Airtime Deficit Round Robin (ADRR) Packet Scheduling Algorithm*

Roberto Riggio, Daniele Miorandi, and Imrich Chlamtac
CREATE-NET
Via alla Cascata 56/C, 38100, Trento, Italy
name.surname@create-net.org

Abstract

In this paper we present Airtime Deficit Round Robin (ADRR) a novel scheduling algorithm for IEEE 802.11-based wireless mesh networks. ADRR enhances the Deficit Round Robin scheduling discipline by taking into account the channel quality experienced by the transmitting node. The devised algorithm addresses the IEEE 802.11 performance anomaly, preventing a node which experiences poor channel conditions from monopolizing the wireless medium lowering the performance of the whole system. The proposed approach combines link scheduling with measurable routing metrics typically available in WMNs. Results show the ability of the ADRR scheduler to achieve performance isolation among links characterized by heterogeneous channel conditions. The proposed solution has been implemented and tested over an IEEE 802.11-based wireless mesh network. Source code has been released under a BSD License making it fully available to the research community.

1 Introduction

In the IEEE 802.11 protocol the main mechanism used to access the wireless medium is called Distributed Coordination Function (DCF). The standard defines also a contention-free access method called Point Coordination Function (PCF). The PCF scheme, being optional, is not implemented in most commercial cards, and in what follows we will then focus on the DCF access scheme only. According to the DCF scheme, a station that wants to transmit a packet must first monitor the channel until an idle period equal to the Distributed Inter-Frame Space (DIFS) is detected. Then, the station generates a random back-off counter. The back-off counter is decremented as long as the channel is idle, frozen when a transmission is detected, and reactivated when the channel is sensed free for a DIFS interval. The station transmits when the back-off counter time reaches zero. Initially, the back-off timer is drawn uniformly in the back-off window $[0, aCW_{min}]$, where aCW_{min} is a parameters which depends on the PHY level being used. For each retransmission (due to collisions or errors), the back-off window is doubled until it reaches the maximum value of $2^m(aCW_{min} + 1) - 1$. Once it reaches its maximum value, the back-off window remains the same until it is reset as result of a successful transmission or because the Maximum Retransmission Limit (MRL) has been exceed. The whole process is sketched in Fig. 1. Thus, due to the back-off procedure, the concurrent transmission of alien stations affects the service time of IEEE 802.11: the higher the number of transmitting stations, the larger the overhead [13].

The half-duplex nature of IEEE 802.11 devices requires the sender to wait for an acknowledgment (ACK) signal after transmitting each frame. If the transmitting station does not receive the ACK it reschedules the transmission like if a collision occurred. If a node sustains repeated unsuccessful transmission it may degrade its transmission bit-rate in order to employ more robust but less efficient modulation schemes. As a result, since the CSMA/CA algorithm gives to each node the same channel access probability, nodes transmitting at low bit-rates will capture the wireless channel for long periods of time at the expenses of the nodes transmitting at higher bit-rates. Such behavior combined with the First-Come First-Served (FCFS) scheduling policy implemented in most commercial AP leads to the well known *"IEEE 802.11 performance anomaly"* extensively discussed in [9].

Wireless Mesh Networks (WMNs) are particularly susceptible to the *"IEEE 802.11 performance anomaly"*. In [3] the causes of packet loss in a large outdoor WMNs are analyzed. The authors conclude that, the loss rate distribution is substantially uniform across the whole range of loss rates and that a large number of links are characterized by intermediate loss rates. Such links can greatly reduce the

*This work was partially supported by MIUR within the framework of the Wireless Mesh Network for Next-Generation Internet (WING, grant number RBIN04M292) project.

978-1-4244-2575-4/08/$20.00

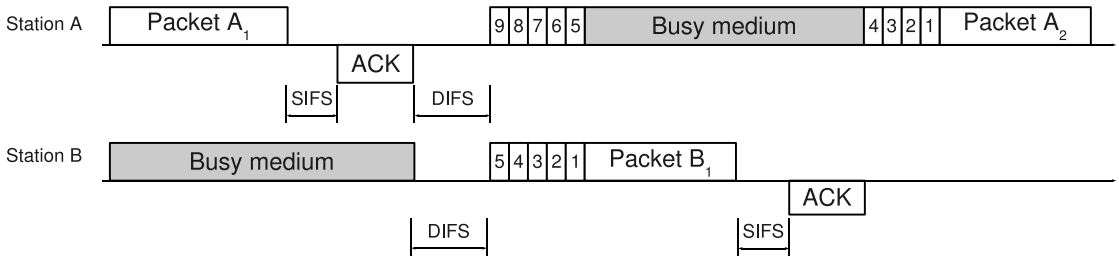

Figure 1. IEEE 802.11 access scheme based on the DCF.

performances of all the nodes sharing the wireless medium with special regard to the nodes experiencing good channel conditions.

