Analytical Networking Project: Disrupting Wireless Networks

Martijn Meijer[1] Dennis Marinus[2]

02 July 2004

Abstract

In some situations, it is required to control the possible use of wireless networks at a geographical level. In these situations, wireless communication needs to be prevented through shielding, jamming or a combination of these two. In this report we will discuss the effects of shielding wireless networks and clarify some of the issues that remained unclear from previous research. Additionally, we present some findings of using different hardware to make measurements concerning wireless networks and show the behaviour of wireless networks when confronted with multiple streams.

[1] vlasbaard@os3.nl, 9544976

[2] dennis@os3.nl, 0353914

Table of contents

1 Introduction

Wireless communication is becoming more and more accessible over the last few years. There are some situations in which it would be better if wireless communication could be prevented. For instance in theatres, libraries and during exams.

There are roughly three stages in the prevention of wireless communication: detection, location and disruption.

When only detection of wireless communication is available, the only solution would be to ask all individuals to stop using wireless communication, without being able to target one individual directly. In this situation it is not possible to hold anybody responsible.

With location, the wireless communication device can be found. This usually means that the person responsible can be found and can be held accountable.

Both detection and location still require people to monitor if there is any wireless communication in use. With disruption, this is no longer needed as the use of wireless communication is no longer possible. Disruption can be done in two ways: shielding or jamming.

Shielding will offer protection from communication between parties on the inside and parties on the outside. Since some communication protocols (for instance GSM) rely on large installations on the outside to communicate, it is realistic to assume that, when shielded, communication between two parties on the inside is also disrupted.

However, since this is not the case for every wireless communication protocol, jamming wireless communication might be needed. Jamming floods the ether, the medium for wireless communication, with noise or garbage information so that communication becomes impossible. However, jamming will need to be contained within the area of application, for both practical and legal reasons. This might be achieved by combining jamming with shielding.

1.1 Problem definition

In this report, we will focus on wireless networks and try to answer the following questions:

1. How do various types of shielding impact the signal strength and transmission rates?
2. How does this behaviour change at different distances and across different floors?
3. In some previous research it appeared that shielded behaviour is sometimes better then unshielded.[3] The speculation was that an access point adapts to shielding situations. We'll try to reproduce the results and explain what was happening.

Following the measurements that we did, we decided to add the following:

4. What does the hardware we have allow us to measure?
5. How does a second wireless connection on the same access point influence connectivity?

We planned on doing some research on jamming as well, but unfortunately there was no time to gather the needed hardware for doing these experiments.

[3] This was a preliminary conclusion from a previous research on this subject by Claudia Eriksen et al.

2 Research setup

2.1 Available hardware

We had access to a couple of Wireless network interfaces (WLAN NICs), which featured the following chipsets:

Medion XG-701A:	Intersil/Conexant Prism Frisbee
Dell built-in:	Atheros AR5001X+
Elsa AirLancer MC-11:	Lucent/Agere ORiNOCO
Avaya Wireless world card gold:	Lucent/Agere ORiNOCO
SMC EZ Connect 2635W:	Admtek ADM8211
3com 3crshpw192:	Atmel AT76C502

There was a 3com OfficeConnect 3crwe454a72 802.11a/b/g access point available, some chicken wire, aluminium foil and a cardboard box.

2.2 Methodology

In order to measure the effect of the different shielding types, we needed to make fully enclosed setups. The chicken wire is strong enough to provide its own robustness, but the aluminium foil not. Therefore it was glued inside the cardboard box, which provided a nearly complete aluminium foil shielding. These enclosures are based on the principle of Faradays cage, which states that an electromagnetic signal cannot enter an area surrounded by a conducting material. Although this says nothing about signals leaving this area, it should prevent signals from entering and thus reaching the access point. The principle of Faradays cage also states that the effect of the cage is unaffected by whether the cage is grounded or not.

We decided to measure the signal and noise strength at three different points in the setup: one inside the shielding itself, in order to have a reference signal strength (*within*); one right outside the shielding, in order to determine how much the shielding was disrupting the signal (*local*) and one at a varying distance to determine what the effect of distance to the access point is when using different shielding setups (*remote*). Also, we measured transmission speed along with the signal strength at the remote location to see the effect of bad reception on transmission speeds.

2.3 Determining the hardware setup

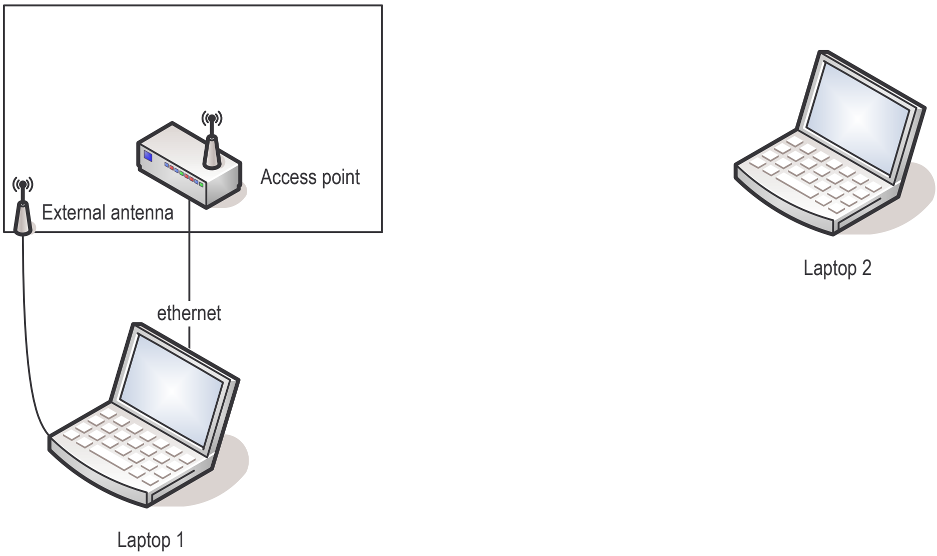
The access point was placed inside the shielding. The Prism NIC was chosen to be used inside the shielding because of its physical separation from the host. Since only the ORiNOCO cards support reading the noise level, we were unable to measure the noise level inside the shielding, only the signal level.

Only 802.11b was used. To achieve this, we forced the access point to only allow 802.11b connections.

For the signal and noise measurements we have used NetStumbler, and for protocol analysis WildPackets AiroPeek NX. We have chosen these software packages because NetStumbler was the only software that seemed to be able to measure the noise levels, and AiroPeek was the only software able to read low level packets like beacon messages. For some reason, NetStumbler gives different signal and noise readings when exporting to a text file. These values can be converted to regular dBm values by subtracting 149.[4] Of all the available NICs, AiroPeek was only able to measure signal strength for each packet if the Atheros NIC was used.

We have created the following setups in order to measure the signal strength, noise strength and transmission speed.

