Introduction

Networks play a fundamental role in the adoption and growth of Internet applications. Penetrating enterprises, homes, factories, and even cities, networks sustain modern society. While assuring responsive and performant networks in today’s hybrid, distributed and containerized application environments occurs behind the scenes in intangible clouds and diagrams abstractions, network glitches are more visible and unforgiving than ever.

Keeping everything running and functional is at the core of IT Operations, but the role of monitoring doesn’t stop there. There is a shift to using a more advanced network monitoring analytics platform, coupled with machine learning and AI frameworks, to enable not only addressing enterprise network problems in real-time but also predicting where network problems may be brewing and identifying underutilized bandwidth that could be monetized. Network monitoring takes DevOps and IT Operations beyond cost of doing business, turning it into a business unit that seeks not only effectiveness and efficiency but also profitability.

Network monitoring – pillars for success and growth

Today’s applications are architectured to work on hybrid, componentized, containerized, distributed, mobile/sensor environments, and so do the networks that sustain all these applications together. Complexity, traffic volume, intolerance to performance degradation and inefficiency will only increase, demanding real-time and holistic monitoring. Metrics as a Service inside organizations is already a reality and is allowing operations to better articulate necessary investments. Facing the new paradigm, DevOps and IT Operations teams must be able to answer both cost-centric and profit-centric questions.

<table>
<thead>
<tr>
<th>Cost Centric</th>
<th>Profit Centric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Are network resources available?</td>
<td>Are network resources fully utilized (provisioned vs. use)?</td>
</tr>
<tr>
<td>How is the network performing (latency, error rates, bandwidth consumption)?</td>
<td>How are KPIs impacting revenue generation?</td>
</tr>
<tr>
<td>Is root-cause identified promptly?</td>
<td>Is the Mean Time to Repair (MTTR) impacting user-experience and causing churn?</td>
</tr>
<tr>
<td>Who/what is driving consumption?</td>
<td>Are there network misuse or new revenue generation use-cases?</td>
</tr>
<tr>
<td>Do I have the right tools to acquire actionable insight and prediction models?</td>
<td>Are mission-critical services KPIs being effectively tracked and tackled?</td>
</tr>
</tbody>
</table>
Network monitoring can be used to substantiate business decisions answering the questions listed above. There are several sources of instrumentation for network monitoring data such as: endpoints metrics, events, logging, sensor data, traffic flow analysis, packet inspection, and synthetic tests. However, often information is collected and visualized in silos, consuming IT resources without providing a complete view of the overall application environment health, all while leaving gaps that can lead to flaws and oversights when assessing network and resource utilization.

The basics of network monitoring

Foundation for network performance monitoring

Accommodating all monitoring requirements for a modern application environment is certainly not a trivial task since a global as well as granular view of the network impact on services is needed. Additionally, to avoid introducing yet another halfway solution, a sound approach must account for supporting multi-type data and cross-measurements analytics that is able to accommodate requirements from the whole organization.

DevOps and IT Operations deals with complex and demanding environments; nonetheless, establishing centralized network performance management can be simple and easy. Three network performance monitoring pillars separate the unmonitored from the monitored network — and if you feed this data into a centralized time series platform you will be able to enable holistic network performance monitoring.

1. **Network Availability** - This refers to host reachability. If unavailable, there could be something related to the endpoint health or network path (such as a load balancer’s sessions limit or expired SSL) preventing traffic from reaching the host.

2. **Network Responsiveness (latency and packet loss)** - Latency refers to the time it takes for traffic to cross the network to a target, and packet loss determines the error rate experienced. Latency and error rates above certain thresholds render the network unsuitable for all or some more sensitive applications. For instance, high latency will completely undermine unified communications (voice & video) services.

3. **Network Bandwidth Consumption (who/what, when, and how much)** - This tracks metrics from the network interface and from the traffic flow providing important information about bandwidth load, which can be used to set up alerts before the interface is completely saturated and applications are impacted. Correlating the network interface metrics data with flow analysis from network traffic analyzer appliances provides precious insights about sessions and IPs/protocol/port troublemakers that are causing saturation and bandwidth struggles, as well as identifying resource misuse and potential Denial of Service (DoS) attacks.
Platform approach to network monitoring

Don’t be consumed by monitoring data. Centralize, analyze and automate.

