On Efficient Airtime–based Fair Link Scheduling in IEEE 802.11-based Wireless Networks

Karina Gomez, Roberto Riggio, Tinku Rasheed, and Imrich Chlamtac
CREATE-NET
Via alla Cascata 56/C, 38123, Trento, Italy
`name.surname@create-net.org`

***Abstract*—In this paper, we present the Airtime Deficit Round Robin (ADRR), a novel scheduling algorithm for IEEE 802.11-based wireless networks. The ADRR mechanism 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. Simulation analyses have shown that the proposed scheme can achieve performance isolation among links characterized by heterogeneous channel conditions. A real prototype has been implemented and evaluated over a small scale testbed confirming the simulation results.**

***Index Terms*—IEEE 802.11, wireless mesh networks, opportunistic scheduling, performance anomaly, experimental measurements, performance evaluation.**

I. INTRODUCTION

Wireless Mesh Networks (WMNs) based on IEEE 802.11 standard are particularly susceptible to the *"IEEE 802.11 performance anomaly"* [1]. In the IEEE 802.11 protocol, the main mechanism used to access the wireless medium is the Distributed Coordination Function (DCF). According to the DCF scheme, a station that has 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.

The half-duplex nature of IEEE 802.11 devices requires that the sender waits for an acknowledgement (ACK) signal after transmitting each frame. If the transmitting station does not receive the ACK it reschedules the transmission. 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 provides 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 expense of the nodes transmitting at higher bit-rates. Such behaviour combined with the First-Come First-Served (FCFS) scheduling policy implemented in most commercial AP leads to significant performance degradation in WiFi networks [1].

The analysis of the causes of packet loss in a large outdoor WMN concludes that [2], 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 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 policies, 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 prevents 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 the routing metric. However, it is worth stressing that, our implementation can be easily extended to other routing metrics such as the Estimated Transmission Count [3] or the Weighted Cumulative ETT [4] metrics. This work extends our previous work on link scheduling in wireless and mesh networks [5] by simplifying the communication protocol and by validating the proposed architecture using both simulations and a real world prototype.

The rest of the paper is structured as follows. A brief overview of the state of the art is presented in Section II. Section III sketches the DRR algorithm and introduces the ADRR policy. In Section IV, we introduce the proposed network architecture. Section V presents the results of the simulation evaluation, while the results from our prototype–based study are reported in Section VI. Finally, Sec. VII draws some conclusions and presents the future works.

II. RELATED WORK

Fairness provisioning in both wired and wireless networks has been extensively addressed in several previous works. In this section, we survey solutions related to the proposed ADRR scheduler. The simplest 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+ [6], and DRR [7]. 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 in 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. Its main drawbacks are the channel symmetry assumption and the tight coupling with the Wireless NIC driver.

In [9], the authors have introduced a Virtual Flow Queuing (VFQ) packet scheduling in order to improve the performance of TCP connections over IEEE 802.11 WLANs. The scheduler has a inter-layer approach wherein the IP layer takes into account the information coming from both the transport and MAC layers in order to schedule packets over the wireless link. The scheduler also considers the transmission time needed to transfer both TCP and ACK segments at the MAC layer.

On the other hand, the ADRR scheduler presented in this paper 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. 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.

III. PROVIDING INTRA-CELL AIRTIME FAIRNESS

This section briefly introduces the DRR (Deficit Round Robin) algorithm [7] before describing the ADRR scheduler in detail. DRR is a modified weighted round robin scheduling discipline that 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 associated counter called *Deficit Counter* (DC_i), which indicates the amount of resources the flow can use.

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*. 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.

The proposed ADRR scheduler exploits the Estimated Transmission Time (ETT) metric [10] in order to estimate the channel time spent serving each non–empty queue. It is worth noticing that, albeit originally devised to work in conjunction with a link quality routing metric, such as the ETT metric, the ADRR scheduler can be easily adapted to traditional single-hop wireless networks. As a matter of fact, in the simulation model, whose outcome are described in Sec. V, the link quality metric has been derived from the statistics collected by the transmission rate adaptation module. It is worth remembering that the ETT metric aims at estimating the time required to successfully deliver a unicast frame over a wireless link and to receive the corresponding acknowledgement thus taking into account re-transmissions and the link quality in both directions. The metric is computed as follows:

$$M_{\text{ETT}} = \frac{1}{d_{rev}R} \tag{1}$$

Where R is an estimate of the highest effective throughput achievable in the forward direction, and d_{rev} is the delivery probability of the the ACK signal in the reverse direction. 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:

$$R = max(r_1, r_2, r_{5.5}, r_{11}), \quad r_x = d_{fwd}x \tag{2}$$

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 modelling 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:

$$d_{rev}(t) = \frac{count(t-W,t)}{w/\tau} \tag{3}$$

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. Each probe sent by a node contains the number of probes packets received by the same node from all its neighbours 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.

