A new trend in the telecom industry is shifting where network functions are deployed - from hardware-based equipment to software applications running in a virtualized environment on standard high volume servers. Network service providers are beginning to deploy virtualized network elements in order to drive higher resource utilization by sharing hardware across multiple network functions. Additionally, by decoupling hardware and software elements, new services can be deployed more quickly once servers are in place because a new service gets deployed as a simple software push. Moreover, this software-focused model can yield dramatic cost savings by introducing telecom standard, high-volume equipment based on Intel® processors.

In support of this transition, Intel and Tail-f* have demonstrated how NETCONF can be used to manage a virtualized network edge element. In this proof point, a typical broadband edge workload was combined with a virtualized service and managed via Tail-f’s ConfD* and NCS* solution.

Tail-f*, a leading supplier of configuration management software, has extended this approach to make it easier for independent software vendors (ISVs) to integrate new software services into existing network infrastructure. Today, it is difficult to integrate new network devices into operation support system (OSS) environments found in most service provider networks. The paper describes how Tail-f overcomes these issues by creating an abstraction layer that simplifies data model mapping and interfaces between OSSs and ISV applications. The solution incorporates the NETCONF standard protocol and the YANG modeling language to increase the portability and reusability of interface software.
**Historical Perspective**

Telecom network operators began offering telephone service in the late 1870s, long before computers. Since early on, operator networks have served vital government and public interests; thus, they could never be turned off for upgrading, testing, etc. As a result, the network management aspects have evolved slowly and are presently very difficult to change. This includes the OSS, which supports functions for fault, configuration, accounting, performance and security (FCAPS) per the ISO Telecommunications Management Network (TMN) model.

OSS began as a manual, paper process. Orders for new services were written by hand and copies distributed to the dozen or so groups that needed the information. This process created a lot of paper records. When mainframe computers came along in the late 1980’s, the first task was to store these OSS records. Mainframes also enabled operators to computerize the transfer of the records from one step of the process to the next, thus automating many tasks, such as order processing, line assignment, fault management, asset tracking and billing.

Over time, customer-facing activities were split off and migrated to a new a system called the business support system (BSS). The OSS and BSS are critical systems that play complementary roles in the delivery of services spanning voice, data, VoIP and multi-media.

Eventually, each major department at a typical operator acquired its own mainframe, leading to a large hodgepodge of systems that required a lot of software development to create and maintain the interfaces. Moreover, a significant amount of effort was needed to stitch together workflows, especially as they grew longer and more complex. Since the entire system was live, turning it off to make changes was still not an option.

Today, much of this infrastructure still exists. There has been standardization on exchange formats (e.g., XML) across systems, but there hasn’t been much advancement in abstracting or creating models of various network elements that need to be managed. This means that for every new element added to the system, multiple interfaces must be supported for each stage of OSS, as depicted in Figure 1. Most of the interfaces are configured using vendor-specific command-line interfaces (CLI), which were never designed for machine-to-machine communication. This leads to a situation where the communication with the network resources is fragile, error-prone and expensive to maintain.

**Application Management Challenges**

It is common for third parties to deliver network functions to network service providers in the form of discrete appliances. Appliances perform a wide range of functions, including content delivery, WAN acceleration, security applications, video rendering and coding, application acceleration and routing. They usually have their own element management system (EMS), which communicates with the OSS. This model has scalability limitations, in that it requires a separate integration between the OSS and each new EMS. Furthermore, the OSS and EMS often have disparities (i.e., semantic mismatches) between data models and operation functions, which increase the scope of the integration task. Overall, a lengthy OSS integration can delay the introduction of new services.

Fast forward to the near future when network functions are now deployed as software loads where the goal is to introduce many more applications and services into the network. This introduces new challenges -

- to ISVs, who must understand, develop, test and maintain multiple management interfaces associated with different sets of network service provider requirements.
- to service providers, who must spend considerable time integrating to a much larger number of device interfaces as the number of new services deployed increases.
A Data Modeling Approach to Management
The aforementioned application management challenges can be addressed by a solution that uses standard data models northbound to the OSS and a simple method for integrating new elements on the southbound interface. The value of modeling approaches such as this one will become increasingly important as more and more services get added to the network in the form of virtualized software applications. Software applications delivered to network service providers as virtualized network elements need to support management capabilities on two different layers:

- Embedded management: The management agent framework is bundled together with the application to provide a manageable virtual appliance.
- Network abstraction layer: This layer interfaces different software- and hardware-based network devices to OSS systems and to human operators.

