

## Feedback-based Adaptation for Improved Power Consumption

Christos Bouras<sup>1,2</sup>, Vaggelis Kapoulas<sup>1,2</sup>, Kostas Stamos<sup>1,2</sup>, Nikos Tavoularis<sup>1,2</sup>

<sup>1</sup>Computer Technology Institute and Press  
“Diophantus”, Patras, Greece

<sup>2</sup>Computer Engineering and Informatics Dept.  
University of Patras, Patras, Greece  
{bouras, kapoulas, stamos}@cti.gr,  
tavoularis@ceid.upatras.gr

Georgios Kioumourtzis

Center for Security Studies, Athens, Greece  
gkioumou@gmail.com

Nikos Stathopoulos

Computer Engineering and Informatics Dept.  
University of Patras, Patras, Greece  
stathopou@ceid.upatras.gr

**Abstract**— In this paper we present a feedback-based adaptation mechanism that adjusts the transmission power of a wireless card on commodity PCs depending on the quality of the connection. Our purpose is to manage the available power in order to achieve lower power consumption without negatively affecting the user’s perception of connection quality. We based our implementation on an existing theoretical model and focused on resolving problems and removing assumptions which made it inefficient in real life implementation. The initial model manages to minimize the power consumption in networks with exactly two nodes. In this paper, we extend the model to consider the possibility of the existence of a base station, where any number of nodes can be connected. Our objectives for the base station are to minimize the power consumption and guarantee continuous connectivity for all mobile nodes. We implement the adaptation mechanism for a specific adapter with open sources drivers thus allowing necessary modifications. We conduct a number of real world experiments. The results indicate that power consumption can be significantly reduced for nodes that are either almost stationary or slowly moving (e.g. at walking speed), without any significant increase in packet loss. The results are quite important as nowadays mobile devices with limited battery life time use tethering to become base stations for other devices like in ad-hoc networks.

**Keywords**—power management, wireless, adaptation, RSSI

### I. INTRODUCTION

As networks and connected devices become more mobile and thus energy constrained, and as the requirements for lower energy consumption become more demanding, the issue of consumed power by wireless network cards is becoming an intensively researched topic. Furthermore, concerns by the public on electromagnetic fields increase the interest of such research. When using a wireless connection, increasing the strength of a signal may result in better quality, as perceived by the end user application, but it also results to more energy consumption and stronger electromagnetic fields. On the other hand, if the signal strength is reduced too much, there is the risk of reduced performance or even losing connectivity completely.

This paper deals with the issue of developing and testing in actual environments a mechanism that manages

transmission power on wireless network cards in order to guarantee both acceptable connection quality and the lowest possible power consumption and electromagnetic field. Similar solutions have been proposed in the literature with an emphasis on theoretical treatments of the subject. [1] proposes a transmission control power algorithm and tests it with the ns-2 [2] network simulator. [3] presents a decentralized power control protocol that manages to improve the throughput performance. The proposed protocol is evaluated with the same simulation software. The 802.11h standard [4] includes a transmission power control mechanism [5]. However, the 802.11h standard refers to 5 GHz band networks and its purpose is to eliminate the interference with satellites and radar systems. In [6] proposes another power control mechanism that reduces the influence of interference on cellular networks. Power control is also developed for 802.11b wireless networks in [7]. This model operates between two nodes at the transport (TCP) layer and it is not functional in any kind of topologies with more than one node. Finally, radio resource management features are implemented in Cisco products [8]. These features include various transmission control power algorithms.

Our mechanism adjusts the transmission power by utilizing the Received Signal Strength Indication (RSSI; a measurement of the strength of a received signal). Several prior researches have taken place in the areas of power optimization and RSSI utilization for link quality estimation. In [9], the authors propose a power management mechanism that is used for routing packets in ad hoc networks with power efficiency. In [10], the RSSI is one of the metrics used to improve routing efficiency in a wireless network. In [11], information transferred in a multi-hop path includes power information in order to guide the power management mechanism. The mechanism described in [12] adapts power levels according to information that is available locally. In [13], RSSI is used to estimate wireless channel state and feeds an algorithm that optimizes MAC layer parameters. The suitability and limitations of RSSI as a link quality metric are discussed and evaluated in [14]. Most of prior research has not focused on the issue of aggregating the power transmission needs of multiple direct receivers of the data stream.

