Downlink Packets Scheduling in Enterprise WLAN

Dong Zhao*, Ming Zhu*, Ming Xu*, Jiannong Cao†
*School of Computer, National University of Defense Techonology, Changsha, China
Email:{dongzhao, zhuming, xuming}@nudt.edu.cn
†Department of Computing, The Hong Kong Polytechnic University, Hongkong
Email: csjcao@comp.polyu.edu.hk

***Abstract*—Enterprise WLAN consists of many APs connected to wired backbone network. Conventional DCF mechanism can not completely prevent conflicts between APs which cause large quantities of transmission failure and retransmission overhead. However, to keep compatible with 802.11-compliant clients, it is difficult to discard DCF. In this paper, we proposed a framework called *DPS*, which incorporates centralized scheduling with DCF. In *DPS*, downlink packets are not forwarded as soon as they arrived at APs. Instead, a central controller schedules their forwarding. APs still use DCF to send downlink packets according to the controller's instructions. *DPS* preserves DCF and requires no modifications to 802.11-compliant clients. Under *DPS* framework, we proposed a scheduling algorithm, ensuring downlink packets be sent with high success rate. In this way, we can significantly reduce transmission failure and retransmission overhead. Experiment results indicated that *DPS* can significantly improve the performance of enterprise WLAN compared with DCF without any scheduling.**

I. INTRODUCTION

In recent years, the number of wireless APs we can find in public places grew rapidly. Enterprise WLAN connects these APs together, and manages APs in a centralized fashion, providing Internet access service for 802.11 devices in enterprise-class environment such as industrial district, office building, and university campus. In enterprise WLAN, the coverage area of APs must overlap to ensure continuous coverage. Furthermore, we tend to deploy a few overlapping APs to share traffic load in high density client areas such as auditoriums and lecture halls. There exists interference among APs when they are operating on the same channel and an efficient channel access mechanism is quite critical to coordinate APs' transmission.

Nowadays, IEEE 802.11 protocol family has become industry standard of WLAN's physical layer and link layer. Distributed Coordination Function (DCF), based on CSMA/CA , is the primary channel access mechanism in 802.11 standard, and has been universally adopted by most of wireless devices. However, DCF has been proved to be unsuitable in dense wireless networks. Due to unnecessary waiting time incurred by random backoff and overhead of coping with some unavoidable collisions [1] and retransmission, only a small proportion of time is spent on effective transmission and the total air time utilization is quite low, leading to low-level performance of network. As APs became denser, the degradation of network performance became increasingly severe.

The distributed characteristic of DCF limits its ability to efficiently solve contention and significantly reduce unnecessary conflicts. So, centralization of channel access mechanism in enterprise WLAN is proposed. In practice, there exists a centralized channel access mechanism in 802.11 standard, that is Point Coordination Function (PCF). In PCF, AP polls every associated station on a polling list in turn for data transmission. This working style can provide contention-free access for the network internally, but performs poor when co-existing with other nearby networks. In addition, PCF is designed to be used in single-AP networks, and adopting PCF in multi-AP network is problematic. MiFi [2] augment PCF to multi-AP scenario, but MiFi's optimization object is mainly to solve fairness problem, rather than coordinating the transmission of APs to improve throughput.

On other hand, DCF has its own advantages. DCF is simple and robust, and it can provide adequate performance in many scenarios, especially when contending rarely happens. Moreover, 802.11-compliant client devices such as laptops, cellphones and tablets, are very widespread. It is impossible for us to discard DCF completely for the sake of compatibility with these off-the-shelf devices.

A variety of studies have found that downlink traffic accounts for nearly 80% of the whole traffic in enterprise WLAN [3], [4]. Motivated by this observation, Shrivastava et al. proposed a framework called Centaur [5], which uses centralized scheduling coupled with DCF to mitigate the performance loss of downlink packets in enterprise WLAN. In Centaur, all the downlink packets from Internet enter through an edge router, and a central controller resides in the router controls the time of forwarding downlink packets consciously, rather than forwarding downlink packets as soon as the router receives packets. Through judiciously choosing packets that are simultaneously sent during every period of time, Centaur can solve some problems such as *"hidden terminal"* and *"exposed terminal"* that DCF suffers from.

