

NETCONF/YANG

- [Let's get started](#)
 - [What is NETCONF?](#)
 - [What is YANG?](#)
- [NETCONF](#)
 - [What is the difference?](#)
 - [Network Management Datastore Architecture \(RFC 8342\)](#)
 - [configuration datastore:](#)
 - [NETCONF Operations](#)
 - [How does it work?](#)
- [YANG](#)
 - [Build-in data types](#)
 - [Common data types](#)
- [References/Links](#)

Let's get started

What is NETCONF?

A network managing **protocol** to configure network elements.

It transports xml (json) via SSH.

```
<rpc-reply message-id="1" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
      <interface>
        <name>eth0</name>
        <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">ianaift:ethernetCsmacd</type>
        <enabled>true</enabled>
        <ipv6 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
          <address>
            <ip>2001:db8:c18:1::3</ip>
            <prefix-length>128</prefix-length>
          </address>
        </ipv6>
      </interface>
    </interfaces>
  </data>
</rpc-reply>
```

What is YANG?

A data modeling **language** for the definition of data sent over the NETCONF network configuration protocol.
It is a schema to validate xml.

```
container interfaces {
  description
    "Interface configuration parameters.";
  list interface {
    key "name";
    leaf name {
      type string;
    }
    leaf type {
      type identityref {
        base interface-type;
      }
      mandatory true;
    }
    leaf enabled {
      type boolean;
      default "true";
    }
  } // list interface
} // container interfaces
```

NETCONF

NETCONF was designed based on Operators Requirements, which are documented in [RFC3535](#). It addresses security topics and operational and maintenance topics.

What is the difference?

Protocol:	SNMP	NETCONF	SOAP	REST	RESTCONF	gRPC	What comes next ...
Transport Stack	UDP (connections)	SSH TCP	SSL HTTP TCP	SSL HTTP TCP	SSL HTTP TCP	HTTP/2 TCP	
Encoding	BER	XML (new JSON)	XML	XML, JSON	XML, JSON	binary	
Resources (filter mechanism)	OIDs	Path (XPath on server)		URLs	URLs	URLs	
Data models	MIBs	YANG modules	WSDL, XSD		YANG modules	Protocol Buffers	
Data Modeling Language	SMI	YANG	WSDL, XSD	Undefined, (WSDL), WADL, text...	YANG	Protocol Buffers	
Management Operations	SNMP operations	NETCONF operations	In the XML Schema, not standardized	HTTP operations	HTTP operations	HTTP/2 operations	
SDO (like)	IETF	IETF	W3C	W3C	IETF	Google	

Network Management Datastore Architecture (RFC 8342)

configuration datastore:

The datastore holding the complete set of configuration data that is required to get a device from its initial default state into a desired operational state.

running configuration datastore:

A configuration datastore (<running>) holding the complete configuration currently active on the device. The running configuration datastore always exists.

candidate configuration datastore:

A candidate datastore (<candidate>) that can be manipulated without impacting the device's current configuration and that can be committed to the running configuration datastore. Not all devices support a candidate configuration datastore.

startup configuration datastore:

The startup datastore (<startup>) holding the configuration loaded by the device when it boots. Only present on devices that separate the startup configuration datastore from the running configuration datastore.

operational datastore:

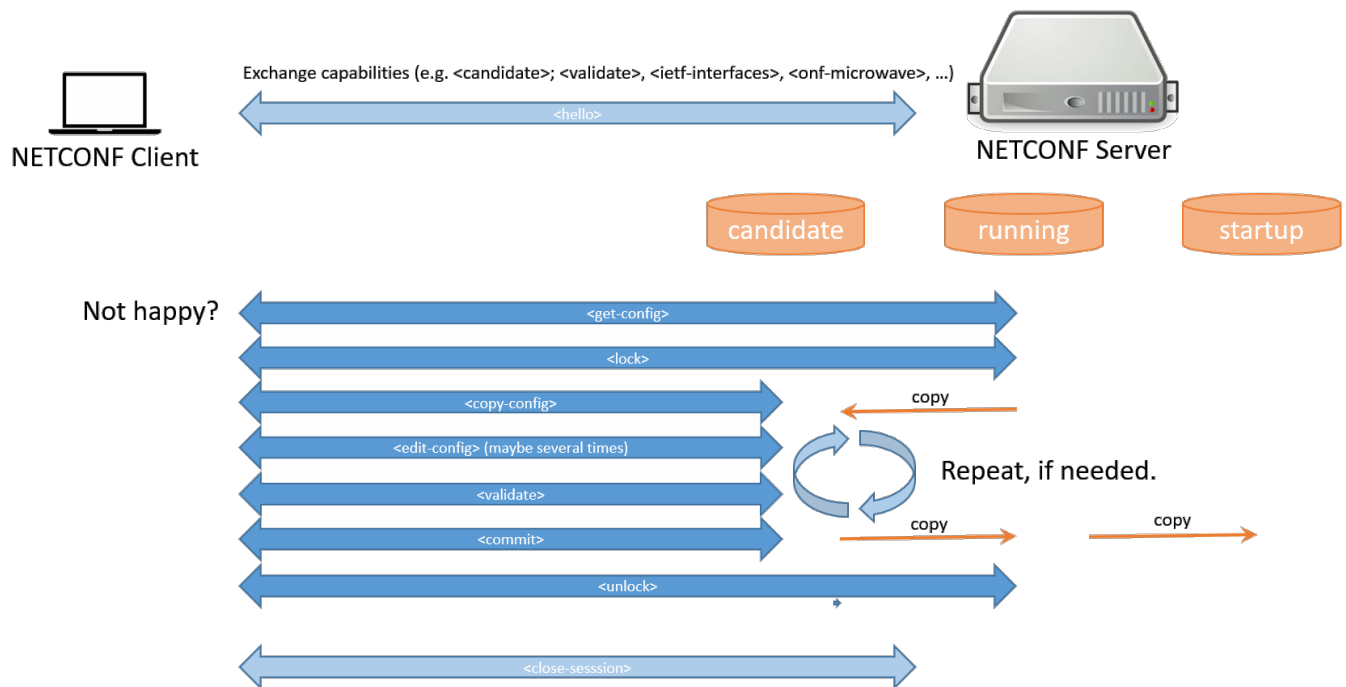
The operational state datastore (<operational>) is a read-only datastore that consists of all "config true" and "config false" nodes defined in the datastore's schema.

