IEICE TRANS. COMMUN., VOL.E90–B, NO.8 AUGUST 2007

PAPER

Co-DRR: An Integrated Uplink and Downlink Scheduler for Bandwidth Management over Wireless LANs

Huan-Yun WEI[†a)], **Ching-Chuang CHIANG**[†], ***Nonmembers**, and* **Ying-Dar LIN**[†], ***Member***

SUMMARY Bandwidth management over wired bottleneck links has been an effective way to utilize network resources. For the rapidly emerging IEEE 802.11b Wireless LAN (WLAN), the limited WLAN bandwidth becomes a new bottleneck and requires bandwidth management. Most possible existing solutions only exclusively focus on optimizing multimedia traffic, pure downlink or pure uplink fairness, or are incompatible with IEEE 802.11b. This study proposes a cooperative deficit round robin (co-DRR), an IEEE 802.11b-compatible host-based fair scheduling algorithm based on the deficit round robin (DRR) and distributed-DRR (DDRR) schemes, to make the uplink and downlink quantum calculations cooperate to simultaneously control uplink and downlink bandwidth. Co-DRR uses the standard PCF mode to utilize the contention-free period to compensate for the unfairness in the contention period. Numerical results demonstrate that: co-DRR can scale up to 100 mobile hosts even under high bit error rate (0.0001) while simultaneously achieving uplink/downlink long-term fairness (CoV<0.01) among competing mobile hosts.

key words: *scheduling, fair queuing, bandwidth management, deficit round robin, 802.11b*

1. Introduction

Bandwidth management has been a mature technology in wired access links to alleviate the drawbacks caused by the bandwidth mismatch between LAN (large bandwidth) and WAN (scarce bandwidth). A large number of commercial or open-source bandwidth management gateways installed at the bottlenecked LAN-to-WAN links can allocate bandwidth according to administrative policies. Thus their important, interactive, or mission-critical traffic such as voice over IP (VoIP), e-business, and ERP flows are not blocked by less important traffic like FTP.

Due to the rapidly emerging WLAN whose bandwidth is limited, the bottleneck sometimes shifts to the LAN side. Three reasons are: (1) ideally the maximum bandwidth of IEEE 802.11b can achieve 11 Mbps, which is relatively scarce compared to the widely-deployed 100 Mbps Fast Ethernet in a corporate network; (2) the WLAN medium is shared among competing uplink/downlink traffic flows within a wireless cell (half-duplex); (3) the acknowledgement overhead of CSMA/CA compared to CSMA/CD consumes additional bandwidth. Our recent benchmarking results among 9 vendors reveal that most commercial access point and WLAN card can only achieve 6–7 Mbps only when using UDP traffic pumping from 100 Mbps to a "single" 11 Mbps mobile host.

Because of the half-duplex nature of WLAN, the bandwidth manager residing at the access point (AP) must simultaneously control uplink and downlink traffic. Otherwise, a single AP transmitting downlink traffic can only grab little bandwidth when contending with multiple mobile hosts (MHs) transmitting the uplink traffic. Besides, if both directions are only independently controlled, fairness and sharing of bandwidth between uplink and downlink are difficult to achieve. Due to the growth of public WLAN services in airports, hotels, libraries, coffee shops, and network gaming shops, the need for fair bandwidth sharing among hosts becomes obvious.

To manage bandwidth over wired access links, various packet scheduling algorithms have been proposed and are extensively studied in [1]–[3]. However, adapting the wired fair queuing schemes to WLAN domain is non-trivial because of WLAN problems such as location-dependent burst errors, varying bandwidth, and contention in the half-duplex link. Therefore, an important research issue is on how to extend the existing solutions from wired networks to wireless networks.

In recent years, fair scheduling and Quality of Services (QoS) in WLAN have received significant attention from the networking research community. Various solutions have been proposed. These schemes can be categorized into QoS-aware MAC protocols [4] and modified queuing solutions [5]–[10]. The former schemes are difficult to deploy unless they are standardized by IEEE for widespread usage. Nowadays, the IEEE 802.11b standard [11] is the most widely used and mature WLAN standard. For compatibility and popularity, the latter schemes comply with the 802.11b standard and modify the existing queuing methods to render Class Based Queuing-Channel State Dependent Packet Scheduling algorithm (CBQ-CSDPS) [6], Wireless Packet Scheduling algorithm (WPS) [9], Channel Independent Fair Queuing algorithm (CIF-Q) [10], etc. However, all of them only consider downlink scheduling. Though some of them claim to support uplink scheduling, they require introducing new protocol messages to continuously monitor the status of queues in the MHs. The current IEEE 802.11b standard does not support exchanging explicit information. Besides, some proposed designs [6], [12] only focus on optimizing the quality of multimedia transmissions without considering the fairness of other traffic.

This study proposes an IEEE 802.11b-compatible scheme to integrate uplink and downlink scheduling. The scheme extends the deficit round robin (DRR) [13] and the

Manuscript received June 23, 2006.

[†]The authors are with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan.

a) E-mail: hywei@L7.com.tw

DOI: 10.1093/ietcom/e90–b.8.2022

distributed-DRR (DDRR) [12] schemes to achieve the purpose, hence called cooperative Deficit Round Robin (co-DRR). For each MH, co-DRR employed at the AP correlates two quantum values, one for downlink and the other for uplink. The two assigned quantum values are to determine the bandwidth allocation among MHs and between uplink and downlink. Furthermore, if one MH has no downlink data, the AP will redistribute its downlink bandwidth to its uplink, and vice versa. By making the two quantum values cooperate with each other using the Point Coordination Function (PCF), co-DRR can achieve the objective of host-based fair scheduling by utilizing the contention-free period to compensate for the unfairness in the contention period.

The rest of the work is organized as follows. Section 2 outlines the IEEE 802.11b access control mechanism and the DRR/DDRR queuing methods. Section 3 investigates the proposed co-DRR scheme in details. The simulation model and the simulation results are presented in Sect. 4. Finally, conclusions and future works are given in Sect. 5.

2. Background

2.1 IEEE 802.11b WLAN Standard

The IEEE 802.11b standard defines the functionality of MAC and PHY. This study focuses on the MAC sublayer. The architecture of the MAC sublayer on infrastructure network, as shown in Fig. 1, provides the Point Coordination Function (PCF) through the services provided by the Distributed Coordination Function (DCF). Both the DCF and PCF can operate concurrently to provide alternating contention and contention-free modes.

2.1.1 Distributed Coordination Function (DCF)

The fundamental access method of the MAC sublayer is DCF, which employs Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. CSMA/CA is a contention-based protocol similar to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) of IEEE 802.3 Ethernet. The major difference between CSMA/CA and CSMA/CD is that the former requires the receiver to reply a positive acknowledgement (ACK) back to the transmitter when a frame is correctly received because collision cannot be actively detected. If no ACK returns, the sender will retransmit.

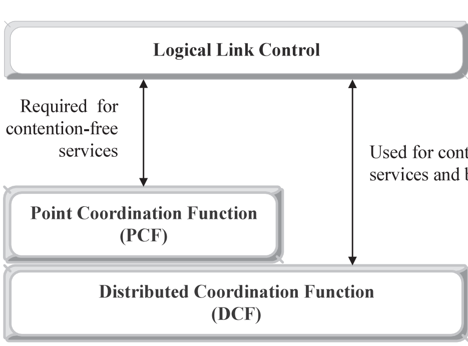


Fig. 1 The MAC architecture of 802.11b.