Our purpose is to expand the model described in [7] and make it more functional for real world implementation. We consider the problem of the existence of a base station with a number of connected wireless nodes. In this situation the base station has to decide the appropriate level of the transmission power in order to minimize the consumption, but it is also important to guarantee the continuous connectivity of the network. In addition, we consider that the rest of the nodes can also change the power levels. At the beginning of the operation the base station keeps the transmission power at a minimum level. When it ascertains that any of the nodes have moved sufficiently away of its range, it has to increase the transmission power in order to ensure the continuous communication. A very important aspect of the operation is the adaptive behavior of the base station, in which it tries to minimize the transmission power whenever it is possible. The nodes around the base station change their transmission power according to the algorithm presented in [7].

The contribution of the paper lies both in resolving problems and removing assumptions in the actual implementation of the model proposed in [7], as well as extending the model and the implementation to work in a setting with a base station (access point) and multiple wireless devices connecting to it.

In an actual environment, the base station may be any wireless router which wants to minimize power consumption. This includes mobile and portable devices that use tethering to become local WiFi access points for other devices in their vicinity.

Another case is in ad-hoc networks in which the nodes participating in the network act also as routers forwarding packets that are received by other nodes. Especially, the proposed mechanism can find its place in wireless sensor networks in which power consumption is a big issue.

Results indicate that the implementation of the extension of the model in [7], can lead to considerable power savings that might extend the battery life-time of such devices, especially in the case of slowly moving nodes (e.g. walkers).

The rest of the paper is organized as follows. Section 2 describes the feedback-based adaptation algorithm. Section 3 provides an overview of the implementation in real world mobile devices. We present the conducted experiments with performance evaluation results in Section 4. Finally section 5 concludes the paper with a summary and thoughts for future work.

## II. FEEDBACK-BASED ADAPTATION

Our mechanism adjusts the transmission power by utilizing the Received Signal Strength Indication (RSSI). RSSI is the measurement of the strength of a received signal.

If the RSSI is lower than a specific threshold ( $P_{thr}$ ), the decoding of the signal cannot be guaranteed due to low Signal to Noise ratio (SNR). Additionally Path loss is the reduction of the power between the sent ( $P_{tx}$ ) and received ( $P_{rx}$ ) signal [7].

$$\text{Path Loss} = P_{tx} - P_{rx} \quad (1)$$

It is obvious that the power of the received signal must be higher than the threshold. Thus the minimum transmission power can be calculated as follows.

$$P_{tx\ min} = \text{Path Loss} + P_{thr} \quad (2)$$

However that value is not tolerant to any fluctuations to RSSI and would cause continuous errors in received packets. To solve that problem we increase the above value by a constant factor ( $P_c$ ) that provides a comfortable buffer zone for RSSI fluctuations.

$$P_{tx} = \text{Path Loss} + P_{thr} + P_c \quad (3)$$

To calculate the transmission power it is important for the senders to have knowledge of the RSSI of their own packets. Thus the receivers have to send a feedback message to the senders containing the RSSI value.

This value can also change temporarily due to environmental factors that are independent of the movement of the nodes. These factors are not likely to affect the next frames and should therefore not affect heavily the mechanism calculations. For that reason, using the single last RSSI value to calculate the path loss is not a sufficiently robust approach. Instead, we calculate an average value using the Exponential Weighted Moving Average (EWMA) [15].

A moving average is the average value of a fixed size list. Every time that a new value is appended to the list, it is placed at the head and the tail is removed (FIFO). Then the average value is recalculated. To emphasize the more recent values we use different weights for each data value according to its place on the list. The more recent data have greater weights than the older ones.

Using this method to calculate the path loss we remain consistent to the previous values but give more importance to more recent values. Additionally, we have smooth plots and avoid significant changes to the path loss. These changes are more likely to occur due to environmental factors rather than the movement of the nodes.

