An Empirical Characterization of Instantaneous Throughput in 802.11b WLANs*

Arunchandar Vasan and A. Udaya Shankar
Department of Computer Science and UMIACS
University of Maryland
College Park, MD 20742.
CS-TR-4389, UMIACS-TR-2002-69

Abstract— **We present an empirical, i.e, measurement-based, characterization of the instantaneous throughput of a station in an 802.11b WLAN as a function of the number of competing stations sharing the access point. Our methodology is applicable to practically any wireless MAC protocol. Our findings show that as the number of stations increases, the overall throughput decreases and its variance increases. Furthermore, the per-station performance depends significantly on the wireless card implementation and does not depend as much on the station's processing capacity.**

I. INTRODUCTION

Wireless LANs (WLAN) are being rapidly deployed as a solution for mobile last-hop connectivity. The IEEE 802.11b [9] standard, allowing for transmission speeds upto 11Mbps in the 2.4GHz ISM band, has been implemented by many vendors. An accurate model of the attainable throughputs in such WLANs is needed to facilitate capacity planning and network deployment.

We present an empirical, i.e., measurement-based, characterization of the instantaneous throughput of a station in an 802.11b WLAN as a function of the number of competing stations sharing an (AP) access point. We call this characterization a *profile*. The profile, once obtained, has various uses: to quantify the performance available to a user, to identify problems in wireless cards and their drivers, to compute the performance of applications running over WLANs, and so on.

Our measurements were done by sniffing the medium, rather than instrumenting the hosts or access points. This ensures that there is no instrumentation overhead or artifacts. It accounts for the quirks of the wireless cards and drivers. It makes the profiling methodology applicable to practically any wireless MAC protocol and independent of the hosts and access points.

We study the performance of an 802.11b WLAN BSS (Basic Service Set) in the infrastructure mode, i.e, when a number of stations are associated with a single AP. Previous works [5], [6], [8], [11], [12] have characterized the performance of 802.11 networks through analytical methods and simulations. Existing measurement studies [13] measure the performance of transport protocols over 2Mbps 2.4GHz FHSS pure CSMA/CA WaveLAN system, which predates 802.11b. Reference [4] measures the throughput available on one wireless link between two APs operating at 2Mbps. A measurement study which characterizes user behaviour in public 802.11 WLANs and its impact on network performance has been reported [3]. However, to the best of our knowledge, there exists no work which actively *measures* the attained throughput in 802.11b WLANs with many stations.

A. Specific Results

- The maximum overall throughput of an 802.11b WLAN in our experiments is about 6.45834 Mpbs with a standard deviation of 0.02649 Mbps for a single station. This throughput is at the lowest level and includes the 802.11 layer headers. The corresponding goodput (application level throughput) is 6.205 Mbps with a deviation of 0.0248 Mbps. As the number of stations increases, the throughput falls and the standard deviation increases. For example, with 5 stations, the overall throughput reduces to 5.74568 Mbps and the deviation increases to 0.540433 Mbps.
Our experiments validate the results reported in [5], which analyzed IEEE 802.11 operation under various assumptions such as time-independent modeling, geometrically distributed packet size, etc. Those results also showed that the IEEE 802.11 standard operates at rates lower than a theoretically possible 7.2754 Mbps.
- The maximum instantaneous throughput attained by a machine in the ad-hoc mode was 6.3204 Mbps with a deviation of 0.02677 Mbps.
- Furthermore, the per-station performance depends significantly on the wireless card implementation and does not depend as much on the processing capacity of end hosts.

---

*Supported in part by the UMIACS/LTS contract

B. Roadmap

Section II describes the profile for 802.11b and its applications. Section III describes our experimental setup. Section IV describes our validation of the throughput measurements. Section V examines the performance drop caused by varying hardware. Section VI concludes.

II. PROFILE OF 802.11B

A profile characterizes the relationship between instantaneous metrics of a system. Typically, the systems of interest are too complex for the profiles to be analytically obtained. Simulation methods do not capture real world idiosyncracies. So, we focus on obtaining profiles empirically by exercising the system in various regimes, collecting traces, and analyzing them. Empirical TCP profiles [10], characterizing the instantaneous throughput of TCP as a function of instantaneous RTT and instantaneous loss-rate, have been used to compute the performance of TCP/IP networks.