In this paper we propose Airtime Deficit Round Robin (ADRR), a novel scheduling discipline aiming at providing intra-cell *airtime* fairness as opposed to the bandwidth fairness provided by traditional scheduling policy, i.e. Fair Queuing or, in case of equally sized data packets, Round-Robin. ADRR enhances the Deficit Round Robin (DRR) scheduling discipline by taking into account the channel quality experienced by the transmitting node. The devised algorithm addresses the *"IEEE 802.11 performance anomaly"*, preventing a node affected by high packet losses from monopolizing the wireless channels lowering the performance of the whole system. Our approach combines link scheduling with measurable routing metrics typically available in WMNs. As a proof-of-concept we have implemented and tested the ADRR scheduling policy on a IEEE 802.11-based WMN exploiting the Estimated Transmission Time (ETT) as routing metric. However, it is worth stressing that, our implementation can be easily extended to other routing metrics such as the Estimated Transmission Count [6] or the Weighted Cumulative ETT [7] metrics.

The rest of the paper is structured as follows. In Sec. 2 we discuss some related works. Section 3 briefly sketches the DRR scheduling algorithm. The proposed ADRR scheduler is analyzed in details in Sec. 4. Section 5 describes the experimental setup of the WMN testbed. The outcome of our measurements campaign are reported in Sec. 6. Finally, Sec. 7 concludes the paper pointing out future research directions.

2 Related work

The literature on fairness provisioning in both wired and wireless networks is extensive. In this section we will only survey solutions which are related to the proposed ADRR scheduler redirecting the readers to [16] for a more comprehensive analysis.

The most simple discipline is the First-Come First Served scheduling. In this case a single queue exists, thus the order of arrival of the packets determines the order in which they are forwarded to the output link. In order provide fairness among heterogeneous links, each outgoing link must have its own queue. In such a scenario, the Generalized Processor Sharing (GPS) scheduling discipline is known to provide fair allocation of the network resources among backlogged queues. However, due to the assumption of fluid traffic (i.e. infinitesimal packet sizes), it is not possible to implement the GPS algorithm, leaving it as a useful benchmark against which realizable service disciplines can be measured. There are several scheduling disciplines which tries to approximate GPS as for example round robin, WF^2Q+ [4], and DRR [14]. However, such algorithms aim at providing bandwidth fairness as opposed to the *airtime* fairness required to address the *"IEEE 802.11 performance anomaly"*.

In [8] the authors propose the Deficit Transmission Time (DTT) scheduling discipline. DTT aims at ensuring a fair usage of the wireless medium by the stations participating an infrastructure BSS. The proposed scheduler is implemented in a centralized way as part of the access point queuing discipline. The authors propose two ways of estimating the quality of the link between the access point and the stations. The first method exploits the SNR values reported from the Wireless NIC's device driver in order to compute the maximum expected throughput toward each station. Its main drawbacks are the channel symmetry assumption and the tight coupling with the Wireless NIC which requires a calibration procedure in order to build the SNR-to-throughput table used by the scheduler. The second method directly measure the overall amount of time needed to send a frame including retransmissions and rate reductions. In order to measure such a time, the Wireless NIC's device driver has been extended implementing a feedback mechanism which reports to the channel monitor any ACK frame reception or MRL exceed event. This information is then used to compute the optimal schedule list.

