For Service Providers, the network is responsible for driving revenue streams. For Enterprises, the network is critical to all business operations.

Both organizations must implement network automation to simplify the process of provisioning and controlling network applications and services. Tail-f Multi-vendor solution, NCS, provides network service programmability to achieve important business objectives:

- Faster development and deployment of new network applications and services
- Faster turn-around times for new features
- Real-time dynamic network capacity allocation
- Better quality due to less human errors, less repetitive manual work
- Vendor independence
- Substantial capital and operating savings

Overview

The following are examples of new network applications and services that have been developed in NCS:

- Layer 3 VPNs
- Layer 2 VPNs
- E-LINE and E-LAN
- BGP management (including peering policy management)
- Layer 2 virtual networks
- Access Control Lists (ACLs) and firewall policy rules
- Layer 4-7 Service chaining
NCS provides a single network-wide interface to all network devices and all network applications and services, as well as a common modeling language and datastore for both services and devices.

A transaction engine handles transactions from the operations at the service layer to the actual deployment of configuration changes in the network.

Designed to be a generic solution, NCS supports the implementation of network applications and service on a wide variety of networking devices, both traditional hardware-based devices and virtual software appliances. NCS supports OpenFlow-based applications through a built-in OpenFlow controller, Tailflow; alternatively, applications built on top of other OpenFlow controllers can be easily integrated into NCS.

### Service-aware

NCS provides a specification of how a network service shall be applied to the network infrastructure. This greatly facilitates the mapping of service configuration changes to device configuration commands. The entire service life-cycle is supported including creating, modifying and deleting service instances.

### Model-driven

Services, device configurations and Openflow applications (flowlets) are specified in declarative YANG data models.

### Fail-safe

NCS applies all service changes towards the network as an atomic change-set, using distributed transactions. This ensures that the network is always in a consistent state and can automatically recover from failed configuration changes.

### Real-time

NCS maintains an accurate and synchronized copy of the network configuration state. Orchestration and management systems can be kept in sync with the network in real-time using the publish-subscribe APIs of NCS.