

Performance Analysis of a Wireless MAN

José Carlos Amaro, Rui Pedro Lopes

(rufino@ipb.pt, rlopes@ipb.pt)

Tel: +351 273 303 000; Fax: +351 273 313 051

Polytechnic Institute of Bragança – ESTiG, 5301-857 Bragança, Portugal

Keywords: Wireless Networks, IEEE802.11b, Performance Analysis, Digital City.

Abstract

A wireless network with metropolitan coverage was created to fulfill the data communication requirements for the Bragança Digital City program, part of the portuguese Digital Cities initiative. The wireless Metropolitan Area Network, based on IEEE 802.11b, still provides both a performant and an economical solution, tailored to the application scenario. Performance measurements have been done in order to apprehend the real network throughput under typical user applications and to have insight into other network characteristics, typical of the wireless medium. A comparison with a wired counterpart is also provided.

1 Introduction

The project leading to this paper – the Communitarian Data Network (CDN) [1] – is one of the main projects of the Bragança Digital City (BDC) program [2], part of the portuguese Digital Cities initiative¹ [3]. The project aims at providing a data communication backbone to the BDC partners, as well as a shared Internet access. In other words, the CDN is a Metropolitan Area Network (MAN), linking every partner of the BDC initiative in a kind of Intranet with a single common gateway to the Internet.

Several tests with different equipment brands revealed that the (then) recent IEEE 802.11b standard [4] matched the BDC requisites. IEEE 802.11b based equipment operates in the 2.4 GHz unlicensed band, using Direct Sequence Spread Spectrum (DSSS), and being capable of rates up to 11 Mbps.

Our main focus in this paper is to evaluate the performance of the wireless MAN, in operation since mid April 2000. We wanted to find out, on the field,

¹ The Bragança Digital City initiative, which hosts this network, was co-financed by the portuguese government, the FEDER European program and other project leading partners.

how wireless technology performed, since network throughput in real world conditions seldom matches nominal levels. Moreover, we wanted to answer questions like: How does topology affect throughput? What is the effective transmission rate when compared with wired networks? How does the packet size affects throughput? What is the maximum throughput and saturation level? Is there a block size for which the throughput is maximized? How much overhead does the communication protocol layer introduces?

The performance tests were done by using a FTP application and the `netpipe` tool [5], the former providing a (somewhat) qualitative approach by measuring the network performance from a user's application perspective and the latter giving us a more rigorous insight into other network parameters.

2 Performance Analysis

The network throughput in real world conditions seldom matches the one announced by the equipment manufacturer. Moreover, the *data throughput* is always smaller than the *link throughput* because of the various protocol overheads. In this paper we will restrict ourselves to the measurement of the *data throughput* in a small (but representative) sub-set of network paths. The wireless network throughput (*data throughput*, from now on) depends on several factors, including *signal quality*, *interferences*, *speed of the systems* and *type of data transfer* [6].

Signal quality and *interferences* were assumed as uncontrollable factors given that our wireless network operates on the unlicensed 2.4 GHz band and so it must accept possible outside influences. However, proper antenna alignment and placement has been taken into account in order to fit these parameters under correct operating limits. We have also conducted our tests in the same day, under the same weather conditions. By using the same hardware, software and firmware we minimized the variance of the *speed of the systems*. On selected situations, we tested the impact of the *type of data transfer*, because a bidirectional transfer places more load on the network, making possible to best evaluate the network throughput in more realistic scenarios.

2.1 Test-bed

The network has two cells with omni-directional antennae and a directional link between them using non-overlapping channels (Figure 1).

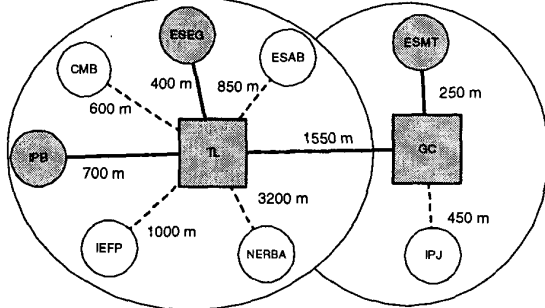


Figure 1 – Test-bed topology.

We have chosen four classes of network paths:

- 1) *Point – to – Point* (PtP) link between the cells (TL – GC),
- 2) *Point – to – Multipoint* (PtM) link inside the same cell (IPB – TL),
- 3) *Point – to – Multipoint – to – Point – to – Point – to – Multipoint – to – Point* (PtMtPtPtMtP) link across the cells (IPB – TL – GC – ESMT),
- 4) *Point-to-Multipoint-to-Point* (PtMtP) link inside the same cell (IPB – TL – ESEG).