Similar with Centaur, we also attempt to preserve DCF and proposed a centralized framework to help DCF mitigate some problems that DCF can not solve independently. One of our design goal is that, the scheme should require no modifications to 802.11-compliant clients, and then enterprise WLAN can coexist with off-the-shelf WiFi networks. Our primary optimization object is to maximize the throughput of downlink packets for the whole enterprise WLAN..

In this paper, we proposed a framework called *DPS* (***D**ownlink **P**ackets **S**cheduling*). Compared with Centaur [5], the advantages of *DPS* is two-fold. Firstly, *DPS* goes beyond epoch-based scheduling method used by Centaur. In *DPS*, a waiting packet may be forwarded as soon as some activated links retreat from the set of activated links and give the corresponding link of the waiting packet a high *packet reception* ratio, rather than waiting to a batch of sending packets all finish

978-1-4673-5939-9/13/$31.00 ©2013 IEEE

their sending. Secondly, unlike other study including Centaur, our algorithm is based on a more accurate interference model — graded SINR-based model. In this model, the impact of multiple interferers are considered together, and the reception of packet at receiver is stochastic. More accurate interference model complicated the scheduling problem, posing a new challenge.

The contributions of this paper are as follows:

- Proposes a framework called *DPS* that combines centralized scheduling with DCF to mitigate performance loss caused by uncoordinated channel access method in enterprise WLAN.
- Proposes an efficient downlink packets scheduling algorithm under the *DPS* framework.
- Demonstrates that *DPS* can significantly increase performance of enterprise WLAN through extensive simulation experiments.

II. Background and Motivation

In this section, we first analyse the characteristics of DCF when used in enterprise WLAN. Then, we introduce the basic idea of our solution: downlink packets scheduling.

A. Discussion about DCF

DCF is the main channel access mechanism in 802.11 protocols. Despite the widespread use of DCF, WLAN is plagued by some performance problems. Owing to high density of APs and terminals, these problems are more severe in enterprise WLAN .

1) The Limitations of DCF: DCF are based on CSMA/CA protocol. According to the protocol, a sender must sense the channel before transmitting a frame. If channel is busy, the sender should defer transmission, wait for a period of time and try again. This protocol suffers from the well-known "*hidden terminals*" problem. In *hidden terminals* scenarios, contending stations that can not hear each other may send packets simultaneously, resulting in collisions. The occurrence of collision indicates high packet loss rate.

Solving packet loss and retransmission causes considerable expense, the overhead is quite great. DCF requires every transmission should be responded with an ACK. Sender won't be aware of packet loss until after a period of timeout. Further, retransmission needs to take time to backoff and compete to access channel again.

DCF includes an optional "virtual carrier sense" mechanism. When "virtual carrier sense" is enabled, sender and receiver require to exchange RTS and CTS handshake frames to reserve channel for subsequent data frame transmission. This mechanism can solve "*hidden terminals*" problem and reduce collisions significantly, but RTS and CTS frames itself bring about extra overhead. When hidden terminals are not very common, the overhead will tend to be a waste of time, even inducing congestion [6], and the loss may outweigh the gain. It is difficult to determine when to enable RTS/CTS. The most common strategy is to disable "RTS/CTS" most of time, and enable it only when sending a very long frame.

Many work attempted to optimize DCF's efficiency through several aspects, including adaptive carrier sense threshold [7], alternative backoff scheme [8] and reduced slot size [9]. In addition, some work such as ZigZag decoding [8] and CSMA/CN [10] are proposed to reduce the overhead of resolving collision. However, DCF can not avoid collision completely due to its distributed characteristic. In dense enterprise WLAN with heavy load, the competition between APs is very fierce, transmission failure and retransmission increase sharply, so the limitations of DCF become more severe and the optimization method described above become less efficiency.

