Exploiting the Full Potential of Multi-AP Diversity in Centralized WLANs through Back-pressure Scheduling

Anfu Zhou, Min Liu, Tong Shu, Yilin Song, Zhongcheng Li
Insitute of Computing Technology
Chinese Academy of Sciences, Beijing, China, 100190
Email: {zhouanfu, liumin, shutong, songyilin, zcli}@ict.ac.cn

Abstract—**Centralized WLANs widely deployed in enterprise environment or university campus often have high density of Access Point (AP). The high density leads to multi-AP diversity, which brings possibility to improve network performance. Previous studies have proposed different schemes to exploit multi-AP diversity, however, these schemes are all based on heuristic and cannot guarantee an optimal exploitation of multi-AP diversity. In this paper, we propose a Theory Based Centralized Scheduling (TBCS) to exploit the full potential of multi-AP diversity. TBCS is based on the well-known back-pressure scheduling. Although back-pressure scheduling is proved to be throughput-optimal, most of previous studies are purely theoretical. To make a practical use of the theoretical back-pressure scheduling, we design new mechanisms in TBCS to handle the problem caused by the wired/wireless mixed scenario of centralized WLANs and to synchronize the scheduling. We evaluate TBCS through NS-2 simulations and show that compared with previous methods, TBCS can support the largest capacity region and greatly improves the throughput of a network.**

I. INTRODUCTION

Recently, centralized WLANs have been rapidly deployed in enterprise or university campus environment. These networks often have high density of Access Point (AP) for extended coverage and higher throughput support [1-3]. The high density results in multi-AP diversity, which brings possibility to improve network performance by exploiting the diversity.

There are plenty of studies [4-7] on exploiting multi-AP diversity. For example, Zhu *et. al.* in [4] and Miu *et. al.* in [5] propose to pick a correct copy or combine corrupted copies of data packets received by multiple APs. However, the proposals in [4][5] just take advantage of the backup function of multiple APs, that is, they just use multiple paths to transmit/receive multiple copies of the same packet. Other studies [6][7] make further use of the multi-AP diversity to distribute packets across multiple paths. The work in [6] distributes packets through a round-robin deterministic policy, and the work in [7] 'intelligently' distributes a packet to another arbitrary path if the current used path has fallen into a bad state. However, the packet distribution policies in these studies are all based on heuristics and hence they *cannot* guarantee an optimal exploitation of multi-AP diversity. In [8], Ahmed *et. al.* proposes to use a centralized scheduling to exploit multi-AP diversity in a dense deployment of APs. However, the work focuses on describing the opportunity and the challenges of the exploitation, and they do not propose the specific solution.

In parallel, there are many theoretical results [9-15] that address the issue of optimal utilization of a network based on the classical back-pressure scheduling, which is first proposed by Tassiulas and Ephremides [16]. We refer to [9], [10] for a comprehensive survey. It is well known that back-pressure scheduling can optimally exploit multiple-paths in a network to support the largest capacity region. However, most studies on back-pressure scheduling are purely theoretical and the back-pressure scheduling is rarely adopted by practitioners because of its unrealistic assumptions. For example, previous studies usually assume that the interference graph of a network, which is hard to obtain in practice, is known in advance. In addition, the scheduling in back-pressure is supposed to operate in a centralized way, which is of great difficulty especially in multi-hop wireless networks.

In this paper, we adapt the back-pressure scheduling for practical use, that is, to optimally exploit multi-AP diversity in centralized WLANs. We observe that centralized WLANs, compared with multi-hop wireless networks, have some advantages for applying back-pressure scheduling. First, it is convenient for a centralized WLAN to obtain the complete interference graph of the network (as done in [3,17,18]). Second, it is natural to perform centralized scheduling in centralized WLANs. However, despite these advantages, there are also many technical difficulties. For example, the first problem is that the existing IEEE 802.11 based WLANs adopt `'single association'` architecture, that is, each client node associates with only one AP. Clearly, we must remove this restriction before exploiting multi-AP diversity. A more important problem is caused by the wired/wireless mixed scenario of centralized WLANs. In the WLANs, packets of a network will queue up on APs. However, the APs contend for access of the wireless medium in a distributed and random

---

This work has been supported by the National Basic Research Program of China (No. 2011CB302702), the Beijing Nova Program and the National Natural Science Foundation of China (No. 60803140, No. 60970133, No. 61070187 and No. 61003225).

way, so the back-pressure scheduling cannot be guaranteed.

In this paper, we first break up the single association architecture of the existing IEEE 802.11 system and construct the complete interference graph including unassociated links. Then we propose a mechanism called `'Queuing in Front'` to handle the wired/wireless mixed scenario, and an AP-feedback mechanism to synchronize the scheduling. Putting it all together, we present our Theory Based Centralized Scheduling (TBCS) system to exploit the full potential of multi-AP diversity. The TBCS works without any modification of client nodes and therefore can be deployed conveniently.

We implement the TBCS system in NS-2 and evaluate the performance of TBCS with extensive simulations. Compared with previous methods in [6][7], TBCS supports the largest capacity region and improves the throughput of centralized WLANs greatly. Our simulation results show that the improvement of throughput can be up to 200%. As far as we know, our work is the first to apply the theoretical back-pressure in WLAN scenario. One of our aims is to demonstrate the advantage of a theory-based solution.

Note that there are studies [19][20] on applying back-pressure scheduling in a different scenario of multi-hop wireless networks. However, in multi-hop wireless networks, it is hard to obtain the interference graph and to do centralized scheduling. So these studies adopt heuristics-based alternative schemes to handle the interference and use approximate (but non-optimal) scheduling schemes. Further, the work in [19] does not consider multi-path diversity and the work in [19][20] cannot handle problems caused by interference such as hidden terminal problem.

The remainder of this paper is structured as follows. In Section II, we point out the potential of multi-AP diversity and give a brief review of back-pressure scheduling. In Section III, we present the design of TBCS system. Section IV evaluates the TBCS with NS-2 simulation and Section V concludes the paper.