In IEEE 802.11b standard, a time interval between frames is called Inter-Frame Space (IFS). Before transmitting any frame, a host should determine that the medium has been idle for the specified interval through the carrier-sense function. Four IFSs, from the shortest to the longest, namely short IFS (SIFS), DCF IFS (DIFS), extended IFS (EIFS), and PCF IFS (PIFS) are defined (Fig. 2 and Fig. 3). For example, if an ACK assigned with SIFS and a new data packet assigned with DIFS are simultaneously waiting for the busy channel, the ACK will be transmitted first.

2.1.2 Point Coordination Function (PCF)

The optional priority-based PCF provides Contention-Free Periods (CFP) for processing time-critical information transfers. The PCF is a centralized access mode based on polling mechanisms. This access mode uses a Point Coordinator (PC) within the AP to determine which host is currently having the right to transmit. Figure 3 depicts several frame transfers during a typical CFP. The PC senses the medium at the beginning of each CFP and defers the PIFS interval for accessing the medium. Then, the PC sends a beacon frame announcing the beginning of the CFP and notifying the beacon interval (superframe time) and the maximum length of the CFP (CFPMaxDuration). Thus, all hosts then update their Network Allocation Vector (NAV) to the

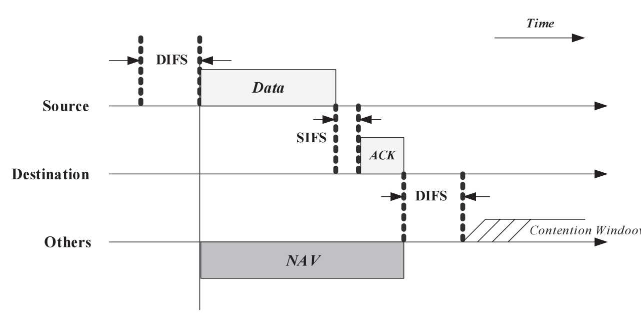


Fig. 2 Basic access method.

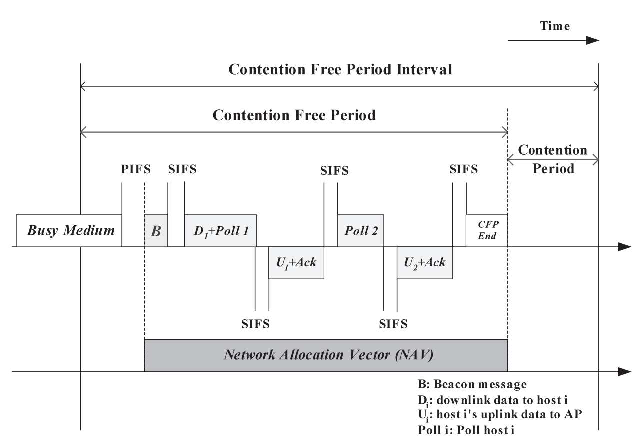


Fig. 3 An example of PCF frame transmissions.

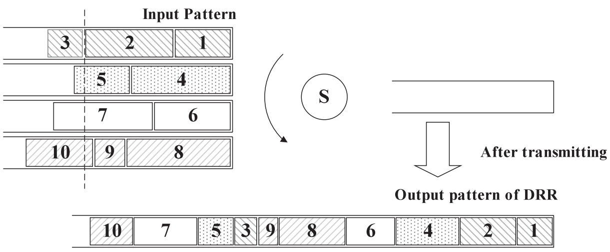


Fig. 4 The transmission state of packets in DRR.

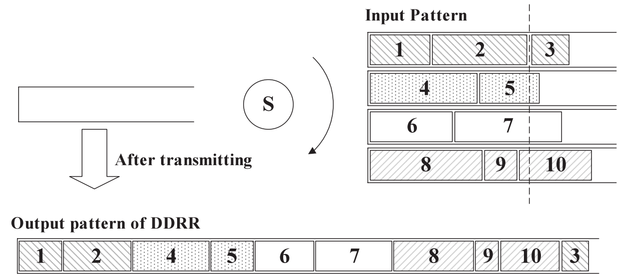


Fig. 5 The transmission state of packets in DDRR.

CFPMaxDuration. An AP must be deterministically set to reserve at least one data frame for the Contention Period (CP) in a superframe time, thus the maximum value for CF-PMaxDuration is limited. After the initial beacon frame, the PC waits for at least one SIFS time, and then transmits one of the following frame types: a data frame, a CF-Poll frame, or a Data-CF-Poll frame. The PC may terminate any CFP by sending a CF-End frame at or before the CFPMaxDuration based on the availability of traffic and the size of the polling list.

2.1.3 Polling List Management (PLM)

The PC maintains a "polling list" to select next eligible host for receiving CF-Polls during the CFP. However, the standard does not specify the details. Thereby we can develop our mechanism for managing the polling list that follows the standard while achieving our objectives.

2.2 Packet Schedulers

The co-DRR scheme utilizes the Deficit Round Robin (DRR) and the Distributed Deficit Round Robin (DDRR) to solve the problem. This section briefly reviews the two schemes.

2.2.1 Deficit Round Robin (DRR)

The DRR [13] algorithm is an extension of the round robin algorithm. It can achieve long-term fairness in terms of throughput. DRR requires only O(1) complexity to process a packet. Moreover, DRR is amenable to variable-length packets and simple enough to implement without any timestamp computation. The mechanism allocates a counter called deficit counter for holding the current value of the credit for each queue. The credit can be accumulated from former rounds if the credit is not run out. So DRR keeps track of the deficits, and those queues with deficits are compensated in the succeeding rounds. Figure 4 provides a DRR example that schedules four queues with equal quantum size. When servicing the second queue, the deficit is insufficient to schedule the packet 5. Thus the DRR turns to select packet 6 from the third queue and results in the final output pattern.

2.2.2 Distributed Deficit Round Robin (DDRR)

In the CFP, the AP polls the mobile hosts. If the polled host has any data, the host will reply an uplink data frame to the AP. However, the AP would never know the size of the upcoming data until the frame arrives at the AP. For example, when the scheduler S in Fig. 5 sends a polling frame to the MH_2 (the second queue). The MH_2 replies with packet 4. The scheduler S does not know the residual deficit is insufficient for packet 5. The AP then polls the MH_2 again to fetch packet 5 and will get a negative deficit after deducting the size of the packet 5. The scheduler then stops polling the MH_2 and turns to poll the MH_3. This method is known as the Distributed Deficit Round Robin [12] (DDRR). DDRR traces the deficits so that those queues with negative deficits are paid back in the succeeding rounds.

3. Cooperative Deficit Round Robin (Co-DRR)

This section describes the co-DRR mechanism based on the PCF mode to (1) manage the uplink/downlink wireless channel access among many mobile hosts and (2) decrease the waste of bandwidth caused by the overheads of the protocol.

3.1 Motivations

The proposed scheme is motivated by using the DRR for managing the downlink traffic while using the DDRR for the uplink traffic. The first problem is how to simultaneously do the downlink scheduling as in Fig. 4 while doing the uplink scheduling at the same time by polling as in Fig. 5. Consider an oversimplified example in Fig. 6 that the AP always works in the PCF mode. To achieve weighted fairness among the MH_1, MH_2, and MH_3, co-DRR maintains two sets of cooperative deficit counters, Dd_i for the downlink traffic towards MH_i, and Du_i for the uplink traffic from MH_i. The maintenance of Dd_i and Du_i basically follows the DRR and the DDRR, respectively. However, when adding downlink's quantum Qd_i to the Dd_i, the Du_i is also added by the uplink's quantum Qu_i. This design ensures that the chances between uplink and downlink traffic are the same.

