This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

Dual Queue Management for Improving TCP Performance in Multi-rate Infrastructure WLANs

Qiuyan Xia, Xing Jin and Mounir Hamdi
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Email: {xiaqy, csvenus, hamdi}@cse.ust.hk

***Abstract*—In IEEE 802.11 wireless local area networks (WLANs), TCP suffers unfairness between uplink and downlink flows due to its asymmetric reactions towards data and ACK losses at the AP (Access Point), and the AP's inability to distinguish itself from other contending stations accessing the medium. In this paper, we propose a novel dual queue management (DQM) scheme with ECN (Explicit Congestion Notification) at the AP's downlink buffer to improve TCP performance in infrastructure WLANs. Our approach maintains two queues for TCP ACK and data respectively, with their total length controlled by a PI (Proportional Integral) controller to prevent congestion. ACK/data packets are marked/dequeued dependent on uplink/downlink time usage of the wireless channel. We also propose an opportunistic scheduling mechanism of the two queues exploiting "multi-rate capability" and "multi-user diversity" for more efficient link utilization. We evaluate the proposed approach in ns-2 and our simulation results demonstrate that this design with few states can significantly improve TCP congestion control, fairness performance and link utilization in WLANs.**

I. Introduction

In an infrastructure WLAN based on 802.11 [1], an AP (Access Point) is responsible for all traffic going through or inside the WLAN. According to the direction of the traffic flow, we classify them into downlink and uplink flows, illustrated in Fig. 1. Since the wireless link from an AP to all receiving stations has limited, time-varying capacity, it easily becomes congested, causing the AP's downlink buffer overflows, which degrades TCP performance. Hence, active queue management (AQM) is required to aid TCP congestion control. However, AQM schemes developed for wired networks cannot be directly applied to wireless network since they may aggravate packet loss which is very costly in wireless case. The alternate is to use AQM with ECN (Explicit Congestion Notification) when the link is detected congested, which turns out to be effective [2]. In addition, since wireless link capacity is time-varying due to fading effects and contentions, AQM schemes controlling the queue length to a target value may lead to large queueing delay when the link capacity is low.

Previous research work also showed that TCP is unfair towards uplink and downlink flows [3]: the sending stations (UP_STAs) obtain substantially larger bandwidth than the receiving stations (DN_STAs), which tend to starve and sometimes could not start altogether. Both MAC and TCP factors account for the critical uplink/downlink unfairness [4] [5]. On the other hand, when multiple users exist in a WLAN, there is a large chance that different bit-rates (controlled by rate adaptation algorithms such as [6]) are employed by different users transferring uplink/downlink flows, referred to as "multi-rate capability" and "multi-user diversity".

Given above analysis, the challenges involved in the design of an AQM scheme for congestion avoidance, uplink/downlink fairness and high link utilization are: (1) control queue occupancy level with ECN which can adapt to link capacity variations; (2) control source rates of uplink/downlink flows according to "temporal fairness"; and (3) as long as "temporal fairness" is satisfied, transmissions at high bit-rates are preferred to achieve high link utilization.

In this paper, we propose the "Dual Queue Management" (DQM) approach, which separates two queues, the TCP ACK queue and TCP data queue, in order to facilitate the mark/dequeue decision regarding uplink/downlink flows. We base our dual queue management on the PI (Proportional Integral) control algorithm [7] that controls the queue length toward adaptive target values with bounded queueing delay. When congestion is anticipated, ECN (or ECN echo for TCP ACK packets) mark is enabled to inform the source which reduces its sending rate. A selection rule is designed to pick the packet for ECN mark, instead of simply marking the incoming packet, based on the channel access time of uplink/downlink flows. When dequeuing a packet, opportunistic scheduling is proposed which decides to serve the head-of-line packet with high bit-rate if both queues (representing uplink and downlink flows) have not reached their fair share. We show later that the methodologies adopted in the proposed DQM can improve TCP performance significantly.