2.3.1 Problem: transfer & measurement on the same card



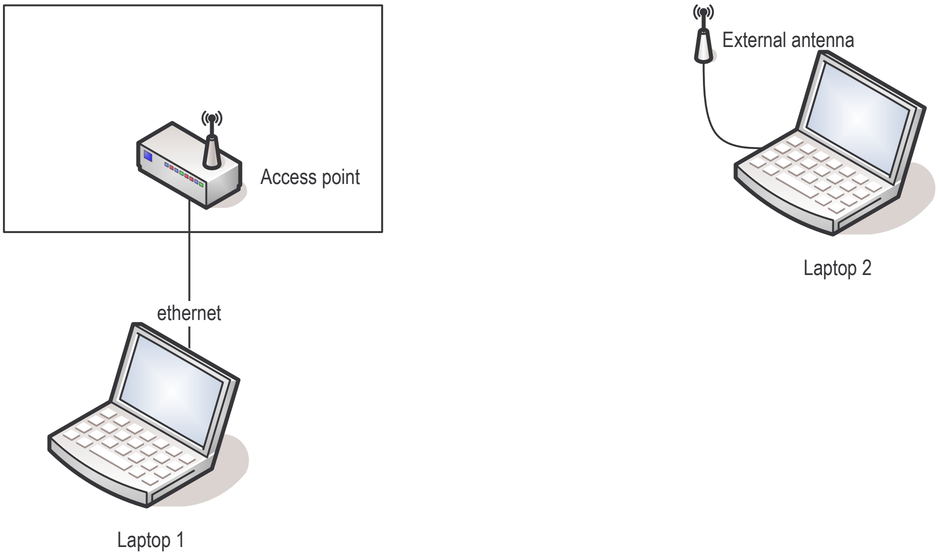
Network adapters per machine			
Laptop 1		**Laptop 2**	
Transmit/receive	Ethernet	Transmit/receive	ORiNOCO
Signal/noise measurement	ORiNOCO	Signal/noise measurement	
Signal measurement inside	Prism	Packet capture	Atheros

[4] According to the moderator of the NetStumber forum: http://www.netstumbler.org/showthread.php?t=9798 .

In this setup we used one card for both transmit/receive and signal/noise measurements in laptop 2. We've noticed that while measuring the signal/noise values, the transmit/receive rates dropped to about 50% compared to the transmit/receive rates measured without measuring the signal/noise level.

We believe that this is because NetStumbler continuously scans all non-overlapping channels (channel hopping) and therefore leaves less bandwidth available for other applications. There was no setting in NetStumbler to scan only one channel. Since NetStumbler is the only noise measurement software we found, we concluded that it would not be possible to use one card for both signal/noise measurements and data transmission.

2.3.2 Problem: USB transfer rate

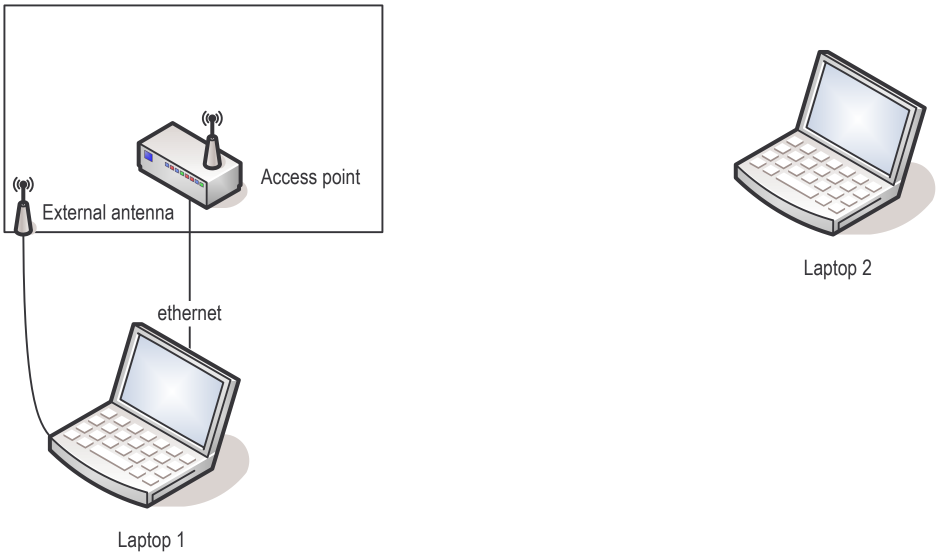


Network adapters per machine			
Laptop 1		**Laptop 2**	
Transmit/receive	Ethernet	Transmit/receive	Prism
Signal/noise measurement	ORiNOCO	Signal/noise measurement	ORiNOCO
		Packet capture	Atheros

In this setup we noticed that the transmit/receive rates while using the Prism USB adapter were about one third of what can be achieved when using an ORiNOCO based adapter.

We believe this is due to the USB1.1 controller in laptop 2, since the Prism adapter is a 54 Mbit USB2 device capable of much higher speeds (at USB 2).

2.3.3 Problem: 802.11b or not

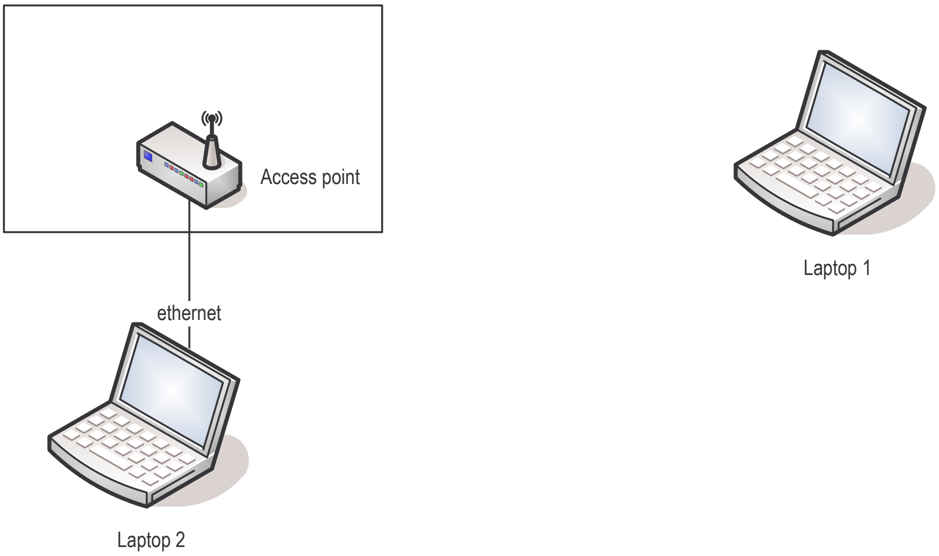


Network adapters per machine			
Laptop 1		**Laptop 2**	
Transmit/receive	Ethernet	Transmit/receive	Atheros
Signal/noise measurement	ORiNOCO	Signal/noise measurement	ORiNOCO
Signal measurement inside	Prism		

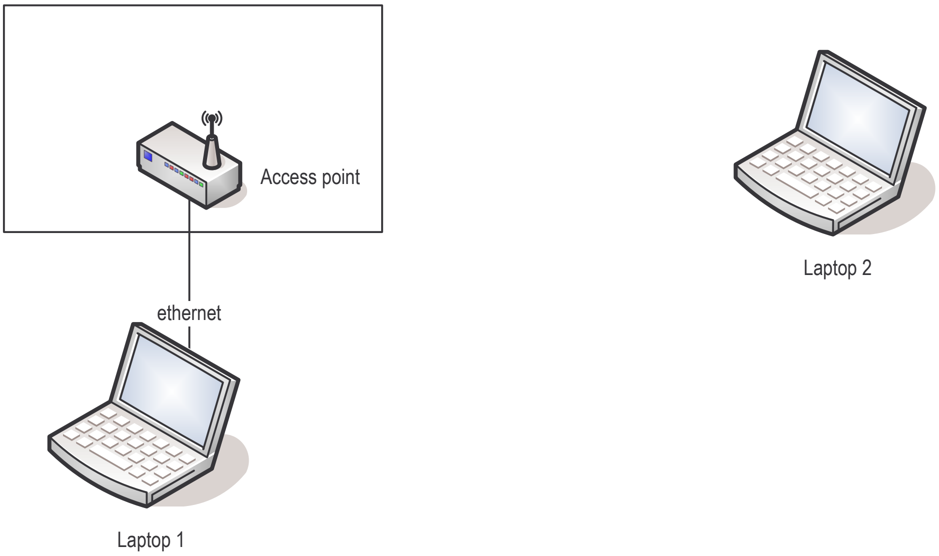
In this setup we noticed that when using the Atheros 54 Mbit adapter for transmit/receive, transfer rates were above what is theoretically possible using an 11 Mbit network: up to 14,75 Mbit/second.

