Perytons™ Protocol Analyzer Architecture

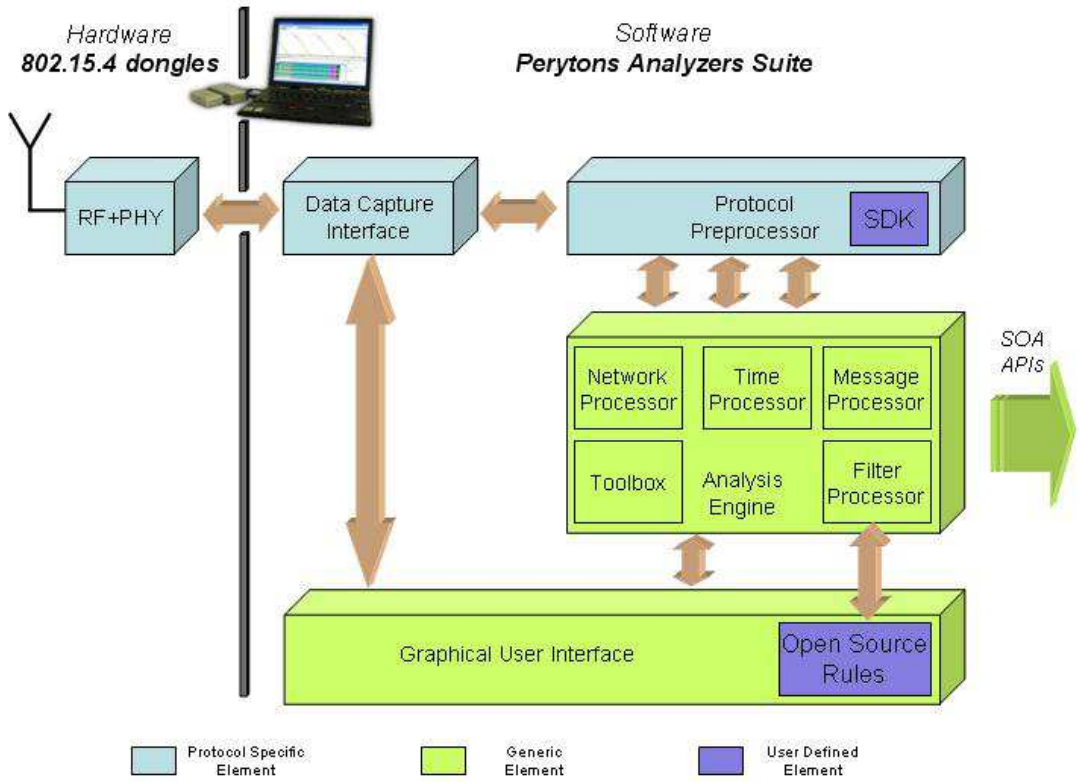
The various Perytons™ Protocol Analyzer models are based on a similar core engine, a fact that enables the easy upgrade between the different models and ensures easy cross operation and work between all models of the Perytons™ Protocol Analyzer family.

The Perytons™ Protocol Analyzer is built-up of two main components – the analyzer core and the protocol specific elements.

The Perytons™ Protocol Analyzer core includes the GUI (Graphical User Interface), analysis engine and analysis toolbox. This part is shared by all Perytons™ Protocol Analyzer models.

The protocol specific elements include RF front-end, data capture interface, and the protocol processor. This part is protocol-specific.

With the exception of the RF front-end, which requires dedicated hardware (supplied with the analyzer), all other elements are implemented via software running on the user's PC.



In addition to the graphical user interface, the user can add ‘code like’ elements to the analyzer, either by using the Peryton-Monitoring Add-On (with which he can create Open Source Rules and incorporate them into the analyzer environment) or through the Peryton-SDK Add-On that allows

the addition of entire protocol layers into the analyzer environment (see "Flexibility" and Open Source Rule" whitepapers for more details).

When monitoring of multiple networks for performance and potential problems is needed, user defined charts and events, are accessed by NOC (Network Operations Center) operators, via SOA (Service Oriented Application) interfaces built into the analyzer engine via APIs (Application Programming Interfaces).

- Protocol specific elements
  - RF front-end

    A USB 802.15.4 dongle is used as the Perytons™ RF front-end for 802.15.4 related traffic. For other protocols (e.g. One-Net and PLC) another off-the-shelf hardware is used. The dongle is tuned to the right channel set by the user or automatically by the analyzer software, and receives all 802.15.4 message traffic transmitted on the channel. The dongle forwards to the Data Capture I/F module each received packet together with side information on channel number , time of reception, signal strength etc. Multiple dongles can be used to enable simultaneous data capture from multiple channels, or improve performance by using antenna diversity on a single channel.
  - Data capture interface

    The data capture interface manages the dongles and brings their data into a common structure used by the analyzer core. It is responsible for pre-capture tasks such as activity search and time synchronization between multiple dongles; for data capture management and for post capture processing (FCS calculation, duplicate message filtering in case of antenna diversity etc).
  - Protocol Preprocessor

    The PHY/MAC preprocessor is responsible for decoding message contents, retrieving field values, and interpreting the information. The Software Development Kit allows users to add their proprietary protocol or application layers into the Preprocessor.

- Common elements

  The analyzer core includes a set of utilities and services used for all protocols
  - Analysis engine

    Provides the tools and libraries used by the Protocol Preprocessor for decoding the protocol. The Analysis Engine is built-up of the Time Processor that builds the time related events (e.g.

beacon), Message Processor that builds the message fields, and the Network Processor that analyses the network topology, and builds it into an easy to view layout.

- Analysis toolbox

  Includes a set of analysis services such as message compare, search, statistics, file export, and workspace management.

- Graphical User Interface

  Displays the captured data in the Time View and in Message View windows. The GUI allows the user to handle and manage all the analysis options and services.
  The GUI module also includes the User Preferences that provides the user with the flexibility to set view details (colors, lines thickness, etc.) in the different analyzer view windows.

- Monitoring tools

  Allow setting Open Source Rules on received messages. The result of the rule can be filtering messages that will be displayed by the analyzer, generating events or alarms (including the option to send e-mail upon high severity alarms), building relevant parameters into user defined statistics and many more (for further information about Open Source Rules please see the 'Perytons™ Open Source Rules' whitepaper).

- SOA APIs

  Enables the easy connection of a centralized server to collect Monitoring results from many analyzers positioned in the field, giving the operator a summary on performance and failures in each of the monitored networks.