Instrumentation of monitoring data is just the beginning of the journey. Thinking from a strategic standpoint is mandatory in order to serve the needs of various constituencies in the organization, while avoiding drowning in your own data. Silos are therefore a no-go since they demand dedicated personnel and additional development work to extract the full benefit of data from multiple sources.

A platform to which all monitoring data can converge for storage, alerting and processing is how monitoring should be approached strategically. Corroborating the concept of taking a strategic approach, the Open Networking User Group (ONUG) Monitoring and Analytics Working Group approached network monitoring from a perspective that it cannot be restricted to any single data source or type, nor be thought in isolation.

The functional architecture of the InfluxData monitoring platform consists of four components:

- **Telegraf** - the collection agent with 200+ plugins and a vast client library, that can source metrics directly from the system it’s running on, pull metrics from third-party APIs, listen for pushed metrics/events or via streaming consumer services. Telegraf as a monitoring collector supports protocols such as ICMP/Ping, SNMP, NETFlow, SFlow, and Syslog.
- **InfluxDB** - the database and storage engine purpose-built to handle time series data. A perfect store with a powerful query engine for processing multiple data sources to help you avoid a siloed approach.
- **Chronograf** - the visualization tool with pre-canned dashboards with the standard baseline for network monitoring, and a query language that is powerful, scriptable and easy to use.
- **Kapacitor** - the rules engine for processing, monitoring, and alerting.
Below is a graph representation of the functional architecture of InfluxData platform:

Purpose-built design
To support network performance monitoring in an efficient and effective way, and to accommodate the types and volume of data generated in a modern application environment, it is important for a monitoring platform to be built for that purpose. In contrast, solutions without an architecture that can fully accommodate unified monitoring will always narrow your horizon to fit into their capabilities, instead of allowing you to extract full benefits from a monitored network.

Key architectural feature 1: Pull & Push collection mechanisms
When implementing a network monitoring framework, it is important to consider both pull as well as push data gathering methods. Pull methods are a good choice for collecting performance and usage metrics at regular intervals; however, push mechanism would be necessary in order to collect actionable events when they happen.

Furthermore, the choice for pull or push also impacts the traffic on those networks. Network bandwidth and endpoint resources could be completely claimed if monitoring traffic is not well-planned, leading to performance degradation and diminishing benefits from network monitoring. A combination of pull and push monitoring mechanisms is usually necessary. These factors are particularly relevant for IoT and in case of very chatty monitoring protocols.
Key architectural feature 2: Scalable and durable

With all data converging to one platform, the inability to scale is a show stopper. Your solution must be able to ingest in high rates (millions writes per second), store a very large amount of data (hundreds of millions of time series measurements), its query engine must be as well performant in order to support real time alerting and analytics across measurements, and it must make the data durable for as long as it is needed.

Key architectural feature 3: Flexible, multi-type and multi-dimensional data model

Flexible data models allowing multidimensional fields and metadata to be collected are fundamental in order to be resource-efficient at the collection phase, but also to allow for visualization at granular grouping tiers, which facilitates isolating and directing the relevance of the monitored measurement to the right audience. Also, support for multi-type data, for instance, numeric and non-numeric, allows to accommodate the various monitoring use-cases from different business units which may include tracking a string (error codes) or boolean data (true/false status) type.

How to use InfluxDB for network monitoring

Everything starts with collecting the relevant network monitoring data. The following protocols and tools provide a foundation for gathering network monitoring data:

- Ping
- SNMP
- Syslog
- NetFlow, IPFIX & sFlow

Telemetrics can also be streamed from leading network appliances into Telegraf for batching and normalization, or even straight into InfluxDB as is the case for Cisco Pipeline, LibreNMS, Nagios Core, and ntopng appliances. Bringing all these network data together from different protocols, appliances, and methods supports the earlier statement that only with a unified solution can you properly be able to understand and manage your solutions fully.

Ping for network monitoring

Ping is an IT administration software utility used to test the reachability of a host on IP networks. Ping operates by sending Internet Control Message Protocol (ICMP) echo request packets to the target host and waiting for an ICMP echo reply. It is available as a CLI tool in practically all OS platforms and is supported with a Telegraf Ping plugin. Telegraf Ping plugin allows you to measure the round-trip for ping commands, response time, and other packet statistics which helps you understand network availability, latency, and packet data loss. The plugin can be deployed as a central collector to ping/query all devices, per data center, per security zone or per subnet.
SNMP for network monitoring

Simple Network Monitoring Protocol (SNMP) is used for collecting information about devices (routers, switches, servers, etc.) on IP networks. It is widely used in network management and monitoring and exposes the system status and configuration which can be queried. SNMPv1 is the original version with SNMPv2c and SNMPv3 adding improvements in performance, flexibility, and security.

