Review of Metageek's Eye P.A. - Visualizing 802.11 Frames

A blog post by Keith R. Parsons

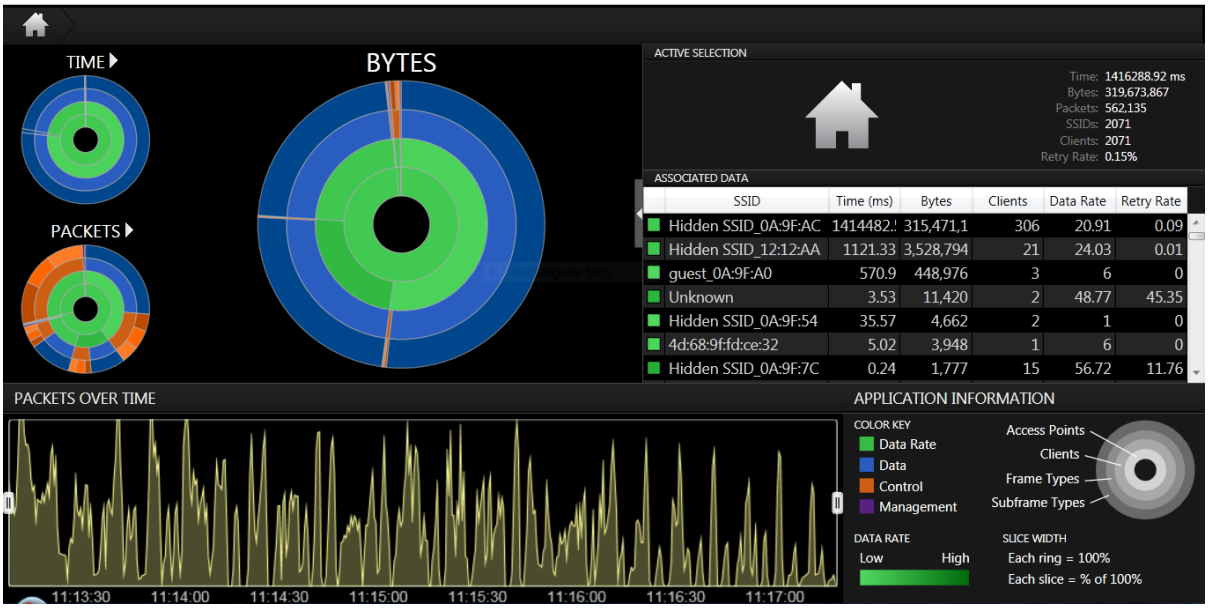
The premise behind Metageek's new product, Eye P.A. is to help visualize 802.11 frames to make it easier to troubleshoot your Wireless network.

You can use the free software Wireshark to capture 802.11 frames – and then look through them, the proverbial needle in a haystack scenario. Or if you can get good at Wireshark and get some percentages of different types of frames. If you are really good at Wireshark you can write filters to help narrow down your search.

Other vendors show these same frames as percentages or even as pie charts. What sets Metageek's Eye P.A. apart is the Tree Pie – the mixture of showing multiple statistics on a single circular view. Other software vendors, especially in the "show me what's on my hard drive" category have used this for years.

Eye P.A. Overview

The concept is fairly simple. Collect data into a standard PCAP format. Then Eye P.A. does it's magic by first parsing all the data into a special database format. Then the collected data is displayed in a visual way to help users better understand the big picture.



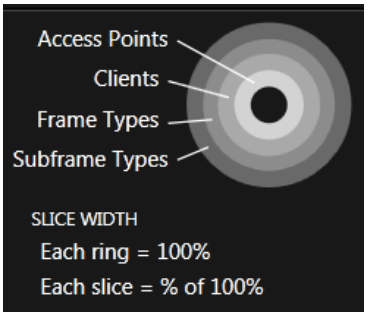
The high level view in Eye P.A. shows three Tree Pies – one for Time, one for Packets and the final one for Bytes. These three can be rotated around so the larger slot can show either of the views.

Note: The guys at Metageek started to call these Tree Pie's – treepees – and so when you talk with them on the phone, you'll now know what they are referring to.

There is also a tabular section – this showing some statistics and from the data collected. Like the Tree Pies, this tabular data reflects the selection you have chosen.

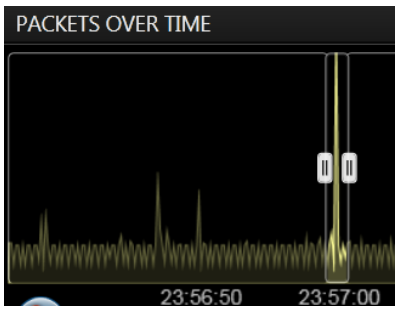
Below is the Packets over Time area – showing volume of packets over the data collection period.