Since the access point was configured to allow only 802.11b traffic, we can only conclude that this combination somehow bypasses these settings. The same setup in Linux provided us with regular speeds, so we suspect a driver problem.

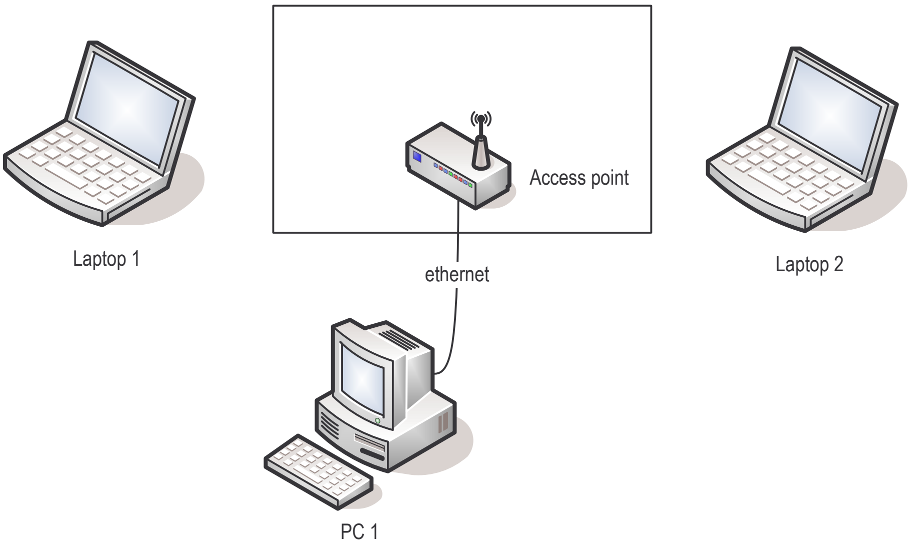
2.3.4 Problem: heavy traffic & noise measurement



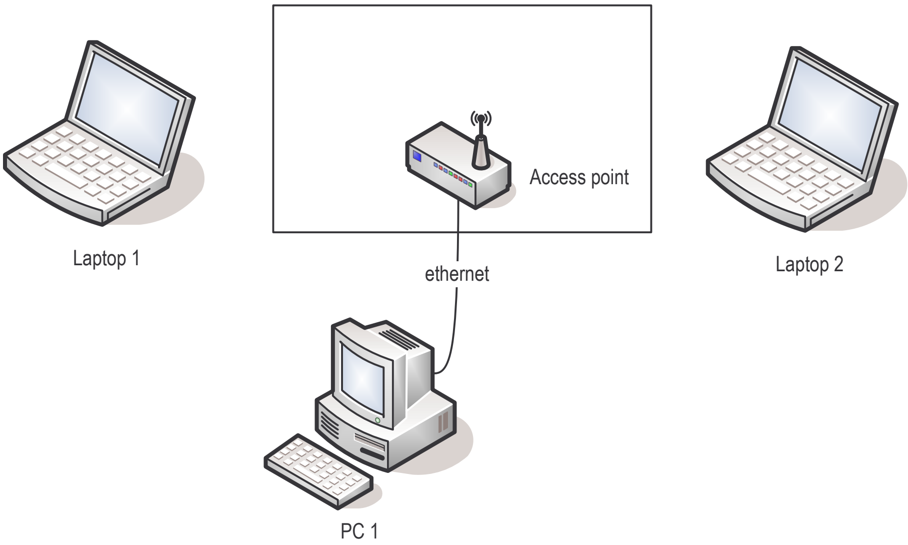
Network adapters per machine			
Laptop 1		**Laptop 2**	
Transmit/receive	Atmel	Transmit/receive	Ethernet
Signal/noise measurement	ORiNOCO	Signal/noise measurement	ORiNOCO
		Packet capture	Atheros



Network adapters per machine			
Laptop 1		**Laptop 2**	
Transmit/receive	Ethernet	Transmit/receive	ORiNOCO
		Signal/noise measurement	Atmel
		Packet capture	Atheros



Network adapters per machine			
Laptop 1		**Laptop 2**	
Transmit/receive	ORiNOCO	Signal/noise measurement	ORiNOCO
		Packet capture	Atheros

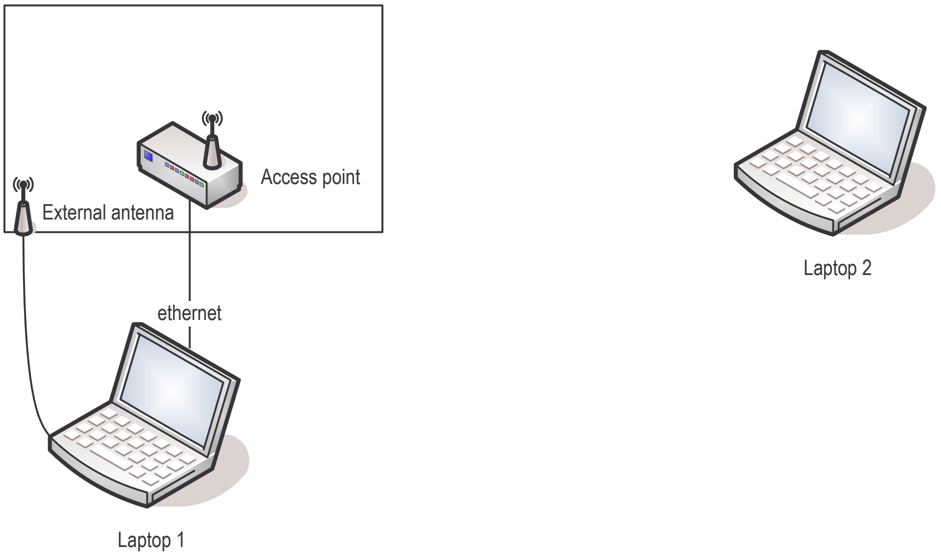


Network adapters per machine			
Laptop 1		**Laptop 2**	
Signal/noise measurement	ORiNOCO	Transmit/receive	ORiNOCO
		Packet capture	Atheros

In the last two setups we used an extra PC connected to the access point with an Ethernet adapter. This PC was only used for transmitting and receiving data and not for measurement purposes.

During these four setups, we noticed that during heavy traffic periods the signal/noise measurement software was unable to function. It would simply return blank values until the heavy traffic stopped. This proved to be repeatable behaviour, during any setup. Therefore we can only conclude that this is either a software bug, or a bug in all hardware we have used, which seems unlikely.

2.3.5 Final setup



Network adapters per machine			
Laptop 1		**Laptop 2**	
Transmit/receive	Ethernet	Transmit/receive	ORiNOCO
Signal/noise measurement	ORiNOCO	Signal/noise measurement	
Signal measurement inside	Prism	Packet capture	Atheros

2.4 Procedure

For each measurement, we started with a measurement of the noise strength next to the shielding and at the measuring distance, as well as the measurements of the signal next to, and inside, the shielding. NetStumbler only measures the signal strength of packets coming from the access point, not the strength of packets from the NIC. In the resulting graphs these variables are averaged and drawn as constants.

We assume that the noise is not dependant on the amount of traffic, and fairly constant over time. Therefore we do not consider this a problem. The local signal strengths might be problematic, but since only the Atheros allowed us to measure during transmission, which was not available at the local machine, this problem could not be resolved.

For each of these four measurable values, we have determined the average and the standard deviation. We have used

$$error = \frac{SD}{\sqrt{n}}$$

as an approximated error, where *SD* is the standard deviation and *n* is the number of measurements. See appendix B for a complete overview of the averages and standard deviations.