In the example, Qd_i=Qu_i=1000 bytes and the AP coordinates all the transmission in the CFP mode. Initially, the

AP adds Qd_i to Dd_i and adds Qu_i to Du_i. Then the AP sends a data frame of 100 bytes with a poll to MH_1. The MH_1 replies a data frame of 1000 bytes. Thus Qd_1 and Dd_1 are updated as 1000−100=900 and 1000−1000=0, respectively. The uplink deficit equals to zero (the uplink deficit is inadequate). Assume that the next downlink frame towards MH_1 is larger than 900 (the downlink deficit is inadequate). The AP begins to serve MH_2. After adding the quantum values to the uplink/downlink deficit counters, the AP continues to send downlink frames with polling to the MH_2 until (1) Dd_2 is inadequate for the next frame to be sent and (2) Du_2 becomes non-positive. The reason of (2) is that the AP would not know 300 bytes is inadequate for the next uplink frame so the AP sends a pure polling and gets a 600-byte frame. It is only when the Du_2 becomes non-positive that the AP will know that the uplink deficit is insufficient. This cycle repeats among the three hosts. In this way, the uplink/downlink of each AP-MH_i link can be controlled.

3.2 Architecture

Based on the motivation, Fig. 7 depicts an overview of the co-DRR architecture. The co-DRR residing at a PCF-capable AP consists of a Downlink Manager (DM) using DRR and a Polling List Manager (PLM) using DDRR for the uplink traffic. Administrative policies can define the proportion of bandwidth among the MHs, or even the proportion of bandwidth between downlink and uplink traffic within an AP-MH_i pair by assigning different quantum values to Du_i and Dd_i. The Du_i, Dd_i, Qu_i and Qd_i have been defined in Sect. 3.1.

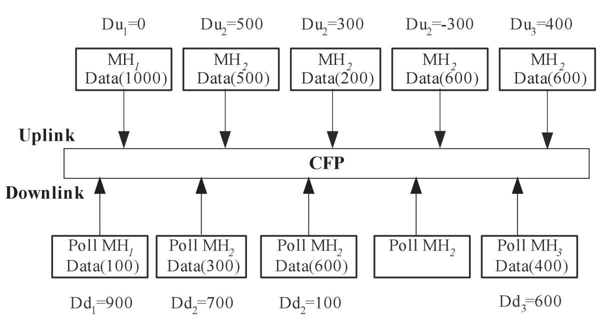


Fig. 6 Oversimplified example of co-DRR.

3.3 Co-DRR Algorithm

The PLM and the DM cooperating in Fig. 6 always works in the PCF mode. However, the example is over-simplified because the IEEE 802.11b standard defines that the CP must exist for the new MHs to join up. In the CP mode, AP and all the MHs contend to transmit frames. The AP cannot control any access but can only compete for the channel. Thus, co-DRR principally uses an extra synchronization mechanism during the CFP to compensate for the uncontrolled behaviors in the previous CP. As long as the CFP is longer than the CP, co-DRR can always compensate for the uncontrolled behaviors. The following sections describe the co-DRR algorithms for PLM, DM, and the synchronization mechanism both in CP and CFP.

3.3.1 Polling List Management (PLM)

During the CP, MHs actively contend for the channel access to transmit data to the AP. Because the AP cannot block the actively transmitted traffic, the deficit counter corresponding to the sender is decreased by the amounts of bytes received. During the CFP, the PLM decides whether to poll a MH_i or not by examining the corresponding deficit counter Du_i. If $Du_i > 0$, MH_i will be polled. After the uplink data arrives, Du_i is decreased by the amount of bytes received. If the polled MH_i has nothing to send, the MH_i will respond a NULL data packet to the AP [11]. Figure 8 illustrates the

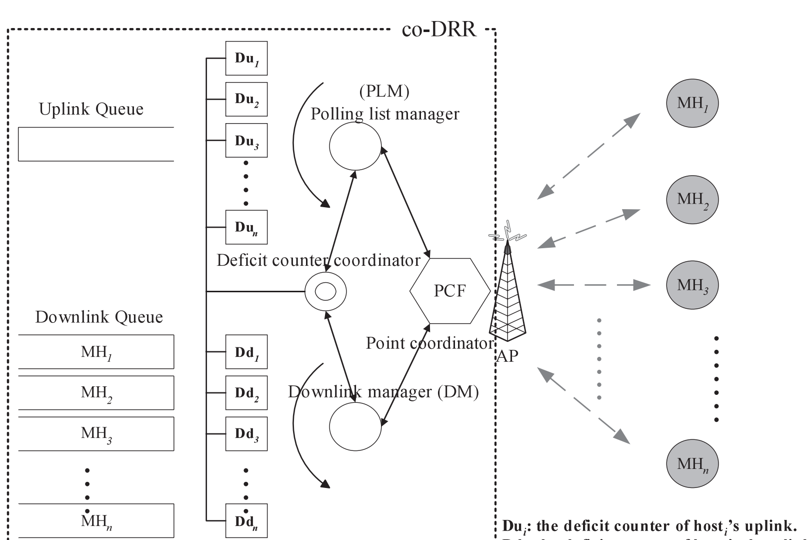


Fig. 7 The architecture of co-DRR.

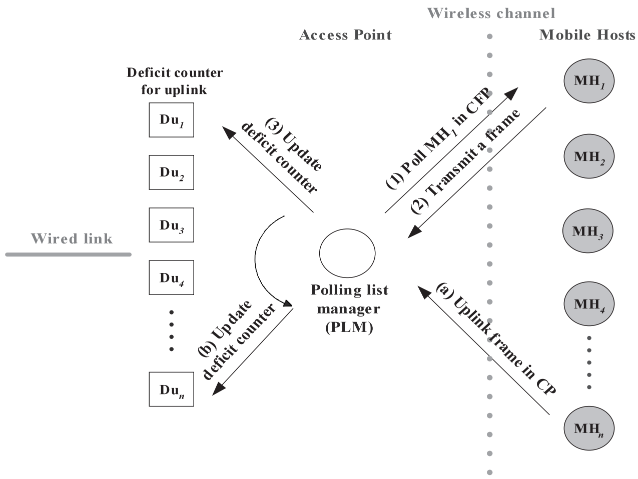


Fig. 8 Working flow of the uplink polling list manager.

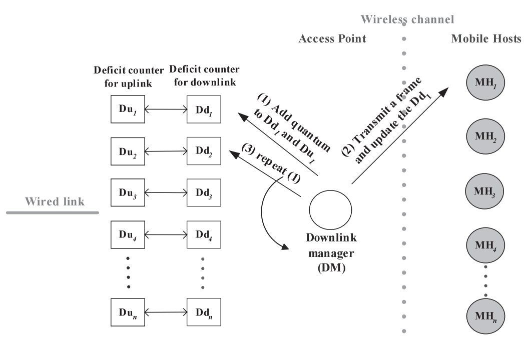


Fig. 9 Working flow of the downlink manager.

CP and the CFP working flow of the PLM. Arrows (a), (b) and arrows (1), (2), (3) represent the flow of the CP and the CFP, respectively.