2) Why DCF should not be Replaced Completely: In spite of the limitations described above, DCF is still a fairly good channel access mechanism. First, DCF is quite robust. Benefiting from its distributed characteristic, DCF shows strong adaptability in a large range of scenarios, especially coexisting with other wireless networks. The combination of physical carrier sense and virtual carrier sense mechanism is sufficient in detecting the interference signal of external networks including non-802.11 network. Second, nowadays, 802.11-compliant devices ,including APs and mobile terminals, are quite ubiquitous, and we have to consider backward compatibility with these off-the-shelf clients. A brand-new channel access mechanism which completely do away with DCF certainly will require modifications to clients, and that will be costly and unpractical.

Therefore, new-developed enterprise WLAN in the near future should consider the compatibility with 802.11-compliant clients that are using DCF. We should not discard DCF completely, and our design goal is to require no modifications to these clients by preserving DCF.

B. Solution: Downlink Packets Scheduling

In enterprise WLAN, all the APs are connected via wired backbone network. Downlink packets, mostly from Internet, are supposed to be sent to their respective clients. Taking advantage of this structural feature, we seek to maintain a moderate competition between activated APs at high-level in a centralized fashion by coordinating and scheduling the forwarding of downlink packets. We ensure every downlink packet be sent with high reception success rate, thereby reducing retransmission and resulting overhead. Since downlink traffic accounts for nearly 80% of the whole traffic in enterprise WLAN [3], the increase of efficiency of downlink traffic will significantly improve the performance of the whole network.

III. DPS Framework

DPS is a downlink packets scheduling framework operating in enterprise WLAN. As shown in Fig. 1, all the APs are connected to a central controller via a fast wired backbone network. We designed a protocol that operates in the wired network connecting APs and the controller. When a downlink packet arrives at an AP(enters through AP's *wired* interface), the AP won't immediately forward the packet to its wireless interface. Instead, AP defers the packet's forwarding, and informs the controller of the arrival of the packet. The controller collects the reports from all the APs, constructing a global view of the whole network, and decides which packet to be sent and when. APs won't send any downlink packet until it receives the instruction from the controller. DCF is still preserved to be used in the communications between APs and clients: upon receiving instruction from controller, AP forwards the packet to wireless interface, and the wireless interface sends the appointed downlink packet to corresponding client using DCF protocol; similarly, client uses DCF to send uplink packets to its associated AP's wireless interface. The controller is also informed when an AP's wireless interface successfully sends a downlink packet (receives the MAC-layer ACK). Therefore, the central controller monitors the status of all downlink packets, knowing which packets are waiting in APs' buffer to be sent, which packets are being sent in the air by APs' wireless interface, and which packets have been successfully received by client.

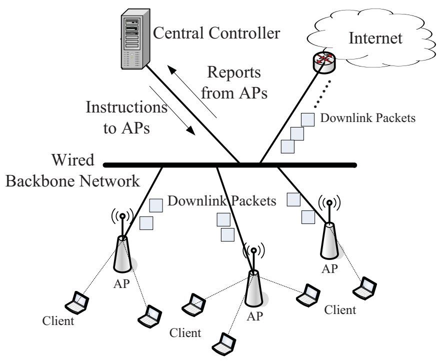

Fig. 1. Architecture of the *DPS* framework. All the APs are connected to a central controller via a fast wired backbone network, and a central controller is responsible for scheduling downlink packets.

As you can see, our design requires no modifications to 802.11-compliant clients. Even it needs not to modify 802.11 MAC protocol which operating in APs' wireless interfaces. Therefore, the advantages of DCF is preserved, and the compatibility with off-the-shelf devices is maintained. The scheduling of central controller coordinates the transmission between APs. The Responsibility of scheduling algorithm operating in controller is to reduce the occurrence of conflicts and retransmission and resulting overhead.