After these initial measurements, the transmission rate, as well as the signal strength at the remote machine was measured. This was done two times: once from the access point to the remote machine, and once the opposite way. These are the data that will be presented in the next chapter. To be able to measure the signal strength of each packet, AiroPeek was used. As we've already seen, this unfortunately limits our choices in hardware.

For transfer rate measurements we have used TTCP, of which a Windows version is made available by Microsoft. We used the version that came with Windows Server 2003. A downloadable version comes with the IPv6 Technology Preview for Windows 2000.[5] We have used the following parameters:

- sender: `ttcp -t -l5000000 -n1 <host>`
- receiver: `ttcp -r -l5000000 -n1`

With these parameters, TTCP sends 1 TCP stream of 5000000 bytes. We tried to use UDP to prevent protocol interference, but we encountered various problems doing so.

1. packet loss: when sending large amounts of data from the wireless machine we experienced very inconsistent packet loss rates.
2. link speed difference: when sending large amounts of data from the Ethernet connected machine to the wireless machine we experienced roughly 90% packet loss. We believe this is due to receive buffers overflowing in the access point.

The measuring distances were determined at a later time, since they depended on the surroundings in which the measurements would be made. We decided on three measuring distances, at half a meter, at ten meters and at twenty-five meters.

2.4.1 Addition to the procedure

After not being able to find interesting results with these 3 shielding methods at 3 distances because the signal was not getting weak enough, we did some additional testing around corners and on different floors.

We placed the access point in large, heavy metal machinery that was around the corner from our previous measuring location. We measured the signal around the next corner, so that there was a closed room between the

[5] http://msdn.microsoft.com/downloads/sdks/platform/tpipv6/readme.asp

measuring point and the access point. However, since placing the laptop which was connected (wired) to the access point in or near the machinery was impractical, we have no local measurements for these additional tests.

Since that still gave no disruption of the signal, we measured at a lower floor and behind some heating pipes. These situations provided us with data about a nearly disrupted network.

Also, we did two tests with background traffic. The first was measured while another machine was continuously transmitting UDP packets at maximum speed, and the second was only a short UDP burst halfway through our measurement. These measurements where made to see how much other 801.11b traffic will influence the transfer speed.

3 Results

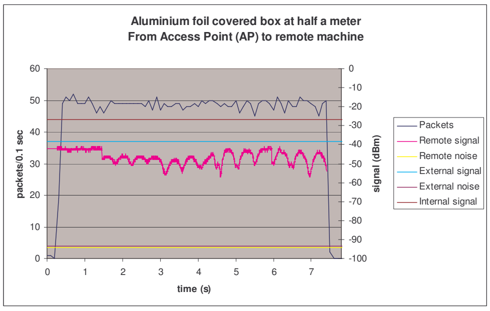
In the following graphs, the noise at both local (external) and remote machines, as well as local internal and external signal strength, are drawn as constants. These are for reference only, since they were actually measured at a different time then the two non-constant variables, as is described in the procedure. For the transfer rate one needs to look at the left Y-axis, and for the various signal strengths at the right Y-axis.

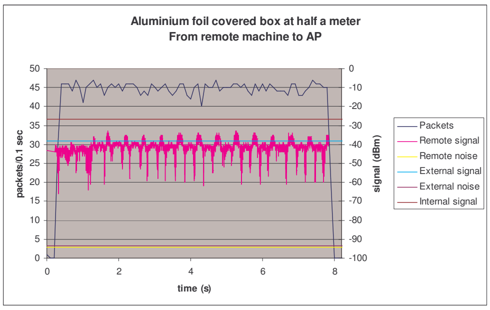
Please keep in mind that we are interested in two things. Firstly, a connection between the SNR (which equals *signal / noise*), which may be observed by looking at the variables within a single graph. Secondly, the influence of the different methods of shielding and measuring distances on the signal strength, which is shown by the differences between the different graphs.

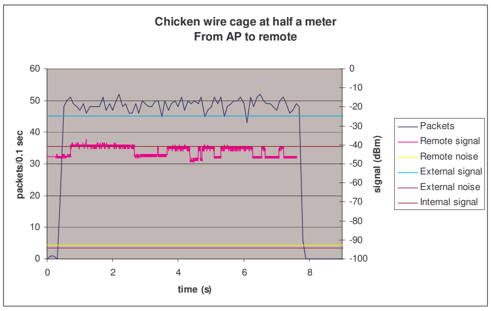
These results will be discussed in the next chapter.

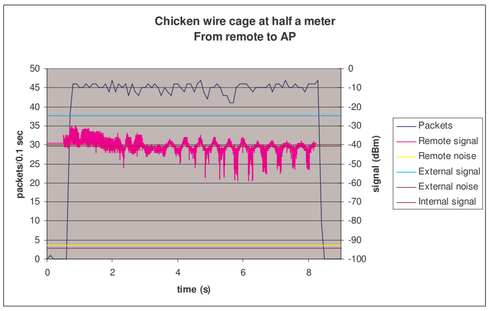
Since the results in the first 9 setups don't show interesting results, we've decided to make the 18 graphs that represent them smaller than the later, more interesting ones.

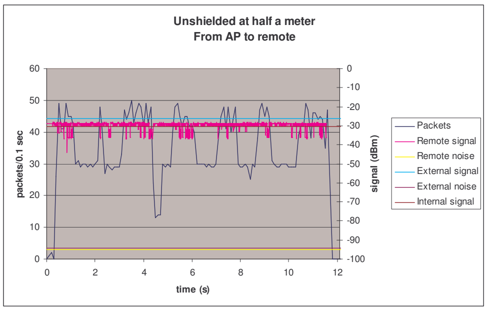
3.1 Various shielding types, half a meter