The PLM resets the Du_i when receiving a NULL data frame from the MH_i. For bandwidth borrowing between uplink and downlink of each AP-MH_i link, the remaining uplink deficit Du_i is added to Dd_i. However, if the downlink queue of the host is also empty, the Dd_i is set to zero.

3.3.2 Downlink Manager (DM)

The DM acts almost the same in both CP and CFP. In Fig. 9, the DM chooses a MH_i to send the data based on the DRR algorithm. To manage both the downlink and uplink bandwidth, when adding the downlink quantum to Dd_i, Du_i is also added by its corresponding uplink quantum. The values added to Dd_i and Du_i can be different according to administratively specified proportions between uplink and downlink of MH_i. Arrows (1), (2), and (3) in Fig. 9 represent the working flow of the downlink manager.

If no pending data queued for the MH_i's downlink queue in a round, the Dd_i of the downlink queue will also be reset to zero. For bandwidth borrowing between uplink and downlink of the same MH_i, remaining downlink deficit Dd_i belonging to the MH_i is added to Du_i. However, if the uplink queue of the host is also empty, the Du_i is set to zero and the DM then processes the next output queue.

3.3.3 Synchronization Mechanism

During the CP, the DM can only service the downlink traffic but cannot send the polling message for retrieving any uplink traffic. Suppose the DM has serviced the downlink queues from i to j (assume $i < j$) in the CP using the DRR. Because the AP cannot poll from i to j in CP, when entering the CFP, the AP first polls from MH_i to MH_j to see if they need any compensation for the uplink traffic by checking whether they have a positive uplink deficit ($Du_i > 0$). If the uplink deficit of a host k ($i \leq k \leq j$) is non-positive, the MH_k must have already successfully transmitted uplink traffic and consumed the deficit Du_k in the previous CP. After polling the last queue j, the DM and the PLM are synchronized to point to the same mobile host MH_j. Afterwards, the co-DRR can simultaneously service the uplink and downlink traffic of MH_j by scheduling out MH_j's downlink data (if $Dd_j > 0$) with a polling frame towards the MH_j to fetch any uplink data (if $Du_j > 0$) as in Fig. 6.

To synchronize the PLM and the DM, co-DRR uses a *SYNC* variable to represent $j - i$. Whenever the DM steps one MH, the *SYNC* is increased by 1. Whenever the PLM steps one MH, the *SYNC* is decreased by 1. Once entering the CFP, if *SYNC* > 0, the DM will wait for PLM to point to the same mobile host MH_j. Thus the PLM polls from MH_i to MH_j to catch up the DM. The *SYNC* is reduced by 1 whenever the PLM steps one MH. When the *SYNC* equals to zero, the PLM and the DM are synchronized to point to the same host MH_j. Then the AP may send the data to (if $Dd_j > 0$) and poll (if $Du_j > 0$) the same host at the same time.

3.3.4 Pseudo-Code

Figure 10 shows the algorithm of processing deficit counters. Figure 11 illustrates the algorithm of synchronization between the PLM and the DM.

3.4 A Running Example

Figure 12 gives a running example of a possible superframe time. Three MHs and one AP are transmitting bidirectional data. All uplink/downlink quantum values are set to 1000 bytes. The Qu_i and Qd_i are added to the Du_i and Dd_i respectively whenever the DM steps to MH_i. Suppose initially all the Du_i and Dd_i are zero, and both the PLM and the DM point to MH_3.

In the CP, all MHs and the AP freely contend for the channel (the CP in Fig. 12). The bandwidth usage of each host is computed and reflected on the deficit counters kept by the AP. Initially the DM points to the MH_1. So Qd_i and Qu_i are added to Dd_i and Du_i respectively. The MH_1 first successfully transmits an uplink 500-byte frame so the Du_1 becomes 500. Subsequently, the AP successfully transmits

```
Downlink:
  p=dequeue( queue(i) );           // dequeue downlink packet p
  IF (p= =NULL)                    // the queue is empty
      Du_i = Dd_i + Du_i ;              // add downlink to uplink
      Dd_i  = 0;                        // reset downlink
  ELSE IF (Dd_i  >= sizeof(p))          // deficit sufficient for p
      Dd_i  = Dd_i – sizeof(p);         // update deficit
Uplink:
  IF (NULL data received after polling)      // if no uplink traffic
      IF (is_empty( queue(i) ))         // if no downlink traffic
          Du_i = 0;  Dd_i = 0;          // reset both
  ELSE
              Dd_i = Dd_i + Du_i ;            // add uplink to downlink
  IF (non-NULL packet p received)
      Du_i=Du_i-sizeof(p);              // update deficit
```

Fig. 10 Maintenance of deficit counters.

```
Synchronization in CFP:
Fetch i from PLM
Fetch j from DM
Loop until CFP end
  IF (SYNC > 0 AND Du_i > 0)          // only poll uplink
      Send pure polls to host i until Du_i<0;
      i=(i+1) mod (max i);         // point to the next MH
      SYNC=SYNC-1;                      // i=i+1 while j remains
  ELSE IF (SYNC == 0)             // now i==j
      Send data or data+poll or poll to host i until inadequate Du_i and Dd_i
      i=(i+1) mod (max i);         // point to the next MH
      j=(j+1) mod (max j);         // point to the next MH
      Du_j=Du_j+Qu_j, ; Dd_j=Dd_j+Qd_j ;   // because DM steps one MH
Contention in CP:
Fetch j from DM
Loop until CP end
      Send pure data until inadequate Dd_j
      j=(j+1) mod (max j);         // point to the next MH
      Du_j=Du_j+Qu_j, ; Dd_j=Dd_j+Qd_j ;   // because DM steps one MH
      SYNC=SYNC+1;                      // j=j+1 while i remains
```

Fig. 11 Synchronization in CFP and contention in CP.

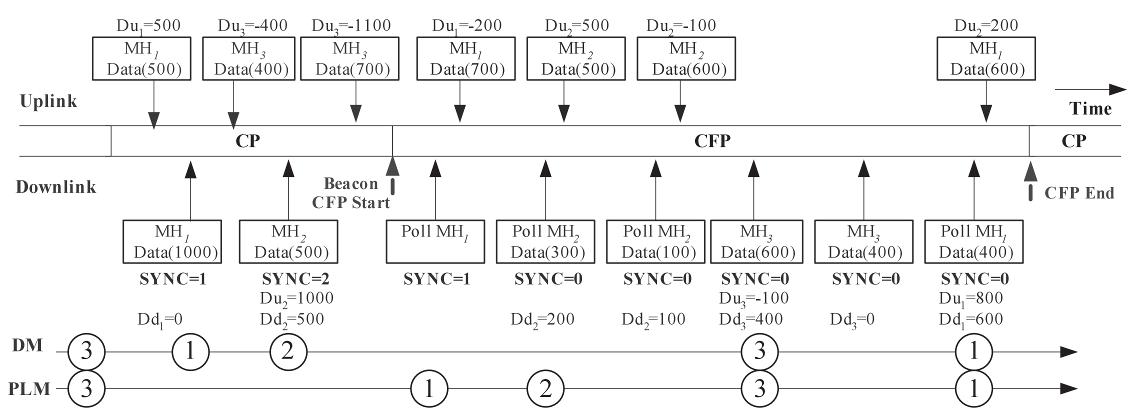


Fig. 12 A running example.