The ADRR scheduler here presented differs from the previous approach in that it leverages bidirectional link quality statistics already maintained by the routing layer in order to compute the optimal schedule list. As it will be clear in Sec 4, ADRR can cope with asymmetric links

and does not need a calibration phase. Moreover, being designed in such a way to exploit measurable link metrics, ADRR requires no changes to the Wireless NIC's device driver and can be readily implemented using off-the-shelf components. However, it is worth noting that, a low level control of the Wireless NIC is required by the routing layer in order to properly compute the routing metric. In particular the device drivers must expose raw 802.11 frames to user space applications allowing the routing layer to send broadcast frames at arbitrary rates and to get per-packet statistics. Such features, albeit not available in most bundled solution (e.g., the Intel Centrino platform) are widely supported by carrier class IEEE 802.11 devices such any product based on the Prism 2.5 or the Atheros chipsets.

3 Deficit Round Robin Scheduling

This section briefly introduces the DRR algorithm, for a detailed analysis the reader is referred to [14]. Analytical bounds to DRR latency and fairness are analyzed in [12]. DRR is a modified weighted round robin scheduling discipline. It can handle packets of variable size without knowing their mean size. According to the DRR algorithm, each flow contending for a link has a corresponding queue i fed with all the packets belonging to this flow. Each queue i has a counter associated called *Deficit Counter* (DC_i), which indicate the amount of resources the flow can use in a round. Flows are visited in a round robin fashion. Each flow is visited only once during each round. Upon each visit the flow's deficit counter DC_i is increased by a fixed quantity Q called *quantum*. If Q is not smaller than the maximum packet length allowed in the network, then the algorithm complexity is $O(1)$ [14]. A packet is sent only if its length is smaller than the deficit counter's current value, otherwise the flow is skipped. After a packet is sent the deficit counter is decreased by the size of the transmitted packet. Only backlogged flows are served. When a flow is not backlogged its deficit counter is set to zero.

4 Providing intra-cell airtime fairness

The proposed ADRR scheduler has been implemented on top of Roofnet [2] and exploits the Estimated Transmission Time (ETT) metric in order to evaluate the channel time spent serving each nonempty queue. Roofnet is an experimental IEEE 802.11b-based WMN deployed at Cambridge, Massachusetts (USA) by the Massachusetts Institute of Technology (MIT). Roofnet routes packets using a modified version of DSR [10] called SrcRR [5] exploiting ETT as routing metric. The ETT metric aims at estimating the amount of time required to transmit a packet over a wireless link (including re-transmission). The ETT metric is computed as follows:

$$M_{\text{ETT}} = \frac{1}{P_{ACK} R} \tag{1}$$

Where R is an estimate of the highest effective throughput achievable in the forward direction, and P_{ACK} is the delivery probability of the the ACK signal in the reverse direction (d_{rev}). Being r_x the estimated throughput of broadcast packets in the forward direction at the transmission rate of x Mb/s, the parameter R can be computed as follows:

$$R = max(r_1, r_2, r_{5.5}, r_{11}) \tag{2}$$

$$r_x = d_{fwd} x \tag{3}$$

Where d_{fwd} is the link delivery probability in the forward direction.

In order to compute the forward (d_{fwd}) and reverse (d_{rev}) link delivery ratios each node periodically broadcast a sequence of five probes: one short probe aimed at modeling the ACK transmission and one long probe for each available transmission rate (1, 2, 5.5, 11 Mb/s)[1]. Each node keeps track of the number of probes received during an observation window W. At any time, d_{rev} is then given by:

$$d_{rev}(t) = \frac{count(t - W, t)}{w/\tau} \tag{4}$$

Note that $count(t - W, t)$ is the number of probes received during the observation window W and w/τ is the number of probes that should have been received.

Finally each probe sent by a node contains the number of probes packets received by the same node from all its neighbors during the last observation window. Such a design choice allows the receiver to compute the forward delivery ratio d_{fwd} toward the node from which the probe was originated. Using two probes to estimate data and ACK delivery ratios separately allows the routing layer to properly model asymmetric links and to cope with the hidden node phenomena. In fact, probes lost at the receiver side due to interference, are taken into account during the computation d_{fwd} at the transmitting side by exploiting the information piggy-backed into each probe.

Let L_{Probe} be the size of the probe used to compute d_{fwd}, the expected transmission airtime TX_{AIRTIME} for a packet S bytes long is then given by:

$$TX_{\text{AIRTIME}} = M_{\text{ETT}} \frac{S}{L_{Probe}}$$

The building blocks of the ADRR scheduler and their relationships are sketched in Fig. 2. The pseudo code of the enqueue and dequeue processes is given respectively in Alg. 1 and Alg. 2. Variables and data structure exploited by both algorithms are summarized in Table 1.