The last area in the bottom right corner shows the legend, or key to the visualizations. Each type of 802.11 Frame is shown in a different color. You'll soon get used to seeing these colors and what they represent. Within each color there are shades, the shades show a sub-set of the frame type.



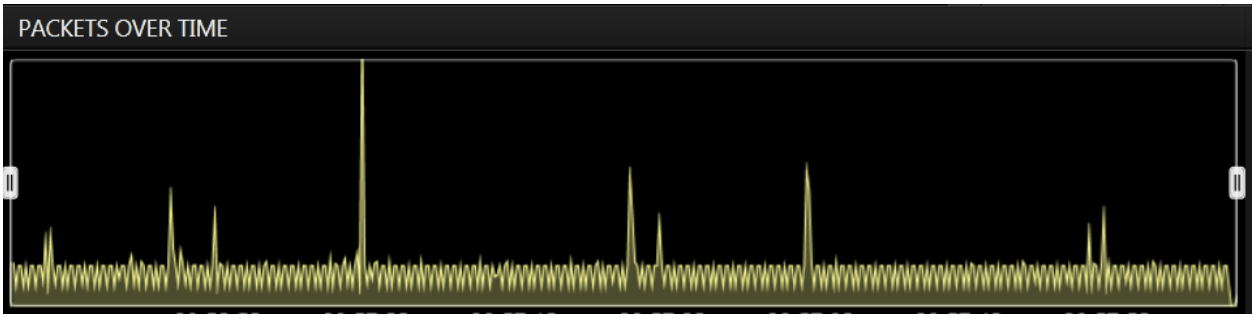
There is also a key showing what each of the circles represents. Access Points on the inside circle, then moving out through clients, then frames and finally subframe types. The width of each slice also connotes meaning. Each slice show a percentage of the total as represented by the entire ring.

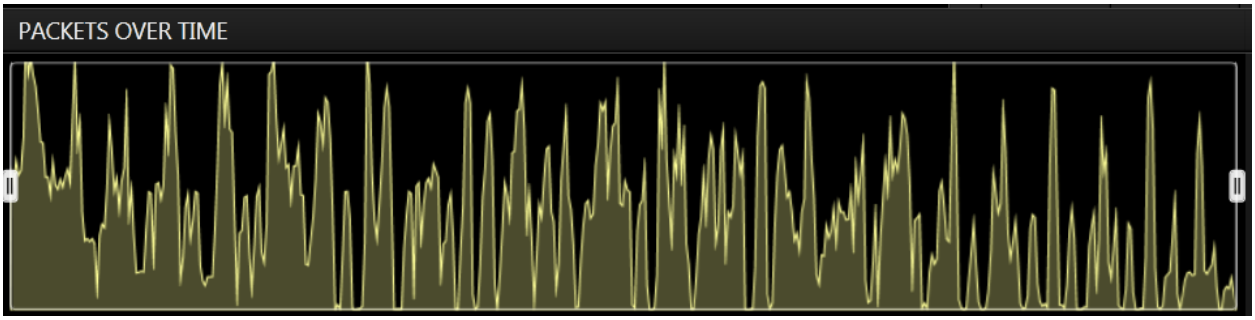
You can choose to select a subset of the entire packet capture, to 'drill down' on a specific chunk of time. Just use the slider bars on either side to re-parse on that specific subset of time/packets.



With each of the visualizations used in Eye P.A. there is logic, and information being displayed.

Note the differences between these two data captures. Without looking at any of the details or Tree Pies… you can instantly tell there was a very different load on the network during the captures.





Using Eye P.A.
But the real value of Eye P.A. isn't showing volume of frames… you can also get that by just looking at the packet counts in Wireshark.

What you can't easily see in Wireshark is the relationships between the different 802.11 frame types. For years I've been attempting to teach learners the value of **really** understanding the 802.11 protocol. Because if you have a great understanding of how the protocol works, troubleshooting isn't guess work – but actually getting to the solution. But first you need to know what the problem is – compared with what is 'normal' on your network.

As a classroom exercise, I use the following chart to emphasize the point of what is 'normal' and what you should expect based on differing scenarios.