a 1000-byte frame and the Dd_1 becomes zero. The Dd_1 is insufficient for the next frame to send so the DM points to the MH_2 and adds the quantum values to the deficit counters. Finally in the CP, the Du_i for MH_1, MH_2, and MH_3 are 500, 1000, −1100 while the Dd_i for MH_1, MH_2, MH_3 are 0, 500, 0. *SYNC* is incremented twice because the DM steps through two hosts during the CP. When entering the CFP where the AP can govern all the channel access, the AP sees a positive *SYNC*. So the AP polls the MHs in the following orders:

1. **Polling MH_1:** The PLM steps to MH_1 so the *SYNC* is first reduced by 1. Since *SYNC* and Du_1 are both positive, the AP continues to send a pure polling to MH_1. The AP then receives 700 bytes data from MH_1 and decreases the Du_1 to −200. The PLM then steps to MH_2 because of $Du_1 < 0$.
2. **Polling MH_2:** Reducing the *SYNC* again before the polling makes the *SYNC*=0 (synchronized). AP then checks both Du_i and Dd_i and finds that they can satisfy the requirement of the algorithm (Du_1 is positive and Dd_2 is greater than or equal to the size of first pending packet in downlink queue). Thus, the AP sends a 300 bytes data with a polling to MH_2 and receives

500 bytes from MH_2. Hence, the new Du_2 and Dd_2 are changed into 500 and 200, respectively.

3. **Polling MH_2 again:** Since $Du_2 > 0$ and Dd_2 is adequate for the next downlink packet (100 bytes), step 2 repeats again. By sending 100 bytes of data and receiving 600 bytes of data, the new Du_2 and Dd_2 becomes −100, 100, respectively. Though $Dd_2 > 0$, Dd_2 is insufficient to send the next pending packet in the downlink queue of MH_2. The AP then tries to choose the next possible MH to poll.
4. **Polling MH_3:** Since the DM steps to MH_2, Qu_2 and Qd_2 are added to Du_2 and Dd_2 respectively. Though Du_3 is still less than zero, Dd_3 is greater than the size of the next pending frame. So the AP only sends the downlink data to MH_3 and reduces Dd_3. This repeats until Dd_3 is insufficient.
5. **Polling MH_1:** After all the MHs are polled, the AP starts a new round to poll from MH_1. Of course the Du_1 and Dd_1 are respectively added by the Qu_1 and Qd_1 when DM steps to the MH_1.

The algorithm stops polling when CFP is terminated.

4. Simulation and Numerical Results

4.1 Simulation Environment

This section presents ns-2 [14] simulation results to investigate the fairness, the scalability and the performance of the proposed scheme. The topology of the simulation network is a simple model (see Fig. 13).

Three mobile hosts MH_1, MH_2, and MH_3 are uniformly distributed around an AP and they are uploading their packets to wired hosts W_1, W_2, and W_3 via the AP, respectively. The wireless channel is an 11 Mbps bottleneck link (IEEE 802.11b).

We used the default values given in [11] for all the DCF and PCF related attributes. Table 1 illustrates the important parameters of the simulation setup. We calculate the effective throughput, also known as the goodput [15], using the data successfully received at the receiver divided by the testing period (120 seconds).

Since we focus on fair sharing of bandwidth, to minimize variances between different experiments, we use the same packet size in all experiments. How packet size affects throughput efficiency, which caused by characteristics of protocol itself, has been discussed in [16].

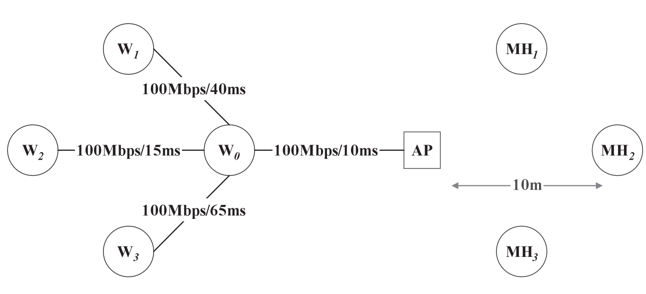


Fig. 13 Simulation model for small scale tests.

4.2 Baseline Experiments

This section demonstrates the inability of (I) pure IEEE 802.11b without any scheduling and (II) IEEE 802.11b with downlink scheduling using DRR. The traffic flows of simulations are shown in Table 2, and all are constant bit rate traffic with fixed 1000-byte packet. For easier observation on how co-DRR performs, we just use UDP flows at present.

For scenario (I), both the uplink and downlink traffic contends for the access channel, and the downlink traffic is fed into a single first-in-first-out (FIFO) queue. Hence, the results shown in Fig. 14(a), are totally unfair. For scenario (II) downlink scheduling is replaced by the DRR instead of the FIFO. Figure 14(b) presents the same priority level for all MHs with quantum value setting to 1000 bytes. During 0 to 40 seconds, the three downlink flows can be controlled to fairly share the bandwidth. However, when the three uplink flows join at the 40th second, the downlink DRR can just manage the leftover bandwidth (three downlink flows all obtain about 750 kbps). Uplink traffic grasps 3/4 of the WLAN bandwidth because three out of the four belong to uplink traffic. Consequently, an AP employing even more perfect downlink schedulers than DRR can still control only the leftover bandwidth. In Fig. 14(c), we alter their quantum values to 1000, 2000, and 3000 bytes, respectively. The uplink flows still occupy 3/4 of the WLAN bandwidth and only the downlink bandwidth is 1:2:3 shared. In summary, managing the uplink traffic is essential when uplink traffic becomes the dominant.

4.3 Effectiveness of Bandwidth Allocation

This section evaluates the effectiveness of co-DRR's bandwidth allocation. The traffic model approximates the setup in Table 2. However, to simultaneously observe the effec-

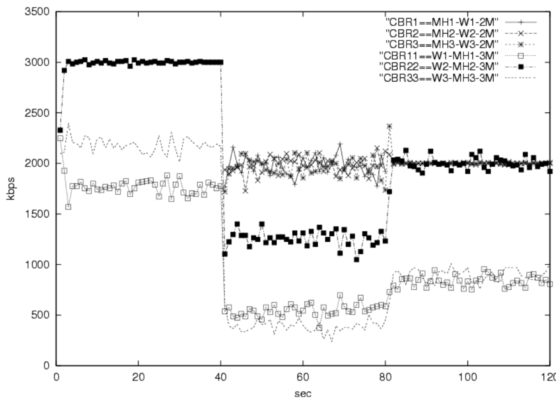
Table 1 Parameter values used in our simulations.

Parameter	Value	Parameter	Value
DIFS	50 µs	Superframe	20 TU*
PIFS	30 µs	Min. CP	For 2346-bytes frame transmission
SIFS	10 µs	Max. CFP = Superframe – Min. CP	

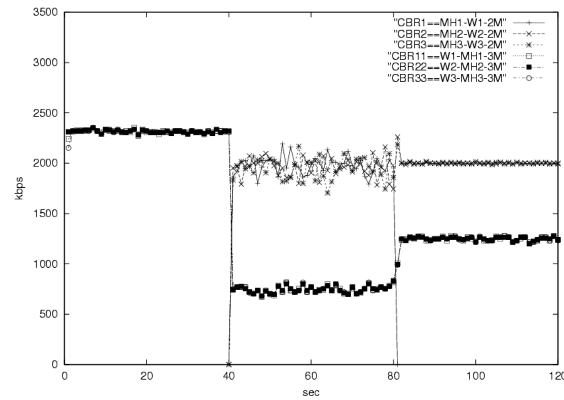
*1TU = 1024 µs

Table 2 Traffic flows: 3 uplink flows and 3 downlink flows.