Computing systems were placed near the wireless router in each testing point. The systems, running a fresh installation of Linux RedHat 6.2, were based on an Alpha 266 MHz processor with 128 Mb of RAM, 1.2 GB SCSI2 hard disk and a DEC 21140 NIC.

We started the performance analysis with the FTP application, which provides a qualitative approach more close to the user's perspective. These results were further completed with a more precise network performance evaluator tool.

2.2 FTP application

We choose FTP for two main reasons:

- 1) to check if new firmware offered noticeable performance improvements over previous FTP tests we had already done [7],
- 2) to see how close the network performed to the FTP throughput levels presented by the manufacturer [6].

We have followed the procedure given in [8], by using FTP in the following way: `ftp> put big_file /dev/null`. We choose `big_file` to be 1048576 bytes (1 Mb) size, perfectly enough to enforce fragmentation at the TCP level, allowing us to see the resulting effect on the network performance.

The first transfer time was always discarded to allow for the file to be placed on the sender's file-system cache. Also, by using `/dev/null` as the

destiny, we tried to avoid the disk writing latency of the file, at arrival.

Unidirectional file transfers were conducted ten times, separately, in both ways, and their average transfer time was used to get the throughput. For the bidirectional scenario, FTP transfers were "simultaneously" launched on both end-systems. Their clocks were previously synchronized with `rdate` and the transfers were scheduled via `crontab`. The transfer times collected at each end, T_x and T_y , were then used to obtain the bidirectional throughput via the formula:

$$\frac{1048576 + \min(T_x, T_y) \times 1048576 / \max(T_x, T_y)}{\min(T_x, T_y)} \times 8 \times 2^{-20}$$

As in the unidirectional case, we were interested in the average of all the bidirectional throughputs. Notice that the formula takes into account only the time period in which the transfer is occurring, simultaneously, between both ends. As we know, one transfer typically ends before the other and therefore the remaining time should not enter the calculation. Of course, the formula assumes that both transfers begin at roughly the same time and the data transfer rate is constant during the transfer (Figure 2).

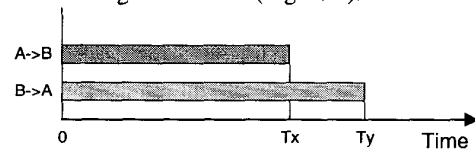


Figure 2 – Bidirectional transfer scenario.

The obtained values are presented in Figure 3.

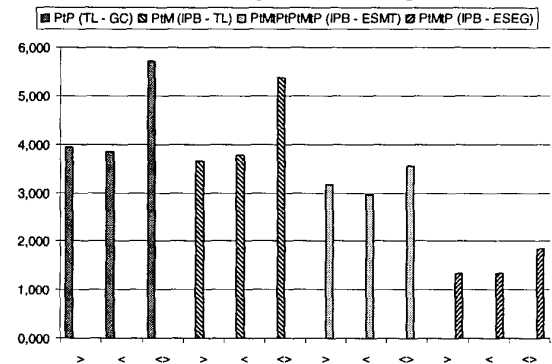


Figure 3 – FTP data throughput (Mbps).

1) *Point-to-Point* (PtP) link between the two cells (TL – GC). Recent firmware optimizations brought, accordingly to the manufacturer, a 20% increase in the *link throughput* for a point-to-point direct link, and so we were expecting this scenario to produce the best *data throughput* results, as indeed happened. Oddly, there is a slight advantage for the TL → GC direction, but further analysis of the S/N ratio at both routers did not reveal significant discrepancies and so the reason for the TL advantage remains to be identified. This scenario is also the best

in terms of bidirectional/unidirectional throughput ratio, where the bidirectional throughput is almost 50% better than the unidirectional case.

2) **Point-to-Multipoint (PtM) link inside the same cell (IPB – TL).** This scenario presents also a good performance, but smaller than the previous one. This may be due to differences at the signal to noise ratio caused by the omni directional antenna, which presents lesser gain.