IV. Downlink Packets Scheduling in *DPS* Framework

DPS framework provides a mechanism that control the forwarding of downlink packets. Under this framework, a downlink packets scheduling algorithm is critical to improve performance. In this section, we introduce our scheduling algorithm that determines the timing of every packet's forwarding . The basic idea of scheduling algorithm is to maintain moderate competition between APs to ensure high transmission success rate, thereby reducing retransmission and then resulting overhead. We first provide the details of interference estimation, which is essential in evaluating competition for scheduling algorithm. Then, a downlink packets scheduling algorithm is proposed.

A. Interference Estimation

All the scheduling-related problems have to tackle a crucial problem that how to determine whether a certain set of links can be activated at the same time. In essence, we require an interference model to predict whether a given link will be interfered with by other activated links and estimate interference degree. Interference model has been proved to have significant impact on the complexity of many optimization problems in wireless networks, such as scheduling and channel assignment.

1) Interference model: The most primitive models includes k-hop model [11] and protocol model [12]. These models have a common traits: they consider every interferer independently, ignoring the fact that interference is a accumulated effect. These models are called pairwise model. Physical interference [12], or called SINR-based model captures the accumulated effects of multiple interferers. In SINR-based model, whether a packet is correctly received depends on the Signal-Interference-plus-Noise Ratio (SINR) at the receiver. Some SINR-based model argues that packet is correct received only when SINR at receiver is greater than a specific threshold value.

All the models above are under the assumption that interference is a binary phenomenon. Graded model further captures the phenomenon that the reception of packet under interference is stochastic, and argues that the success rate of packet reception rises with the decrease of SINR. In this model, interference degree is assessed by *Packet Reception Rate* (PRR), which is a function of SINR. Several works studied how to construct the PRR-SINR curve [13], [14] in off-line fashion. Considering the practicality, The studies of Padhye et al. [15] and Ahmed et al. [16] proposed sending some test packets to measure PRR in single-hop AP-based networks. In real-deployed testbeds, several works proposed some measurement-based model to predict PRR through measuring received signal strength (RSS) of radios [17], [18].

2) PRR computation: For the sake of accuracy and practicability, we choose graded SINR-based model to construct interference relation of APs. To calculate SINR, we need to get the power of useful signal, interfering signals, and background noise. Furthermore, in graded SINR-based model, PRR is a function of SINR value, so we need to construct the PRR-SINR curve. We benefit from many existing work [18], [19], which develops measurement-based techniques to estimate interference. Moreover, We incorporate off-line RSS measurement with online probing technique to construct interference relation of the whole network.

Given a target link and a set of simultaneously activated links, we need to estimate the PRR of the target link can achieve under the interference of these interfering links. Let $l_{i,j}$ denote a link, i is the source node and j is the destination node, $P_r(s)$ is the received power of signal from node s, at node r. Using the approach proposed by Reis *et al.* [18] and Padhye *et al.* [15], we can get $P_r(s)$ value of any node s and r. So, we obtain the SINR of $l_{i,j}$:

$$SINR_{i,j} = \frac{P_j(i)}{N_j + \sum_{l_{k,m} \in \mathbf{I}} P_j(k)} \quad (1)$$

where N_j is the background noise at node j, and $\mathbf{I}$ is the set of all simultaneously-activated links.

With the SINR, we next attempt to estimate PRR of link using PRR-SINR curve, and the PRR-SINR curve can be obtained by theoretically analyzing [20] or experimental fitting [13], [14]. To simplify the work, we choose the method of [20].

B. Downlink Packets Scheduling Algorithm

The central controller maintains two sets of packets: the set of sending packets, and the set of waiting packets. In addition, it also records the set of activated links. During network's operating, controller updates these three sets continuously according to the reports from APs and instructions it own has issued.

