ITRAINONLINE MMTK

WIRELESS TROUBLESHOOTING HANDOUT

Developed by: Alberto Escudero Pascual/ IT +46

Table of Contents

1. About this document

These materials are part of the ItrainOnline Multimedia Training Kit (MMTK). The MMTK provides an integrated set of multimedia training materials and resources to support community media, community multimedia centres, telecentres, and other initiatives using information and communications technologies (ICTs) to empower communities and support development work.

1.1 Copyright information

This unit is made available under the Creative Commons Attribution-NonCommercial-ShareAlike 2.5 Sweden. To find out how you may use these materials please read the copyright statement included with this unit or see http://creativecommons.org/licenses/by-nc-sa/2.5/se/.

1.2 Degree of Difficulty

The degree of difficulty of this unit is “Medium” with some additional “Advanced” parts.
All “Advanced” sections are marked with a red frame to make the reader aware of a higher degree of difficulty.

---

2. Introduction

This unit proposes a methodological approach to the troubleshooting of wireless networks. The main problem of troubleshooting any communication network is to identify what is going on when things "go wrong". Rather than rebooting everything that is attached to a power cord or blaming the weather conditions, we propose to follow the OSI model to try to find out the cause of the problem.

The OSI *(Open Systems Interconnection)* Reference Model, created by ISO (International Standards Organization), is an abstract description for computer network (communication) protocol design. The model splits different communication functions into seven different layers that can work independent of each other.

The Internet protocol design follows a similar structure to the OSI model. Each protocol *layer* only uses the functionality of the layer below and provides functionality only to layers above.
This structure is of great help when trying to troubleshoot a problem as it helps us to isolate where the problem is located. The first thing that we always need to do when things go wrong is to try to identify in which "layer" the problem appears and which layer that is the cause of the problem.

For example, users will always complain that an application "x" is not working! (OSI Layer 7) but the cause of the problem can be in any of the layers below. For example, it can be related to lack of radio signal (OSI Layer 1) or lack of IP address (OSI Layer 3).

Layer	***OSI***	***TCP/IP***
7	Application	Application
6	Presentation	
5	Session	Transport (TCP)
4	Transport	
3	Network	Network (IP)
2	Data Link	Media Access Control
1	Physical	

Table 1: OSI model versus TCP/IP protocol suite.

3. Methodology

Depending on the information that we have in advance we can take two approaches:

3.1 Top-down troubleshooting

When there is a "problem", top-down troubleshooting starts by checking the application's configuration settings and finishes by checking whether there is wireless interference or a low signal level in the radio receiver.

3.2 Middle-top, middle-down troubleshooting

When there is a "problem", this approach starts by checking whether there is IP connectivity to the requested service or the border router, and depending on the result attempts to troubleshoot the layers below or above.

This approach is the most popular, ping <the service>, ping <the router>.

Unfortunately most of the time this only helps to identify who to blame, rather than troubleshooting the actual problem. If “ping” to the border router fails then we can blame the wireless carrier, if “ping” to the service fails then we can blame the international carrier. If none of them fail then we blame the user or the operative system.

Whatever approach we take to troubleshooting a problem it is important that we are familiar with the tools that are appropriate when analysing each of the functional layers of our network.

The ultimate goal of having a methodology is that it will allow you to describe *troubleshooting procedures* and be able to identify which problems that require higher levels of expertise.

3.3 Practical example

Let's take an example to illustrate the approach. If someone calls you and screams “*I can not read my Hotmail!”* you need to be able to have a method to identify the cause without calling in your best network engineer.

If we follow the *first* of the proposed methods (top-down) we will ask the following questions trying to identify where the problem is:

- What program do you use to check your e-mail? (Checking for application problems)
- Can you check the proxy settings of your program?
- Can you reach any other Internet sites? (Checking for DNS problems)
- Does your application time out? (Checking for session TCP problems)
- Have you authenticated with the access-control server? (Checking for Authentication Problems)
- Can you reach our router/provider web site? (Checking for routability problems)
- Do you have an IP address? (Checking for IP problems)

If we follow the *second* proposed method (middle-top/down) we will ask the following questions:

- Can you ping hotmail.com?
- Can you ping <IP address of the border router of the WISP>?

If both answers are “no”:

- Do you have an IP address?
- Have you authenticated with the access-control server?

Classifying problems is not an easy task, and problems vary from network to network – but the *methodology* we use to troubleshoot is always the same.