3) **Point-to-Multipoint-to-Point-to-Point-to-Multipoint-to-Point (PtMtPtPtMtP) link across the two cells (IPB – TL – GC – ESMT).** This is the network path with the largest hop number (six). A packet traverses four routers, one for each end point and one for each cell center. Nonetheless, the performance still is reasonable (although slower than the previous scenarios), because this network path uses three non-overlapping radio channels, allowing for simultaneous transmissions in those channels (more packet fragments may be transmitted while the first fragments still have not arrived to the destination but are already traveling through a different radio link). Again, we find one particular direction (IPB → ESMT) gaining throughput over the reverse one, which may have its roots at the TL → GC advantage we already noticed. Notoriously, the bidirectional throughput falls almost to unidirectional levels, offering only around 12% increase over the latter.

4) **Point-to-Multipoint-to-Point (PtMtP) link inside the same cell.** With an end-to-end link between satellite systems within the same cell, the available bandwidth must be shared between three communicating entities. This is why the throughput falls well behind (more than 50%) the one from the point-to-multipoint scenario (the communication took place only between two entities) and even from the scenario 3 (although involving more entities, different radio channels provided separate transmissions). The bidirectional transfer still offers a better throughput.

When comparing the results of the present test round to a former study [7], we observe now a better throughput under common scenarios (scenario 3 and 4). Scenario 3 improvements range from 21.6% to a 26.7% increase in throughput for the unidirectional transfers, whereas the bidirectional throughput increases as much as 19.7%. Scenario 4 presents also better results, although more modest: around a 14% improvement for the unidirectional transfers and 12.7% for the bidirectional case. We should stress that our 1st test round [7] was based on heterogeneous computing systems, most of them even less (albeit not significantly) performant than those we have used in the present tests. Nevertheless, it seems that the firmware updates played well its role, allowing for performance gains, not far from those announced by the manufacturer (around 20% in average).

2.3 Performance evaluator tool

Although other tools could have been used, such as *netperf* [9], *netsec* [10] or *ttcp* [11], we choose *netpipe* [5] due to its protocol independence and variable time benchmark. We also had to do minor changes on *netpipe* source code to adapt the stop condition to networks with larger latency.

The gathered data enabled us to draw a graph with the throughput versus the block size, so that we could check the existence of a best block size (Figure 4). We also performed the tests on a direct 10 Mbps cross-link Ethernet connection between two hosts.

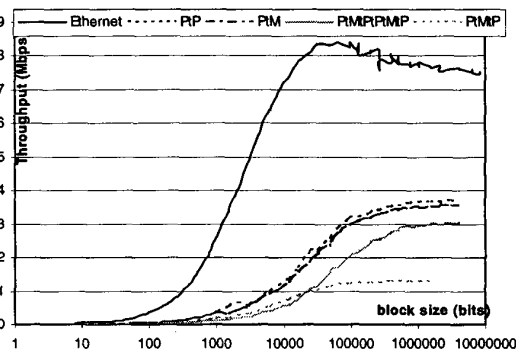


Figure 4 – Throughput evaluation.

It is easily pointed a peak value for Ethernet: 8.4 Mbps at a block size of 8195 bytes. For the wireless network, it seems that the larger is the packet the higher is the effective rate (until a specific limit, of course). The IEEE 802.11b collision avoidance mechanism based on the four-way handshake confirms this fact because the overhead introduced by the RTS/CTS and ACK diminishes for larger packets.

Maximum values for the throughput are very similar to those obtained from the FTP test (Table 1).

Network/Path	Throughput (Mbps)	Block size (Kbytes)	FTP (Mbps)
Ethernet	8.405061	8.002	
PtP	3.694231	512	3.94
PtM	3.558416	383.997	3.65
PtMtPtMtP	3.063639	512	3.16
PtMtP	1.283579	48	1.34

Table 1 – Maximum throughput for the available data.

A noticeable fact is that for relatively small blocks, the throughput on the PtMtP situation is higher than the PtMtPtPtMtP but, for larger blocks, the situation reverses. This is more clearly observed in the network signature graph (Figure 5).

The starting value at the times axis is the latency for the corresponding network/path, which is higher for the PtMtPtPtMtP situation. We can thus conclude that, although using overlapping transmission frequencies, the delay introduced by routers (hop

number) translates into a higher cost for small packets than the MAC protocol overhead.

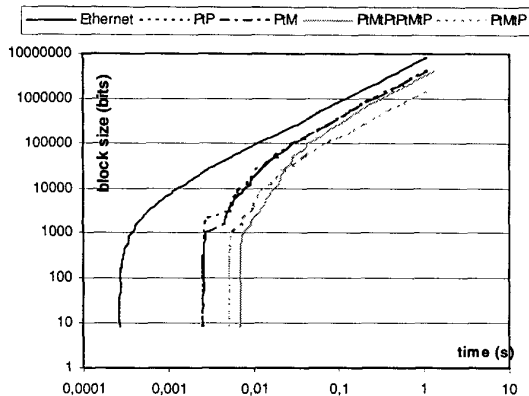


Figure 5 – Network signature graph.