The rest of the paper is organized as follows. In Section II, we briefly review the related work. The details of our proposed "Dual Queue Management" are presented in Section III. Section IV shows the simulation set up and results of our approach. Finally, this paper concludes with Section V.

II. Related Work

Considerable research has been done on AQM. RED [8] is the first one introduced which marks/drops packets in proportion to the average queue length to keep it low while allowing occasional bursts. Subsequently, several variants of RED are proposed. CHOKe [9] discriminates against non-responsive

[1]This work was supported by a grant from Research Grants Council (RGC) under contract CERG (610307).

flows by adding random samplings into RED. REM [10] defines a price function as a congestion measure based on the difference between input rate and link capacity, and calculates the marking probability exponentially to the price. AVQ [11] maintains a virtual queue (VQ). At each packet arrival, it enqueues a fictitious packet and updates the VQ's capacity. AVQ marks/drops a real packet only if the VQ overflows. PI AQM [12] calculates the drop/mark probability as a function of the difference between current and reference queue length, the difference between old queue length and reference queue length, as well as previous drop/mark probability. It has been shown that PI (Proportional Integral) controller is superior to RED in robustly regulating the steady-value of the queue level. All these AQM schemes are originally designed for wired networks. They can not be directly applied to wireless networks unless tuned to wireless characteristics. One example is Proxy-RED [2], which utilizes RED together with ECN to perform AQM functionality at the gateway on behalf of the AP. VQ-RED [13] maintains a set of virtual queues (VQ) for each flow where VQ is a data structure kept track of only by its length. Each VQ management algorithm works similarly to RED. We note that only VQ-RED is developed for infrastructure WLAN and can achieve good fairness performance. However, VQ-RED requires to keep per-flow state and it neither consider bit-rate adaptation of each user nor bit-rate differences among multiple users.

III. Dual Queue Management

A. System Model

The system model is shown in Fig. 1. We assume that bit errors of wireless link can be resolved by retransmissions at the MAC layer. In our model, a number of wireless stations are associated with an AP and establish TCP connections with a wired host in a high-speed fixed network. We assume all nodes are ECN capable and both uplink and downlink TCP sources are "greedy". In particular, we consider N_{dn} wireless stations downloading data from the wired host (downlink flows) and N_{up} wireless stations uploading data to the wired host (uplink flows). The AP is connected to the wired station via a 100 Mbps Ethernet link with a $25\,ms$ propagation delay. The stations and AP are working under 802.11b capable of four bit-rates. AARF [6] rate adaptation is implemented at the MAC layer which adapts bit-rate according to channel conditions. The performance figures that we consider are listed in Table I, including aggregate TCP throughput R_{total}, mean per-flow throughput ratio γ and Jain's fairness index FI.

B. Dual Queue Management

In DQM, the interface queue of the AP is divided into two queues, the TCP ACK ($qAck$) and TCP data ($qData$) queues. The total length of two queues is controlled by the PI (Proportional Integral) algorithm which decides the packet mark probability. The packet for ECN mark is picked in a way such that the uplink or downlink flows which get more channel access are marked. An incoming packet is enqueued to (if not dropped when the buffer limit is exceeded) one of the

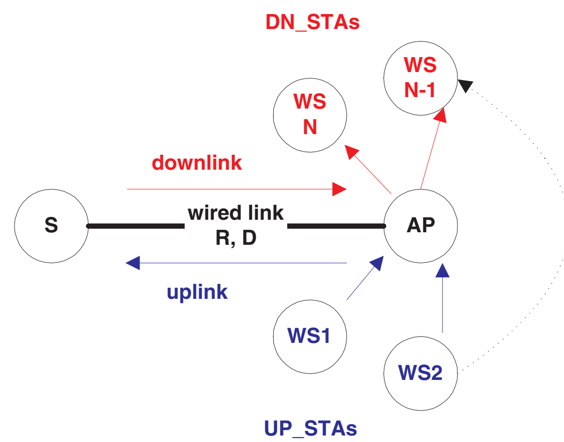


Fig. 1. Infrastructure WLAN model. The AP is connected to a wired station. There are N_{up} UP_STAs and N_{dn} DN_STAs associated with the AP.