If the sender is a base station and multiple nodes are connected, then the reduction of the transmission power is not efficient before we guaranty the connectivity to the rest of the nodes. In order to achieve this, we use a table containing the nodes that are connected to the base station and the path loss for each one of them. The base station sets its power according to the maximum path loss. If we receive an RSSI feedback message we have to update that table. Also if the sender of this message is the node who defines the transmission power we have to recalculate the power. We present the pseudo code of our mechanism, as follows:

```

message_received(packet) {
    rssi=extract_rssi(packet);
    rssi_avg=average(rssi,mac)
    send the rssi_avg to node with that mac
    address

    previous_rssi=get_previous_rssi(packet.mac)
    if (change(rssi_avg,previous_rssi)>MAX_CHANGE)
    {
        send rssi to node with that mac address
        set_previous_rssi(packet.mac,rssi_avg)
    }
}

```

```

get_rssi_information(rssi) {
    Path_loss=calculate_path_loss(rssi).
    Ptx=calculate_ptx(path_loss)
    set_transmission_power(Ptx)
}

```

The above pseudo code contains two main functions:

- The `message_received` function runs when a packet has been received, and then it informs the sender about its RSSI. This method runs at the receiver.
- The `get_rssi_information` function runs when RSSI information has been received. Then it calculates the path loss and the appropriate transmission power. This method therefore runs at the sender of the packets.

Since every node can be both a sender and also a receiver, these two functions run in parallel.

The base station, implements a different `get_rssi_information` function, as follows:

```

get_rssi_information_for_station() {
    rssi=wait_until_get_rssi_packet();
    Path_loss=calculate_path_loss(rssi)
    PrevMaxPathLoss=getMaxPathLoss(table);
    getMaxPathLoss(table)
    update_table(packet.mac,Path_loss)

    if(PrevMaxPathLoss != getMaxPathLoss(table)) {
        Ptx=calculate_ptx(path_loss)
        set_transmission_power(Ptx)
    }
}

```

This function updates the table and ensures that the transmission power would be set according to the node that is located further away (the one with the maximum path loss). If the maximum path loss has changed, we also recalculate the transmission power. This is possible in two cases: The first case refers to the scenario in which any node moved too far away and we have to increase the power. The second case refers to the scenario in which the node that was located further away has come closer. That means that we can minimize the power according to the path loss of the node with what used to be the second largest distance to the base station. In other words the algorithm sets the power at a level that would be suitable even for the most distant node at any moment.

Every node sends periodically a keep-alive message to the base station. If the base station stops receiving that messages from a node, it may consider that node inactive and it removes it from its node table. It also minimizes its power

if necessary, in order to accommodate the next most-distant node.

### III. IMPLEMENTATION ISSUES

Retrieving the RSSI per packet and setting the transmission power is not supported by all wireless adapters and their drivers. In our work we used Cisco Aironet cb21ag cardbus adapters [16] and the ath5k driver for Linux [17]. This driver is open source software which allows us to perform a number of changes.

The hardware supports a variety of transmission power levels. The default transmission power without our mechanism is the maximum value. The ath5k driver can extract the RSSI per packet instead of calculating the average value like most drivers do.

Our mechanism consists of two components. First we change the driver to create a virtual file at the proc filesystem [18]. This file contains information about the RSSI and the MAC address of every new packet received. Secondly, at the application level we create a service that reads the information from the virtual file and communicate with the nodes using a known port. This service calculates the exponential moving average for every node and sends a UDP packet whenever the average RSSI has changed significantly. It listens also to that port, revives the packets and changes the transmission power using system commands. As we use UDP packets for feedback messages it is possible that some of them may be lost. For this reason when we receive an RSSI packet we send an acknowledgment message containing the RSSI value that we have received. If the sender of the RSSI message does not receive the acknowledgment or the value on the acknowledgment packet is wrong, it resends the RSSI message.

This implementation divides the very fundamental element on driver level and implements the basic mechanics on an application making it simpler and easily expandable.

To calculate the average value we use the Linux implementation of EWMA.

Our implementation of the proposed mechanism is open source code which can be downloaded, tested and modified freely [19].

### IV. EXPERIMENTS AND RESULTS

To measure the quality of the network we use the MTR (My traceroute) utility, which is a computer application that combines the functionality of the traceroute and ping programs in a single network diagnostic tool.