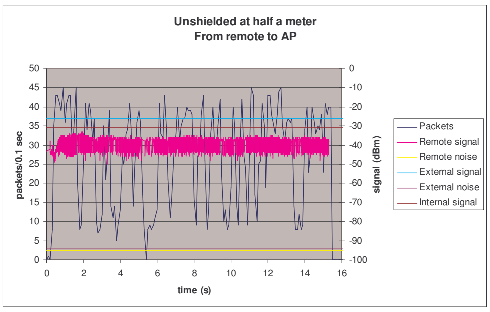




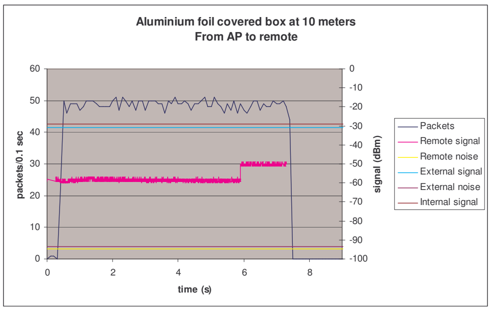


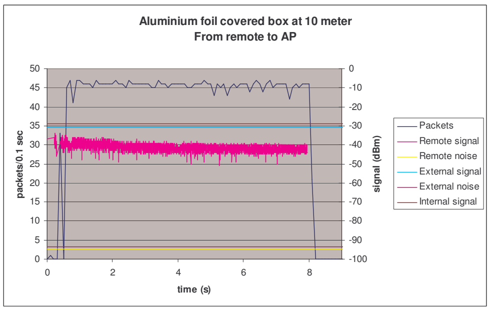


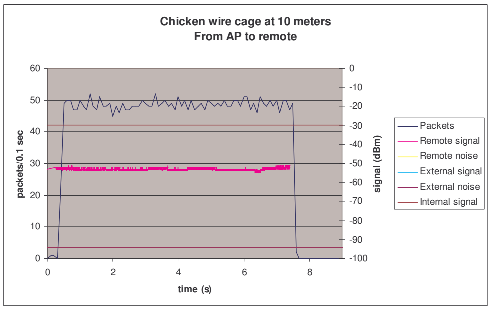


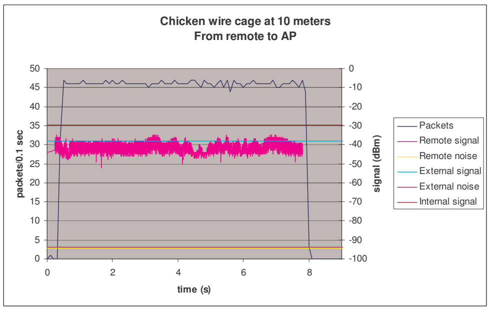


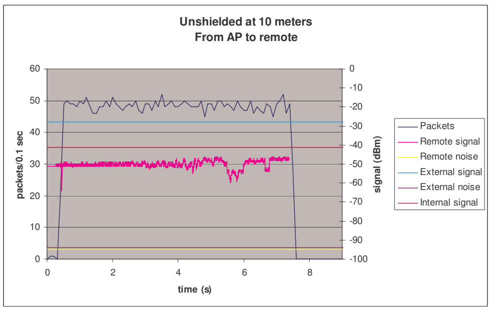
3.2 Various shielding types, 10 meters

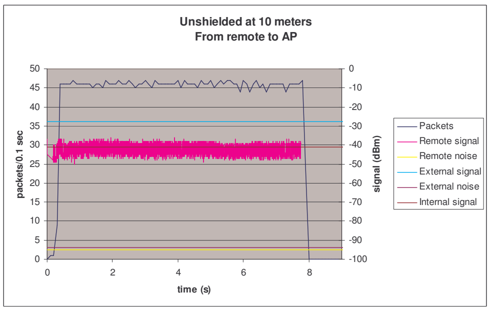




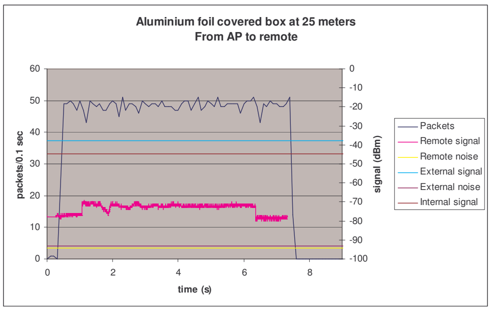


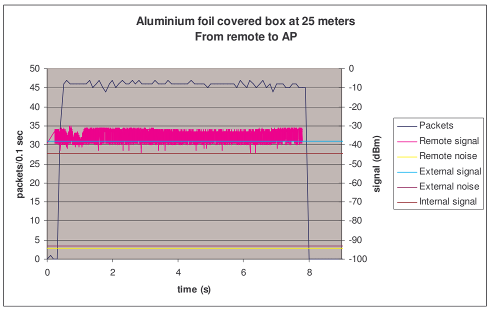


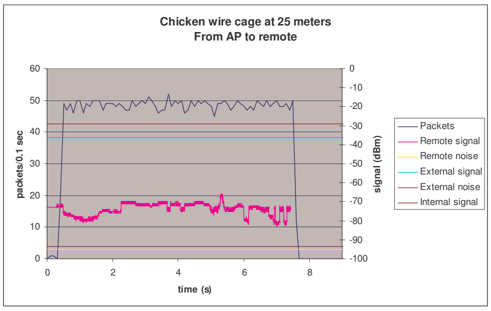


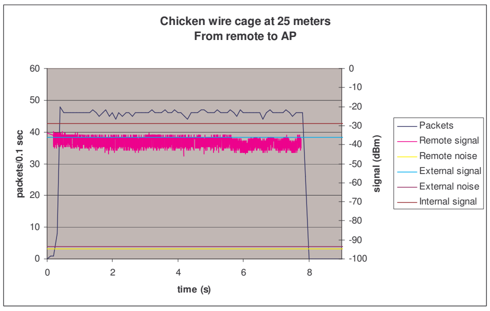


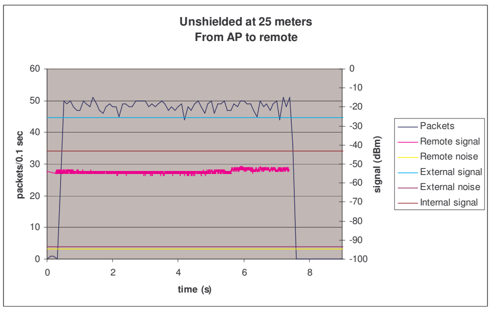
3.3 Various shielding types, 25 meters

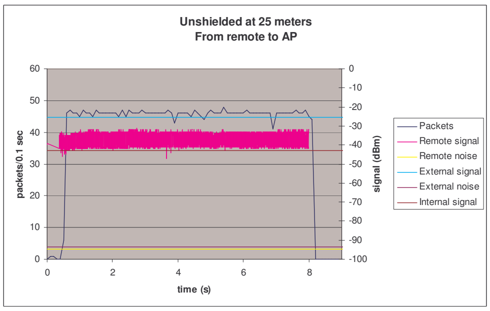




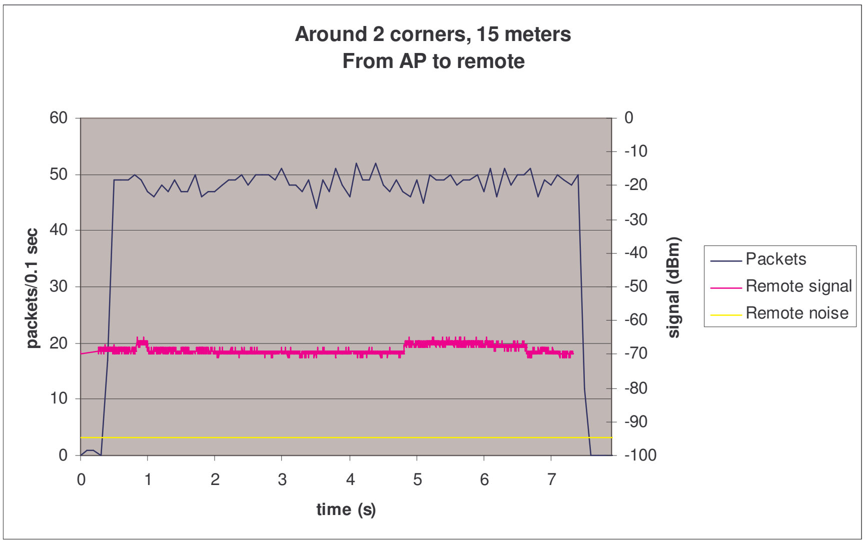


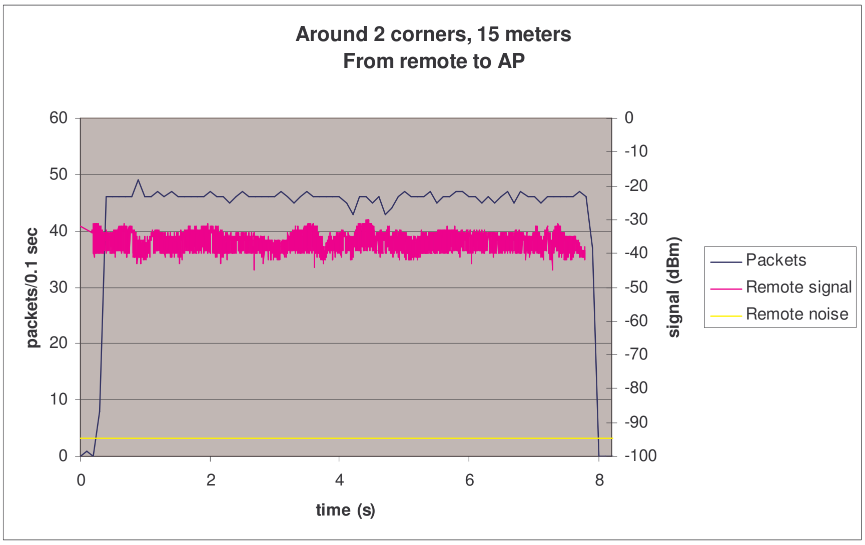




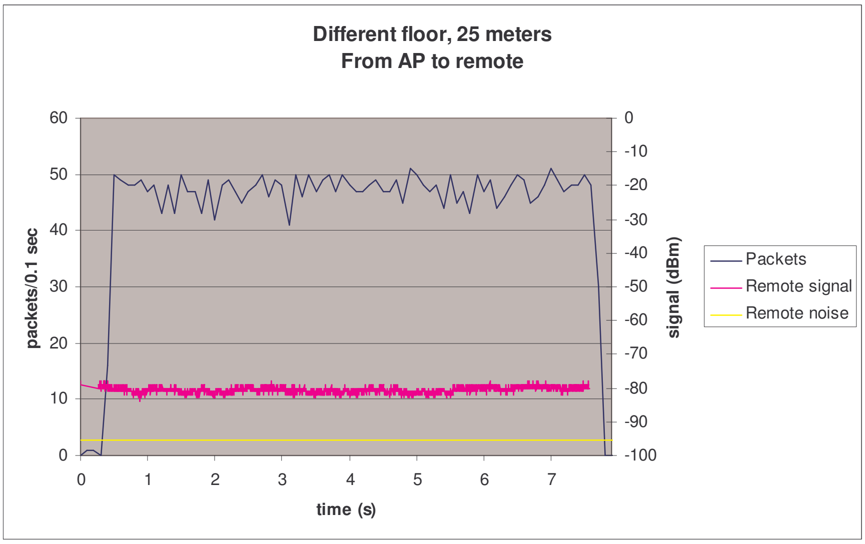


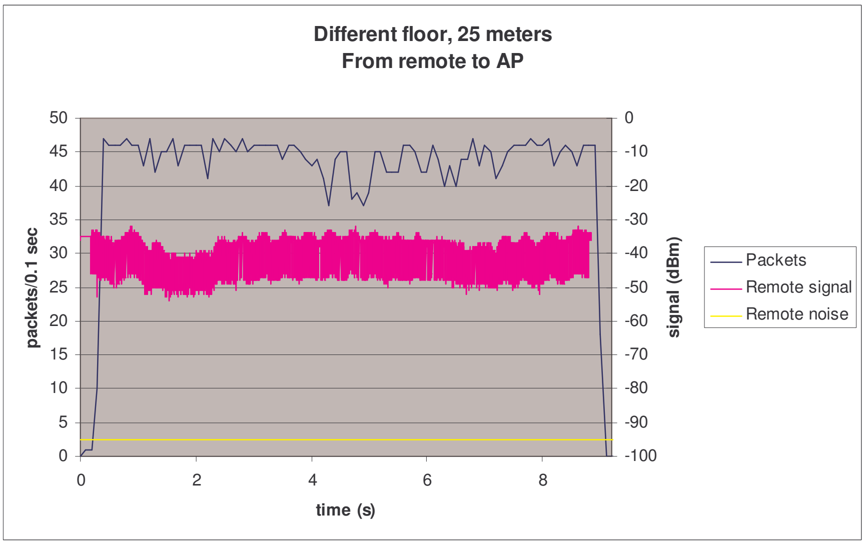
3.4 Around corners

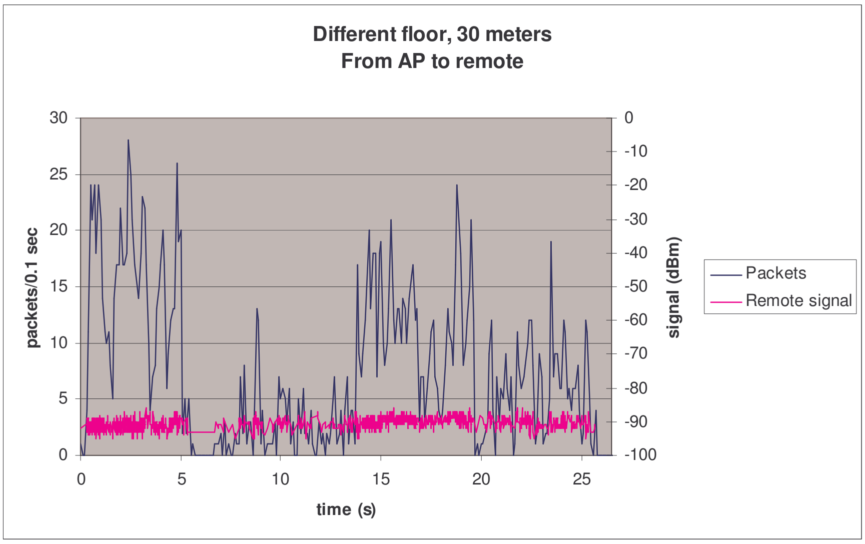




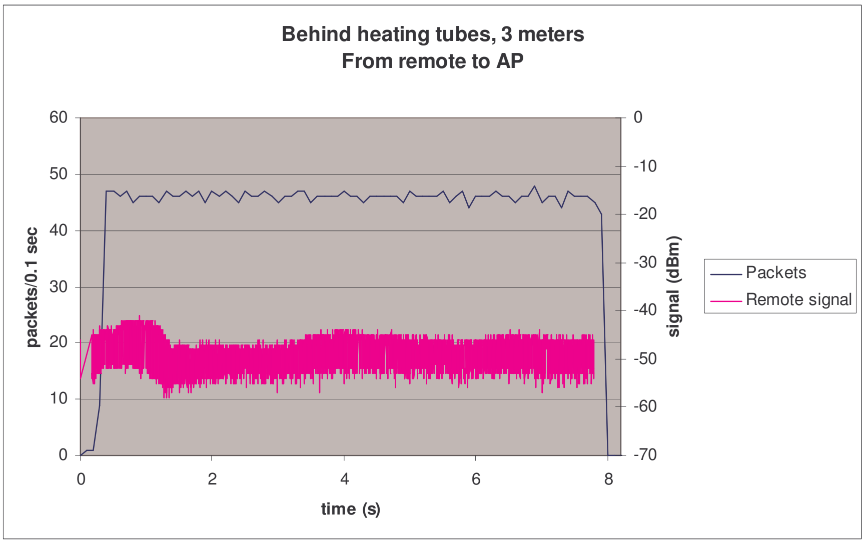
3.5 Different floor

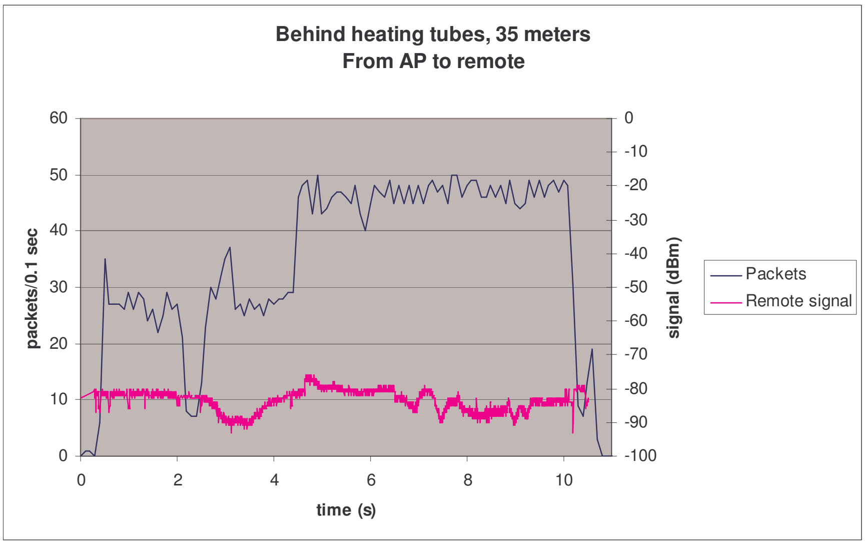


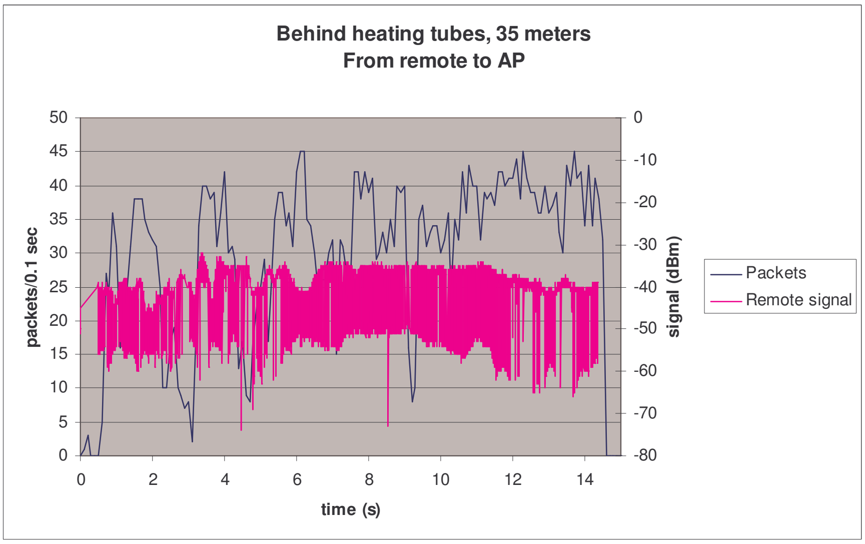



Different floor, 30 meters
From AP to remote
packets/0.1 sec
30
25
20
15
10
5
0
0
5
10
15
20
25
time (s)
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
signal (dBm)
Packets
Remote signal

3.6 Behind heating tubes



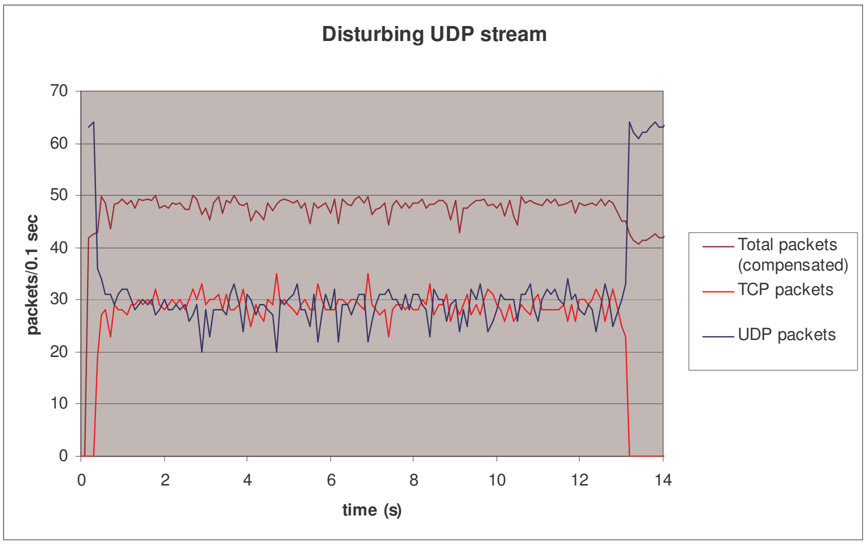


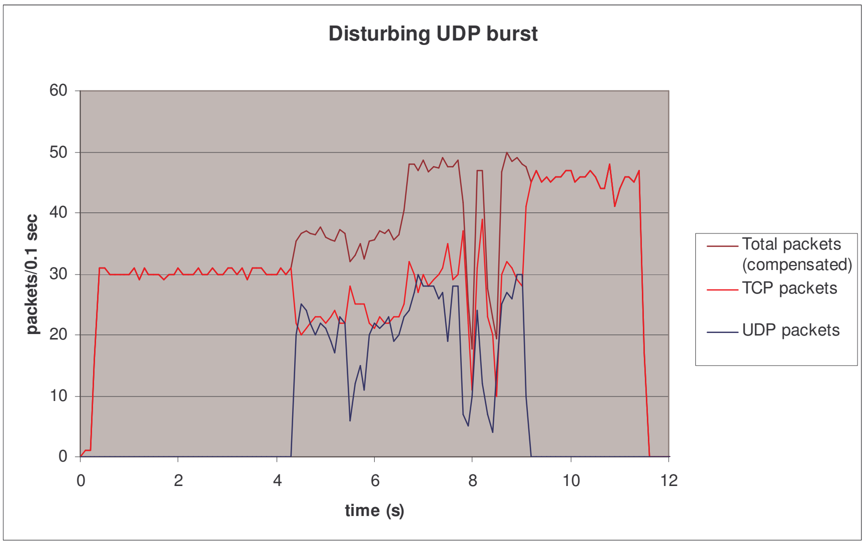

Behind heating tubes, 35 meters
From remote to AP
packets/0.1 sec
50
45
40
35
30
25
20
15
10
5
0
signal (dBm)
0
-10
-20
-30
-40
-50
-60
-70
-80
0
2
4
6
8
10
12
14
time (s)
Packets
Remote signal

3.7 Interfering traffic