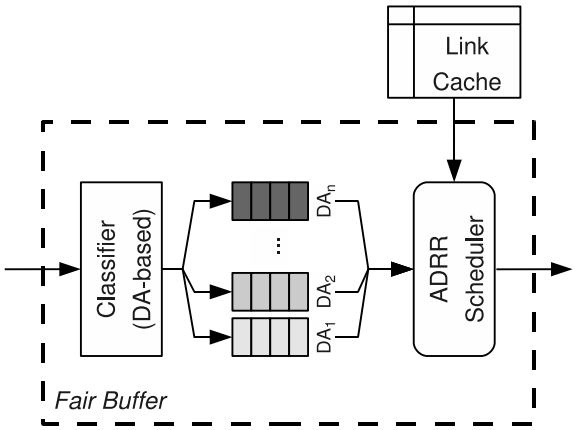

Figure 2. Block diagram for Aggregation Buffer with Airtime DRR Scheduler

Table 1. Variables and data structure used by the ADRR algorithm

Variable	Default	Description
$ActiveQueue$	$\{\emptyset\}$	List of currently backlogged queues
Q	$12000\mu s$	Quantum value
$DC(i)$	0	Queue i deficit counter

The scheduler maintains a linked list of currently backlogged queues ($ActiveQueue$). Incoming data frames are first classified according to their next hop (Alg. 1, row 2) and then fed to the corresponding queue (Alg. 1, row 6). If such an queue does not yet exist, it is created dynamically by the scheduler (Alg. 1, row 3 through 5). Probe frames have higher priority than data frames and are granted preemptive access to the link bypassing the ADRR scheduler.

At each round, the deficit counter of the currently visited queue $DC(i)$ is increased by a fixed quantity Q (Alg. 2, row 3). The ADRR scheduler only serves packets whose expected transmission time is smaller than the deficit counter (Alg. 2, row 6 through 8). After a packet is sent the deficit counter is decreased by the expected transmission time of the transmitted packet (Alg. 2, row 9). A frame whose transmission time exceed the deficit counter is held back until the next visit of the scheduler (Alg. 2, row 11). Empty queues are removed from $ActiveQueue$ and their deficit counter is set to zero (Alg. 2, row 14 through 16).

[1]Broadcast frame are not acknowledged nor retransmitted by IEEE 802.11 devices.

Algorithm 1 Enqueuing process.

1: **for** each incoming packet p **do**
2: i = p.nextHop()
3: **if** i not in ActiveQueue **then**
4: ActiveQueue.pushBack(i)
5: **end if**
6: ActiveQueue(i).enqueue(p)
7: **end for**

Algorithm 2 Dequeuing process.

1: **if** $ActiveQueue$ is not empty **then**
2: i = ActiveQueue.next()
3: DC(i) = DC(i) + Q
4: **while** true **do**
5: $airtime$ = ActiveQueue(i).computeTxAirtime()
6: **if** $airtime <$ DC(i) **then**
7: p = ActiveQueue(i).dequeue()
8: p.send()
9: DC(i) = DC(i) - $airtime$
10: **else**
11: break
12: **end if**
13: **end while**
14: **if** i is empty **then**
15: ActiveQueue.remove(i)
16: **end if**
17: **end if**

5 Evaluation Methodology

Our WMN is based on the Roofnet platform developed by the MIT. Routing is implemented using the Click modular router [11], developed at MIT. A Click router is built by assembling several packet processing modules, called elements, forming a directed graph. Each element is in charge of a specific function such as packet classification, queuing, and interfacing with networking devices. Click comes with an extensive library of elements supporting various types of packet manipulations. Such a library enables easy router configuration by simply choosing the elements used and the connections among them. Finally, a router configuration can be easily extended by writing new elements. The Click modular router is available as both Linux Kernel Module and user-space driver, allowing straightforward porting of an user-space implementation to kernel-space. We extended the default Roofnet configuration by implementing the additional elements responsible for packet scheduling and classification. All the developed software has been released under the BSD License[2].