We have obtained a *profile* for 802.11b WLANs, expressing the instantaneous throughput attained by a station in an 802.11b WLAN as a function of the number of competing stations using the same channel, when all stations send data as fast as possible. Let N be the number of stations in the WLAN's AP cell. Let $B(N)$ denote the overall throughput for N stations, i.e., the sum of the per-station throughputs Our findings show that (1) as N increases, $B(N)$ decreases and its variance increases, and (2) the instantaneous throughput attained by a single station is clustered around $B(N)/N$.

A. Processing traces and profile computation

Given a WLAN access point with N associated stations sending data as fast as possible, a sniffer collects a trace of the 802.11b packets transmitted along with the time instants they were transmitted. From this trace, we compute a set of instantaneous throughputs, both overall and per-station, at various points in time.

The usual way to define the instantaneous overall throughput at time t is K/T where K is the number of packets transmitted over the medium in the time interval $[t, t+T]$ and T is a specified duration (of the order of a second or smaller). Similarly, the instantaneous throughput of a station at time t is obtained by using the number of packets sent by that station over $[t, t+T]$. However, this definition becomes problematic when a packet starts transmission but does not finish within the interval.

We solve this problem by computing the instantaneous throughput based on the time duration needed to transmit a fixed number of packets. So, if a packet transmission starts at time t and $K-1$ packets are transmitted after that, with the transmission of the Kth packet ending at time s, then the instantaneous overall throughput at time t is defined to be K divided by $s-t$. For the instantaneous throughput of a particular station at time t, we simply replace K by K_i, where K_i is the number of packets sent by the station out of those K packets.

Machine	Max Throughput Mbps	Deviation Mbps
madras	6.45834	0.026909
delhi	6.33004	0.0148434
cairo	6.3097	0.03173
ouzo	6.2190	0.0314
bombay	6.25208	0.288062

TABLE I
MAXIMUM THROUGHPUTS

We process the traces to obtain $B(N, t)$ sampled at various points in time. We varied K, the number of packets over which we computed the instantaneous throughputs, and, as expected, found that as K increases the deviation of all the sample throughputs decreases. We chose $K = 500$ packets which gives a deviation of about 3.9% from the mean in typical experiments. The results for throughputs attained by single stations are shown in table I. The actual configuration of the machines is described in section III and summarised in table II. A typical variation of instantaneous throughput of a single machine (madras) with time is shown in figure 2.

We repeated the same experiment with many stations. All stations were programmed to send as fast as possible. We show the effect of increasing the number of sending stations on the overall throughput in figure 1. We computed the throughputs over increasing values of K. As K increases, the deviation of the overall throughput reduces due to averaging over longer intervals.

All points of form $< V_i, T_i >$, where V_i is a sample value of instantaneous overall throughput and T_i is the corresponding instantaneous throughput of one station, are plotted in the same figure to generate the profile. The profile points are in packets/sec, where each packet has a UDP payload of 1472 bytes and a size of 1532 bytes along with the 802.11 headers (the rationale for this choice is explained in section III). In order to distinguish between clusters, we draw the outlines of the clusters instead of the clusters themselves.

Figure 3 shows the profile for the number of stations N varying from two through ten computed over $K = 500$ packets. Figure 4 shows the profile for the number of sta-