For the final two graphs, things are a bit different. Since they concern a second stream, noise levels are not as important. They do, however, clutter up the graph and make it unclear, so we removed them. Both graphs are with traffic from the remote machine to the access point.

In the graphs we have plotted both the amount of TCP packets, as well as the amount of UDP packets of the second connection. For the total amount of traffic however, we had compensate since a different packet size was used for the UDP traffic (1036 bytes in stead of 1536 bytes for TCP). We compensated this difference in the number of UDP packets before adding them to the total packets.





4 Discussion

From most graphs, it can be derived that the signal strength within the shielding is strongest, followed by the external and lastly by the remote signal. This is what one would expect, even though in the half meter-cases the remote machine was not much further away then the local machine.

It is clear that the signal that is received from the access point, especially in the later examples, is weaker then the signal send by the remote machine. Since the signal strength is being measured at the remote machine, this was to be expected.

The remote signal strength measured when sending from the remote machine, where the signal is generated by the NIC, is very much the same in all measurements. This is because the transmitting NIC and the measuring NIC, while both remote, are at the same distance in all measurements. This shows that the power used to send a packet is not dependant on the situation. It is always sending at full power.

The noise levels play no role of significance in the first 9 experiments. This is why we needed to expand the procedure.

In some of the later experiments (mostly the 30 meters on a different floor and the 35 meters behind the heating tubes), it can be seen that the transfer rates become more erratic when the signal levels come close to the noise levels. Some of the high peaks in those experiments still reach the optimal speeds. In these two setups, NetStumbler did not detect the noise anymore. Apparently it couldn't find the access point any longer in the time it is listening to a single channel during the channel hopping.

When comparing the different graphs together, one that immediately catches the eye is the unshielded at half a meter. The transfer rate is very erratic, while the signal strength is quite stable and very high. First we were thinking that this might be a timing problem, resulting of a minimum distance in the 801.11b standard. However, since this problem does not arise at the same distance, but shielded, this cannot be the case. We assume that the signal is actually to strong for the sensor within either the access point or the NIC.

Also, while there is loss in signal strength to be seen because of the shielding provided by the aluminium foil covered box and the chicken wire cage, they do not seem to affect the transmission speed at the distances we measured.

When measuring the 30 meters distance on a different floor, we observed when moving just 1 meter further, the connection would drop. This while the transfer rate was about half of what it was other setups. The area where there was a non-full speed connection was very narrow.