MTR cat uses ICMP or UDP protocol to take statistics from the network. By using MTR we are able to calculate a number of network metrics such as the average latency, the worst latency, the packet loss, etc in a given amount of tries. The performance evaluation of the implemented mechanism is measured by the aforementioned metrics. We are particularly interested in the levels of power consumption and the packet losses as they indicate the level of potential trade-off that the mechanism may achieve. Packet losses provide an indication of the way throughput will be affected under a given situation with and without the operation of the

proposed mechanism. Increased packet losses cause retransmissions, more bandwidth usage and increased delay, leading to an overall reduction of Quality of Service (QoS).

For the first experiment we use only two nodes simulated by two laptops. We hold the first laptop (which serves as the base station) on a static location and we move the second around the area of range of the first one. On some predefined locations we take statistics of the quality of the network using the MTR utility as described above. We choose these locations to represent a variety of situations (w.r.t. to the distance between the nodes, and obstacles in the line of sight) so that the laptops would have different transmission power in every case. We repeat each experiment twice; the first time by using our adaptation mechanism and the second time without using any radio resource management mechanism. The purpose of this experiment is to determine how efficient our mechanism is compared to the default transmission method. The experiment is conducted in an open space; this is the most common use case with moving nodes (i.e. outdoors). It also reduces the possibility of interference of walls and other factors.

The following table shows the experimental results from several experiments with different random movements, repeated both with and without the adaptation mechanism. Other parameters that are also measured such as latency indicate no variation with or without the adaptation mechanism, as the transmission distance always remained at 1 hop.

TABLE I. COMPARISON OF EXPERIMENTAL RESULTS

No adaptation		Using adaptation	
<i>Tx power (avg)</i>	<i>Packet loss</i>	<i>Tx power (avg)</i>	<i>Packet loss</i>
20	1.6%	19.5	2.6%
20	0.2%	13.5	3.6%
20	0.0%	5	0.1%
20	0.0%	5	0.0%

According to the results there is no important variation on packet loss which means that the adaptation mechanism does not affect connection quality.

However, the average transmission power has been reduced significantly, which demonstrates that the mechanism is capable of achieving significantly lower energy consumption, while at the same time maintaining connection quality characteristics at the same level as before. The energy savings can range from 2% to 75% depending on the distance of the transmitting and receiving nodes.

The behavior of the adaptation mechanism in these experiments satisfies our design goals. The power savings level is higher when the distance between the base station and the moving node is small. At the same time the packet loss ratio values are negligible.

Obviously, the mechanism is very suitable for smaller distances. For moving nodes in larger distances from the base station the power savings may be smaller or even negligible (near the outskirts of the coverage area).

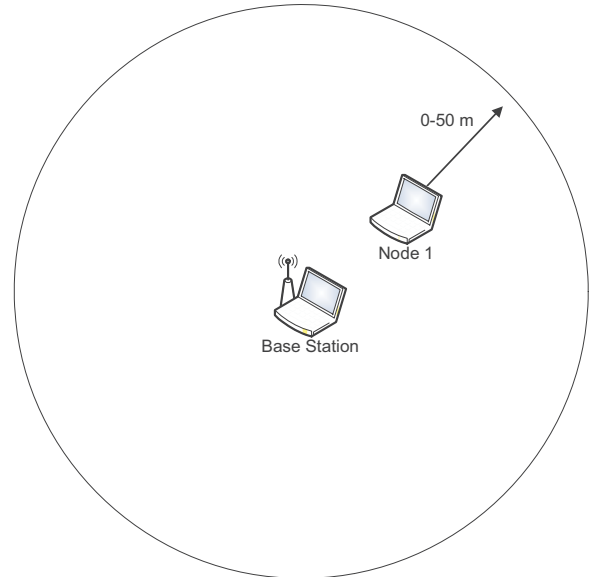


Figure 1 Experiment with a single receiver

Figure 2 presents the results from a similar experiment conducted while the node was slowly moving (at walking speed) in a straight direction away from the base station (as shown in Figure 1), in order to evaluate the adaptation mechanism's response to a deteriorating environment.