TABLE I
Performance Figures

$R_{uptotal}$	total uplink throughput
$R_{dntotal}$	total downlink throughput
R_{total}	sum of uplink & downlink throughput
$R_{up} = R_{uptotal}/N_{up}$	mean throughput of uplink flows
$R_{dn} = R_{dntotal}/N_{dn}$	mean throughput of downlink flows
$\gamma = R_{up}/R_{dn}$	mean per-flow throughput ratio
$FI = \frac{(\Sigma_{i=1}^{N} R_i)^2}{N\Sigma_{i=1}^{N} R_i^2}$	Jain's fairness index

queues with respect to its type. Packets from different queues are dequeued with opportunistic scheduling.

To achieve TCP uplink/downlink temporal fairness, AP monitors the number of active uplink/downlink flows (N_{up} and N_{dn}) as well as the channel access time of these flows (T_{up} and T_{dn}). Note that we do not keep per-flow state but only the states of two classes (uplink/downlink). In addition, AP is also aware of the bit-rates towards different destinations. Based on these information, DQM selects a packet for mark and a packet to be dequeued, which can be from different queues. The implementation details are given in the next sections and the pseudo code is described in Algorithm 1.

1) Enqueue with Bounded Queueing Delay: A packet arrives at the AP's downlink buffer when the buffer limit ($q.lim$) is exceeded will be dropped. Otherwise, it is enqueued to one of the queues according to its type. To control the total length ($q.len = qAck.len + qData.len$) towards the desired value ($q.ref$), packets are marked with different probabilities regarding current queue occupancy level in regular intervals in the function $calculate\_p()$. After enqueueing a packet, DQM refers to the calculated mark probability ($q.prob$) and draw a random value to decide whether a packet needs to be marked. If this is true, the head packet of one of the queues will be marked. The criteria is to mark the packet from the queue that get more average per-flow access time of the channel (T_{up}/N_{up} vs. T_{dn}/N_{dn}) in current observation

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

window ($T_w = (N_{up} + N_{dl}) \cdot T_{fair}$). In this way, we are trying to maintain the "temporal fairness" between uplink and downlink flows, so that in each observation window, neither uplink flows nor downlink flows can consume more bandwidth than their fair share ($N_{up} \cdot T_{fair}$ and $N_{dn} \cdot T_{fair}$).

Wireless links have limited and time-varying capacity, which are error-prone and shared by multiple stations. Packets may corrupt due to poor channels and/or congestions, which are retransmitted at the MAC layer, leading to extended round trip time viewed by TCP. Hence, controlling average queueing delay (D_{ref}) is more effective since it reflects both the queue size ($q.ref$) and underlying average channel access time for transmitting a packet (T_{pkt}). The value of $q.ref$ can be obtained using the following equation:

$$q.ref = D_{ref}/\overline{T_{pkt}}, \tag{1}$$

where T_{pkt} is the MAC layer expected transmission time for a packet including backoff, interframe spacing, PHY overhead, MAC payload transmission time (possibly retransmissions) and transmission time for MAC layer acknowledgement. When estimated at the AP, AP can explicitly calculates T_{pkt} for outgoing packets since it knows the backoff time as well as the number of retransmissions. For incoming packets, AP has to approximate T_{pkt} assuming an average backoff time with minimum contention window and no retransmissions. Note that T_{pkt} for downlink data and corresponding ACK contribute to T_{dn}, while T_{pkt} for uplink data and corresponding ACK account for T_{up}. T_{pkt} for packets in the AP's downlink buffer is averaged ($\overline{T_{pkt}}$) using exponentially weighted moving average in order to reflect variations in downlink channels. Hence, $q.ref$ is updated at regular time intervals by (1).