The measurements campaign has been carried out ex-

[2]http://www.wing-project.org/

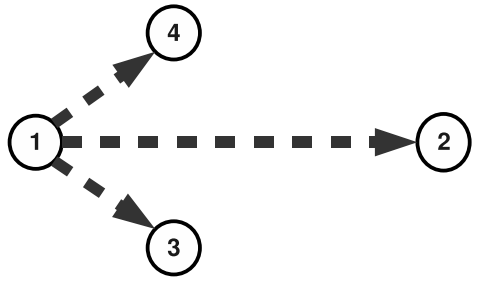

Figure 3. Reference network topology.

Table 2. Parameters of the CBR connection exploited during the measurements campaign

rotocol	Packet Interval	Payload	Bitrate
UDP	2 ms	1460 bytes	≈6 Mb/s

ploiting a 4-nodes wireless testbed implementing a flat WMN. Testbed's nodes are based on the PCEngines WRAP processor board. Each node is equipped with a 233MHz CPU, 128MB of RAM, and one IEEE 802.11a/b/g wireless interfaces with RTC/CTS disabled (the board supports up to two wireless interfaces). All measurement were run with the IEEE 802.11 interfaces operating in "b" mode.

Traffic is generated at node number 1, which acts as gateway, using the Jugi's Traffic Generator (JTG), a freely available synthetic traffic generator [1]. JTG can generate and inject different traffic patterns over TCP and/or UDP sockets. Traffic is then collected at the receiver side (nodes number 2, 3, and 4) where suitable tools are available for analysis. All measurements were performed over a 1 minute interval; results are averaged over 10 runs. The experimental setup is sketched in Fig. 3.

In order to validate the ADRR scheduling discipline, node number 2, 3, and 4 have been fed with a CBR connection generated at node number 1. We have modeled each CBR connection as a single UDP stream with constant interdeparture time and packet size. Table 2 summarizes the parameters of the UDP stream. Measurements have been repeated using two other scheduling policies: FCFS (which is the default packet scheduling policy implemented in most IEEE 802.11 devices) and DRR. For the DRR scheduling discipline the quanta has been set to 1500 bytes which is the MTU (Maximum Transmission Unit) supported by an Ethernet LAN.

6 Performance Measurements

Measurements have been carried out exploiting three deployments scenarios differentiate by the channel condition experienced by node number 2. Notice that in each deployment all nodes are in radio range. However, while node number 3 and 4 are kept close to the gateway, node number 2 is positioned in such a way to experience channel condition raging from *Good* to *Poor* with an intermediate *Medium* quality.

Table 3 through 5 summarize the outcomes of our measurements campaign. Results show that the proposed scheduler is capable of addressing the *"IEEE 802.11 performance anomaly"* maintaining a high throughput over reliable links (Node number 3 and 4) as opposed to both the FCFS and the DRR policies that fail to achieve performance isolation when node number 2 starts to experience poor channel conditions.

As you can see from table 3, when channel condition for node number 2 are still good the available resources are evenly shared among all the nodes. However, it is worth noting that the average throughput achieved by each node using the ADRR is slightly higher than the throughput achieved using the both the FCFS and the DRR scheduling disciplines.

We postulate than the ADRR scheduler is capable exploiting channel fluctuation by opportunistically allocating more airtime to links that experience better channel condition. We recall that the feedback mechanism embedded in the routing metric gives the transmitting station (Node 1 in our case) the capability to schedule for transmission links experiencing better channel conditions. Such considerations are supported by the theoretical finding in [15] channel fluctuations can instead be exploited by transmitting information opportunistically when and where the channel is strong.

As node number 2 moves away from the gateway (Table 4 and 5), the ADRR is capable of allocating more resource to the nodes experiencing better channel conditions, while the other scheduling policies degrade the aggregated throughput. In the extreme case where node number 2 experience poor channel condition ADRR outperform both FCFS and DRR by delivering a higher aggregated throughput (1.1 Mb/s w.r.t. the baseline scenario) and by allocating to node number 3 and 4 a percentage of the bandwidth which is only slightly lower the the optimal case.

7 Outlook and Future Work

In this paper we proposed an opportunistic scheduler capable of addressing the the *"IEEE 802.11 performance anomaly"*. The proposed architecture is capable of providing performance isolation in IEEE 802.11-based WMNs.

Table 3. Average throughput for different scheduling disciplines (Good channel conditions). Results are in Kb/s.