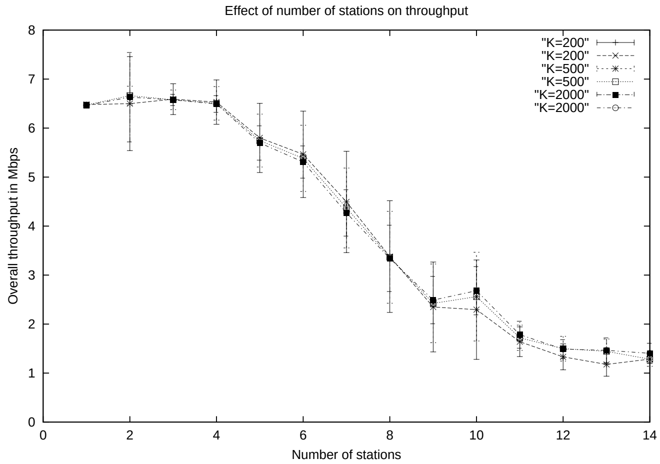

Fig. 1. Throughput with increasing number of stations

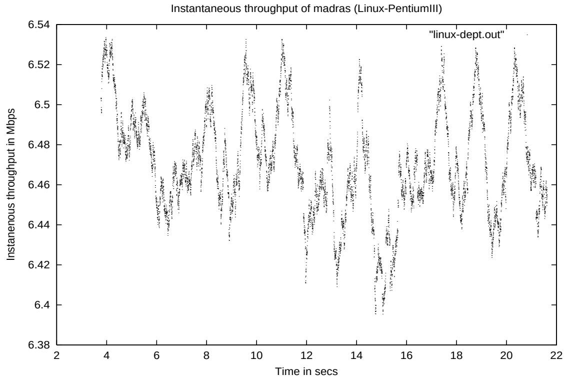

Fig. 2. Instantaneous throughput vs. time for madras (Linux-PentiumIII)

tions N varying from three through nine computed over $K = 10000$ packets. Figure 5 shows the actual clusters for $N = 3$, $N = 7$, and $N = 9$ computed over $K = 500$.

The clusters for $K = 10000$ are more concentrated as the throughputs over longer time periods would vary less, while short term throughputs have higher variance.

B. *Discussion of observations*

When all stations in a 802.11b WLAN try to send as fast as possible, each station gets an instantaneous throughput within a cluster of points around $B(N)/N$. The reason for such a behaviour is the ability of the 802.11b MAC to distribute bandwidth almost evenly **on an average**. The variation in the instantaneous throughput for the same background traffic suggests that the distribution of bandwidth could be unfair in the short term as has been reported in the literature [7].

As N increases, the overall throughput decreases and the spread of a cluster increases significantly (due to the effect of collisions and backoffs).