2) Dequeue with Opportunistic Scheduling: Since we implement two queues at the AP, a scheduling algorithm is required when dequeuing a packet. Also, rate adaptation is adopted at the MAC layer of each station which reacts to channel variations by tuning its bit-rate. Hence we propose an opportunistic scheduling approach to exploit the chance that packets with high bit-rates appear at the head of the queue. As long as the fair share of both queues ($N_{up} \cdot T_{fair}$ and $N_{dn} \cdot T_{fair}$) is not exceeded, the head-of-line packet with larger bit-rate will be served. Otherwise, the queue that gets less channel access time than its fair share will be served. By this way, "temporal fairness" between uplink and downlink flows is not violated and the link utilization is improved.

IV. Performance Evaluation

Simulations have been carried out in ns-2 simulator (version 2.29.3) [14]. We consider the network topology shown in Fig. 1. All configurations and tunings are at the AP. The wireless stations use the default settings of the IEEE 802.11b PHY/MAC parameters. Table II summarizes the parameters used in simulations. We assume that TCP sources always have data to send. We compare the performance with BASE (the baseline scheme used in current WLANs) and dual queue management (DQM), subject to varying number of wireless stations and channel capacity.

Algorithm 1 Enhanced Dual Queue Management

```
enqueue(pkt)
if isTcpAck(pkt) then
    curq = qAck;
else
    curq = qData;
end if
q.len = qAck.len + qData.len;
if q.len >= q.lim then
    drop(pkt); return; {drop pkt due to overflow}
else if drop_early(pkt, q.len) && drop_tgt then
    drop(drop_tgt);
end if
curq.enqueue(pkt);

dequeue()
if (T_up < N_up · T_fair) && (T_dn < N_dn · T_fair) then
    if bitRate(Head(qData)) > bitRate(Head(qAck)) then
        dq = qData;
    else
        dq = qAck;
    end if
else if (T_up < N_up · T_fair) then
    dq = qAck;
else if (T_dn < N_dn · T_fair) then
    dq = qData;
else if (T_up/N_up < T_n/N_dn) then
    dq = qAck;
else
    dq = qData;
end if
dq.dequeue();

pickPacketForECN(pkt)
if !empty(qAck) && (T_up/N_up > T_dn/N_dn) then
    apkt = Head(qAck);    return apkt;
else if !empty(qData) && (T_up/N_up ≤ T_dn/N_dn) then
    apkt = Head(qData);    return apkt;
else
    return pkt;
end if

calculate_p()
q.prob = q.a * (q.len − q.ref) − q.b * (q.old − q.ref) + q.prob;
q.old = q.len;
<set an interrupt with frequency ω to invoke this procedure.>

drop_early(pkt, q.len)
if (u = uniform()) ≤ q.prob then
    mark(pickPacketForECN()); {ECN mark}
end if
return 0;
```

TABLE II
Simulation Parameters

$(q.a,\ q.b)$	$(1.822e^{-5},\ 1.816e^{-5})$
$\omega,\ D_{ref}$	160 Hz, 0.05 sec
T_{fair}	0.005 sec
TCP Packet size	1000 Bytes
AP's downlink buffer size	100 packets
maximum TCP congestion window	43 packets

A. Performance with Uniform Bit-rate

We first study the performance with fixed number of uplink/downlink flows and uniform bit-rate 11 Mbps (by

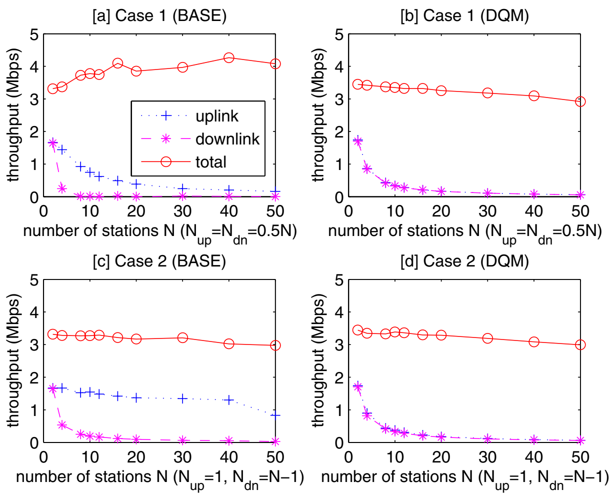


Fig. 2. Throughput performance for Case 1 and Case 2.

