Introduction

At the core of the ATLAS-Daq infrastructure there are 3 distinct computer networks responsible for the data transfers between the system's subcomponents. More than 200 switches and routes interconnect around 3500 hosts to build a real-time filtering system for particle collision events.

For the ATLAS Networking Team the operational goals are to prevent network downtime and to be able to track down events on post-mortem network issues as fast as possible. A complex software solution has been developed to help networking experts accomplish their goal while providing relevant and up-to-the-minute system information for other related Daq teams.

The network monitoring software is designed as a modular solution around a central database (N-CDB). The N-CDB acts as a shared data structure for the various modules and is kept up-to-date by a set of modules performing topology discovery and statistics collection.

The information is accessible either from a web-based user interface or through a custom ATLAS Data API Mechanism (ADAM) for data exchange. Additionally, a self-check mechanism warns about any module not functioning as expected, by sending detailed mail messages to experts.

Topography Consistency

An accurate representation of installed devices and network connections in the N-CDB is assumed by every module and is vital for a network behavior of the system. To keep the topology representation up to date, a full network discovery process runs periodically and is complemented by additional data.

The network discovery is based on MAC-address information, being able to identify network device uplinks and end node connectivity by physical address. However, a pure network discovery does not solve the problem of a wrong geographical location or external databases are queried to complete the topology view (e.g., ATLAS Back-End, DTF Lab, BNL, Gaia, Collabaid).

Self-check Mechanism

The reliability of the system may be affected by a monitoring module. The purpose of the self-check mechanism is to define the steady state, to identify and notify experts in real-time about any faulty modules. To make the self-check mechanism more robust, the checking of module execution status and the module reporting have been implemented as two distinct components.

The execution status check is implemented using a set of scripts that monitor each module's functionality and expected results, and then report the status to the N-CDB. A module can fail its reporting and send a detailed message. It anticipates an imminent problem and is an expert intervention. The warning component then checks the active module's status information and sends alert messages to notify status updates.

The User Interface

The user interface is designed to provide all the data an expert needs to investigate a problem and yet be simple enough to be used by experts. A set of web-based applications are integrated into a portal, offering access to real-time statistics and related services about network status and complementing systems. For offline visualization of the current topology, a comprehensive PDF report is generated daily to be downloaded.

Another option to access the N-CDB data is through a CLI-like interface. The CLI tool was developed as a tool for users to access and change the N-CDB data manually without the need of affecting data flow consistency.

Statistics Collection

Gathering statistics about traffic flows and network delays is mandatory for monitoring the systems' performance. The network overview shows the primary statistics of information for external and internal network problems. These statistics are collected by a set of statistics developed software, which polls SNMP counters, gathers statistics, and forwards every 30 seconds. Once network is located, an in-depth analysis is performed by looking for relevant traffic spikes collected via the SFlow protocol.

Sometimes external factors, such as environmental conditions or faulty systems, affect the network's normal behavior. Making information from directly related systems available to network monitoring modules gives the networking team an advantage in understanding the influence of physical or external events. Currently, a full control environment (Nagios, computer monitoring, PVE, and ENSIS) is required information system accessible through the same user interface used for network monitoring.

The ADAM API

The information stored in the N-CDB is of interest to other systems and analysis. To facilitate programmatically access to information, a generic interface for data exchange has been implemented. The creation of a generic interface has been a joint effort between several ATLAS teams and led to the definition of the ATLAS API Mechanism (ADAM) interface exchange.

The ADAM API is fully implemented in the network monitoring solution and provides network traffic information, computer status and environment conditions to external applications on demand. Through this API the network monitoring software can be a data provider for other external data analysis applications.

Current status & future plans

Started as an independent network monitoring software, the current ATLAS-Daq network monitoring solution has grown by integrating additional system information. This extension improved the understanding of overall system behavior by enabling the correlation between networking events and external factors. Two years after its deployment it is the main tool used by both experts and non experts to analyze network problems. It has proven to be better suited for system diagnostics than the alternative commercial solution used for performance monitoring.

Future plans to improve the current solution include the extension of the N-CDB by storing network event logs from multiple sources and the addition of an event processing engine. Such a feature will bring together the networking events and correlate with the collected statistics that provide a more complete picture of the circumstances associated with the generated event.