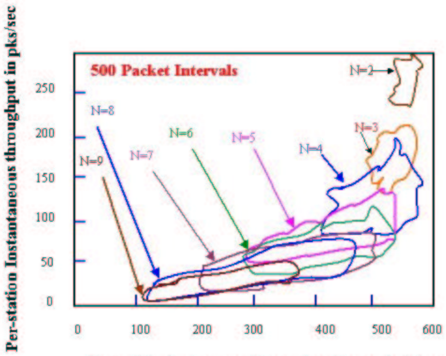

Fig. 3. Profile of 802.11b computed over $K = 500$

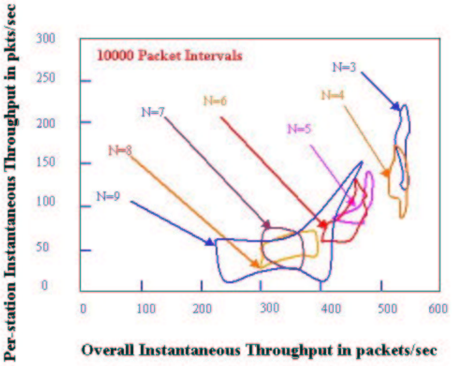

Fig. 4. Profile of 802.11b computed over $K = 10000$

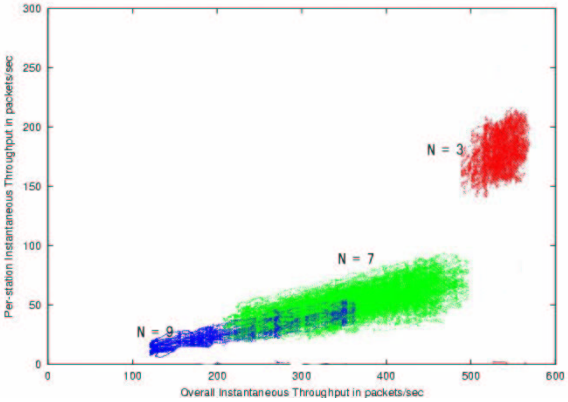

Fig. 5. Clusters for $N = 3, 7, 9$ computed over $K = 500$

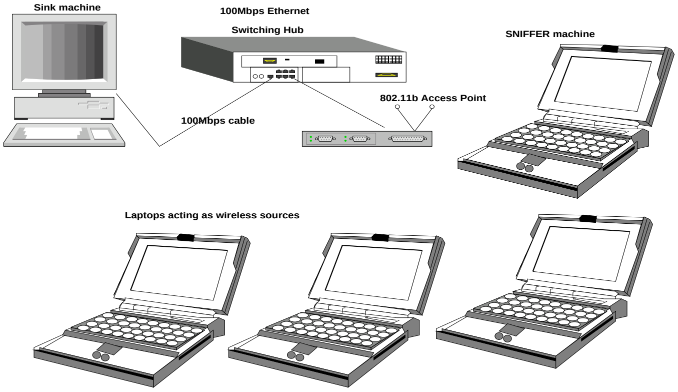

Fig. 6. Network Topology

C. *Applications of the Profile*

Given the number of active stations, the profile immediately yields the minimum throughput per-station. This is because any workload is subsumed by our workload, where all stations send as fast as they can. The profile can be used to evaluate different proposed or implemented MAC protocols. This is general enough to be applicable to most CSMA/CA media access methods (where throughput depends not only on the background traffic but the number of sources that generate the traffic also) as we have not made any specific assumptions about the MAC protocol to be measured.

Profiles can be used to model thoughput attained by different transports when source stations send at a rate less than the maximum (typically due to some higher-level control like TCP congestion control).

III. INFRASTRUCTURE USED FOR EXPERIMENTS

The network topology is shown in figure 6. Upto 14 wireless stations were associated with the AP. Both the AP and a machine called **sink** are connected to a 100Mbps ethernet switch. There was no other station connected to the switch. The wireless stations are the traffic sources and **sink** executes the sink program, which receives all the data sent.

A. *Stations, Cards, AP*

We used nine laptops and five Compaq IPAQs for our experiments. The laptops ranged ranging in processing speed from Pentium-4 1.2GHz through Pentium 166 MHz with main memory ranging from 256Mb to 64Mb. The IPAQs had a 200MHz Intel StrongArm processor and main memory of 32Mb. The machine **sink** had a Pentium-4 processor with 256Mb memory.

The AP used was Lucent Orinoco. The cards used for sending traffic were Compaq WL110 and Lucent Orinoco Silver, both of which have the same chipset. We disabled WEP on the cards and the AP in order to avoid any potential overhead. RTS/CTS usage was also disabled on all cards.

Name	Processor	OS
delhi	Pentium-4 1.2 GHz	Windows2000
bombay	Strong Arm 200 Mhz	Linux
madras	Pentium-III 850 MHz	Linux
cairo	Celeron 1.1GHz	Linux
ouzo	Pentium-III 850 MHz	Windows2000

TABLE II
TYPICAL CONFIGURATION OF VARIOUS MACHINES USED

B. *Software*

The OS on the sink was FreeBSD. It ran a DHCP server, which assigned an IP address to each of the associated wireless stations when they came up.

The OSs available on the laptops were Windows 2000 and Linux with kernel 2.4.7-10 or above. All IPAQs used Linux. The Linux machines used the `wvlan_cs` driver for Lucent Orinoco and Compaq WL110 cards, both of which have an identical chipset. The Linux driver used the `pcmcia-cs` package and was a loadable kernel modules. The Windows 2000 drivers used were provided by the card manufacturers. Table II summarizes the configuration of typical machines used in the experiment.

C. *Capturing Packets*

Our approach to packet capture was to sniff the medium in monitor mode. We used the sniffing capability available in the firmware of Compaq WL100. All timestamping of packets was done at the sniffer, thereby eliminating the problem of clock skews between different capturing hosts.

The sniffer was a high end laptop with a Pentium III 850MHz processor, 256MB of RAM, and a Compaq WL100 card used in *monitor* mode. In this mode, the card can listen to all data on a particular channel without being associated with any AP. We used the `linux-wlan` [2] driver as a loadable module for Linux 2.4.16 to set the card in the monitor mode. We used `ethereal` [1] to capture packets from the wireless interface.

D. *Accessing the WLAN*

As we used an isolated AP for our experiments, we had complete control and exclusive access to it. All stations were associated to the same AP in channel 6. Our logs showed that our stations were the only ones on the channel, and the only other traffic in the channel was from the AP (beacons, etc.) and constituted 0.9-1% of the total traffic in a typical experiment.

E. Traffic Generation

We used custom UDP-based source and sink programs to generate and transfer traffic from the stations to the sink on the other side of the AP. The sink listens to a port, receives packets, accounts them, and drops them. The source sends packets of certain size as fast as it can; the actual rate would depend on the kernel protocol stack bandwidth.

We did not use traditional traffic generation tools because such tools use expensive timestamping routines at the endpoints, and this slows down the stations too much. All our timestamping is done by the sniffer.