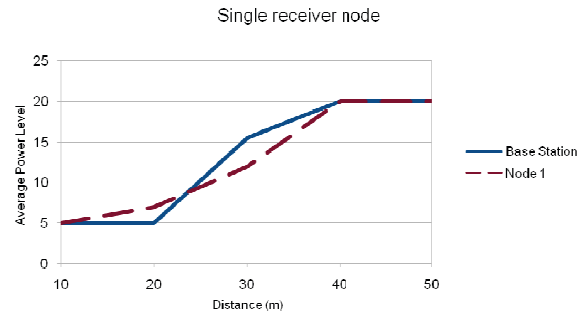


Figure 2 Average power level for a single receiver slowly moving away

Figure 2 depicts the dependence between the average power transmission levels and the distance between the base station and the moving node. We can observe how the power level increases as the distance increases, until it peaks just before the node is about to lose connectivity with the base station. The difference in the power used by the base station and the moving node reflects the different estimation done based on the (slightly) different feedback they get. Still both follow a similar pattern of adapting the transmission power as their distance increases.

Figure 3 shows how the packet loss ratio is simultaneously affected. It can be seen that until the 50m distance (where communication was lost), packet loss increased very slowly, as a result of both nodes increasing the transmission power accordingly (i.e. adapting to the changing distance between the nodes). After the moving node reaches the limit of the coverage area of the base

station, connection is lost (as even with maximum transmission power it could not be maintained).

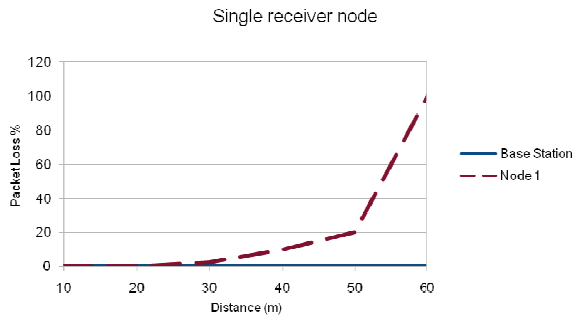


Figure 3 Packet loss for a single receiver slowly moving away

In our next experiment, we use two nodes connecting to a base station (Figure 4). We are moving the nodes in order to measure the quality of the network connection as well as the average transmission power of each node at the new positions. We consider the case that one node moves while the other remains stationary.

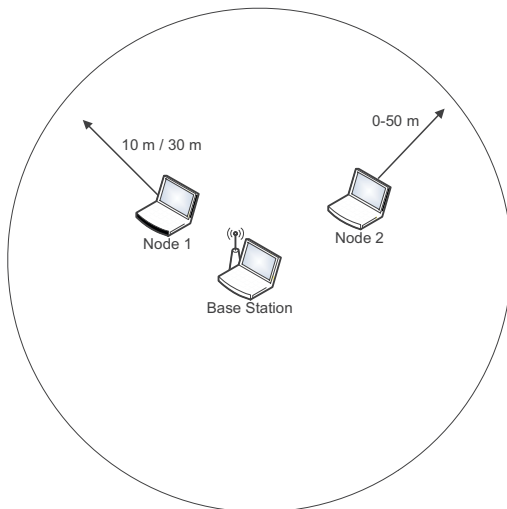


Figure 4 Experiment with multiple receivers

The following figures present the results split in two periods. During the first period, Node 1 stays 10m away from the base station, while Node 2 slowly moves away from the base station. Then Node 1 moves to 30m away from the base station, and Node 2 repeats the above described movement sequence.

As can be seen in Figure 5, the power level for the moving Node 2 gradually increases while it distanced itself, whereas Node 1 power level always remained at a comfortable low level. Obviously, the power level for the base station changes like the one for Node 2, so that connection can be maintained with the node farthest away.

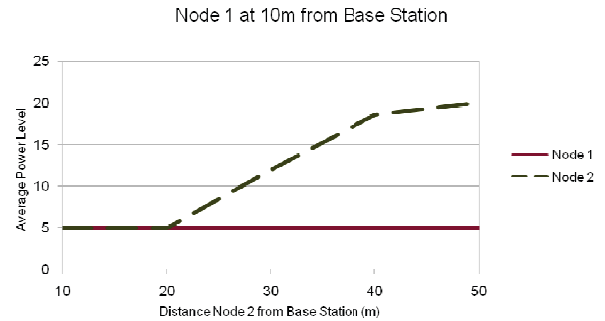


Figure 5 Average power level for two receivers, one staying near the base station and the other slowly moving away