SNMP network monitoring is supported with a Telegraf SNMP plugin, which can also be used to collect endpoint metrics such as CPU, memory, storage, disk I/O, power, environmental, uptime, etc. This can be done by configuring the Telegraf SNMP plugin to fetch the specific hierarchies also described as Management Information Base (MIBs). The hierarchical namespace contains object identifiers (OID), and each OID identifies a variable that can be read or set via SNMP.

Syslog for network monitoring

Syslog is a protocol standard that describes how log messages should be formatted and transmitted. It is used by devices and applications for message logging to send data about status, events and diagnostics to a central server, a.k.a. Syslog server. A listener process gathers the data sent over UDP or TCP.

Components in the message include an originator process ID, a timestamp, and the hostname or IP address of the device. Syslog standard also has built-in severity levels ranging from an emergency event to debugging purpose messages providing valuable insights for monitoring and alerting. Syslog network monitoring is supported with a Telegraf Syslog plugin.

NeTFlow, IPFIX & SFlow for network monitoring

NetFlow, IPFIX and SFlow are standards used for congestion control, troubleshooting, and network management, among other things. From a network performance management perspective, these protocols allow you to collect bandwidth data per traffic flow and per IP/protocol/port.

- NetFlow was introduced by Cisco and supported by vendors like Juniper and VMWare to provide the ability to collect IP network traffic as it enters or exits an interface.
- IPFIX (Internet Protocol Flow Information Export) is a standard derived from Cisco’s proprietary NetFlow v9 used to export traffic information from L2-L7. It was designed to address shortcomings from NetFlow and be vendor-neutral.
- SFlow (short for “sampled flow”) is an industry standard for packet sampling on Layer 2-7 and provides a means for exporting truncated packets for the purpose of network monitoring. This standard is supported by many vendors including and not limited to Cisco, Arista Networks, Fortinet, Juniper, and IBM.
Although these protocols are not currently supported by a Telegraf plugin, an InfluxData partner, ntop, has built a ready-to-use solution to collect network traffic NetFlow, IPFIX and sFlow messages that is stored in InfluxDB.

**Other network monitoring data**

This is just scratching the surface when it comes to network metrics and events. And you may find that there is monitoring data that you and your organization understand and would like to see added to your InfluxDB instance. What’s important to note is that as a set of open source projects, you can easily work with the community or contribute directly yourself to InfluxDB or to Telegraf. Telegraf’s popularity is due to its ease of use and deployment, use case versatility to support a number of datasources, and finally, to the ease to which you can contribute to a new Telegraf plugin that supports gathering the data that you need for your monitoring purposes. Writing your own Telegraf plugin is easy due to its well-documented and relatively simple process. With the source code being open and a vast library of plugins already implemented, there are a lot of examples to draw insights and guidance from. For a starting point, InfluxData provides a webinar about writing a Telegraf plugin.

**Conclusion**

Choosing a platform that can handle a unified monitoring strategy, accommodating the use-cases that address the needs of various organization constituencies, is fundamental. So is paying attention to efficiency, which can be an important aspect for IoT support. Monitoring starts with identifying the information that matters. After that, InfluxData platform provides the tools to move along the series of steps that comprises a comprehensive monitoring strategy: Collect > Store > Process > Visualize & Alert > Automate.

**About InfluxData**

InfluxData is the creator of InfluxDB, the open source time series database. Our technology is purpose-built to handle the massive volumes of time-stamped data produced by IoT devices, applications, networks, containers and computers. We are on a mission to help developers and organizations, such as Cisco, IBM, PayPal, and Tesla, store and analyze real-time data, empowering them to build transformative monitoring, analytics, and IoT applications quicker and to scale. InfluxData is headquartered in San Francisco with a workforce distributed throughout the U.S. and across Europe.

Learn more.
InfluxDB documentation, downloads & guides

Download InfluxDB
Get documentation
Additional case studies
Join the InfluxDB community