Finally, 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:

---

[1] Broadcast frames are not acknowledged by IEEE 802.11 devices.

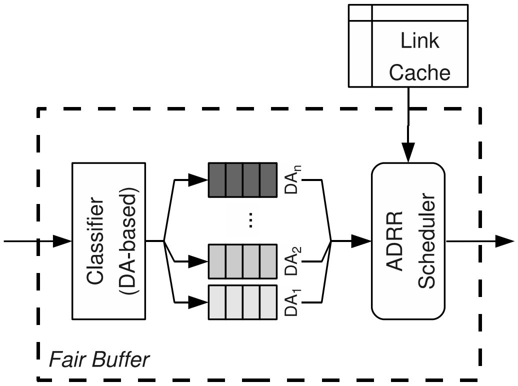


Fig. 1: Block diagram for the Airtime DRR Scheduler

TABLE I: Data structure used by the ADRR algorithm

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

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

IV. SYSTEM ARCHITECTURE

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

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 a queue does not 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 exceeds the deficit counter is held back until the next visit of the scheduler (Alg. 2, row 11). Empty queues are removed from the $ActiveQueue$ and their deficit counter is set to zero (Alg. 2, row 14 through 16).

V. SIMULATIONS

In this section, we present the analysis of the ADRR scheduler. In particular, we evaluate the bandwidth efficiency of our scheduling policy (*ADRR*) in comparison with a baseline scenario where no link scheduler is used (*FCFS*), and thus the packets are served in a FIFO fashion, and in scenario where the DRR scheduling policy is used (*DRR*).