assuming stations carrying these flow are close enough to the AP). We discuss the following cases (as proposed in [5]): [1] $N_{up} = N_{dn} = N/2$; [2] $N_{up} = 1$, $N_{dn} = N - 1$.

Fig. 2 plots the average per-station uplink/downlink throughput and the system aggregate throughput with BASE and DQM in Case 1 and 2. The fairness performance is shown in Fig. 3. With BASE, the figures in Case 1 fluctuate a lot and are hardly predictable, while the ones in Case 2 are more stable (the throughput ratio increases linearly with N_{dn}). This is mainly caused by the increased contention level for medium access, coupled with bursty data losses due to buffer overflow in Case 1, when N_{up} increases with N. On the other hand, with DQM, the throughput performance is much more stable both in Case 1 and 2, even when N gets larger. The bandwidth is fairly distributed between uplink and downlink stations, where the throughput ratio is very close to the ideal value 1. Hence, DQM can substantially improve fairness by regulating the channel access time of uplink/downlink flows.

We also study the global fairness performance by Jain's fairness index (FI) in Fig. 3. As N increases, with BASE, FI drops more smoothly in Case 2 than in Case 1, but both achieve very low values around 0.203. On the other hand, although fairness among flows in the same direction is not enforced in DQM, with uniform traffic, it can achieve the global fairness very close to 1. We consider that this result is due to two reasons: first, each flow has the same simulation configuration; second, the distributed medium access assures the same access probabilities to wireless stations, including those in the same direction; finally, our scheme reduces bursty data losses (hence retransmissions and timeouts) by marking those flows that get more channel access.

B. Performance with Rate Adaptation

"Temporal fairness" is more desirable when nodes employ rate adaptation. For this end, we study DQM performance in a scenario where the uplink stations move far away from the AP at the time of t_0 (100 second here), hence their bit-rates change from 11 Mbps to 2 Mbps. Fig. 4(a) shows that before t_0, the average per-flow throughput of uplink and downlink flows is very similar. After that, the uplink flows converge to lower average throughput based on the "temporal fairness" rule. Hence, opportunistic scheduling alleviates the well known phenomenon of "performance anomaly" where all stations get the same throughput with the presence of low bit-rate senders. Fig. 5(a) shows that after t_0, the reference queue length decreases due to larger mean transmission time. Meanwhile, the queueing delay still fluctuates around the reference value. We mention that if both uplink and downlink flows have various bit-rates, the gain of multi-user diversity will be mitigated, since we only utilize the average bit-rate of them. We consider this as our future work to exploit multi-user diversity to a larger extent.

C. Performance with Varying Flow Number

In this section, we consider DQM performance in a scenario where downlink flows start at the very beginning, while uplink flows start in the middle (at $t_0 = 100$ second) and all have bit-rate 11 Mbps. In Fig. 4(b), before t_0, downlink flows fairly share the whole bandwidth; after t_0, uplink flows start traffic and the average throughput of uplink and downlink flows converges to the same value. In Fig. 5(b), when there is only downlink traffic, the link is under-utilized, with smaller queueing delay and queue size than the reference values. After uplink flows start, the link becomes saturated and both queueing delay and queue size are controlled around the reference values.