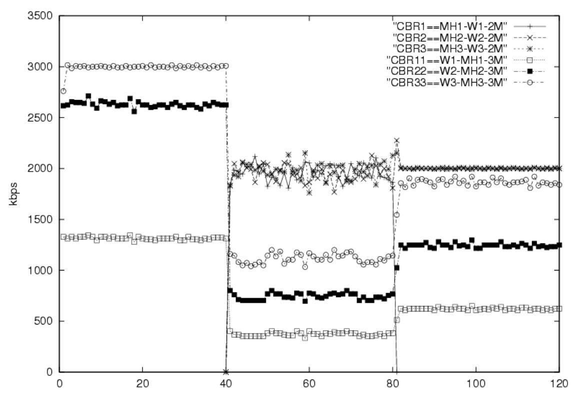
Hosts	Uplink Traffic		Downlink Traffic	
MH_1	***CBR1***	MH_1→W_1 (2Mbps)	***CBR11***	W_1→ MH_1 (3Mbps)
	Start at 40sec. / Stop at 80 sec.		Start at 0sec. / Stop at 120 sec.	
MH_2	***CBR2***	MH_2→W_2 (2Mbps)	***CBR22***	W_2→ MH_2 (3Mbps)
	Start at 40sec. / Stop at 120 sec.		Start at 0sec. / Stop at 120 sec.	
MH_3	***CBR3***	MH_3→W_3 (2Mbps)	***CBR33***	W_3→ MH_3 (3Mbps)
	Start at 40sec. / Stop at 120 sec.		Start at 0sec. / Stop at 120 sec.	



(a) FIFO queue.



(b) DRR queue.



(c) DRR queue with priority.

Fig. 14 Pure access and downlink scheduling.

Table 3 Quantum values assigned to hosts. ($Host_1$: $Host_2$: $Host_3$ = 1: 2: 3)

Hosts	Uplink Quantum (bytes)	Downlink Quantum (bytes)
MH_1	1000	1000
MH_2	2000	2000
MH_3	3000	3000

tiveness of bandwidth allocation and bandwidth borrowing, we modify the start and stop time of the flows. All flows start at the 0 second, and *CBR1* stops at the 20 second, *CBR11* stops at the 40 second, *CBR22* stops at the 80 second and others stop at the 120 second. The priorities assigned to the hosts are proportional to their quantum values, as shown in Table 3.

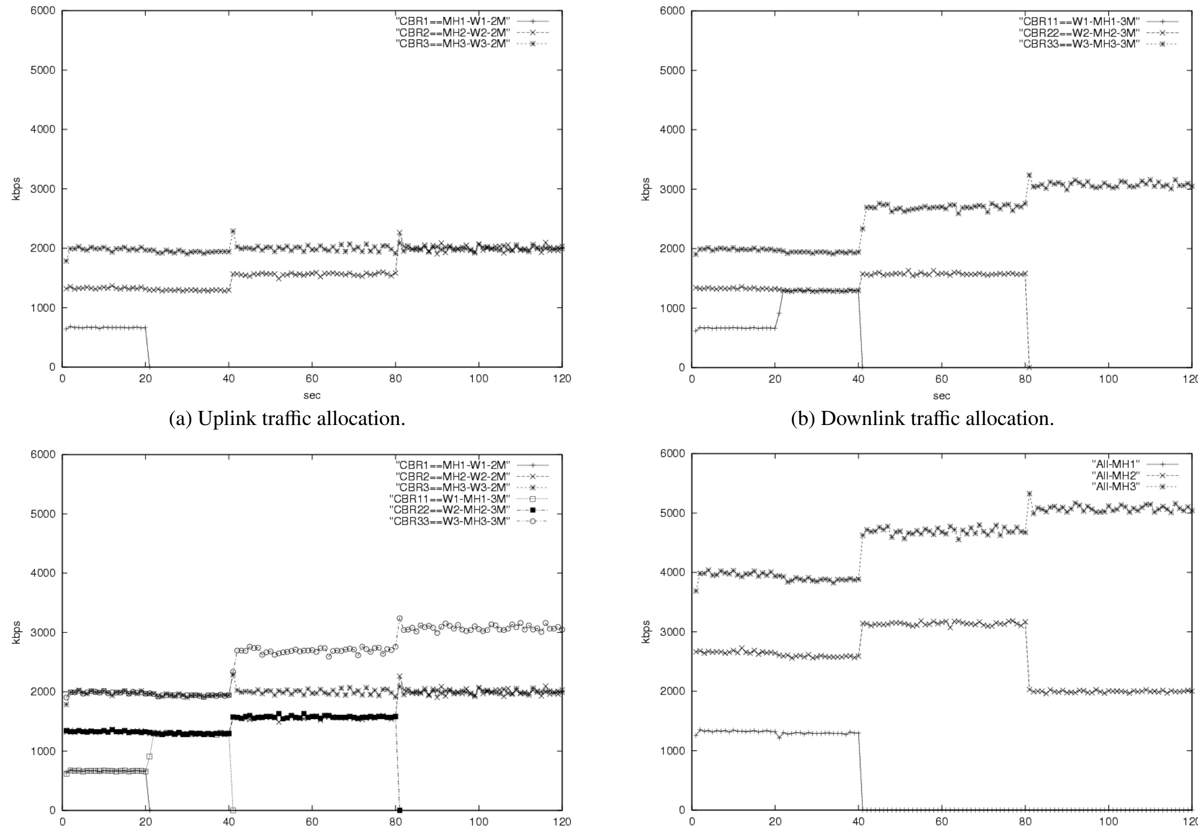
Figure 15 is divided into 4 parts. Figures 15(a) and (b) show the uplink and downlink traffic respectively. For readability, Fig. 15(c) combines the results of Figs. 15(a) and (b). Figure 15(d) shows the bandwidth borrowing among the MHs.

During the first 20 seconds, Figs. 15(a) and (b) show that bandwidth shared among uplink and downlink traffic flows are proportional to our quantum assignment (see Table 3). As *CBR1* stops at the 20 second, the assigned bandwidth is relocated to its downlink traffic, *CBR11*. At the 40 second, the *CBR11* also stops and then the newly available bandwidth is relocated to MH_2 and MH_3 proportional to the quantum values assigned to them. When *CBR22* stops at the 80 second, the wireless channel becomes not congested. All remaining flows (*CBR2*, *CBR3*, and *CBR33*) get enough bandwidth. In Fig. 15(d), each MH can obtain the bandwidth proportional to the assigned quantum values. Compared to the results in Fig. 15, co-DRR not only achieves the fairness among hosts, but also the fairness between uplink and downlink.

4.4 Scalability of Co-DRR

This section tests the scalability of co-DRR. In a wireless network, a host shares the access channel with other hosts. When the number of hosts increases, the effectiveness of co-DRR may degrade. So we aim to examine whether co-DRR can perform well under typical number of MHs. The fairness is defined as the coefficient of variation (CoV) of the bandwidth obtained among the MHs[†]. CoV is used because standard deviation is insufficient. The importance of having a 1 Mbps variance among the 10 MHs is different from that of having a 1 Mbps variance among the 100 MHs. For the 100 MHs, each MH can get at most

[†]The CoV is defined as the standard deviation divided by the mean.



(a) Uplink traffic allocation.

(b) Downlink traffic allocation.

(a) (c) Uplink and downlink traffic (merge (a) and (b)).

(d) Total traffic (uplink and downlink) of each host.