II. THE POTENTIAL OF MULTI-AP DIVERSITY

Now we demonstrate the potential of multi-AP diversity with a simple example. Consider the topology of Fig. 1, which consists of four APs and three client nodes. We denote the i_{th} AP by AP_i, the j_{th} client node by n_j. If a client node is in the transmission range of an AP, we consider there is a link between them. we denote the k_{th} link by l_k. Note that there are four links in Fig. 1. We arrange the relative position of nodes so that n_2 has multi-AP diversity, that is, it is in transmission range of both AP_2 and AP_3. As in [17][18], we say that two links interfere with each other if the two links cannot transmit simultaneously. In this topology, we assume that l_1 interferes with l_2, and l_3 interferes with l_4, and obviously l_2 interferes with l_3.

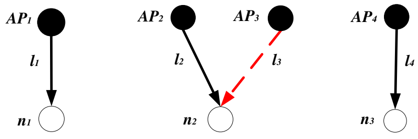

Fig. 1. A simple multi-AP topology

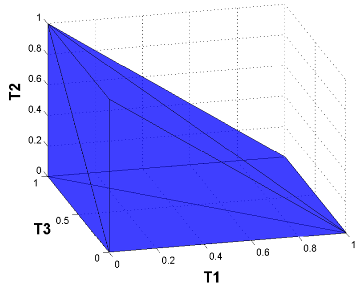

Fig. 2. Capacity region without multi-AP diversity

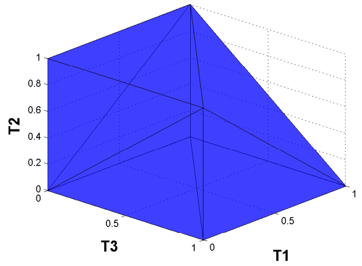

Fig. 3. Capacity region with multi-AP diversity

It is well known that traditional IEEE 802.11 based WLANs adopt 'single association' architecture, that is, each client node associates with only one AP (without loss of generality, we suppose n_2 is associated with only AP_2). Now we examine the capacity region under the single association architecture. Denote the throughput of the three nodes n_1, n_2, n_3 by T_1, T_2 and T_3 respectively. It is easy to know that all feasible $\{T_1$, T_2, $T_3\}$ must satisfy the constraint as follows [1]:

$$\begin{cases} 0 \leq T_1, T_2, T_3 \leq 1 \\ 0 \leq T_1 + T_2 \leq 1 \end{cases} \tag{1}$$

On other hand, if we break the single association architecture and let both AP_2 and AP_3 can transmit packets to n_2, then the constraint of all feasible throughput vectors $\{T_1$, T_2, $T_3\}$ becomes

$$\begin{cases} 0 \leq T_1, T_2, T_3 \leq 1 \\ T_2 \leq (1 - T_1) + (1 - T_3) = 2 - T_1 - T_3 \end{cases} \tag{2}$$

We plot the capacity region (i.e., all feasible $\{T_1$, T_2, $T_3\}$) described by (1) and (2) in Fig. 2 and Fig. 3 respectively. Clearly, the capacity region with exploitation of multi-AP diversity in Fig. 3 is much larger than that without exploitation of multi-AP diversity in Fig. 2 (Quantitatively speaking, the volume of Fig. 3 is $\frac{5}{3}$ times of that of Fig. 2). For example,

---

[1] Note that we suppose the capacity of each link is 1. For simplicity, we omit the transmission overhead such as control message, backoff time etc. We will take into account all these overhead in our simulations in section IV.

$\{T_1 = 0.5, T_2 = 0.8, T_3 = 0.5\}$ is not feasible without exploitation of multi-AP diversity, since $T_1 + T_2 > 1$, and l_1 and l_2 interfere with each other and cannot transmit simultaneously. On the other hand, $\{T_1 = 0.5, T_2 = 0.8, T_3 = 0.5\}$ can be supported with exploitation of multi-AP diversity since the packets to n_2 can be transmitted on both l_2 and l_3. Specifically, we can distribute the load of n_2 as follows: 0.4 on l_2 and 0.4 on l_3. Since l_1 and l_3 can transmit simultaneously, and l_2 and l_4 can transmit simultaneously, it is clear that the distribution can make the traffic $\{T_1 = 0.5, T_2 = 0.8, T_3 = 0.5\}$ feasible.

A. Back-pressure Scheduling

It is well-known that back-pressure scheduling, first proposed in [16], is a throughput-optimal scheduling policy. The back-pressure scheduling can support any traffic rate inside the capacity region of a network. Hence the back-pressure scheduling can exploit the full potential of multi-AP diversity.

To understand back-pressure scheduling, we first introduce the network model. Following the convention in [9][10], we denote a wireless network by a graph $\{V, E, I\}$, where V is set of wireless nodes, E is set of links, and I represents the interference constraints of the network. $I(i, j) = 1$ if link i and link j cannot transmit simultaneously.

At the beginning of each scheduling slot, the back-pressure scheduling chooses a maximal feasible schedule using the following policy

$$M^* = \max_{M \in M_E} \sum_{l \in E} q_l \times M(l) \tag{3}$$

Where q_l is the queue backlog of link l. M is a maximal feasible schedule, such that, all links in M can transmit simultaneously and no more links can be activated without violating the interference constraints. M is denoted with a vector in $\{0, 1\}^{|E|}$. If link l is included in a maximal schedule M then $M(l)$ is set to 1, and to 0 otherwise. Let M_E be the set of all possible maximal schedules.

The scheduling policy of (3) is also called Maximal Weighted Scheduling (MWS) in literature. As we can see, MWS is equivalent to finding a maximum weighted independent set of links and its computation complexity is NP-hard [11].