We classify waiting packets into several groups according to their corresponding links. One link is associated with an ordered queue of downlink packets, and a link is identified by its source (AP) and destination (client). The ordered queues are maintained to keep the order of every packet's arrival and ensure that packet arrives first will be sent first. When receiving a report that informs the controller of a packet's arrival, the controller inserts the packet at the back of corresponding queue.

Let $\mathbf{WQ}_{i,j}$ denote the waiting queue of link $l_{i,j}$, $\mathbf{AL}$ denote the set of activated links, $source_p$ and $destination_p$ denote the source and destination of downlink packet p respectively. Our scheduling logic is presented in pseudo code as shown in Algorithm 1. We do not claim that our algorithm is optimal, and our aim is to provide a significant improvement over existing scheme, i.e., DCF without any scheduling, by taking advantage of some intuitive heuristic view.

When a downlink packet arrives at AP, the controller invokes the procedure *HandNewArrivedPacket*, and inserts the new arrived packet into corresponding wait queue. When a packet is successfully transmitted, the controller invokes the procedure *HandSendSuccessPacket* to remove the corresponding link from $\mathbf{AL}$. Once receiving a transmitting failure report of a packet, the controller invokes the procedure *HandSendFailPacket* to insert the failing packet in the front of corresponding waiting queue, so that the failing packet will be sent first in the near future and out of ordering in MAC layer will be prevented. This has particular beneficial effect on the performance of TCP.

Every time procedure *Schedule* is invoked, it iterates through each nonempty waiting queue, and checks whether the packet at the front of waiting queue can be permitted to be sent. If the packet is determined to be sent, the algorithm inserts the packet into a packet set PS. In the end, PS records all the chosen packets scheduled to be sent next. Then, the controller issues instructions to relevant APs.

The basic principle to decide whether a packet can be permitted to be sent is that: the packet can achieve high PPR under the prerequisite of only affecting PRR of links in $\mathbf{AL}$ slightly. Specifically speaking, we have two quantitative criteria: first, if the packet was sent in the air, the sum of PRR of links in $\mathbf{AL}$ and PRR of the candidate packet's corresponding link should not be less than the sum of original PRR of links in $\mathbf{AL}$ when it was not sent; second, if the packet was sent in the air, the PRR of all the activated links including the candidate link and links in $\mathbf{AL}$ should not be less than a threshold value (we set it 90%). The purpose of the second criterion is to ensure already activated links achieve

Algorithm 1

Procedure: HandleNewArrivedPacket(p):
Insert p to the end of $\mathbf{WQ}_{source_p,destination_p}$;
Schedule(); // *Invoke procedure Schedule*
return
Procedure: HandleSendSuccessPacket(p)
$\mathbf{AL} \leftarrow \mathbf{AL} - \{l_{source_p,destination_p}\}$;
Schedule(); // *Invoke procedure Schedule*
return
Procedure: HandleSendFailPacket(p)
Insert p to the front of $\mathbf{WQ}_{source_p,destination_p}$;
Schedule(); // *Invoke procedure Schedule*
return
Procedure: Schedule()
$PS \leftarrow \emptyset$;
$sum \leftarrow 0$;
for each non-activated link whose waiting packets queue $\mathbf{WQ}_{i,j}$ is not empty **do**
  $sum\_prr \leftarrow 0$;
  for each link $l \in \mathbf{AL} \cup \{l_{i,j}\}$ **do**
    $prr \leftarrow$ CalcPRR(l, $\mathbf{AL} \cup \{l_{i,j}\}$);
    if $prr < threshold$ **then**
      goto 12;
      // *Once a link's PRR lower than threshold, skip directly and turn to the next candidate* $\mathbf{WQ}_{i,j}$
    end if
    $sum\_prr \leftarrow sum\_prr + prr$;
  end for
  if $sum\_prr \geq sum$ **then**
    $ca\_p \leftarrow$ Dequeue the packet at the front of $\mathbf{WQ}_{i,j}$;
    $PS \leftarrow PS \cup \{ca\_p\}$;
    $\mathbf{AL} \leftarrow \mathbf{AL} \cup \{l_{i,j}\}$;
    $sum \leftarrow sum\_prr$;
  end if
end for
return PS

high PRR, so that transmitting failures and retransmissions could be reduced.