The saturation graph (Figure 6) shows a plot of the block size versus the transfer time on a logarithmic scale, both for the X and Y axis. The knee of the curve reveals a saturation point, the point after which an increase in block size results in a near-linear increase in transfer time.

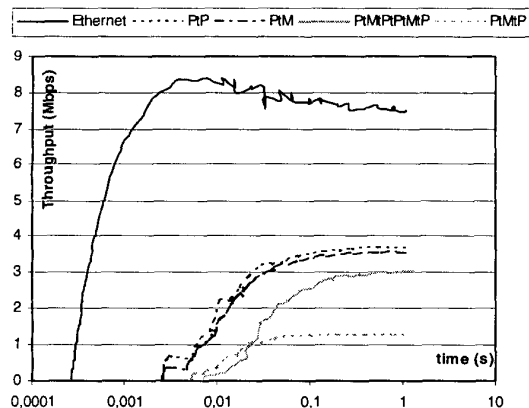


Figure 6 – Saturation graph.

The saturation point is more evident for Ethernet. For wireless, after a clear break around block sizes of 1000 bits, the curves present a higher slope, meaning that the improve on transmission time goes beyond the sizes for Ethernet.77

3 Conclusions

The CDN project provided us with an excellent opportunity to evaluate wireless technology performance. Although based on IEEE802.11b technology, which allows a raw 11 Mbps, we found out that the effective data transmission rates are far from this value. Also, when compared to 10 Mbps Ethernet, it presents higher latency (as expected) and

much lower throughput, due to the overhead introduced by physical and data link layers protocols.

Topological aspects also influence the throughput as well as the latency. Larger network paths (with more hops) increase the latency although the use of overlapping channels accomplish higher throughput for larger block sizes. The opposite occurs for single cell communication.

The CDN is operating since mid April 2000 thus completing over one year with up-time of almost 100%. The meteorological conditions (temperature ranges of -8°C to 38°C mixed with snow, rain, fog and strong wind) did not affect the network connectivity and we only had a down time of some minutes for firmware updates. The overall conclusion is that the technology is stable and cheap when compared to leased lines solutions.

Further performance analysis will cover other applications besides FTP, as well as more sophisticated traffic generation tools, such as MGEN [12] (for real-time multicast flows) and DBS [13] (for multipoint configurations). We expect these applications and tools to provide more insight on the network behavior under a variety of profiles and loads.

4 References

- [1] CDN (<http://rdc.bcd.pt>)
- [2] BDC (<http://www.braganca-digital.pt>)
- [3] Digital Cities (<http://www.mct.pt/CidadesDigitais>)
- [4] IEEE, "802.11b-1999 Supplement to Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific reqs-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher Speed Physical Layer (PHY) Extension in the 2.4 GHz band", 1999
- [5] Q. O. Snell, A. R. Mikler & J. L. Gustafson – "NetPIPE: A Network Protocol Independent Performance Evaluator", Ames Laboratory/Scalable Computing Lab, Ames, Iowa, April, 1996 (<http://www.scl.ameslab.gov/netpipe/paper/full.html>)
- [6] ORiNOCO Sales Bulletin 031/A – "Throughput Performance of an OR-system", June, 2000 (ftp://ftp.wavelan.com/pub/docs/orinoco/bulletin/sales/SB_031.pdf)
- [7] José Amaro, Rui Lopes – "Rede Digital Comunitária: uma Rede sem Fios Metropolitana", CRC2000, Viseu, Portugal, November, 2000.
- [8] Paul Gortmaker – "Linux Ethernet-Howto v2.7", May, 1999 (<http://www.ibiblio.org/mdw/HOWTO/Ethernet-HOWTO.html>).
- [9] Netperf (<http://www.netperf.org/>).
- [10] NetSpec (<http://www.itc.ukans.edu/netspec>).
- [11] tcp (<http://www.epm.ornl.gov/~batsell/NB.html>).
- [12] MGEN (<http://manimac.itd.nrl.navy.mil/MGEN/>)
- [13] DBS: Distributed Benchmark System (<http://shika.aist-nara.ac.jp/member/yukio-m/dba/index.html>)