To avoid the computation complexity of MWS, a simpler scheduling policy called Greedy Maximal Scheduling (GMS) is proposed and extensively studied (see [11] and the reference therein). GMS operates as follows: at the beginning of each scheduling, it first picks link l with the largest backlog; it then discards all links that interfere with link l, it then picks the link with the largest backlog from the remaining links; and this process continues until no links left [11]. Although GMS is a sub-optimal policy in general, many analytical and simulation results show that GMS can achieve the full capacity region in some special network topologies like tree networks [11]. In this work, we will also examine the performance difference between GMS and MWS in the scenario of centralized WLANs.

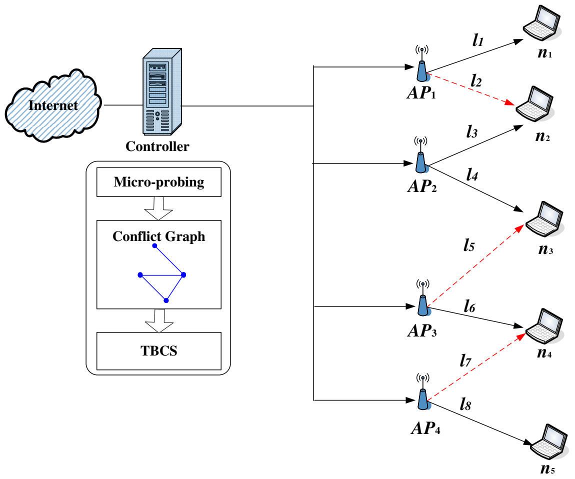

Fig. 4. High-level overview of the TBCS system

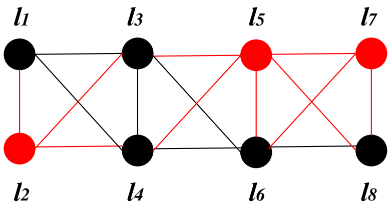

Fig. 5. Interference graph without and with multi-AP diversity

III. DESIGN OF TBCS

In this section, we present the TBCS system which adapts the theoretical back-pressure scheduling to optimally exploit multi-AP diversity in centralized WLANs. Centralized wireless networks are becoming prevalent in enterprise networks, and have received much research interest [3][17][18]. A high-level overview of the architecture of our TBCS system is shown in Fig. 4. The key component of the TBCS system is the controller (co-located at the edge router), which has two main functions. The first is to construct the conflict graph and the second is to perform the back-pressure based scheduling, that is, to optimally distribute packets to APs.

In order to apply the back-pressure scheduling in practice, we are confronted with some technical challenges. The first one is how to take into account the unassociated links, the second is how to handle the wired/wireless mixed scenario of centralized WLANs and the third is how to synchronize the scheduling. We explain the challenges and give our solutions as follows.

A. Construct Interference Graph Including Unassociated Links

As is well known, one prerequisite of back-pressure scheduling is the interference graph of a wireless network. In this work, we adapt Micro-probing proposed in [17] to construct the interference graph of a centralized network. The basic idea of Micro-probing is as follows. The controller sends interference probing requests to APs and APs carry out the interference measurement probings and respond with probing results. For example, to test if two links l_1 and l_2 interfere with each other, Micro-probing initiates a transmission on l_1 (i.e., a transmission between AP_1 and its client n_1), at time t_0. Then Micro-probing instructs AP_2 of link l_2 to send a broadcast frame at the exact same time instant. If AP_1 does not receive an ACK within SIFS, then one can infer a collision at n_1 and further infer that l_1 interferes with l_2. Micro-probing is an efficient interference measurement method. It can rapidly construct the conflict graph even while the network is in use (i.e. online). The overhead of Micro-probing is very low (it takes only a few seconds for a medium scale network of dozens of nodes).

However, the Micro-probing algorithm assumes the single-association architecture and just measures interference relationship among associated links. For exploiting multi-AP diversity, we extend Micro-probing to measure the interference among any links including unassociated links. That is, if an AP is in transmission range of a client node, we treat it as a link existing between them and measure interference between the link and other links. For example, in Fig. 4, suppose there are five associated links l_1, l_3, l_4, l_6, l_8. The interference graph achieved by the original Micro-probing is denoted by the black nodes and black links of Fig. 5. However, in this paper we break up the constraint of single association and then we can add three new links l_2, l_5, l_7 because client node n_2 is in transmission range of AP_1, client node n_3 is in transmission range of AP_3 and client node n_4 is in transmission range of AP_4. The new links and the interference relation are added to Fig. 5 (denoted by red nodes and red links). We then get a complete interference graph including the unassociated links, which provides the basis for exploiting multi-AP diversity.

In the NS-2 simulations in Section IV, we break the single-association by modifying the code of Route Module of NS-2. With the modification, we enable a wireless node to receive packets from multiple APs. In practical wireless networks, it is more difficult to break the single-association, since due to the IEEE 802.11 standard [21], wireless clients nodes need a "association-exchange" with an AP before it can transmit or receive packets from the AP. However, there are many approaches that can solve the problem [22][23]. For example, the authors in [22] propose MultiNet, in which a wireless client node can simultaneously connect to multiple APs by virtualization techniques. MultiNet realizes the multi-association by introducing an intermediate layer below IP, and works well over wireless cards operating according to IEEE 802.11.

B. Queuing in Front

A distinct characteristic of a centralized WLAN is that it is a wired/wireless mixed network (while the controller and APs are connected with wired links, APs and client nods are connected via wireless medium). This characteristic brings a technical problem when applying back-pressure scheduling: since the bandwidth of the wired part of the network is often much bigger than the bandwidth of the wireless part, the downlink packets of the network will queue up on APs. However, the APs contend for access of the wireless medium in a distributed and random way so the MWS policy cannot be guaranteed [19].

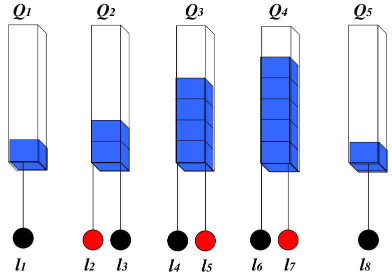

Fig. 6. The relation between queues and the links

To solve this problem, we propose a mechanism called Queuing in Front. The basic idea is that we make all packets queuing on the controller. The controller then distributes packets in batches. In each batch, the controller distributes packets to APs according an optimal schedule. We explain the details of Queuing in Front as follows.