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
```

A. Simulation Environment

The simulation environment was realized in the OMNET++ simulator. The MiXiM model was used in order to simulate the IEEE 802.11–based wireless network. Each wireless node is equipped with an IEEE 802.11 interface derived from the MiXiM framework and operating in the ISM 2.4 Ghz frequency band.

The ADRR scheduling policy has been integrated within the *mac* module. The deficit counter (DC) and the quantum (Q) parameters are initialized at the beginning of each simulation using the setting reported in Table 1; conversely the TX_{AIRTIME} and the M_{ETT} parameters are dynamically calculated by the *mac* module according to the actual channel conditions.

B. Simulation Scenarios

Figure 2 sketches the star-shaped network topology used as the scenario for the simulations. The duration of each simulation was set to 400 seconds. The results reported in this work are the average of 10 runs executed with different seed values for the random number generator. Results refer to a network composed of 1 Access Point and 4 clients distributed over a 500x500 meters square field. A maximum power transmission of 18 dBm, a signal attenuation threshold of 110 dBm and a path loss coefficient of $\alpha = 4$ were set during the whole simulations.

In order to validate the ADRR scheduling discipline, node number 1 through 4 have been fed with a constant bitrate (CBR) connection connection generated at the Access Point.

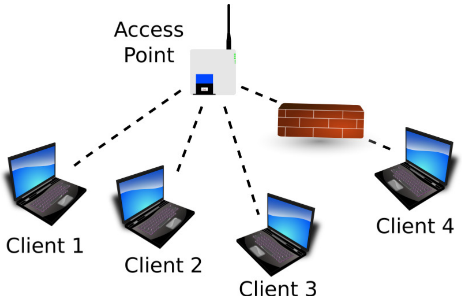


Fig. 2: The star-shaped topology used in the simulations.

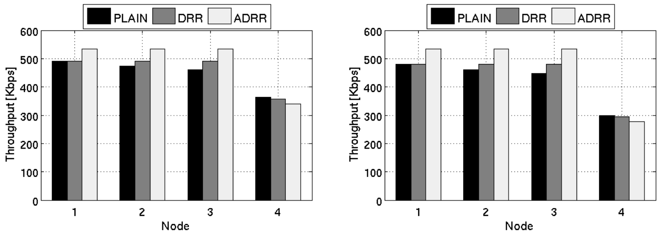
We have modelled each CBR connection as a single UDP stream with inter-departure time and packet size equal to, respectively, 15ms and 1000 bytes resulting in a final bitrate of, roughly, 533 kb/s (i.e., the average bitrate for a standard quality encoded YouTube video). Measurements have been repeated using FCFS (which is the default packet scheduling policy implemented in most IEEE 802.11 devices) and DRR. For the DRR scheduling discipline, the quantum has been set to 1000 bytes.

Simulations have been carried out exploiting three deployment scenarios differentiated by the channel conditions experienced by node number 4. Notice that in each deployment, all nodes are in radio range. However, while node number 1 through 3 are kept close to the gateway, node number 4 is positioned in such a way to experience channel conditions ranging from Good (100% delivery rate) to Poor (60% delivery rate) with an intermediate Medium quality (70% delivery rate). Poor, medium and good channel conditions basically means that channel has variations in the signal-to-noise ratio (SNR) from low to high values. This can happen for several reasons i.e. the presence of obstacles between nodes, the distance, the presence of external interference.

C. Results

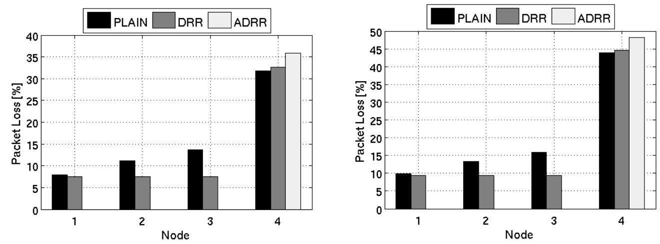
In order to evaluate the performance of the ADRR packet scheduling algorithm, we focused on some network-level metrics which closely reflect the system behaviour at the application level. These metrics are throughput and packet loss.

Figure 3 summarizes the outcomes of the simulations. As it can be observed, the *ADRR* scheduler addresses the *"IEEE 802.11 performance anomaly"* by maintaining a high throughput over reliable links (Node number 1 through 3) as opposed to both the *FCFS* and the *DRR* scenarios where performance falls when node number 4 starts to experience poor channel conditions.



(a) Medium channel conditions. (b) Poor channel conditions.

Fig. 3: Throughput for each client in the network using different scheduling policies (Simulations).



(a) Medium channel conditions. (b) Poor channel conditions.

Fig. 4: Packet Loss for each client in the network using different scheduling policies (Simulations).

As it can be seen from the figure, when node number 4 moves away from the access point, the ADRR scheduler 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 4 experiences poor channel conditions (see Fig. 3b), ADRR outperforms both FCFS and DRR by delivering an higher aggregated throughput and by allocating to node number 4, a percentage of the bandwidth which is only slightly lower than the optimal case.

Packet loss is reported in Fig. 4. As expected, when the channel conditions for node 4 deteriorates, the packet loss significantly increases also for the other nodes. Such behaviour is to be linked to the channel contention mechanism which requires node 4 to perform multiple transmission attempts increasing the channel busy period and thus the collisions experienced by the other nodes whose transmission are probably outside the carried sense range of node number 4.

VI. PROTOTYPE

A prototype has also been implemented and evaluated in order to demonstrate the applicability of the ADRR scheduling policy within a realistic scenario, namely an IEEE 802.11-based WMN. In particular, the mesh backhaul is implemented using WING, an experimental IEEE 802.11 wireless mesh networking toolkit [11], [12] supporting link quality routing through the WCETT metric [4]. Mesh routers are built around the Gateworks Cambria GW2358-4 processing board. Each node is equipped with two IEEE 802.11a/b/g wireless interfaces (Atheros chipset) with RTC/CTS disabled (the board supports up to four wireless interfaces). Routers exploit the OpenWRT as operating system. All the developed software has been released under a BSD License and made available to the community [13].

