TrafficAnalyzer Specification

# Summary

This tool is aimed at analyzing traffic received at a probe in order to check the following information:

* All VLANs from received traffic are documented and all documented VLANs are in received traffic
* All IP addresses from received traffic are documented and all documented IP addresses are in received traffic
* Answer/response messages show symmetry overall
* Answer/response messages show symmetry by pairs of IP addresses, Point Codes, Calling/Called Party, etc.
* Message distribution per VLAN
* IP Decoration is correct (according to files nodes.csv, point-codes.csv, etc.)

This tool tries to complete the existing *validate-network.py* tool by improving the speed of analysis.

# Architecture

## Generic

The program will be a Python program to be run on Windows, input file must be a local .pcap file (originated by a probe capture).

Output should be given in the form of an Excel file with several tabs (in the same way like *validate-network.py* works), and also show some charts with some specific module (PDF format).

Eventually, Python 3 has been chosen in order to obtain better a performance from several needed modules.

## Future trends

It would be wise to create a version for PyPy (Python 2.7) which could be executed on Linux servers. In this way, captured files could be analyzed on-site at the very same probe.

## Packets dissection

When it comes to analyzing TCP/IP captured packets, the following Python modules are available:

* Dpkt -> This is a python module for fast, simple packet creation / parsing, with definitions for the basic TCP/IP protocols
* Scapy -> This module performs packet dissection but can also create packets, send, receive and capture them over the net. It runs natively on Linux, and on most Unixes with libpcap and its python wrappers. The same code base now runs natively on both Python 2 and Python 3.
* Pyshark -> This is simply a python wrapper for tshark

The choice is to use dpkt as this is faster than the rest, this is implemented in file *analyze\_traffic\_dpkt.py* (other versions exists for scapy: *analyze\_traffic\_scapy.py* and pyshark: *analyze\_traffic\_pyshark.py*).

## Charts

Results will be displayed in the form of charts by using the following module:

* matplotlib -> This is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms

# Interface

Program Will receive one of two types of parameters:

-i list of interfaces, to sniff traffic from ports and perform the analysis

-f list of files, to analyze pcap files with pre-captured traffic