IV. Validating the Throughput Measurements

We need to ensure that our throughput measurements are not being subjected to limitations of the sniffer's ability to capture packets or the stations' ability to pump out data due to processor speed, card speed, NIC/host interface speed, OS, driver, and other such quirks. We describe a series of experiments designed to ensure this.

A. Validating the sniffer

The first thing we need to establish is that the sniffer captures most of the packets and is not a bottleneck. We sent packets at the speed described before and accounted for them at the sink, and we found that the sniffer had captured all of the packets received by the AP. This was possible as the sniffer was placed very close to the AP.

B. Validating the sending stations

In order to make sure that a source station's capacity to send was not the bottleneck, we added another station which also sent data at the maximum rate to the sink. We observed that while the throughput increases by 0.01 Mbps, the deviation of the throughput increases significantly by 0.3 Mbps. All stations irrespective of the processing power could operate individually at around 6.2 Mbps. Two or more of any combination of stations was more than enough to saturate the network. This confirms that the sending stations were not a bottleneck and exploited whatever the medium could offer.

C. Comparison with Ad-hoc mode

All data transfer in the infrastructure mode is through the AP. In order to quantify how much the processing capacity of an AP could be a bottleneck, we setup the two of the Linux laptops to operate in the *ad-hoc* mode, which employs basic DCF and stations do not use an AP. We observed a throughput with mean 6.3204 Mbps and deviation 0.02677 Mbps. This shows that the AP's processing capacity in bridging (i.e. converting packets from the 802.11 format to the Ethernet format) the packets to the 100Mbps ethernet interface was not a bottleneck.

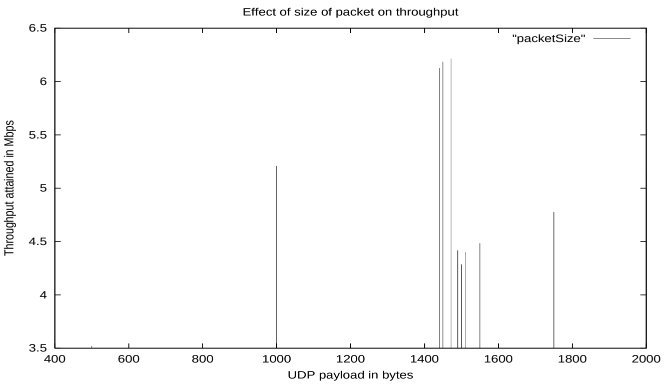

Fig. 7. Variation of throughput with UDP payload

D. Effect of packet size

We varied the packet size while computing the throughput of a single machine, ouzo, to see the effect on the throughput. Figure 7 shows the results on .

As expected, the throughput increases with packetsize. It first attains a maximum at around 1470 bytes (corresponding to MTU of 1500 bytes, IP header of 20 bytes, and UDP header of 8 bytes). It then falls off (due to the onset of fragmentation), and again increases (due to the increasing size of the fragment).

Therefore, all our experiments were conducted using packets with UDP payload 1472 bytes and actual size on the air of 1532 bytes.

V. Impact of Heteregenous Hardware

We experimented with stations having mixed processing speed as well as enabling/disabling the use of RTS/CTS. We observed massive difference in performance when we use cards with different chipsets with and without using RTS/CTS. We observed this when using three machines with the configurations.

- CISCO Aironet340 enabled with RTS/CTS on Pentium-III with FreeBSD.
- Lucent Orinoco Silver without RTS/CTS on Pentium-166 with Linux.
- Lucent Orinoco Silver without RTS/CTS on Pentium-166 with Linux.

All machines were programmed to send as fast as possible. Clearly, the traffic from the Pentium-III could not have been at a disadvantage vis-a-vis the other stations in terms of processing power. However, we see that the throughput drops dramatically for the Pentium-III station. Fig-

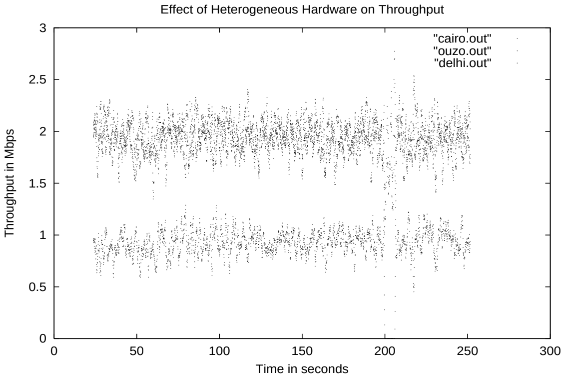

Fig. 8. Effect of Heterogeneous Hardware

ure 8 shows this behaviour where the Pentium-III machine sends at around 1.0 Mps. The average throughput for the Pentium-III machine is 0.9405 Mbps with a deviation of 0.148 Mbps, while those for the other two machines were 1.86462 Mbps with deviation 0.138157 Mps, and 1.9786 Mbps with deviation 0.315 Mbps. These were computed over K=500.