The experimental setup is sketched in Fig. 5. Traffic is generated at node number 1, the mesh gateway, using the Jugi's Traffic Generator (JTG), a freely available synthetic traffic generator [14]. JTG can generate and inject different traffic patterns over TCP and/or UDP sockets. Traffic is then collected at the receiver side (nodes number 1, 2, 3, and 4) where they are analyzed. During our experimental campaign, node number 2 through 4 have been fed with a

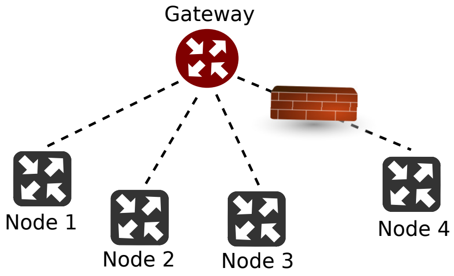
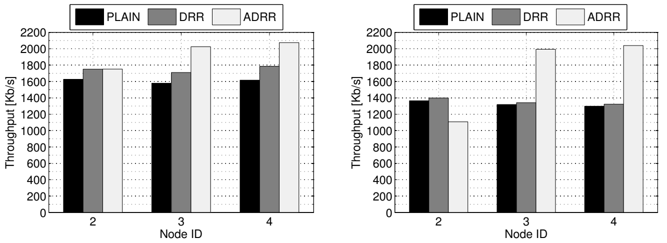


Fig. 5: The topology used for the experimental campaign.



(a) Medium channel conditions.

(b) Poor channel conditions.

Fig. 6: Throughput for each client in the network using different scheduling policies (Testbed).

CBR connection generated at node number 1. Each connection establishes as a single UDP stream with inter-departure time and packet size equal to, respectively, 2ms and 1460 bytes. All measurements were performed over a 1 minute interval. The campaign was carried out with the devices operating in the IEEE 802.11b mode.

Figure 6 summarizes the outcomes of our measurement campaign. As observed from the figure, the experimental results confirm what already confirmed with the simulations, e.g., the fact that the ADRR is capable of delivering performance isolation between competing flows when node number 2 starts to experience poor channel conditions.

Moreover, when channel conditions 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 that the ADRR scheduler is capable of exploiting channel fluctuation by opportunistically allocating more airtime to links that experience better channel conditions. 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 findings in [15] where channel fluctuations are exploited by transmitting information opportunistically when and where the channel is strong.

VII. CONCLUSIONS AND FUTURE WORK

We proposed an opportunistic scheduler capable of addressing the well known performance anomalies of IEEE 802.11 networks. The proposed architecture is capable of providing performance isolation with IEEE 802.11-based wireless mesh networks. The optimal scheduled list is computed exploiting measurable routing metrics. Using both simulations and real experimentation, we showed that the ADRR scheduler is capable of exploiting channel fluctuation by opportunistically allocating more airtime to links that experience better channel conditions.

As future work we plan to extend the ADRR scheduler in order to exploit path diversity when computing the optimal scheduling list. In this context, the residual path metric allows us to differentiate traffic routed over homogeneous paths from traffic that experiences good link condition only locally and that will be routed 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 on the validation of the ADRR scheduler over a larger testbeds (in terms of both number of nodes and network coverage) is also envisioned. This will allow us to obtain further insight into the scalability of the scheduling discipline discussed in this paper.

REFERENCES

[1] 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.

[2] 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*, vol. 34, no. 4, pp. 121 – 132, Oct. 2004.

[3] 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.

[4] 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.

[5] R. Riggio, D. Miorandi, and I. Chlamtac, "Airtime deficit round robin (ADRR) packet scheduling algorithm," in *Proc. of IEEE MeshTech Workshop*, Atlanta, GA, USA, Oct. 2008.

[6] J. Bennett and H. Zhang, "Hierarchical packet fair queueing algorithms," *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, pp. 675–689, Oct. 1997.

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

[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*, vol. 13, no. 4, pp. 481 – 495, Aug. 2007.

[9] P. Starzetz, M. Heusse, F. Rousseau, and A. Duda, "Virtual flow queueing for improving tcp performance over ieee 802.11 wlans," in *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, 31 2008-april 3 2008, pp. 2158 –2163.

[10] 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.

[11] R. Riggio, N. Scalabrino, D. Miorandi, F. Granelli, Y. Fang, E. Gregori, and I. Chlamtac, "Hardware and software solutions for wireless mesh network testbeds," *IEEE Communication Magazine*, vol. 46, no. 6, pp. 156 – 162, Jun. 2008.

[12] R. Riggio, K. Gomez, T. Rasheed, M. Gerola, , and D. Miorandi, "Mesh your senses: Multimedia applications over wifi-based wireless mesh networks," in *Proc. of Secon*, Rome, Italy, 2009.

[13] "WING Toolkit. Wireless Mesh Network for Next Generation Internet." [Online]. Available: http://wing-project.org

[14] "JTG." [Online]. Available: https://hoslab.cs.helsinki.fi/savane/projects/jtg/

[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*, vol. 44, no. 7, pp. 2796–2815, Nov. 1998.