Figure 2 shows how these two management layers fit in the operational environment of a network service provider, using Tail-f's Network Control System* (NCS) as the network abstraction layer and Tail-f’s ConfD as the embedded management component. These layers are described in more detail in the following sections.

Virtualized Network Elements
The highly competitive telecom industry is driving network service providers to find faster ways to respond to evolving market requirements so they can make changes in days rather than the months typical of today’s infrastructure. This can be achieved with virtualized network elements that take advantage of recent Intel® microarchitecture enhancements, particularly those that significantly increase packet processing performance. Previously, service provider networks required specialized hardware to attain the necessary performance, but now software-based networking elements provide sufficient throughput at lower cost and reduced time-to-market for new services.

- Software-based solutions benefit from the global economies of scale of the IT industry. In addition, network functions delivered as software can be deployed in virtualized environments, such as the cloud, further improving the business case and increasing flexibility.
- Service providers can shorten their time-to-market for new services by developing an ecosystem of agile and innovative suppliers. Transitioning to software-based network elements is aligned to potentially more software suppliers than hardware-oriented solutions, thus tapping into a broader talent pool to speed up innovation.

For more information, read the “Software Defined Networking and Software Based Services” white paper on the Intel website.
Providing a path forward, Tail-f’s ConfD management solution enables an ISV to integrate its application to a carrier-grade management agent framework. The solution supports live reconfiguration capabilities and exposes standard northbound management protocols, such as the Network Configuration Protocol (NETCONF) and the Simple Network Management Protocol (SNMP), to the network management system, as depicted in Figure 3. One of the key advantages of providing the software application with a NETCONF management interface is that it enables plug-and-play integration with the network abstraction layer, requiring literally no integration code (see the sidebar “About NETCONF” for more information).

The Tail-f solution is also based on the YANG standard data modeling language that is described in IETF RFC 6020. The use of YANG essentially brings together all of the standard interfaces and various CLIs and greatly reduces the integration effort of network service providers (see the sidebar “About YANG” for more information).

The effort to integrate ConfD into an ISV application consists of:

- Using the YANG modeling language to write data models that describe the configurable parameters of the application
- Writing instrumentation code employing the ConfD API
  - This code subscribes to changes in ConfD datastore and takes appropriate action when configuration parameters are updated.

This integration requires about one man-week for a demo system and between a few man-weeks to a few man-months (depending on the size and complexity of the application) for a production-ready system. ISV’s can choose among different modules of ConfD, depending on their requirements, and use ConfD to either extend an existing management system or implement a new one. The solution enables developers to build carrier-grade applications in less time and with less risk, in large part because it uniquely renders all critical northbound interfaces (NETCONF, CLI, SNMP, REST and Web UI) from a single data model. Designed with a robust infrastructure, ConfD includes transaction management, high availability, security and role-based access control.

**About NETCONF**

NETCONF, published in December 2006, standardizes some essential management functions, such as installing, editing, querying and deleting the configuration of network devices. NETCONF operations are realized on top of a Remote Procedure Call (RPC) layer using an XML encoding scheme to support a basic set of operations. CLI scripting was the primary vehicle for automating configuration changes in the network prior to NETCONF. This approach has several limitations, including the lack of transaction management, no structured error management and frequently changing commands (i.e., structure and syntax) that make scripts fragile and costly to maintain.

Simple Network Management Protocol (SNMP) can also be used to change configurations, but its primary role is supporting performance and monitoring applications. Some SNMP drawbacks are the limitations associated with the lack of useful standard security and commit mechanisms, and the lack of a defined discovery process, which makes it hard to find the correct MIB (Management Information Base) modules.

The NETCONF protocol was designed to address the shortcomings of existing practices and protocols for configuration management, and its features include:

- Distinction between configuration and state data
- Multiple configuration data stores (candidate, running, startup)
- Configuration change transactions
- Configuration testing and validation support
- Selective data retrieval with filtering
- Streaming and playback of event notifications
- Extensible procedure call mechanism

![Figure 3. Tail-f* ConfD* Exposes Northbound Management Protocols](image)
Network abstraction layer

As mentioned earlier, each network element traditionally comes with its own element management system (EMS). Integrating an EMS into the OSS environment of a network service provider is a costly and time-consuming undertaking. With a shift towards software-based networking, this model is even less sustainable because it impedes the ability to quickly launch new network services from innovative ISVs. The key to solving this problem is to provide a network abstraction layer between the OSS environment and the network. This layer needs to perform two main tasks:

- Get the desired service configuration from the OSS environment; calculate the corresponding configuration changes in the network; and provision and activate the service; and deploy the configuration changes into the network.
- Collect status information and alarms from the network; map this to service-level status and alarms; and forward this information to performance and alarm management systems in the OSS environment.