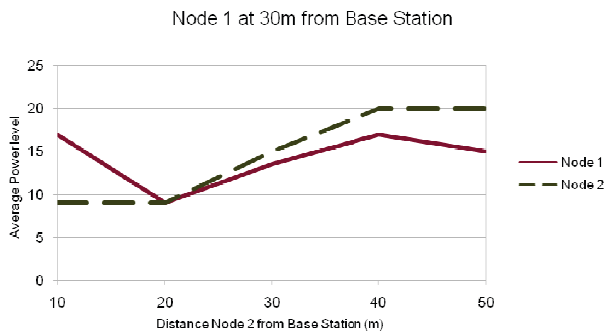


Figure 6 Average power level for two receivers, one staying away from the base station and the other slowly moving away

When Node 1 also moves further away from the base station, its power level increases, initially to a high value, but afterwards settles to lower values that still guaranteed connection quality. This can be verified by Figure 7 and Figure 8, which show that packet losses for neither node increased to a level above 5%, except when Node 2 was about to exit the connection range of the base station at about 50 meters distance from it.

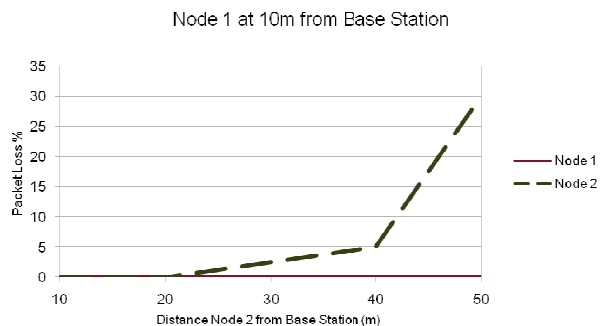


Figure 7 Packet loss for two receivers, for two receivers, one staying near the base station and the other slowly moving away

The only exception is a brief period just after Node 1 moved to 30m distance from the base station, due to the



amount of time that the algorithm needed to adjust power levels. That means that the transmission power adaptation mechanism is more suitable for slowly moving nodes (e.g. walkers) rather than quickly nodes (e.g. cars). Quickly moving targets may be accommodated, with less packet loss, by being conservative and using more transmission power than required. i.e., for quickly moving nodes we can trade-off less packet loss for more transmission power. It may also be possible to tune the mechanism to adapt more aggressively, again at the cost of using more transmission power that necessary in other cases.

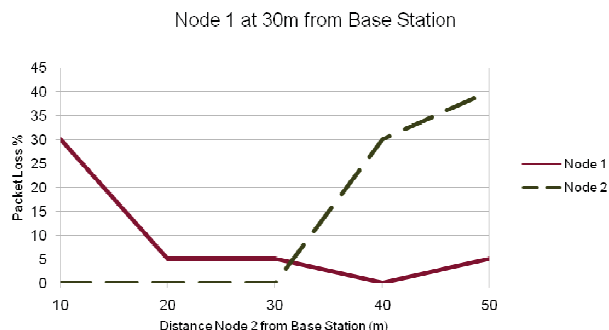


Figure 8 Packet loss for two receivers, one staying away from the base station and the other slowly moving away

## V. CONCLUSIONS AND FUTURE WORK

In this paper we have demonstrated how a power management mechanism that utilizes RSSI for feedback may be implemented in a real-world operating system and be evaluated in realistic settings.

The actual implementation revealed several issues that were not easily identifiable through simulations and allowed accurate performance measurements.

The experiments showed that for slowly moving nodes, the mechanism achieved an important reduction of the transmission power. That reduction can significantly minimize the power consumption, leading to obvious environmental and economic advantages. Also minimizing the signal strength is important for the human health, especially on people with electromagnetic hypersensitivity, although that sector needs more research before we can safely reach conclusions. We expect the implementation of such mechanisms to increase in the near future as mobile computing becomes more wide spread and environmental, economical and health concerns attract a lot of attention. Our current implementation code can be found at [19].

The implementation of such a mechanism in self-powered sensor networks provides also the potential to increase the life-time of the network. This is a very important application area of this and other similar power adaptation mechanisms.

In our future work we intend to investigate support for similar mechanisms in wider sets of wireless adapters and the possibility of an application-level implementation of the

mechanism that may be more easily transferable to different wireless card implementations.