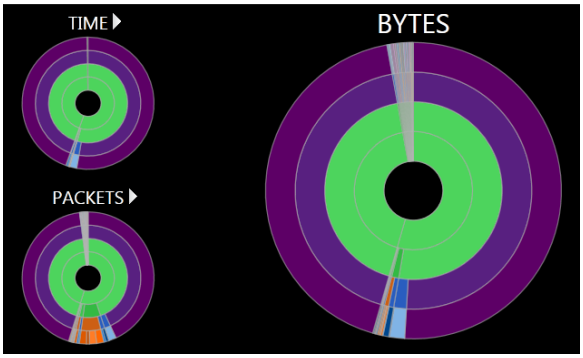
	Size	**Speed**	**Distance**
Control	Tiny	1Mb	Far
Management	Small	1Mb	Far
Data	Huge	Fast	Near

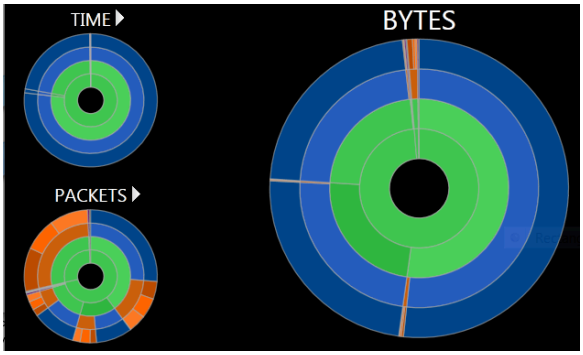
Since you know the sizes of Control vs Management vs Data frames and how they are transmitted differently. This table should make perfect sense.

Now onto some ratios between these different 802.11 Frame Types – and how different scenarios would have different ratios.

	Control	**Management**	**Data**
Idle	small	huge	small
Good	1:1 Ratio	is	Big
Bad	1:3 Ratio	is	Bigger
Hidden Node	3:1 Ratio	Is	Big

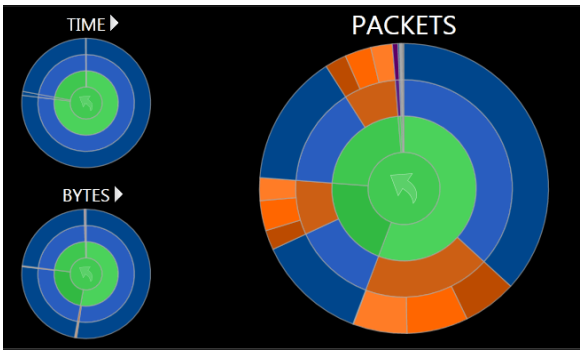
In an Idle network – there will be lots of management packets, and very little of the other types. Here's what this looks like in Eye P.A. – Note all the Purple for Management Frames. There is nothing wrong with this network… it's just idle! In the Time view, almost the entire time is spent in Management Frames.





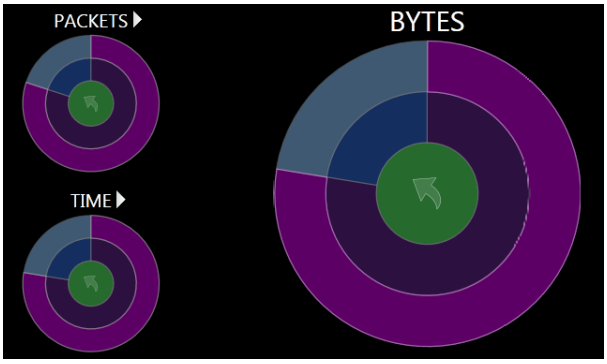
But you'll have an entirely different view of a network that is passing lots of data. The data packets are HUGE compared with the tiny Control and Management Frames. By also looking at the Time view – you can also determine the Wi-Fi network is working very efficiently – with the transport of data frames as the paramount activity on the RF frequency.

In this high volume of data, the Bytes view shows almost entirely Data Frames as a percentage of the Bytes transmitted. This is a good thing. Note below the Time View to see these Data Frames are nearly 100% of the time slots being used in the RF airtime. But note also the Packets View – there is still a standard 802.11 Data/Ack sequence going on. Almost a one to one ratio between Data and Control. Justs what you'd expect in a smoothing running Wi-Fi network.



Here's the same data set, but the focus is on the Packet Counts – see how it emphasizes the differences between Frame Types – and de-emphasizes actual throughput. Use the views appropriately to help you see what is really going on with your network.

In very rare instances, you'll find an equally distributed view between Time, Packets and Byte Views. Normally it's the differences between these that you'll gain information about your network.