Today, this layer is often realized as an assortment of manual processes, ad-hoc scripts and large systems integration projects. A more efficient approach is to implement this layer with a generic network abstraction engine, where the specifics of both services and network configurations are stringently defined in YANG data models.

This network abstraction engine needs to maintain a semantically rich database of both service instances and the current configuration of the network, including bi-directional relationships between service instances and the corresponding network configuration elements. It should also provide fail-safe service activation and provisioning to avoid having the network end up in an inconsistent state if something goes wrong, such as a network policy violation or an error when reconfiguring a device.

Tail-f’s Network Control System (NCS), depicted in Figure 4 on the following page, is such a network abstraction engine. It has the added value of providing zero-code interfaces to most network devices, allowing very quick turn-around times when upgrading network device software. ISV applications using ConfD as their management agent framework are plug-compatible with the NCS, involving no integration work at all.

The NCS greatly simplifies the mapping between data representations used by the OSS and the network elements, respectively. In other solutions, this data mapping requires a significant coding effort. All northbound management interfaces in the NCS are auto-rendered from common data models of both network elements and services. In addition to the advantage of rapid development time, auto-rendering ensures consistent capabilities are exposed across all interfaces.

About YANG

YANG is a data modeling language used to model configuration and state data manipulated by the NETCONF protocol. YANG was published as an IETF standard (RFC 6020) in September, 2010. YANG describes to the content and operations layers in NETCONF, as adopted by several organizations: the Metro Ethernet Forum for Carrier Ethernet, CableLabs® for DOCSIS Converged Multiservice Access Platforms, and the Open Networking Foundation for OpenFlow Configuration.

The rapid industry adoption of NETCONF made it a priority to define a data modeling language to complement NETCONF. Modeling languages such as SMI (SNMP), UML, XML Schema and others already existed. However, none of these languages were specifically targeted to the needs of configuration management. They lacked critical capabilities, like being easily read and understood by human implementers, and fell short in providing mechanisms to validate models of configuration data for semantics and syntax. Hence, YANG was developed, and some of its capabilities are:

- Human readable, easy to learn representation
- Hierarchical configuration data models
- Reusable types and groupings (structured types)
- Extensibility through augmentation mechanisms
- Supports the definition of operations (RPCs)
- Formal constraints for configuration validation
- Data modularity through modules and submodules
- Versioning rules and development support
Simplifying the Management of Virtualized Network Elements in Telecom

Benefits of Tail-f* Management Solutions
What makes Tail-f’s management solution unique is its ability to truly understand hardware- or software-based network elements and translate them into formats used by OSS systems. In other words, the solution can interpret any vendor’s CLI model and present it to network service providers in the form they want to use. The real value of the abstraction layer is its impressive grasp of object properties, which enables the OSS system to manipulate them with minimal integration effort. The Tail-f offering is a complete management solution that provides a wide range of benefits, including:

- Faster Development Time - auto-render management interfaces from a single data model
- Reduced Risk - mature software that is proven from development through deployment
- Support for Key Industry Standards – full NETCONF and YANG support, and SNMP Agent support for v1, v2c and v3
- High Availability - 1:N data replication
- Flexible and Extensible - management API allows development of additional interfaces
- Scalable Performance - support for symmetric multicore processing enabling ConfD and NCS to distribute processing loads over multiple cores to maximize performance

Simplifying Application Management
Until recently, ISVs and network service providers faced an uphill struggle when integrating new services into the OSS environment. Now, this process is greatly simplified with Tail-f management solutions based on the standard NETCONF protocol and the YANG data modeling language. As a result, network service providers can automate management functions, saving the large amount of time needed to write dedicated CLI scripts and mapping data models.

This capability paves the way for ISVs to develop virtualized network elements running on cost-effective, yet very high-performance Intel Xeon processor-based servers. By incorporating Tail-f management solution, ISVs can remove a critical hurdle for network service providers integrating new network elements into their OSS environment. As a result, services can be launched faster and at lower cost, ultimately providing a valuable competitive advantage for network service providers.

For more information about Tail-f Service Provider Solutions, please visit [http://www.tail-f.com](http://www.tail-f.com)

To learn more about Intel solutions for communications, please visit [www.intel.com/go/commsinfrastructure](www.intel.com/go/commsinfrastructure)