Suppose there are N client nodes in a centralized WLAN. For each client node n_i, we set up a queue denoted by Q_i on the controller. The length of Q_i is denoted by $len(Q_i)$. When a packet comes, the controller puts it into the corresponding queue according to the destination address of the packet. Hence all packets queue up on controller, and there are no queue backlogs on wireless links between APs and client nodes. However, as expressed in (3), the back-pressure scheduling needs backlog of every link to compute the optimal schedule. To solve this problem, we set up a virtual queue backlog vq_l on the controller for each link l . If the destination of the link (denoted by dst_l) is client node i, the controller set vq_l to $len(Q_i)$. Then at the beginning of a scheduling slot, the back-pressure scheduling computes an optimal schedule M^* with the virtual backlog information. Note that no two links in M^* will interfere with each other. For each link l belonging to the optimal schedule M^*, the controller delivers q^* packets to the corresponding AP of the link (the AP is denoted by $\mathrm{AP}_{\mathrm{head}_l}$). q^* is the minimum queue length of the links in M^*, that is

$$q^* = \min_{l \in M^*} \mathrm{vq}_l \tag{4}$$

In such way, the controller guarantees an optimal exploitation of multi-AP diversity without modification of the existing random medium access mechanism of IEEE 802.11 on APs.

For example, for the network in Fig. 4, suppose the queue lengths on the controller are $\{1, 2, 4, 5, 1\}$ at the beginning of a scheduling. We plot the relationship between queues on the controller and the virtual queues on wireless links in Fig. 6. From the figure, we can see that the virtual backlog of the wireless links are $\{1, 2, 2, 4, 4, 5, 5, 1\}$. Applying (3) under the interference constraint shown in Fig. 5, the controller decides that the chosen links are l_4 and l_7. Then the controller will pick

TABLE I
Algorithm 1: TBCS algorithm

```
**Procedure** handleIncomingPacket($\mathrm{Pkt}_k$)
    if destination($\mathrm{Pkt}_k$)==i
        put ($\mathrm{Pkt}_k$) into $Q_i$;
    updateVirtualBacklog();
**Procedure** updateVirtualBacklog()
    for each link l
        if $\mathrm{dst}_l == i$
            $\mathrm{vq}_l = len(Q_i)$
**Procedure** Scheduling()
    Compute the optimal schedule $M^*$ with MWS (or GMS)
    $q^* = \min_{l \in M^*} \mathrm{vq}_l$
    for each link $l \in M^*$
        Dequeue $q^*$ packet from $Q_{\mathrm{dst}_l}$;
        Deliver the $q^*$ packets to $\mathrm{AP}_{\mathrm{head}_l}$;
**Procedure** handleFeedback()
    Denote the current optimal schedule with $M^*$;
    for each link $l \in M^*$
        Wait for the report from $\mathrm{AP}_{\mathrm{head}_l}$;
    Scheduling();
```

4 (here $q^* = 4$) packets from Q_3 and Q_4 and then delivers the packets to AP_2 and AP_4 respectively. Finally AP_2 and AP_4 transmit the packets to the client nodes n_3 and n_4. Note that the chosen links by (3) will not interfere with each other, so we do not need to worry about collisions induced by hidden terminal problem or others.

C. Scheduling Synchronization

In the theoretic studies [9][10] of back-pressure scheduling, the scheduling operates in an ideal way. In those studies, a time slotted system is assumed where each slot has the unit length and the transmission time of every packet is exactly one slot. Clearly, this does not hold true in real networks. In current IEEE 802.11 networks, the nodes contend for the wireless medium in a random way and they backoff for a random period before transmission for collision avoidance. So even when packets are of the same length, the transmission of the packets will not finish at the same time.

To synchronize the scheduling at the controller, we propose an AP-feedback mechanism. The idea is that each AP monitors its transmission status, when the transmissions of all q^* packets are finished, the AP reports to the controller. The controller will not do the next scheduling until it gets all the reports from APs. In this way, we make the scheduling synchronized.

The AP-feedback mechanism will not cause much overhead. On one hand, the synchronization report packet is very small. On the other hand, the bandwidth of the wired links between the controller and APs is often much bigger than the bandwidth of wireless links.

In our TBCS system, we do not change the CSMA/CA mechanism on APs and client nodes. In this way, we can deploy the TBCS system easily. In addition, the reservation of CSMA/CA is also useful when a WLAN network is confronted with exogenous interference outside the WLAN. To sum up, the logic of TBCS is presented in the pseudo code shown in Algorithm 1.

Note that TBC-Scheduling is limited to downlink traffic. It is reported in many studies that downlink traffic is the dominant traffic in wireless networks with the fraction of the downlink traffic constituting commonly about 80% of the aggregate traffic [3][24]. In addition, we also evaluate the impact of the existence of uplink traffic by introducing the bidirectional TCP traffic into the network, as shown in Section IV.

D. Issues on Overhead and Security

The schemes in TBCS are more complex than existing schemes. TBCS requires determining the interference graph of a network, as well as the AP-feedback, and the computation of an optimal scheduling. However, we argue that the added complexity is justifiable. As we explained above, Micro-probing can rapidly construct the conflict graph in a few seconds [3,17,18]. Although the computation complexity of MWS is NP-hard, we will find in Section IV that GMS is a good approximation to MWS. GMS has low computation complexity of $O(L^2)$ (L is the number of links) and has close performance to MWS. In addition, we take into account all the overhead in the simulations of Section IV and the simulation results validate the feasibility of new schemes.

TBCS will not affect the security functions in WLANs. The encryption/decryption and other security functions in existing WLAN security standards (e.g., WEP [21], 802.1x [25]) are handled in software [7] and can be easily assembled with TBCS.