There is one easy way to classify any problem in a network:

- Things do not work at all (Why my computer does not <include word here>?)
- Things work sometimes... (or things work, but “badly”) (Why is my computer so slow?)

The first type of problem is normally easier to troubleshoot, as it stems from problems related to a wrong link budget, power loss in the equipment, misalignment of antennas, wrong settings etc.

The second type of problem, especially when related to lower layers of the TCP/IP stack, is more difficult to troubleshoot as it will require you to monitor all the wireless parameters during a period of time while you are trying to identify the cause of the problem.

In the diagram below we include a set of tools that can help you to troubleshoot:

Layer	*Tools*	*TCP/IP*	*Tools*
7	Application	Application	nslookup
6	Presentation		
5	Session	Transport (TCP)	Ntop (Win32/Linux)
4	Transport		Visualroute, traceroute
3	Network	Network (IP)	Nmap Ntop (Win32/Linux) Ethereal Etherape
2	Data Link	Media Access Control	Ethereal (Win32/Linux)
1	Physical		Netstumbler (Win32) Kismet, Wavemon, Wellenreiter **Vendor Specific Management Tools**

Table 2: Tools for troubleshooting for each and one of the seven layers of the TCP/IP protocol stack.

When it comes to identify problems in the wireless media, we can use two types of tools: the ones that work with any IEEE 802.11b compliant product, and those that come with every specific vendor.

Some vendors (e.g. Proxim Orinoco Outdoor Solutions) implement *extensions* to IEEE 802.11b that require very specific monitoring/troubleshooting tools.

4. Tools for troubleshooting

1. Nslookup, dig
2. Ntop
3. Visualroute, traceroute
4. Nmap
5. Ethereal (See Scenario 3)
6. Etherape (See Scenario 2)
7. Netstumbler (See Scenario 1)
8. Kismet
9. Vendor specific management tools

5. Scenario 1: Radio Interferences, Occupied Channels?

There is a not simple and cheap way to monitor all the parameters involved in the “physical layer” of your wireless network. When troubleshooting the “radio” you will always use tools that talk with the “wireless cards” and retrieve a limited set of that information for you.

By using a program like “Netstumbler”, a wireless cards acts as a simple “spectrum” analyser that can scan for existing networks, their signal to noise ratio, modulation technique and operation mode. Netstumbler gathers all that information in a easy to use interface.

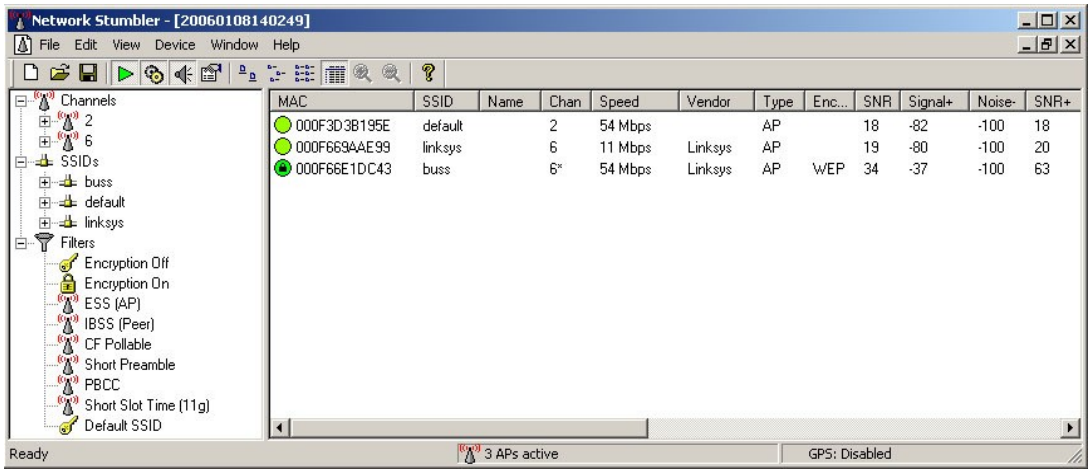


Figure 1: Netstumbler GUI, Source: http://upload.wikimedia.org/wikipedia/en/9/95/Netstumbler.jpg

In the example (Figure 1), we can see that there are three SSID present (default, linksys, buss) in “just” two channels (2 and 6). Two of the access points are operating in the “g” standard of 54 Mbps (default and buss) and one in “b”.

WEP encryption is enable in the network with SSID=buss. All networks are listened with good SNR ratios SNR>10 dB.