We repeated the same experiment with different OSs and encountered the same problem of CISCO Aironet340 cards suffering at the hands of other cards. Therefore, we precluded CISCO cards from all our experiments and used only Lucent Orinoco and Compaq WL110 both of which have an identical chipset.

This experiment suggests that the primary deciding factor in the throughput observed by a wireless end-user is not the processing speed of the machine used, but how well the 802.11b card implements the standard and interacts with other sources.

VI. CONCLUDING REMARKS

We obtained an empirical, i.e., measurement-based, characterization of the instantaneous throughput in 802.11b WLAN as a function of the number of competing stations. Our results confirm the general trends reported in [5]. The overall throughput decreases slightly as the number of stations increases. Also, the variance of the throughput available to a station increases significantly. Network performance of a wireless station is determined more by the wireless card implementation than its processing capacity.

Future directions of our work are measuring the short term fairness of 802.11b MAC, studying the effect of RTS/CTS on throughput, quantifying the effect of RTS/CTS on the maximal profile of 802.11b, and performance modelling of reliable transports like TCP over 802.11b WLANs by integrating empirical models with analytical methods.

VII. ACKNOWLEDGEMENTS

We thank Arunesh Mishra, Moustafa A. Youssef, and Bao Trinh of the Department of Computer Science for their invaluable help in procuring and configuring our experimental setup.

REFERENCES

[1] http://www.ethereal.com.

[2] http://www.linux-wlan.com.

[3] A. Balachandran, G. Voelker, P. Bahl, and V. Rangan. Characterizing user behavior and network performance in a public wireless LAN. In *Proceedings of ACM SIGMETRICS*, 2002.

[4] B. Bing. Measured performance of 802.11. In *Proceedings of Conference on Local Computer Networks*, pages 34–42, 1999.

[5] F.Cali, M. Conti, and E. Gregori. IEEE 802.11 wireless LAN: Capacity analysis and protocol enhancement. In *Proceedings of INFOCOM*, 1998.

[6] S. Khurana, A. Kahol, S. K. S. Gupta, and P. K. Srimani. Performance evaluation of distributed co-ordination function for IEEE 802.11 wireless LAN protocol in presence of mobile and hidden terminals. In *MASCOTS*, pages 40–47, 1999.

[7] C. E. Koksal, H. Kassab, and H. Balakrishnan. An analysis of short-term fairness in wireless media access protocols (poster). In *Measurement and Modeling of Computer Systems*, pages 118–119, 2000.

[8] A. Kopsel, J. Ebert, and A. Wolisz. A performance comparison of point and distributed coordination function of an IEEE 802.11 WLAN in the presence of real-time requirements. In *Proceedings of Seventh Int'l Workshop on Mobile Multimedia Communications, Oct 2000.*, 2000.

[9] The Institute of Electrical and Electronic Engineers Inc. IEEE 802.11, 1999 edition (ISO/IEC 8802-11:1999). *http://standards.ieee.org/getieee802/802.11.html*.

[10] C. T. Popescu and A. U. Shankar. Empirical TCP profiles and application. *7th Intl. Conf. on Network Protocols (ICNP'99)*, 1999.

[11] Y.C. Tay and K.C. Chua. A capacity analysis for the IEEE 802.11 mac protocol. *ACM/Baltzer Wireless Networks*, 7(2(March 2001)):159–171, 2001.

[12] J. Weinmiller, H. Woesner, and A. Wolisz. Analyzing and improving the IEEE 802.11-MAC protocol for wireless LANs. In *Proceedings of the 4th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '96)*, pages 200–206, 1996.

[13] G. Xylomenos and G. C. Polyzos. TCP and UDP performance over a wireless LAN. In *Proceedings of INFOCOM*, pages 439–446, 1999.