IV. EVALUATION

In this section, we perform NS-2 [26] simulations to evaluate TBCS. We compare TBCS with three other scheduling methods. The first one is the existing IEEE 802.11 DCF which cannot exploit multi-AP diversity at all. The other two are Divert [7] and Round Robin [6], which exploit the multi-AP diversity only in a heuristics-based way, as we have introduced in Section I. We also compare the performance of two specific scheduling policies in TBCS, the throughput-optimal scheduling policy MWS and the sub-optimal but low-complexity policy GMS. We denote them with TBCS-MWS and TBCS-GMS respectively. The common parameters at the MAC and physical layers for the NS-2 throughout simulations in this paper are listed in Table II. The lasting time of each simulation is 500 seconds.

We first simulate in a simple topology aimed at clearly presenting and understanding the effect of TBCS. We examine

TABLE II
PARAMETERS OF NS-2 SIMULATIONS

DataRate	11 Mbps
BasicRate	2 Mbps
SIFS	10 μs
DIFS	50 μs
EIFS	364 μs
σ	20 μs
PCLP length	192 bits
MAC header (RTS, CTS, ACK, DATA)	(20,14,14,28) Bytes
Minimum Backoff Window	31
Maximum Backoff Window	1023
Long Maximum Backoff Stage	7
Short Maximum Backoff Stage	4
Transmission Range	250m
Carrier Sense Range	550m

the increase of the aggregated queue length of a network confronted with saturated traffic, and show that TBCS achieves the largest capacity region. Then we simulate in random topologies where nodes are randomly placed. We examine the throughput of TCP and UDP flows in simulations and show that TBCS achieves more throughput than other methods.

In all following simulations we only use static topologies due to space limits. In real wireless systems, client nodes may joint and leave a network dynamically. We think that TBCS can handle real dynamic systems, since in the proposed TBCS algorithm, the interference is probed online with Microprobing, which can rapidly re-construct the conflict graph after the change of network topology caused by node mobility [17]. Once we get the new interference graph, TBCS algorithm can run and exploit the multi-AP diversity as usual.

A. Capacity Region

In the first simulation, we use the topology in Fig. 7 (which is an instance of the multi-AP topology Fig.1). We probe the boundary of the capacity region of the network by scaling the amount of traffic. We set one Poisson UDP traffic to each client node and set the proportion of the three traffic flows to $\{3, 3, 1\}$. Note that all flows in our simulation are from nodes outside the WLAN to the wireless client nodes in the WLAN. The packet size of the UDP flow is set to 1000 Bytes. We increase the load factor in steps of 0.01^2.

We record the total queue length of the network in each simulation and report the simulation results in Fig.8. As we can see, under low traffic load, all five scheduling methods can keep the network stable. However, the network loses stability and the queue length grows fast when the load factor increases beyond certain thresholds. 802.11 DCF and Divert cannot guarantee network stability beyond load factor 0.14, and the threshold of Round Robin is 0.21. TBCS can support the largest traffic load threshold 0.29. Note TBCS-GMS achieves almost the same performance with TBCS-MWS. Although it is proved that GMS only has the efficiency ratio of 1/2 of MWS in the worst case, its actual performance is usually close to the optimal in many practical scenarios [11]. Our simulation result also validates the point in the centralized WLAN scenario. Hence we just use TBCS-GMS as the representative of TBCS in next simulations.

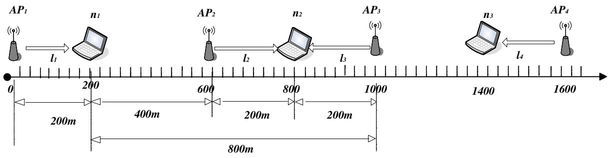

Fig. 7. An instance of the multi-AP topology of Fig. 1

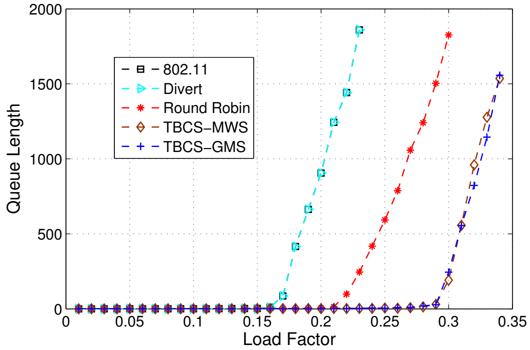

Fig. 8. Total queue length under simple topology

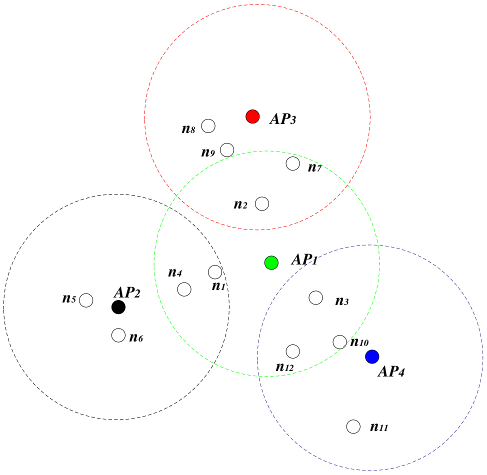

Fig. 9. A random topology with four APs and twelve flows

B. Throughput

In the next simulation, we examine the throughput gain achieved by TBCS in a random topology where the relative positions of the nodes are shown in Fig. 9. The topology consists of four APs (represented by colorful solid circles) and twelve wireless client nodes (represented by hollow circles). We denote the transmission range of an AP with a larger colorful circle in the figure. Note that seven nodes (n_1, n_2,

---

[2] Here the load factor value of 1 corresponds to the maximum throughput that a link can achieve when the link transmits alone without any contention and under saturated input traffic.