Scheduler	Node 2	Node 3	Node 4	Aggregated
FCFS	1869.6	1818.2	1870.7	5558.5
DRR	1857.1	1817.8	1889.2	5564.2
ADRR	2014.2	1995.4	2055.9	6065.5

Table 4. Average throughput for different scheduling disciplines (Medium channel conditions). Results are in Kb/s.

Scheduler	Node 2	Node 3	Node 4	Aggregated
FCFS	1626.2	1579.1	1616.5	4821.9
DRR	1751.0	1709.6	1785.3	5245.9
ADRR	1751.6	2024.8	2074.2	5850.6

Table 5. Average throughput for different scheduling disciplines (Poor channel conditions). Results are in Kb/s.

Scheduler	Node 2	Node 3	Node 4	Aggregated
FCFS	1364.8	1318.7	1298.4	3981.9
DRR	1399.1	1340.9	1323.3	4063.3
ADRR	1107.2	1993.6	2038.8	5139.6

The optimal scheduled list is computed exploiting measurable routing metrics typically available in WMNs. The software implementation of the proposed mechanism has been released under a BSD License, with the aim of providing the reference scientific community with a basis for developing further innovative solutions.

As future work we plan to extend the ADRR scheduler in order to exploit path diversity when computing the optimal scheduling list. In context, the residual path metric allows us to differentiate differentiate traffic routed over homogeneous paths from traffic that experiences good link condition only locally and that will be route over a lossy or congested link a few hops away. In such a case, the end-to-end performance of the flow will not benefits from the the extra airtime allocated by ADRR to the link.

Finally, further efforts will be devoted to the validation of the ADRR scheduler over a larger testbed (in terms of both number of nodes and network coverage). This will allows us to obtain further insight into the scalability of the scheduling discipline proposed in this paper.

References

[1] Jtg. https://hoslab.cs.helsinki.fi/savane/projects/jtg/.

[2] Mit roofnet. http://pdos.csail.mit.edu/roofnet/.

[3] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. *ACM SIGCOMM Computer Communication Review*, 34(4):121 – 132, Oct. 2004.

[4] J. Bennett and H. Zhang. Hierarchical packet fair queueing algorithms. *IEEE/ACM Transactions on Networking*, 5(5):675–689, Oct. 1997.

[5] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In *Proc. of ACM MOBICOM*, Cologne, Germany, 2005.

[6] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proc. of ACM MobiCom*, San Diego, California, USA, 2003.

[7] R. Draves, J. Padhye, and B. Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *Proc. of ACM MOBICOM*, Philadelphia, Pennsylvania, USA, 2004.

[8] R. G. Garroppo, S. Giordano, S. Lucetti, and L. Tavanti. Providing air-time usage fairness in IEEE 802.11 networks with the deficit transmission time (DTT) scheduler. *ACM Wireless Networks*, 13(4):481 – 495, Aug. 2007.

[9] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *Proc. of IEEE INFOCOM*, San Francisco, California, USA, 2003.

[10] D. Johnson, Y. Hu, and D. Maltz. The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4. IETF RFC 4728, Feb. 2007.

[11] E. Kohler, B. C. Robert Morris, J. Jannotti, and M. F. Kaashoek. The Click modular router. *ACM Transaction on Computer System*, 18(3):263 – 297, Aug. 2000.

[12] L. Lenzini, E. Mingozzi, , and G. Stea. Tradeoffs between low complexity, low latency, and fairness with deficit round-robin schedulers. *IEEE/ACM Transaction on Networking*, 12(4):681 – 693, Aug. 2004.

[13] F. D. Pellegrini, F. Maguolo, A. Zanella, and M. Zorzi. A cross-layer solution for VoIP over ieee802.11. In *Proc. of WPMC 2005*, Alborg, Denmark, September, 18-22 2005.

[14] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round robin. In *Proc. of ACM SIGCOMM*, Cambridge, Massachusetts, USA, 1995.

[15] D. Tse and S. Hanly. Multi-access fading channels: Part i: Polymatroid structure, optimal resource allocation and throughput capacities. *IEEE Transactions on Information Theory*, 44(7):2796–2815, Nov. 1998.

[16] L. Wischhof and J. Lockwood. Packet scheduling for link-sharing and quality of service support in wireless local area networks. Technical Report WUCS-01-35, Washington University in St. Louis, Nov. 2000.