Netstumbler is a “passive” software that eavesdrops wireless traffic from the network. Not all the wireless card will allow you to “monitor” wireless traffic promiscuously, before you install Netstumbler check out that your wireless card is supported.

6. Scenario 2: Congested Network? Flooding?

If you want to get a “general overview” of the type of IP connections that are active in your wireless network, you can use the Unix program “EtherApe” in your wired gateway. EtherApe allows you to monitor the *incoming and outcoming* connections routed into your wireless. It can help you not only to identify the type of IP traffic present and the distribution of traffic between your nodes but also how “dynamic” your network is. By observing the traffic graphs with the software, you will be able to detect viruses scanning your clients or the present of heavy peer-to-peer or FTP traffic. There are similar softwares and more sofisticated protocol analysers also under MS Windows (AirDefense, Scrutinizer, SolarWinds, etc) but few of them are free (if any!).

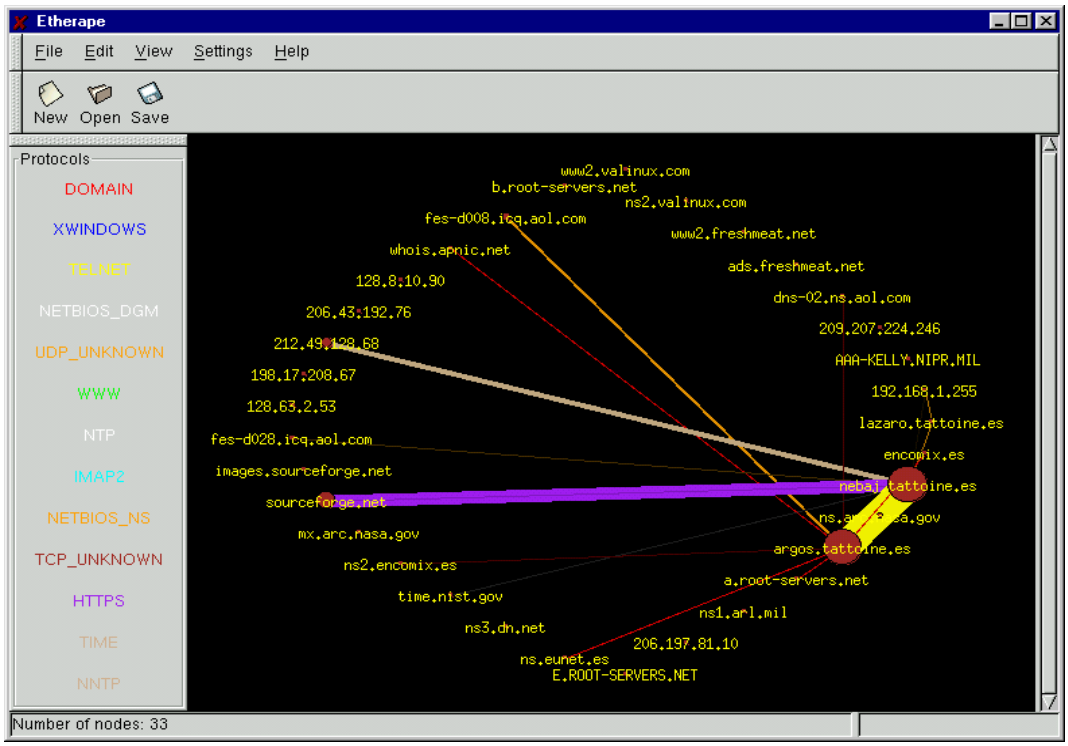


Figure 2: EtherApe GUI, Active IP connections

In the example (Figure 2) we can see that there is big amount of HTTPS (Secure Web Traffic) between the node “nebaj” and “sourceforge” (violet line). We can also see that there is a “Telnet” connection between nodes “nebaj” and “argos” (yellow think line). DNS traffic is working properly from “argos” to “ns.eusnet.es” and “ns2.economix.es” (red lines)