We also plan to investigate the feasibility of extending input metrics for the mechanism beyond RSSI in order to produce combined and more accurate estimations on link quality.

Finally, we intend to investigate the mechanism under faster movement scenarios, in order to investigate ways for reducing its delay in adapting power levels.

## REFERENCES

- [1] C. Bouras, V. Papapanagiotou, K. Stamos, G. Zaoudis, "Efficient Power Management Adaptation for Video Transmission over TFRC" (AICT 2010), Barcelona, Spain, May 9 - 15 2010.
- [2] The network simulator 2: <http://www.isi.edu/nsnam/ns/> (Accessed December 2012).
- [3] J. P. Monks, V. Bharghavan, and Wen-mei Hwu, "A Power Controlled Multiple Access Protocol for Wireless Packet Networks," IEEE INFOCOM 2001, Alaska, April, 2001 pp 219-228.
- [4] IEEE Std 802.11h-2003, Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 5: Spectrum and transmit power management extensions in the 5 GHz band in Europe (Approved 29 December 2003).
- [5] Daji Qiao, Sunghyun Choi, Kang G. Shin "Interference analysis and transmit power control in IEEE 802.11a/h wireless LANs" IEEE/ACM Transactions on Networking (TON) Volume 15 Issue 5, October 2007.
- [6] E. Lopez-Aguilera, J. Casademont Serra, "A transmit power control proposal for IEEE 802.11 cellular networks," 6th International Workshop on Applications and Services in Wireless Networks, 2006. New York: Institute of Electrical and Electronics Engineers, 2006, p. 235-242.
- [7] A. Sheth and R. Han. An Implementation of Transmit Power Control in 802.11b Wireless Networks. Technical report, Dept. of Comp. Science, Univ. of Colorado, 2002.
- [8] Cisco Systems, Inc. "Radio Resource Management under Unified Wireless Networks", Document ID: 71113, May 17, 2010
- [9] Srikanth Krishnamurthy, Tamer ElBatt, Dennis Connors, "Power management for throughput enhancement in wireless ad-hoc networks" U.S. Patent 6,735,448, filed Nov 7, 2000, and issued May 11, 2004.
- [10] William Vann Hasty, Jr., Peter J. Stanforth, "System and method for using per-packet receive signal strength indication and transmit power levels to compute path loss for a link for use in layer 2 routing in a wireless communication network" U.S. Patent 7,672,246, filed Feb 23, 2006, and issued March 2, 2010.
- [11] Philip Orlik, Zafer Sahinoglu, Jinyun Zhang, "System and method for reducing power consumption in a wireless communication network" U.S. Patent 7,035,677, filed Jan 21, 2003, and issued Apr 25, 2006.
- [12] Raissa M. D'Souza, Sharad Ramanathan, Duncan Temple Lang, "Adaptive power level setting in an ad-hoc wireless network", U.S. Patent 6,970,714, filed Apr 30, 2002, and issued Nov 29, 2005.
- [13] Anna Chaltseva, Evgeny Osipov, "On passive characterization of aggregated traffic in wireless networks", WWIC'12 Proceedings of the 10th international conference on Wired/Wireless Internet Communication, 2012, pp. 282-289.
- [14] Angelos Vlavianos, Lap Kong Law, Ioannis Broustis, Srikanth V. Krishnamurthy, Michalis Faloutsos, "Assessing Link Quality in IEEE 802.11 Wireless Networks: Which is the Right Metric?", IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 15-18 Sept. 2008, pp. 1-6.
- [15] Statistical Analysis, Ya-lun Chou, Holt International, 1975, ISBN 0030894220, section 17.9.

- [16] Cisco Systems, Inc., Cisco Aironet 802.11A/B/G Wireless CardBus Adapter.
- [17] Atheros Linux wireless driver Ath5k, <http://wireless.kernel.org/en/users/Drivers/ath5k> (Accessed September 2012)
- [18] T. Killian, “Processes as Files”, USENIX Summer Conf. Proc., Salt Lake City, 1984.
- [19] Implementation code, [http://ru6.cti.gr/ru6/cross\\_layer.php#pman](http://ru6.cti.gr/ru6/cross_layer.php#pman) (Accessed December 2012).