Fig. 15 Results of co-DRR bandwidth management.

110 kbps (11 Mbps/100MHs) so a variance of 1 Mbps is significant. By dividing the standard deviation by the mean, the value is normalized.

In this test, each host simultaneously has uplink and downlink traffic. The quantum values assigned to uplink and downlink traffic are all 1000 bytes. Each flow transmits at a constant bit rate of 1 Mbps. The number of hosts in a test round increases 10 hosts iteratively, with a beginning of 10 hosts. Each round lasts for 120 seconds. The mean goodput of a host is its total data successfully transmitted/received during the test round divided by 120 seconds.

Figure 16 describes the CoV and the normalized total goodput of wireless access link in co-DRR and pure IEEE 802.11b without scheduling. The CoV of co-DRR only arises slightly when the number of hosts increases. However, all CoVs are below 0.01, even under 100 MHs. In addition, when the number of MHs increases to compete for the limited bandwidth, the serious congestion significantly degrades the total goodput in the pure condition. In Co-DRR, though PCF consumes bandwidth, the benefit deserves. The total goodput of co-DRR remains almost the same high until increasing the MH to 100. The congestion in the CP becomes too high that the total goodput and the fairness are degraded.

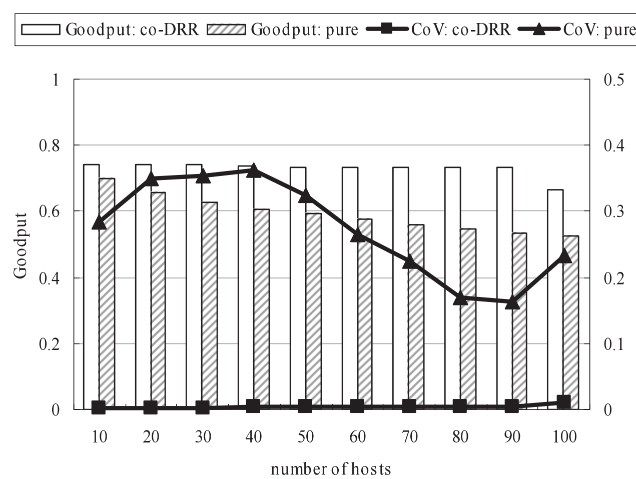


Fig. 16 Fairness among MHs (the CoV) and the total goodput.

4.5 Impact of Burst Errors

The wireless channel is often affected by other signals so it is more realistic to further study the issues under environments with burst errors. The selected channel model in the simulation is the well-known Gilbert channel model [17]. The two-state discrete-time Markov chain shown in Fig. 17

comprehensively describes the model.

Each state in the Gilbert channel model has an associated error probability, $P_e(Good)$ and $P_e(Bad)$, respectively. Whenever the channel is in the Good state, transmissions are always successful ($P_e(Good)$=0), and vice versa ($P_e(Bad)$=1). The channel alternates between the Good state and the Bad state with transition probabilities p and q respectively, and its probability transition matrix is

$$T = \begin{bmatrix} 1-p & p \\ q & 1-q \end{bmatrix} \tag{1}$$

The steady-state probabilities are denoted by

$$\prod = \begin{bmatrix} \pi_{Good} & \pi_{Bad} \end{bmatrix} \text{ with } \begin{cases} \pi_{Good} = \dfrac{q}{p+q} \\ \pi_{Bad} = \dfrac{p}{p+q} \end{cases} \tag{2}$$

The steady state channel BER is then obtained as

$$\begin{aligned} BER &= \pi_{Good} \times P_e(Good) + \pi_{Bad} \times P_e(Bad) \\ &= \frac{q \times P_e(Good) + p \times P_e(Bad)}{p+q} \end{aligned} \tag{3}$$

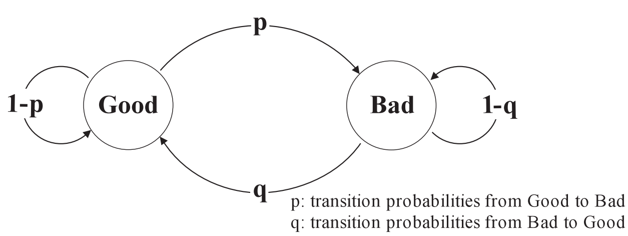


Fig. 17 Gilbert channel model.

Table 4 The parameters of error model.

BER	p	Q
1.0E-0.6	0.000001	0.999999
1.0E-0.5	0.00001	0.99999
1.0E-0.4	0.0001	0.9999

According to the above principal, $P_e(Good)$ and $P_e(Bad)$ are set to 0 and 1 respectively, and we determine the values of parameter p and q to obtain the mean BER (Table 4). The traffic flows and the quantum assignments in this simulation approximate those in Table 2, except that all flows start at the 0 second and stop at the 120 second.

Figure 18 compares pure IEEE 802.11b and the co-DRR. Obviously, the difference is that the latter not only can fairly share the bandwidth between uplink and downlink of a host, but also among all MHs under serious error rates. The results of the pure IEEE 802.11b reveal that the total uplink traffic of each BER setting stably grasps about 3/4 of the total bandwidth. For example, consider the zero BER setting. The total uplink goodput is 2000 + 2000 + 2000=6000 kbps while the total downlink goodput is 150+1380+750=2280. The limited downlink bandwidth is unfairly shared by the three MHs in each BER setting. With co-DRR, the proportion of the uplink/downlink traffic among the three hosts is well controlled.

When it comes to the effective total channel utilization, we sum up the goodputs in Fig. 18 and normalize them as the fraction of the link capacity. In Fig. 19, increasing the BER diminishes the total goodput of the wireless access link both in pure 802.11b and in co-DRR. Under different BERs, the

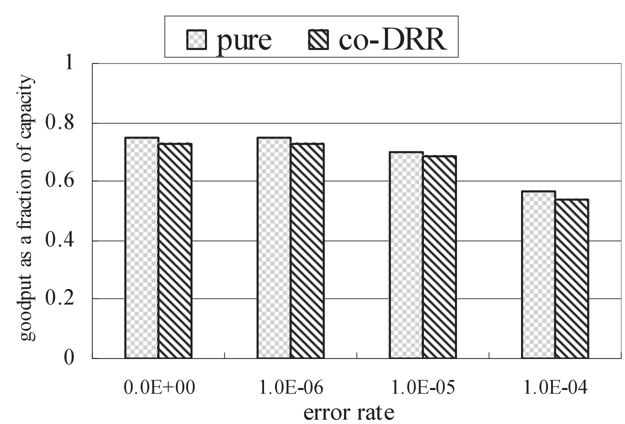


Fig. 19 Total throughput of the access link.