For function *CalcPRR*, it has two parameters: the target link and the set of activated links, and returns the PRR of the target link under the interference of the activated links. We can use the approach mentioned in section IV-A2 to derive the pseudo code of *CalcPRR*.

The waiting queues of all the links are linked as a circular linked list. In procedure *Schedule*, the iteration on waiting queues goes along this circular linked list. As you can infer, the link being visited first will have more possibility to be chosen. For the consideration of fairness, every time the procedure *Schedule* is invoked, the starting point of iteration is forwarded by one position. In this way, we prevent certain links from starvation, because all the candidate links have comparatively fair opportunities to be chosen by scheduling algorithm.

V. Evaluation

We evaluated the performance of *DPS* and downlink packets scheduling algorithm through extensive packet level simulations. Our topology consists of 10 APs and 50 nearby clients, each client associated with one AP. The layout emulates real indoor building environment. The relative position of links covers every possible scenarios including *"hidden terminal"* and *" exposed terminal"*.

We conducted two sets of experiments to examine the impact of *DPS* on UDP flows and TCP flows separately. Every AP generated downlink traffic to its associated clients, and every client also generated uplink traffic to its associated AP. For UDP, the traffic was generated by a constant bit rate (CBR) traffic generator; for TCP, the traffic was generated in a random fashion to emulate the behavior of real Internet-access traces. For each traffic, we fixed packet size at 1024 bytes. The wireless interfaces of APs and clients operated under 802.11g, the PHY rate was fixed at 54Mbps, and RTS/CTS handshake was disabled.

We varied the generated traffic rate to examine the rise and fall of throughput with the increase of offered load. We kept the ratio of downlink and uplink traffic at 4:1, consistent with the trace records collected by prior studies [3], [4].

We first compared the average throughput of all the 50 links in *DPS* with downlink packets scheduling against that in DCF without any scheduling. The results are given in Fig. 2. We found that *DPS* improves the throughput of the whole network significantly. When the offered load is low, both solutions can meet the traffic load and send packets in time, so the improvement of *DPS* is not quite obvious. With the increase of offered load, the capacity of some links can not support the traffic load and these links enter saturated state. At that time we can find in the plot that the increase of average throughput can not match the increase of offered load, and then the improvement becomes obvious. The throughput of DCF without any scheduling displays a decline under heavy load due to the increased transmission failure and retransmission caused by fierce competition. This reflects the poor performance of DCF in heavy load network. On the contrast, *DPS* does not show such kind of decrease owing to its intelligent downlink packets scheduling. On the whole, *DPS* improves average UDP throughput by more than 40% over that of DCF without any scheduling. *DPS* has even greater effect on TCP's performance due to *DPS*'s ability to reduce collisions. For TCP, the overall gain of *DPS* over DCF without any scheduling is nearly 70% in saturated state.

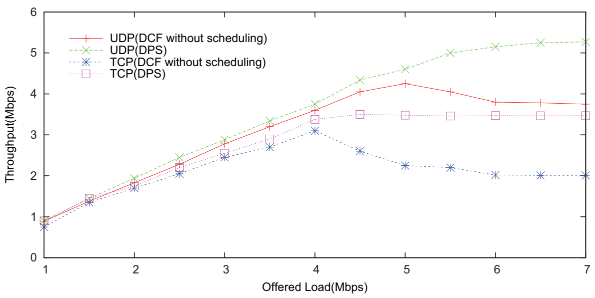

Fig. 2. Throughput for UDP and TCP with the increase of offered load. The plots show average throughput of all 50 AP-client links.