With the disturbing UDP stream the total amount of packets being sent stays roughly the same as with a single connection. Interestingly, the amount of packets sent is more of less equal for both streams. This indicated that the traffic is divided among the two streams according to the number of packets, not according to the amount of bytes.

For the burst measurement, we have no explanation for the total amount of traffic being lower in the first then in the second half. Unfortunately we didn't notice until writing the report, and are therefore not able to redo the measurement.

5 Conclusion

In this report we have tried to answer five questions.

Firstly, we may conclude that chicken wire and aluminium foil are not very suited to shield wireless network signals. They have some influence on the reach of access points, but for complete shielding it is simply not effective enough.

The same is true for different distances. The signal gets weaker, but not by much. However, if measured at a different floor, where the only connection between the floors is through the staircase, the signal is affected significantly, the transmission rate drops and after only a few more meters the signal is lost.

An unshielded access point very close will not give a good signal. When used unshielded, the signal seems to drown itself at distances smaller than 2 meters. This might explain the strange results in previous work where a shielded setup resulted in higher transfer rates than in unshielded setup.

There are big differences in what details wireless network cards allow you to measure. While all used cards were able to measure signal strength, only the ones with an ORiNOCO chipset allowed us to measure noise levels. Only the Atheros could be used to capture 802.11b packets at such a low level that we were able to look at the beacon messages.

If there are multiple wireless streams, the number of packets is divided equally. The drop in transmission speed is therefore dependant on the size of the packets.

There are still quite a few questions about wireless network left. Since chicken wire and aluminium foil are insufficient for shielding, what material will be better suited for this purpose? Is it, in practice, possible to jam a wireless network, and can the jamming be contained? If so, it will enable network administrators to control the use of their wireless networks more precisely.

Appendix A: project proposal

Analytical Networking Project: Disrupting Wireless Networks project proposal

Martijn Meijer[6] Dennis Marinus[7]

14th June 2004

Introduction Wireless networks (WLANs) are becoming common in the networking world. During our Analytical Networking Project we therefore decided to answer some of the more interesting research questions that come to mind:

- How do various types of shielding impact the signal strength and transmission rates?
- How does using a WLAN inside confined quarters relate to open field usage?
- How do different weather types influence WLAN behaviour?
- In what ways is it possible to disrupt a WLAN (specifically, 802.11b) at the data link layer?
- Is it possible to jam WLANs at the physical layer using commonly available WLAN hardware, and how?
- How do access points deal with these problems?

Previous work Some previous experiments concerning wireless networks were done by Claudia Eriksen. Among the experiments she did were:

- Experiments with different types of shielding (chicken wire, aluminium foil, solid metal)
- Signal measurements with varying distances

The (preliminary) results of these experiments were non-conclusive. The signal strength and/or the transfer rate appeared not to be negatively affected by the shielding.

[6] vlasbaard@os3.nl, 9544976

[7] dennis@os3.nl, 0353914

Requirements In order to be able to do the experiments we have determined that we will need the following:

- 2 laptops with WLAN interfaces, preferably with working batteries and dual-boot/Knoppix-STD
- Confined quarters and an open field where the experiments can be done without bothering others (3.U01 vs. park?)
- Access point
- Various small things like network cables and extension cords
- Permission to use commercial software (if needed)

For some of the experiments we have specific requirements:

- Shielding experiments:
  - Chicken wire, aluminium foil, solid metal, shielded room, etc.
  - PDA or Airmagnet for measuring signal strength inside the shielded environment
- Unintentional jamming in the real world:
  - (Old) Cordless phone
  - Microwave oven
- Targeted jamming:
  - Access point that may be taken apart
  - Permission to jam (legal issues?)
- Data link layer disruptions:
  - WLAN card able to send rogue packets

Some of these requirements can be met by using our own equipment. We have laptops, we have an WLAN card that might be usable and the cordless phone can be arranged, but it will take some time to ship it from the Netherlands.

Software We have tried the following software:

- Windows freeware and/or open source:
  - NetStumbler
  - Ethereal with RawPacket NDIS protocol driver
- Windows commercial:
  - Wildpackets Aeropeek
  - McAfee Sniffer Wireless (sort of works)
- iPaq:
  - WiFiFoFum
- Linux:
  - Airsnort
  - Airjack
  - Kismet
  - Wavemon (sort of works)

The following software we tried appears to be broken:

- vxScanner
- Wellenreiter
- wscan

And for the following software we don't have the correct hardware:

- Ministumbler
- Pocket Warrior
- Airmagnet

References

[1] Gregory Rehm, *Homebrew Antenna Shootout*, 2nd November 2003.
`http://www.turnpoint.net/wireless/has.html`

[2] Andy Dornan, *Wireless LAN Analyzers: The Ultimate Hacking Tools?*, 5th March 2003.
`http://www.networkmagazine.com/article/NMG20030305S0001`

[3] Edouard Lafargue, *Wireless Network Audits using Open Source tools*, 2003.
`http://www.sans.org/rr/papers/5/1235.pdf`

[4] Jacco Tunnissen, *Wireless LAN Security & Wardriving (802.11)*, 8th June 2004.
`http://www.wardrive.net/wardriving/tools/`

[5] *WirelessSniffer*, 24th May 2004.
`http://www.personaltelco.net/index.cgi/WirelessSniffer`

[6] *Concept question on wireless jamming*, 14th May 2004.
`http://www.dslreports.com/forum/remark,10218437?mode=flat`

[7] *Windows Packet Capture Library*, 8th March 2002.
`http://netgroup-mirror.ethereal.com/winpcap/misc/links.htm`

Appendix B: noise data

dstance	shielding	time	local signal				local noise				within signal				remote noise			
			n	av	sd	ae	n	av	sd	ae	n	av	sd	ae	n	av	sd	ae
0,5	box	23	24	110,83	2,2	0,45	24	55,62	1,17	0,24	23	122,26	10,09	2,10	24	54,75	0,74	0,15
0,5	cage	22	24	124,17	1,99	0,41	24	54,71	1,16	0,24	23	108,3	7,96	1,66	23	56,08	1,12	0,23
0,5	unshielded	29	25	122,85	3,56	0,71	25	54,67	1,07	0,21	30	118,47	7,85	1,43	31	54,06	0,89	0,16
10	Box	20	22	118,09	0,81	0,17	22	55,5	0,91	0,19	21	120,14	7,88	1,72	20	54,25	1,12	0,25
10	cage	22	22	110,96	5,49	1,17	22	54,91	0,97	0,21	23	119,22	8,9	1,86	22	54,55	1,47	0,31
10	unshielded	23	16	121,31	2,24	0,56	16	55,13	0,62	0,16	24	107,75	7,32	1,49	23	53,96	0,98	0,20
25	box	27	28	111,07	1,25	0,24	28	55,93	0,81	0,15	28	104,5	7,09	1,34	27	54,7	0,72	0,14
25	cage	21	22	112,77	2,74	0,58	22	55,32	1,13	0,24	22	120,14	5,89	1,26	22	54,14	1,17	0,25
25	unshielded	29	26	123,58	1,55	0,30	26	55,56	0,8	0,16	30	105,87	8,12	1,48	29	54,48	1,18	0,22
15	cornered	23													23	54,3	0,93	0,19
25	different floor	51													53	53,77	1,03	0,14
30	heating	33													34	52,09	1,88	0,32

distance Distance between the access point and the remote machine, in meters
shielding Type of shielding used
time The amount of time measurements were made, in seconds
n The number of successful measurements
av The average of the measurements
sd The standard deviation of the measurements
ae The estimated error, as described in the measuring procedure

Shielding types:
box Aluminium covered box
cage Chicken wire cage
unshielded No shielding
cornered Access point in a metal bin, remote machine around two corners
different floor Access point in a metal bin, remote machine on a different floor
heating Access point in a metal bin, remote machine behind large metal pipes