V. Conclusions

TCP performs poorly in infrastructure WLANs, suffering from congestion, unfairness and low link utilization. In this paper, we propose a dual queue management (DQM) scheme, with each queue holding TCP ACK/TCP data packets and representing uplink/downlink flows respectively. The simulation results demonstrate that with proper congestion control, enqueue and dequeue operations, the proposed dual queue management can effectively avoid congestion and provide quite good uplink/downlink fairness while using temporal fairness to achieve high link utilization. Since it is only implemented at the AP's downlink buffer and does not require per-flow/station queue/state or any changes to the MAC layer, it can be easily implemented in practice. In our future work, we will investigate more deeply into the gain of multi-user diversity, as well as provide per-flow/per-station temporal fairness assurance to nonidentical traffic.

References

[1] IEEE Std 802.11-1999, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Std., Aug. 1999.

[2] S. Yi, M. Keppes, S. Garg, X. Deng, G. Kesidis, and C. R. Das, "Proxy-RED: An AQM scheme for wireless local area networks," in *Proc. ICCCN'04*, Chicago, IL, USA, Oct. 2004.

[3] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP fairness over Wireless LAN," in *Proc. IEEE INFOCOM'03*, San Francisco, CA, USA, Mar. 2003.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

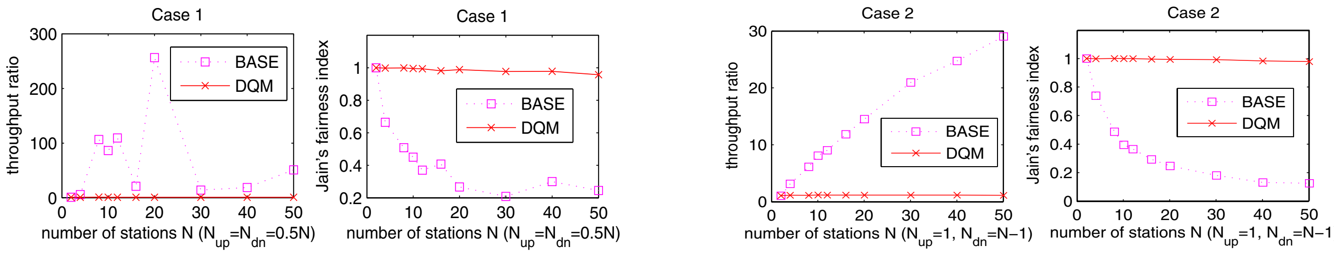
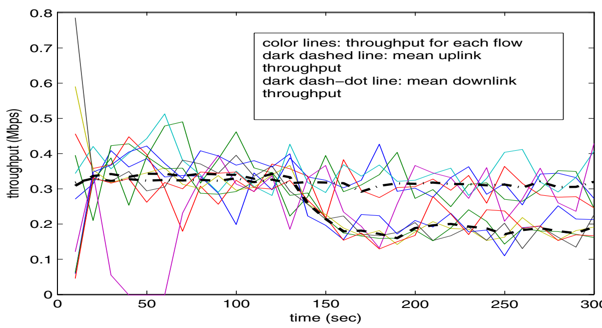
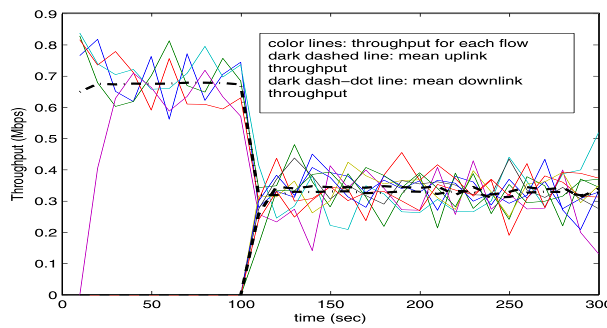


Fig. 3. Uplink/downlink throughput ratio and Jain's fairness index for Case 1 and Case 2.

