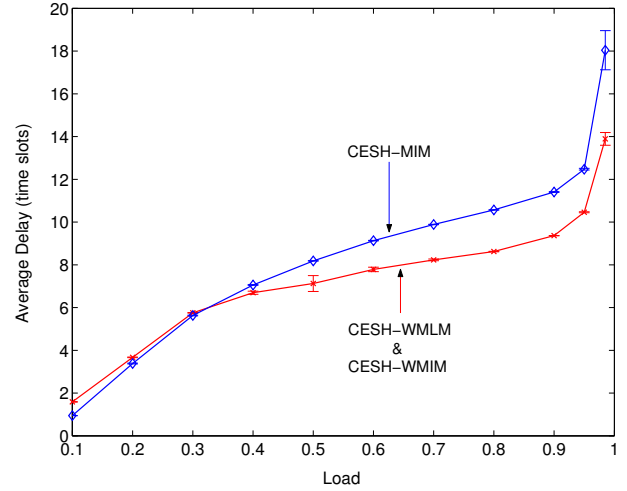


Fig. 5. Average delay under uniform traffic in a 20x6 system.

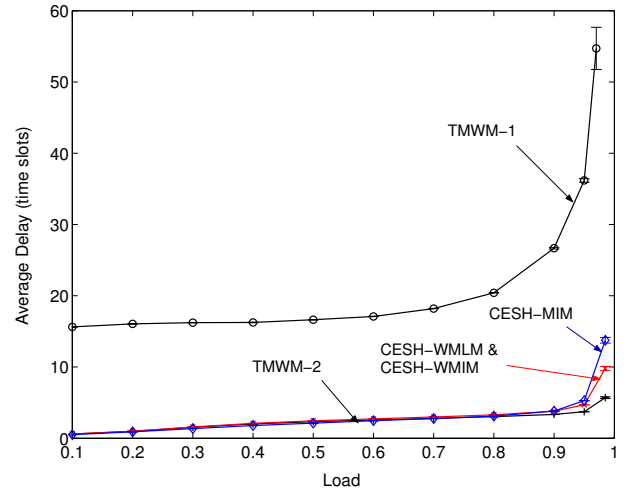


Fig. 6. Average delay under nonuniform traffic in a 6x4 system.

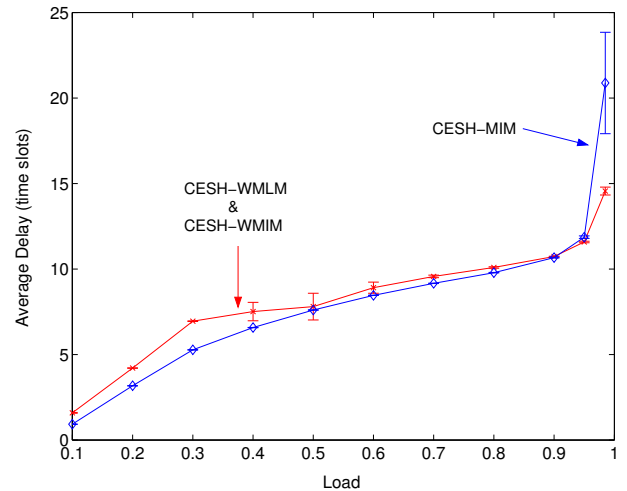


Fig. 7. Average delay under nonuniform traffic in a 20x6 system.

lowest average delay and TMWM-1 always has the highest average delay. The average delays of CESH-MLWM, CESH-WMIM and CESH-MIM are very close to that of TMWM-2 when the system load is low. When the load is very high, CESH-MIM has slightly higher average delay than CESH-MLWM and CESH-WMIM.

The average delays of the three proposed schemes under nonuniform traffic in a  $20 \times 6$  system is shown in Fig. 7. Different from the case in a smaller system shown in Fig. 6, there is a cross-point between the CESH-MIM and CESH-WMLM/CESH-WMIM curves. Compared to CESH-MLWM and CESH-WMIM, the average delay of CESH-MIM is lower when the system load is low, and is higher when the system load is over 95%.