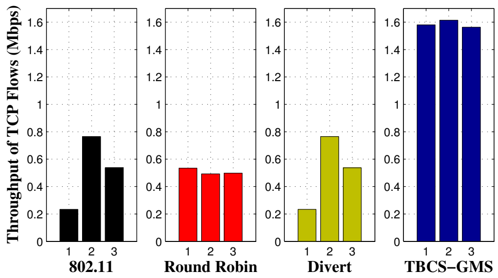

Fig. 10. Throughput of TCP flows

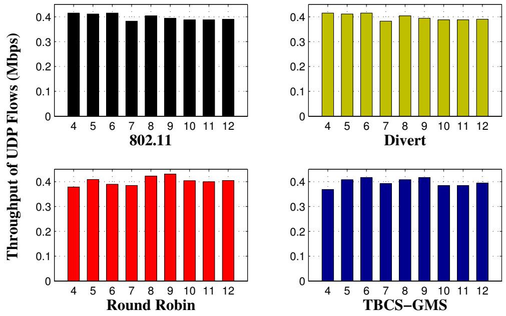

Fig. 11. Throughput of UDP flows

n_3, n_4, n_7, n_{10}, n_{12}) have multi-AP diversity. For clarity, we do not plot the controller and the wired links between the controller and the APs. We create a TCP and UPD mixed traffic scenario, and denote the flow to client node n_i by f_i. Three flows f_1, f_2 and f_3 are set to be TCP, and the other nine flows are set to be UDP. The traffic load of each UDP flow is set to be 0.45Mbps. Then we run each simulation with the four scheduling methods.

We record the throughput of TCP flows in Fig. 10 and the throughput of UDP flows in Fig. 11. We find that TBCS-GMS greatly improves the throughput of TCP flows. The total throughput of the three TCP flows is improved by about 200%, from 1.54Mbps to 4.75Mbps. We examine the trace of the simulation and find the reason for the improvement as follows. Under 802.11 DCF, all packets of the TCP flows are transmitted by AP_1 and the packets cannot be transmitted simultaneously. However, under TBCS-GMS, three packets (AP_2 to n_1, AP_3 to n_2 and AP_4 to n_3) can be transmitted simultaneously so as to improve the total TCP throughput. Note that Divert achieves the same throughput with 802.11 DCF because in NS-2 simulations there is no packet loss due to signal attenuation (all packet losses are due to collision) so Divert does not switch transmissions between APs. We also find that Round Robin improves the fairness among flows. Fig. 11 shows that the exploitation of multi-AP diversity will not hurt the performance of UDP flows. To sum up, we can see that compared with heuristics-based methods, TBCS can greatly improve the throughput of the network.

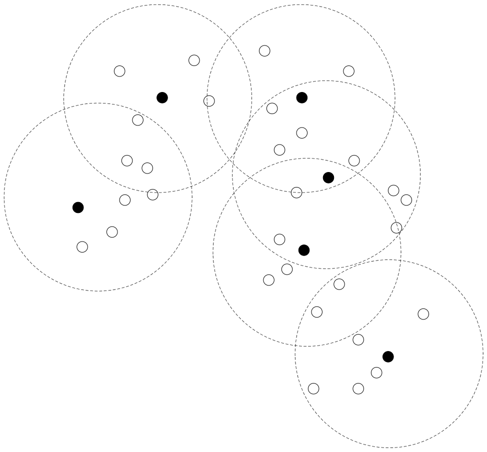

Fig. 12. A larger topology with 6 APs and 30 client nodes

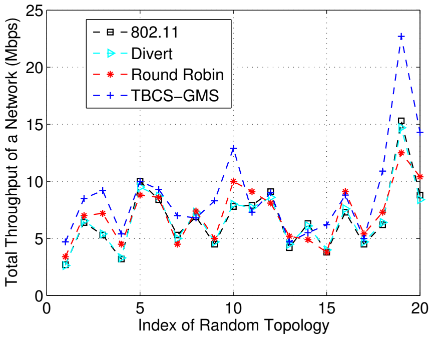

Fig. 13. Network throughput under random topologies

C. Performance under More Random Topologies

Here, we perform simulations under 20 random-generated topologies to evaluate the performance of TBCS. For each scenario, the number of APs is chosen randomly from 4 to 8, and the number of client nodes is chosen from 8 to 50. For each random topology, we first set a number of neighboring APs that have overlapping coverage, and then we set random positions for wireless client nodes around each AP. This cellular WLAN model is very representative and has been used in other research works ([4]-[7]). An example topology is shown in Fig. 12, there are six APs and 30 client nodes. Under each topology, we run simulations with the four scheduling schemes: 802.11, RoundRobin, Divert and TBCS-GMS.

With the simulation results, we compare the total throughputs achieved by the four scheduling schemes under each of the 20 topologies in Fig. 13. Clearly, we can see that the TBCS-GMS can greatly improve the the total throughput of a network. We compute the ratio of the improvement and find that the average improvement ratios of Divert and RoundRobin over all 20 topologies are 0.75%, 10.08% respectively, while that of TBCS-GMS is 35.62%.

As we can see from Fig. 13, the improvement under different topologies shows significant difference. we find the reason as follows: since the topologies are generated randomly,

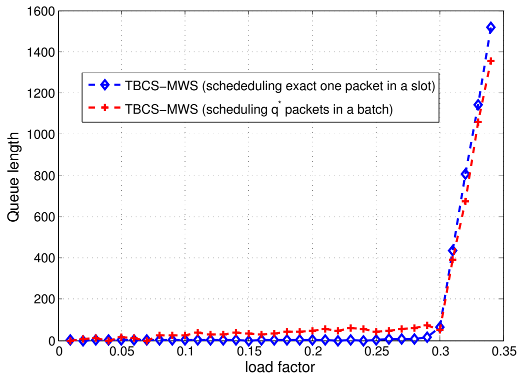

Fig. 14. Queue length of TBCS-MWS with and without batch-scheduling

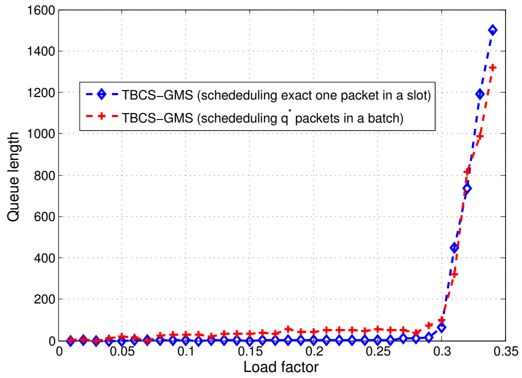

Fig. 15. Queue length of TBCS-GMS with and without batch-scheduling

they have different degree on diversity. For topology with rich diversity (the 19th topology), the improvement is high, while for topology with low diversity (the 6th topology), the improvement is low.