In the 'Bad' network the ratios are based on 802.11 retransmissions. If you have more ACK's than Data – you must be retransmitting lost Data frames. You can also look for the Retry counters shown when hovering over any specific slice of a Tree Pie.

In the Hidden Node scenario, you should show a single Data Frame, and three control frames. (RTS-CTS-Data-ACK) – Three Controls for each Data frame.

You can also learn about your target network by reviewing the Tabular version of the data. In this view, you can sort on any column. Sorting a second time reverses the sort. Use this to find 'top talkers' or those who are using the bandwidth or time the most.

ASSOCIATED DATA

SSID	Time (ms)	Bytes	Clients	Data Rate	Retry Rate
Hidden SSID_0A:9F:AC	1414482.!	315,471,1	306	20.91	0.09
Hidden SSID_12:12:AA	1121.33	3,528,794	21	24.03	0.01
guest_0A:9F:A0	570.9	448,976	3	6	0
Unknown	3.53	11,420	2	48.77	45.35
Hidden SSID_0A:9F:54	35.57	4,662	2	1	0
4d:68:9f:fd:ce:32	5.02	3,948	1	6	0
Hidden SSID_0A:9F:7C	0.24	1,777	15	56.72	11.76

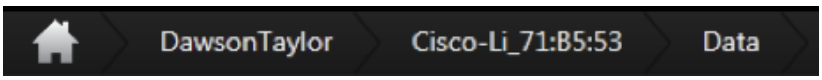
Double clicking on any line in the Tabular view will 'drill down' to that – making the selection focus in the Tabular as well as all the Tree Pies match that specific subset of the collected data.

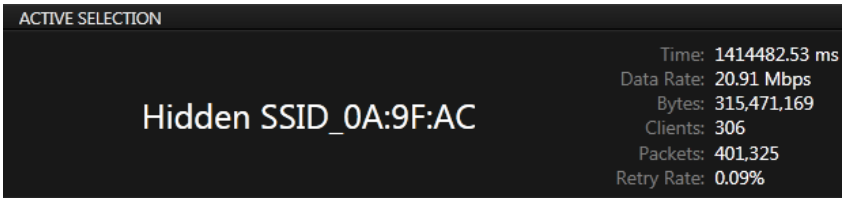
You'll find yourself moving in and out (down and up?) through your data sets to find the most pertinent data in your troubleshooting process.

When you find a specific set of data that needs more detailed analysis – where you need to look inside the actual packets – you can quickly export this subset back to Wireshark for further analysis. (It basically acts like a filter to the PCAP file and returns just those packets that meet the criteria you chose in Eye P.A.) You can choose the Send to Wireshark from Eye P.A.'s File menu.

Time	Source	Destination	Protocol	Length	Info
0.000000	LiteonTe_4f:5e:	Broadcast	802.11	276	Data, SN=90, FN=0, Flags=.p....F.C
0.307268	Apple_1f:6a:6c	Broadcast	802.11	316	Data, SN=94, FN=0, Flags=.pm...F.C
0.307739	Apple_1f:6a:6c	Broadcast	802.11	316	Data, SN=95, FN=0, Flags=.pm...F.C
0.308088	Apple_1f:6a:6c	Broadcast	802.11	316	Data, SN=96, FN=0, Flags=.pm...F.C
0.308611	Apple_54:89:25	Broadcast	802.11	316	Data, SN=97, FN=0, Flags=.pm...F.C
0.309015	Apple_54:89:25	Broadcast	802.11	316	Data, SN=98, FN=0, Flags=.p....F.C
1.228557	Apple_c8:5c:dc	Broadcast	802.11	106	Data, SN=107, FN=0, Flags=.p....F.C

Additionally, at the top of the Eye P.A. screens you can see the Active Selection area at the right side, and a 'Breadcrumbs' view of how you drilled down to this level at the left side. You can also click inside the breadcrumb area to go back up a level.





How to Use Eye P.A.

I've had lots of discussions with various individuals on this very subject. The consensus seems to be based on first having a sound foundation in 802.11 basics, understanding what and how packets flow on your Wireless LAN. Without this knowledge, Metageek's Eye P.A. is a very pretty piece of software with very colorful pictures…

I'd recommend learning at least at the CWNA level of packet flows – personally I think moving on to the CWAP level would be even better. Either way – strong 802.11 protocol knowledge is necessary. This tool makes it simple to visualize the packet flows. But you **need** to know what those mean.

This is a far superior method when compared with just looking at raw packets in Wireshark – but the again, the more you understand what is 'normal' in 802.11 – the better you'll be able to interpret when things are not going as they should.