## V. CONCLUSIONS

In this paper, we proposed a new class of channel-aware polling service-based scheduling schemes, termed Channel-Aware Scheduling with Exhaustive Service and Hamiltonian Walk (CESH), for multichannel WLANs. We described three variations of CESH, namely CESH-MLWM, CESH-WMIM and CESH-MIM, and studied their performance via simulation studies. Compare to MWM-based scheduling (the well-known optimal-achieving scheme), the proposed CESH schemes achieve comparable delay performance, but with significantly reduced computational complexity. This is due to their high service efficiency from the use of exhaustive polling service. The low complexity makes them highly suited for practical implementation in multi-channel WLANs.

## REFERENCES

- [1] P. Kyasanur, J. So, C. Chereddi, and N. H. Vaidya, "Multi-channel mesh networks: Challenges and protocols," *IEEE Wireless Communications*, vol. 13, no. 2, pp.30–36, Apr. 2006.
- [2] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, "A receiver-initiated collision-avoidance protocol for multi-channel networks," in *Proc. IEEE INFOCOM'01*, Anchorage, AK, Apr. 2001, pp.222–233.
- [3] S. Damodaran and K. M. Sivalingam, "Scheduling algorithms for multiple channel wireless local area networks," *Computer Communications*, vol. 25, no. 14, pp.1305–1314, Sept. 2002.
- [4] P. Kyasanur and N. H. Vaidya, "Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks," *SIGMOBILE Mobile Computing and Communications Review*, vol. 10, no. 1, pp.31–43, Jan. 2006.
- [5] K. Kar, X. Luo, and S. Sarkar, "Throughput-optimal scheduling in multichannel access point networks under infrequent channel measurements," in *Proc. IEEE INFOCOM'07*, Anchorage, AK, May 2007, pp.1640–1648.
- [6] W. Tam and Y. Tseng, "Joint Multi-Channel Link Layer and Multi-Path Routing Design for Wireless Mesh Networks," in *Proc. IEEE INFOCOM'07*, Anchorage, AK, May 2007, pp.2081–2089.
- [7] X. Lin and S. Rasool, "A Distributed Joint Channel-Assignment, Scheduling and Routing Algorithm for Multi-Channel Ad Hoc Wireless Networks," in *Proc. IEEE INFOCOM'07*, Anchorage, AK, May 2007, pp.1118–1126.
- [8] M. X. Gong, S. F. Midkiff, and S. Mao, "A cross-layer approach to channel assignment in wireless ad hoc networks," *ACM/Kluwer Mobile Networks and Applications Journal (MONET)*, vol.12, no.1, pp.43-56, Feb. 2007.
- [9] M. X. Gong, S. F. Midkiff, and S. Mao, "On-demand routing and channel assignment in multi-channel mobile ad hoc networks," *Elsevier Ad Hoc Networks Journal*, doi:10.1016/j.adhoc.2007.11.011, in press.
- [10] H. Takagi and L. Kleinrock, "A tutorial on the analysis of polling systems," Computer Science Department, UCLA Report No. 850005, Tech. Rep., Feb. 1985.
- [11] A. Nijenhuis and H. Wilf, *Combinatorial Algorithms: For Computers and Calculators*, 2nd Edition, Academic Press, New York, 1978.
- [12] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling for maximum throughput in multi-hop radio networks," *IEEE Trans. Automatic Control*, vol. 37, no. 2, pp.1936–1949.
- [13] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Communications*, vol. 47, no. 8, pp.1260–1267, Aug. 1999.
- [14] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Networking*, vol. 7, no. 2, pp.188–201, Apr. 1999.
- [15] Y. Li, *Design and Analysis of Scheduling for High Speed Input Queued Switches*, Ph.D. dissertation, Polytechnic Univ., Brooklyn, NY, Jan. 2004.
- [16] Y. Li, S. Panwar, and H. J. Chao, "Exhaustive service matching algorithms for input queued switches," in *Proc. IEEE HPSR'04*, Phoenix, AZ, Apr. 2004, pp.253–258.
- [17] A. Konrad, B. Y. Zhao, A. D. Joseph, and R. Ludwig, "A Markov-based channel model algorithm for wireless networks," *ACM Baltzer Wireless Networks*, vol. 9, no. 3, pp.189–199, May 2003.