The average throughput can only show the average impact on all the links. We choose 20 representative links from all 50 links to show the impact on individual link. As we can see in Fig. 2, when offered load is greater than 6.5Mbps, the average throughput of both UDP and TCP begins to stabilize, indicating that all the links have entered saturated state. So, we examine the throughput of these 20 links in such state and the results are given in Fig. 3. We can find that *DPS* provides significant UDP and TCP throughput gains for all the links over DCF without any scheduling.

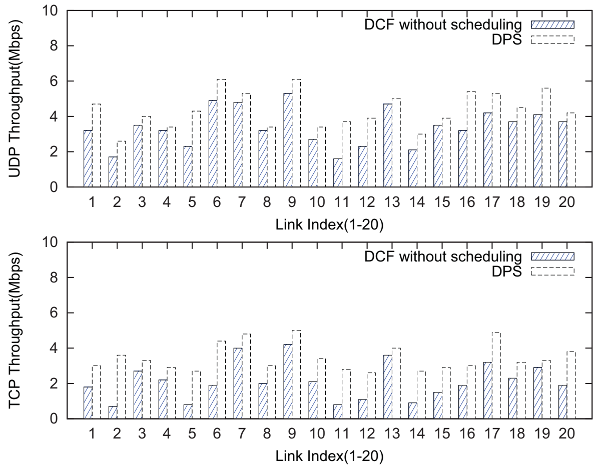

Fig. 3. Throughput gain for every representative link when the offered load is 6.5Mbps. The top is for UDP flow and the bottom is for TCP flow. The plots show every link's throughput and links are indexed by 1 to 20.

We also measured the impact of DCF without any scheduling and *DPS* with downlink packets scheduling on the delay of UDP. We choose the result plot when downlink rate was at 6.5Mbps, because under this load the throughput plot became smooth, indicating that all the links entered saturated state. The results are shown in Fig. 4. *DPS* can reduce the delay across all link by more than 35%. For the links with high delay when using DCF without any scheduling, the improvement is particularly great.

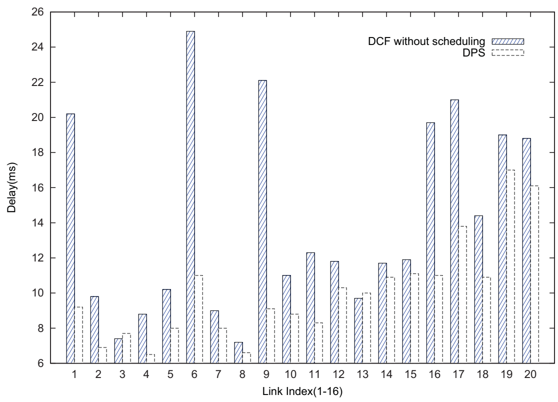

Fig. 4. Delay of UDP when offered load was fixed at 6.5Mbps. Links are indexed by 1 to 20.

The core idea of *DPS* is to schedule downlink packets to maintain high transmission success rate and reduce the overhead of solving collisions and retransmission. To demonstrate this characteristic, we examine the ratio of the number of successfully transmitted packets and the number of transmit attempts. We show the result of UDP in Fig. 5. With the increase of offered load, the ratio decreases sharply when using DCF without any scheduling, indicating large quantities of transmission failure and retransmission. On the contrast, the ratio in *DPS* keeps stable nearly at 0.9. That is the key of *DPS* to improve performance of network.

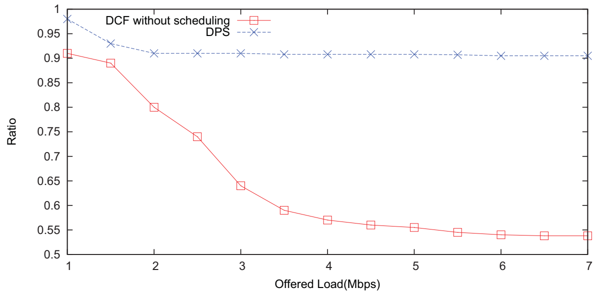

Fig. 5. The ratio of the number of successfully transmitted packets and the number of transmit attempts for UDP traffic with the increase of offered load.