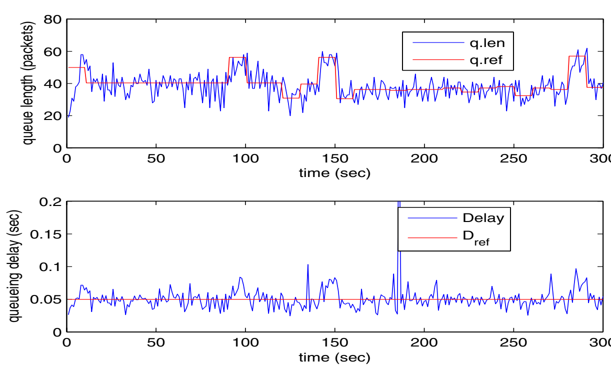


(a) $N_{up} = 5$, $N_{dn} = 5$ with rate adaptation.

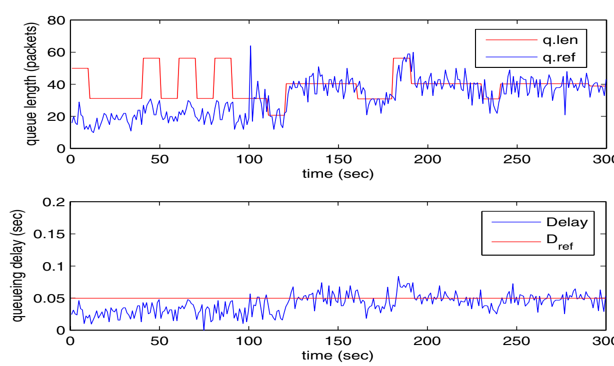


(b) $N_{up} = 5$, $N_{dn} = 5$ with different start time.

Fig. 4. Throughput evolution.



(a) $N_{up} = 5$, $N_{dn} = 5$ with rate adaptation.



(b) $N_{up} = 5$, $N_{dn} = 5$ with different start time.

Fig. 5. Evolution of queue size and queueing delay.

[4] N. Blefari-Melazzi, A. Detti, A. Ordine, and S. Salsano, "Controlling TCP Fairness in WLAN access networks using a Rate Limiter approach," in *Proc. IEEE ISWCS'05*, Siena, Italy, Sep. 2005, pp. 375–379.

[5] E. C. Park, D. Y. Kim, H. Kim, and C. H. Choi, "A Cross-Layer Approach for Per-Station Fairness in TCP over WLANs," *IEEE Transactions on Mobile Computing*, Sep. 2007.

[6] M. Lacage, M. Manshaei, and T. Turletti, "IEEE 802.11 Rate Adaptation: A Practical Approach," in *Proc. ACM MSWiM'04*, Venice, Italy, Oct. 2004, pp. 126–134.

[7] H. Xu, Q. Xue, and A. Ganz, "Adaptive Congestion Control in Infrastructure Wireless LANs with Bounded Medium Access Delay," in *Proc. ACM MobiWac'02*, Fort Worth, TX, USA, Oct. 2002, pp. 44–49.

[8] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[9] R. Pan, B. Prabhakar, and K. Psounis, "CHOKe - A stateless queue management scheme for approximating fair bandwidth allocation," in *Proc. IEEE INFOCOM'00*, Tel Aviv, Israel, Mar. 2000.

[10] S. Athuraliya, V. Li, S. Low, and Q. Yin, "REM: Active Queue Management," *IEEE Transactions on Network*, vol. 15, no. 3, pp. 48–53, May 2001.

[11] S. Kunniyur and R. Srikant, "Analysis and Design of an Adaptive Virtual Queue Algorithm for Active Queue Management," in *Proc. ACM SIGCOMM'01*, San Diego, CA, USA, Aug. 2001.

[12] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Transactions on Automatic Control*, vol. 47, no. 6, pp. 945–959, Jun. 2002.

[13] X. Lin, X. Chang, and J. K. Muppala, "VQ-RED: An Efficient Virtual Queue Management Approach to Improve Fairness in Infrastructure WLAN," in *Proc. IEEE LCN'05*, Sydney, Australia, Nov. 2005, pp. 632–638.

[14] NS2, "The Network Simulator2," http://www.isi.edu/nsnam/ns. [Online]. Available: http://www.isi.edu/nsnam/ns