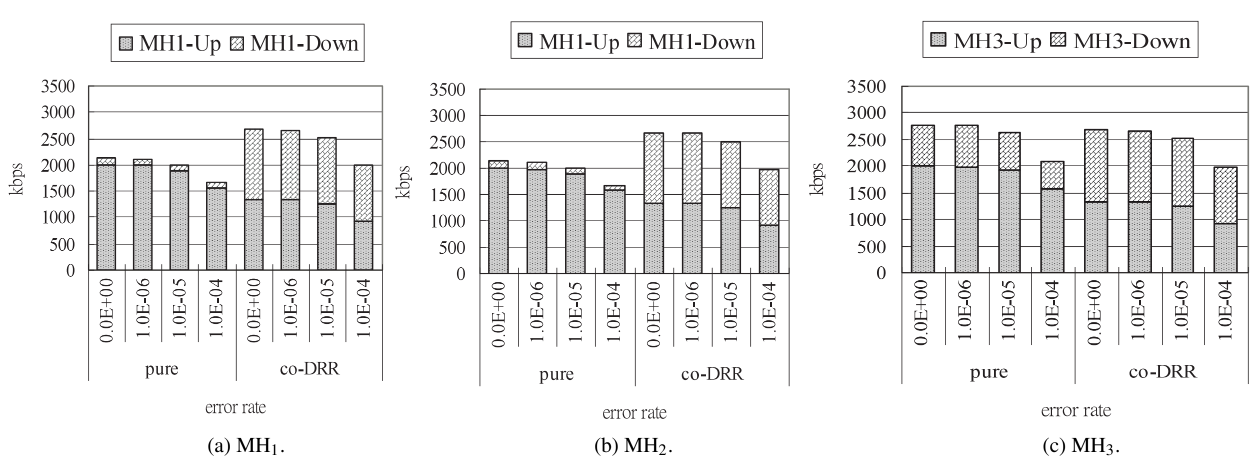


Fig. 18 Impact of burst errors.

PCF protocol overhead only consumes little goodput compared with the pure conditions.

5. Conclusion and Future Work

In this work, the contention between uplink and downlink in WLAN are resolved by our proposed co-DRR. Co-DRR integrates uplink and downlink scheduling by skillfully making DRR and DDRR algorithms cooperate to manage the wireless access link. Co-DRR accomplishes (I) long-term fairness among many hosts, (II) fairness between uplink and downlink, (III) automatic bandwidth borrowing between uplink and downlink, and (IV) simple O(1) complexity. Computer simulations using ns-2 demonstrates that the fairness among hosts, even between uplink and downlink, can be effectively controlled even under error-prone wireless link.

Since co-DRR inherits DRR, co-DRR is O(1) in processing each packet. However, co-DRR also inherits DRR's poor latency. So co-DRR can only achieve long-term fairness. The current co-DRR only focuses on providing bandwidth fairness among MHs and cannot support bandwidth fairness among flows. Therefore, future works include short-term fairness and flow-based control. Although the co-DRR performs well in WLAN, some complicated issues are not addressed in this study:

(I) Link state detection: being aware of the wireless link state can better utilize the wireless channel. Designing and adapting a better link state detection mechanism for the scheduler deserves much concern. For example, the Head-of-Line (HOL) Blocking problem experienced by a single downlink FIFO queue can be alleviated by DRR with per-MH downlink queues. In the face of burst errors, the DRR can be designed to skip serving the current MH to prevent HOL blocking. Events such as missing of link-level ACKs can be used as indications of burst errors. The lost downlink bandwidth due to the burst errors can be compensated in the following rounds by using the deficit counter. In this way, co-DRR can solve the HOL blocking problem and achieve long-term fairness.

(II) Multi-rate tradeoff: fixed bandwidth allocation for each MH is not reasonable because MHs located at different distances have different capabilities to transmit the frames. Bandwidth allocation should be dynamically regulated based on the detected sending rate of different MHs. Tradeoffs between channel utilization and fairness deserves extensive studying.

References

[1] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," ACM SIGCOMM'95, vol.25, no.4, pp.231–242, Oct. 1995.
[2] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," IEEE/ACM Trans. Netw., TON'95, vol.3, no.4, pp.365–386, Aug. 1995.
[3] P. Goyal, H.M. Vin, and H. Cheng, "Start-time fair queueing," IEEE/ACM Trans. Netw., TON'97, vol.5, no.5, pp.690–704, Oct. 1997.
[4] I. Aad and C. Castelluccia, "Introducing service differentiation into IEEE 802.11," IEEE ISCC'00, pp.438–443, July 2000.
[5] P. Bhagwat, P. Bhattacharya, A. Krishma, and S. Tripathi, "Enhancing throughput over wireless LANs using channel state dependent packet scheduling," IEEE INFOCOM'97, vol.3, pp.1133–1140, April 1996.
[6] M. Srivastava, C. Fragouli, and V. Sivaraman, "Controlled multimedia wireless link sharing via enhanced class-based queuing with channel-state-dependent packet scheduling," IEEE INFOCOM'98, vol.2, pp.572–580, March 1998.
[7] T. Nandagopal, S. Lu, and V. Bharghavan, "A unified architecture for the design and evaluation of wireless fair queueing algorithms," ACM MOBICOM'99, pp.132–142, Aug. 1999.
[8] D. Eckhardt and P. Steenkiste, "Effort limited fair (ELF) scheduling for wireless networks," IEEE INFOCOM'00, vol.3, pp.1097–1106, March 2000.
[9] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," ACM SIGCOMM'97, pp.63–74, Oct. 1997.
[10] T.S.E. Ng, I. Stoica, and H. Zhang, "Packet fair queueing algorithms for wireless networks with location-dependent errors," IEEE INFOCOM'98, vol.3, pp.1103–1111, March 1998.
[11] ISO/IEC and IEEE Draft International Standards, "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," ISO/IEC 8802-11, IEEE P802.11/D10, Jan. 1999.
[12] R.S. Ranasinghe, L.L.H. Andrew, and D. Everitt, "Impact of polling strategy on capacity of 802.11 based wireless multimedia LANs," IEEE ICON'99, pp.96–103, Sept. 1999.
[13] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," ACM SIGCOMM'95, vol.25, no.4, pp.231–242, Oct. 1995.
[14] ns2: network simulator version 2.1b8a, June 25 2001, http://www.isi.edu/nsnam/ns/
[15] D. Newman, "Benchmarking terminology for firewall performance," RFC2647, Aug. 1999.
[16] B. Bing, "Measured performance of the IEEE 802.11 wireless LAN," IEEE LCN'99, pp.34–42, Oct. 1999.
[17] E.N. Gilbert, "Capacity of a burst-noise channel," Bell Syst. Tech. J., vol.39, pp.1253–1265, Sept. 1960.

Huan-Yun Wei received the B.S. and Ph.D. degrees in computer and information science from National Chiao Tung University, Hsinchu, Taiwan, in 1998 and 2003, respectively. His research interests include TCP rate shaping for enterprise edge devices, queuing analysis of TCP traffic shaping, high-speed packet classification, and performance analysis of network security gateways. He is especially interested in the design and implementation of FreeBSD/NetBSD/Linux kernels.

Ching-Chuang Chiang received the M.S. in Computer & Information Science from National Chiao Tung University, Hsinchu, Taiwan, in 2002.

Ying-Dar Lin received the Bachelor's degree in Computer Science and Information Engineering from National Taiwan University in 1988, and the M.S. and Ph.D. degrees in Computer Science from the University of California, Los Angeles (UCLA) in 1990 and 1993, respectively. He joined the faculty of the Department of Computer and Information Science at National Chiao Tung University (NCTU) in August 1993 and is Professor since 1999. From 2005, he is the director of the newly established graduate Institute of Network Engineering. He is also the founder and director of Network Benchmarking Lab (NBL), co-hosted by Industrial Technology Research Institute (ITRI) and NCTU since 2002, which reviews the functionality, performance, conformance, and interoperability of networking products ranging from switch, router, WLAN, to network and content security, and VoIP. His research interests include design, analysis, implementation and benchmarking of network protocols and algorithms, wire-speed switching and routing, quality of services, network security, content networking, and embedded hardware software co-design. He can be reached at ydlin@cs.nctu.edu.tw.