D. Discussion on the Impact of Batch-scheduling

Note that the TBCS algorithm schedules packets in batches in order to reduce the overhead brought by the delay between the controller and APs. However, this makes the TBCS algorithm not fully compatible with the original back-pressure algorithm, since the back-pressure algorithm makes a scheduling decision for each packet, not per batch. It is quite possible that, after one packet is transmitted according to the schedule, the decision in the next time slot will be different and involve other links.

In this subsection, we evaluate the impact of batch-scheduling on the capacity region. Here we set the bandwidth to be very large (1Gbps in next simulations) to isolate the impact of the delay caused by backhaul links. In the next subsection, we will add the factor of delay. We still use the representative multi-AP topology of Fig. 7 and perform simulations the same way as that in Section IV.A. We compare the performance of TBCS-MWS algorithm with and without batch-scheduling in Fig. 14. From the figure, we can first see that the capacity region with and without batch-scheduling is nearly the same, which indicates that batch-scheduling won't reduce the capacity region. Second, we have an interesting observation that when the traffic load is inside the capacity region, the average queue length of TBCS-MWS with batch-scheduling is bigger than that of TBCS-MWS without batch-scheduling. For example, when the load factor is 0.2, the average queue length with batch-scheduling is about 46, while that without batch-scheduling is only about 2. This phenomenon could be explained by the queueing theory [27]: the batch-scheduling leads to more fluctuation of service rate, hence the queue length will be larger.

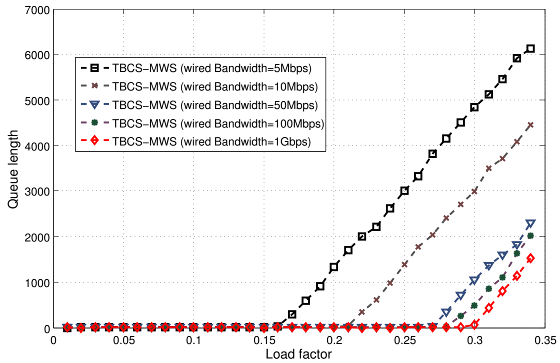

Fig. 16. Impact of finite backhaul capacity on TBCS-MWS without batch-scheduling

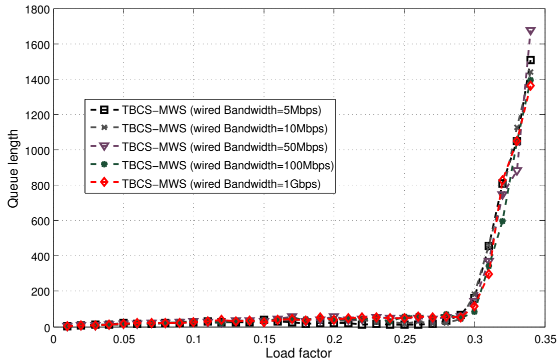

Fig. 17. Impact of finite backhaul capacity on TBCS-MWS with batch-scheduling

We also perform simulations with TBCS-GMS, and plot the results in Fig. 15. It is clear that the behavior of TBCS-GMS is very similar to that of TBCS-MWS.

E. Discussion on the Impact of Finite Backhaul Capacity

Recall that in the proposed TBCS algorithm, all packets are queued on controller and then distributed to APs. Clearly, the delay caused by the finite bandwidth of the wired backhaul links will impact the achieved capacity region of the algorithm. For TBCS without bach-scheduling, for each scheduling decision, the overhead is the packet transmission delay (denoted by T_d) over the backhaul link between the controller and APs. Consequently, an estimation of the utilization ratio of a network (denoted by u) is in the following:

$$u = \frac{T_s}{T_s + T_d} \tag{5}$$

where T_s is the transmission time of a packet over the wireless link between an AP and a wireless client node. Clearly, the smaller the bandwidth of backhaul link, the larger T_d, and the lower the network utilization ratio u. When the bandwith of backhaul link is 5Mbps, and other network parameters are as shown in Table II, we can compute from (5) that the network utilization ratio is about 50%.

We run simulations with TBCS-MWS (without batch-scheduling) under different values of bandwidth of backhaul links (5Mbps, 10Mbps, 50Mbps, 100Mbps and 1Gbps). The simulation results shown in Fig. 16 validate the above estimation. From the figure, the capacity region decreases as the bandwidth of the backhaul link decreases. When the bandwith of the backhaul link is set to 5Mbps, the capacity region threshold is reduced to 0.15, which is about a half the maximum achieved threshold.

Then we apply batch-scheduling in TBCS-MWS and repeat the above simulations, and plot the simulation results in Fig. 17. The simulation results are according to our anticipation: the capacity region rarely changes as the bandwith of backhaul link decreases, which demonstrates that batch-scheduling can effectively deal with the impact of the finite backhaul capacity.

V. CONCLUSION

Previous studies on exploiting multi-AP diversity are usually heuristics-based and hence cannot guarantee the optimal exploitation. In this paper, we develop a theory based centralized scheduling (TBCS) that can achieve the full potential of multi-AP diversity in centralized WLANs. In TBCS, we break the single association in IEEE 802.11 and adapt a theoretical optimal back-pressure scheduling. We also design the "Queue in Front" mechanism to handle the problem caused by the wired/wireless mixed scenario of centralized WLANs and the "AP-feedback" mechanism to synchronize the scheduling. Compared with previous heuristics-based methods, TBCS supports the largest capacity region and improves the total throughput of the network. Our simulation shows that the improvement of throughput can be up to 200%. Importantly, TBCS works without modification of client nodes hence can be deployed conveniently.