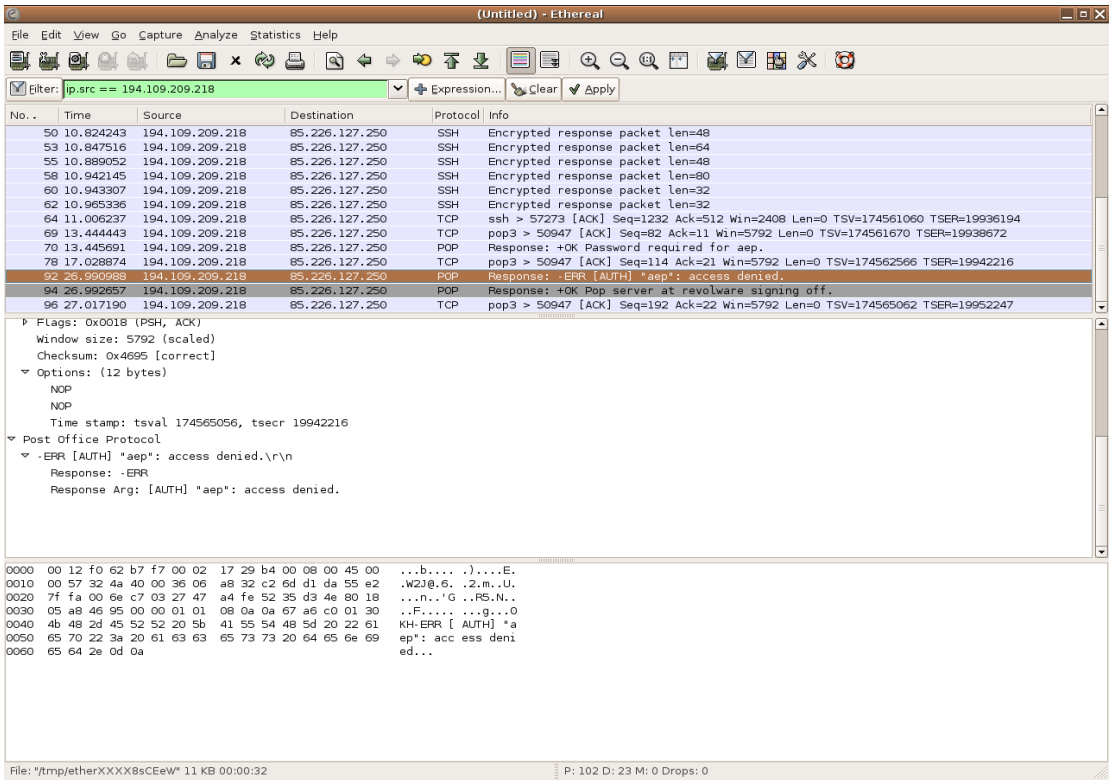
The UDP unkown traffic (brown line) corresponds to ICQ Messenger traffic (connection to fes-d008.icq.aol.com)

7. Scenario 3: Why this network service is not working? Connection Refused?

If you need to have a closer look to what is happening for a specific type of traffic you might consider installing Ethereal. Ethereal will allow you to capture ALL the traffic that is passing by your interface and be able to examine the traffic flows and the bits and bytes of every transaction. Ethereal is very useful to monitor::

- packet loss in TCP connections: that is normally an indication of a congested network, collisions etc
- round trip-times: that is an indication of your network latency. High round trip-times inside of your wireless network is an indication of high level of channel utilization or packet collisions.
- protocol errors: errors that are not normally visible to the user as inappropriate authentication, duplicate IP addresses, network unreachable, ICMP flooding etc. (See: Advanced Networking)

Figure 3: Troubleshooting POP3 Mail Problems with Ethereal



Just to show how powerful Ethereal can be for troubleshooting, in the example (Figure 3), we can see the level of detail that can be obtained from Ethereal.

- After capturing traffic from the network we can apply a filter (ip.src=194.109.209.218) to filter all packets coming FROM the mail POP server. (green box)
- After filtering all the packets we can see the traffic exchange between the mail server and our client (violet). Traffic marked as TCP indicates the connection renegotiation (known as TCP handshake); traffic parked as POP3 corresponds to the Application “POP3”, mail retrieval.
- We can select individual packets of the POP3 Session and see the presence of -ERR (AUTH): “aep access denied”)

With this information we can determine that: the connection from our client to the mail server takes place, the POP3 server is running and the problem takes place when authenticating. An authentication problem can result from a client or a server-side problem: the client sending the wrong password or the server not being able to validate the password correctly.

8. Conclusions

The five main issues you should remember from this unit can be summarized as:

1. The more you know about how things work, the easier to troubleshoot when they do NOT work
2. To understand a problem is not the same that solving a problem
3. Try to apply a logical methodology when things go wrong rather then doing things in random order
4. Whatever approach we take to troubleshooting it is important to be familiar with the tools that are appropriate when analysing each of the functional layers of the network.
5. When identifying problems in the wireless media, we can use two types of tools: the ones that work with any IEEE 802.11b compliant product, and those that come with every specific vendor.