VI. Conclusion

In this paper, we proposed *DPS* — a framework that combines centralized scheduling with DCF mechanism to mitigate performance loss caused by uncoordinated channel access method in enterprise WLAN. In *DPS*, a controller schedules the forwarding of downlink packets. APs still use DCF to send downlink packets to client according the instructions of controller. *DPS* requires no modifications to 802.11-compliant clients, and advances DCF by scheduling downlink packets judiciously. Under *DPS* framework, we proposed a downlink packets scheduling algorithm. This algorithm is under graded SINR-based interference model. The basic principle of the algorithm is that: packet should achieve high PPR under the prerequisite of only affecting the PRR of activated links slightly. Our simulation results demonstrate that, *DPS* can provide significantly improvement to the performance of enterprise WLAN, compared with DCF without any scheduling.

Acknowledgment

This research is partially supported by the National Science Foundation of China under Grant No. 61070211 and No. 61003304.

References

[1] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Communications Magazine*, vol. 43, no. 9, pp. 23–30, 2005.

[2] Y.Bejerano and R.Bhatia, "Mifi: a framework for fairness and qos assurance in current ieee802.11 networks with multiple access points," in *IEEE INFOCOM*, 2004.

[3] Y. c. Cheng, J. Bellardo, P. B. u, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: Solving the puzzle of enterprise 802.11 analysis," in *ACM SIGCOMM*, 2006.

[4] M.Rodrig, C.Reis, R.Mahajan, D.Wetherall, J.Zahorjan, and E.Lazowska. (2005) Crawdad data setuw/sigcomm2004.

[5] 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 *ACM MobiCom*, 2009.

[6] S.Ray, J.B.Carruthers, and D.Starobinski, "Rts/cts-induced congestion in ad hoc wireless lans," in *WCNC*, 2003.

[7] X. Yang and N. Vaidya, "On physical carrier sensing in wireless ad hoc networks," in *IEEE INFOCOM*, 2005.

[8] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans," in *ACM SIGCOMM*, 2005.

[9] E. Magistretti, K. K. Chintalapudi, B. Radunovic, and R. Ramjee, "Wifi-nano: reclaiming wifi efficiency through 800 ns slots," in *ACM MobiCom*, 2011.

[10] S. Sen, R. R. Choudhury, and S. Nelakuditi, "Csma/cn: Carrier sense multiple access with collision notification," in *ACM MobiCom*, 2010.

[11] G. Sharma, R. Mazumdar, and N. Shroff, "On the complexity of scheduling in wireless networks," in *ACM MobiCom*, 2006.

[12] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[13] R. Maheshwari, S. Jain, and S. R. Das, "A measurement study of interference modeling and scheduling in low-power wireless networks," in *ACM SenSys*, 2008.

[14] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of concurrent transmission in wireless sensor networks," in *ACM SenSys*, 2006.

[15] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multi-hop wireless networks,," in *IMC*, 2005.

[16] N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki, "Online estimation of rf interference," in *ACM CoNEXT*, 2008. [Online]. Available: http://doi.acm.org/10.1145/1544012.1544016

[17] A. Kashyap, S. Ganguly, and S. R. Das, "A measurement-based approach to modeling link capacity in 802.11-based wireless networks," in *ACM MobiCom*, 2007.

[18] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-based models of delivery and interference in static wireless networks," in *ACM SIGCOMM*, 2006.

[19] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan, "A general model of wireless interference," in *ACM MobiCom*, 2007.

[20] J. Huang, S. Liu, G. Xing, H. Zhang, J. Wang, and L. Huang, "Accuracy-aware interference modeling and measurement in wireless sensor networks," in *ICDCS*, 2011.