Collecting PCAP Data Files

Metageek's Eye P.A. **requires** a PCAP file in order to operate. It does not come with its own PCAP capture utility. You can use either Windows, or Linux or even Mac OS X for the actual capture process. But as for now, the Eye P.A. product only runs under Windows.

I've used both native Windows as well as VM-based versions running on top of Mac OS X. It works well either way.

The standard Windows drivers for Wi-Fi NICs don't have the proper promiscuous mode/monitor mode that will pick up the necessary headers in order for Eye P.A.'s data parser to work.

One type of header is called the PPI – and it contains a variety of details about the 802.11 frame – you can find details about this here. The other option is more standard and uses a Radiotap Header. This version contains a wide variety of fields describing the 802.11 frame and how it was collected by the Wi-Fi NIC. Details available here.

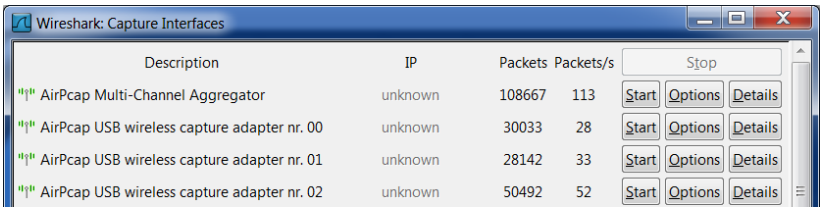
Wireshark can collect either type. See the results in the screen shots below:

```
▽ PPI version 0, 32 bytes
    Version: 0
  ▷ Flags: 0x00
    Header length: 32
    DLT: 105
  ▽ 802.11-Common
      Field type: 802.11-Common (2)
      Field length: 20
      TSFT: 2676709534
    ▷ Flags: 0x0001
      Rate: 24.0 Mbps
      Channel frequency: 5745 [A 149]
    ▷ Channel type: 802.11a (0x0140)
      FHSS hopset: 0x00
      FHSS pattern: 0x00
      dBm antenna signal: -42
      dBm antenna noise: -82
```

```
▽ Radiotap Header v0, Length 25
    Header revision: 0
    Header pad: 0
    Header length: 25
  ▷ Present flags
    MAC timestamp: 4179570803
  ▷ Flags: 0x06
    Data Rate: 6.0 Mb/s
    Channel frequency: 5745 [A 149]
  ▷ Channel type: 802.11a (0x0140)
    SSI Signal: -41 dBm
    SSI Noise: -82 dBm
    Antenna: 0
```

Both contain enough information for Eye P.A. to do it's data parsing and visualization processes.

In order to collect this data on a Windows machine, you'll need access to a Cace Technologies AirPcap card. These are available at their website.

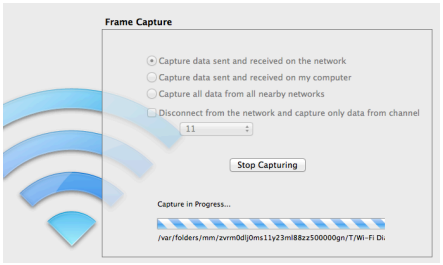


One cool thing with AirPcap cards – they have a driver that aggregates multiple AirPcap data capture flows into a single data stream/PCAP file. So you can capture on multiple frequencies or channels at the same time. In this graphic you can see I can choose between any of the three individual AirPcap devices, or one aggregated virtual device.

On the Mac OS Platform you can use Wireshark – but it is missing some features. You will have to choose the channel you want to scan prior to starting Wireshark. (Usually you'll just be scanning the same channel your Mac is associated to) – You can also use a Terminal session with the Airport command to change channels. The command is `Airport –C[channel].` This will be using the internal Wi-Fi NIC. Check in your System Preference – Network to see if your Wi-Fi NIC is on EN0 or EN1.

***Note**: MacBookAir's usually are on EN0 and MacBookPros on EN1.*

Another method for capturing a PCAP file on Mac OS X is a new utility available in Lion called Wi-Fi Diagnostics. I wrote a different blog post on how to use this utility you can find here.



Conclusion

I believe this tool from Metageek will help many experienced Wireless LAN Professionals to quickly diagnose and troubleshoot issues within their networks. If you understand 802.11 fundamentals and how the different frame types interact – this tool is for you!

Keith R. Parsons
Wireless LAN Professionals.com
Personal Keith@inpnet.org
Twitter http://twitter.com/keithrparsons
Facebook http://facebook.com/keithrparsons
LinkedIn http://linkedin.com/in/keithrparsons
Website http://WirelessLANProfessionals.com