Note that in this paper, we evaluate TBCS using NS-2 simulations. In practical wireless networks, the network environment will be different from simulations. For example, there will be transmission errors due to the signal fluctuation of the wireless channel. Note that the controller in TBCS is supposed to wait the reports of all the APs before scheduling the next batch of packets. When an AP is experiencing the worst channel condition, the controller will wait too much time for the next scheduling. Currently, we are implementing TBCS in practical systems and solving these new challenges.

REFERENCES

[1] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill, "Designing High Performance Enterprise Wi-Fi Networks", in *Proc. USENIX NSDI,* San Francisco, California, USA, Apr. 2008.

[2] R. Murty, J. Padhye, A. Wolman, and M. Welsh. "An Architecture for Extensible Wireless LANs", in *Proc. ACM HotNets,*Calgary, Alberta, Canada, Oct. 2008

[3] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. Mishra, "CENTAUR: Realizing the Full Potential of Centralized WLANs through a Hybrid Data Path", in *Proc. ACM MobiCom,* Beijing, China, Sep. 2009

[4] Y. Zhu, Q. Zhang, Z. Niu, and J. Zhu, "Leveraging Multi-AP Diversity for Transmission Resilience in Wireless Networks: Architecture and Performance Analysis", *IEEE Trans. on Wireless Communications,* 8(10):5030-5040, Oct. 2009

[5] A. Miu, H. Balakrishnan, and C. Koksal, "Improving Loss Resilience with Multi-radio Diversity in Wireless Networks", in *Proc. ACM MobiCom,* Cologne, Germany, Sep. 2005

[6] E. Vergetis, E. Pierce, M. Blanco, and R. Guerin, "Packet-Level Diversity - From Theory to Practice: An 802.11-based Experimental Investigation", in *Proc. ACM MobiCom,* Los Angeles, California, USA, Sep. 2006

[7] A. Miu, G. Tan, H. Balakrishnan, and J. Apostolopoulos, "Divert: Fine Grained Path Selection for Wireless LANs", in *Proc.ACM MobiSYS,* Boston, Massachusetts, USA, Jun. 2004

[8] N. Ahmed, S. Banerjee, S. Keshav, A. Mishra, K. Papagiannaki, and V. Shrivastava, "Interference Mitigation in Wireless LANs using Speculative Scheduling", Extended Abstract, In *Proc. ACM MobiCom,*Montreal, QC. Canada, Sep. 2007.

[9] X. Lin, N. Shroff, and R. Srikant, "A Tutorial on Cross-layer Optimization in Wireless Networks", *IEEE J. Sel. Areas Commun.,* 24(8):1452-1463, 2006.

[10] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource Allocation and Cross-Layer Control in Wireless Networks", *Foundations and Trends in Networking,*1(1): 1-144, 2006

[11] C. Joo, X, Lin, and N. B. Shroff, "Understanding the Capacity Region of the Greedy maximal scheduling Algorithm in Multihop Wireless Networks", in *Proc. IEEE INFOCOM,* Phoenix, AZ, USA, Apr. 2008

[12] A. Gupta, X. Lin, and R. Srikant, "Low-complexity Distributed Scheduling Algorithms for Wireless Networks", in *Proc. IEEE INFOCOM,* Anchorage, Alaska, USA, May. 2007

[13] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar, "Throughput and Fairness Guarantees through Maximal Scheduling in Wireless Networks", *IEEE Trans. Inf. Theory,* 54(2):572-594, Feb. 2008.

[14] G. Sharma, N. B. Shroff, and R. R. Mazumdar, "On the Complexity of Scheduling in Wireless Networks", in *Proc. ACM MobiCom,* Los Angeles, CA, USA, Sep. 2006

[15] X. Lin and N. B. Shroff, "Joint Rate Control and Scheduling in Multihop Wireless Networks", in *Proc. IEEE CDC,* Atlantis, Paradise Island, Bahamas, Dec. 2004

[16] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling for Maximum Throughput in Multihop Radio Networks", *IEEE Transactions on Automatic Control,* 37(12):1936-1949, 1992.

[17] N. Ahmed and S. Keshav, "SMARTA: A Self Managing Architecture for Thin Access Points", in *Proc. ACM CoNext,* Lisbon, Portugal, Dec. 2006

[18] N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki, "Online Estimation of RF Interference", in *Proc. ACM CoNext,*Madrid, Spain, Dec. 2008

[19] A. Warrier, S. Janakiraman, S. Ha, and I. Rhee, "DiffQ: Practical Differential Backlog Congestion Control for Wireless Networks", in *Proc. IEEE INFOCOM,* Rio de Janeiro, Brazil, Apr. 2009

[20] B. Radunovic, C. Gkantsidis, D. Gunawardena, and P. Key, "Horizon: Balancing TCP over Multiple Paths in Wireless Mesh Network", in *Proc. ACM MobiCom,* San Francisco, California, USA, Sep. 2008

[21] "IEEE 802.11b/d3.0 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification", IEEE Standard, Aug. 1999.

[22] R. Chandra, Paramvir Bahl, and Pradeep Bahl, "MultiNet: Connecting to Multiple IEEE 802.11 Networks using a Single Wireless Card", in *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 2004

[23] A. J. Nicholson, S. Wolchok, and B. D. Noble, "Juggler: Virtual Networks for Fun and Profit", *IEEE trans. Mobile Computing*, 9(1): 31-43, 2009

[24] "Advanced RF Management for Wireless Grids", White-paper from Aruba Networks, http://www.arubanetworks.com/pdf/rf-for-grids.pdf

[25] "IEEE Standards for Local and Metropolitan Area Networks: Standard for Port Based Network Access Control", IEEE Standard, Oct. 2001.

[26] NS-2 Network Simulator, http://www.isi.edu/nsnam/ns.

[27] Leonard Kleinrock, "Queueing Systems volume 1: Theory", *John Wiley & Sons*, 1975