NETCONF Operations

Operation	Description
get	Retrieve running configuration and device state information.
get-config	Retrieve all or part of a specified configuration datastore.
edit-config	Loads all or part of a specified configuration to the specified target (<running>, <candidate>) configuration datastore. operation-types: merge, replace, create, delete, remove
copy-config	Create or replace an entire configuration datastore with the contents of another complete configuration datastore.
delete-config	Delete a configuration datastore. The <running> configuration datastore cannot be deleted.
(partly-) lock	It allows the client to lock the entire configuration datastore system of a device.
(partly-) unlock	Releases a configuration lock, previously obtained with the <lock> operation.
commit	Sets the running configuration to the current contents of the candidate configuration.
validate	This protocol operation validates the contents of the specified configuration.
close-session	Request graceful termination of a NETCONF session.
kill-session	Force the termination of a NETCONF session.
(hello-message)	(exchange of yang capabilities (yang modules) between server and client)

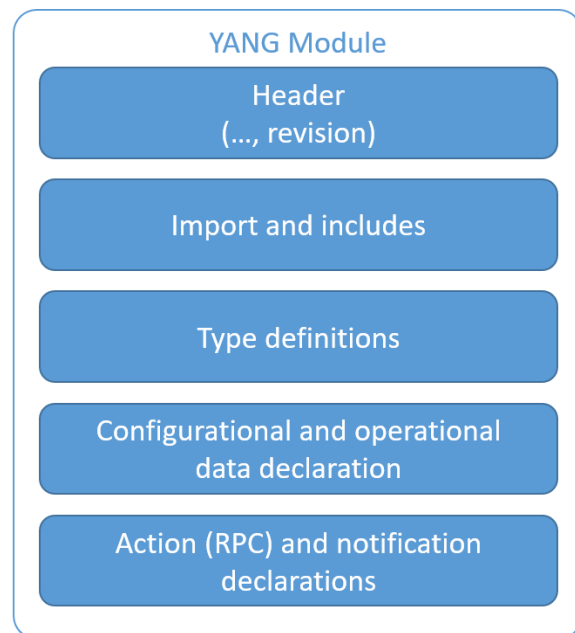
How does it work?

(one of x possibilities)



YANG

- YANG is a data modeling language for NETCONF (and RESTCONF) configuration and state data.
- It defines the syntax of the exchanged xml (and json) data between a NETCONF client (e.g. OpenDaylight) and a NETCONF server (a device)



Build-in data types

Name	Description
binary	Any binary data
bits	A set of bits or flags
boolean	"true" or "false"
decimal64	64-bit signed decimal number

Common data types

ietf-yang-types	Equivalent SMIv2 type
counter32	Counter32 (SNMPv2-SMI)
zero-based-counter32	ZeroBasedCounter32 (RMON2-MIB)
counter64	Counter64 (SNMPv2-SMI)

ietf-inet-types	Equivalent SMIv2 type
ip-version	InetVersion (INET-ADDRESS-MIB)
dscp	Dscp (DIFFSERV-DSCP-TC)

empty	A leaf that does not have any value
enumeration	Enumerated strings
identityref	A reference to an abstract identity
instance-identifier	References a data tree node
[u]int [8 16 32 64]	[8 16 32 64] -bit [un]signed integer
leafref	A reference to a leaf instance
string	Human-readable string
union	Choice of member types

zero-based-counter64	ZeroBasedCounter64 (HENUM-TC)
gauge32	Gauge32 (SNMPv2-SMI)
gauge64	CounterBasedGauge64 (HENUM-TC)
object-identifier	OBJECT IDENTIFIER
object-identifier-128	
yang-identifier	
date-and-time	
timeticks	TimeTicks (SNMPv2-SMI)
timestamp	TimeStamp (SNMPv2-TC)
phys-address	PhysAddress (SNMPv2-TC)
mac-address	MacAddress (SNMPv2-TC)
xpath1.0	
hex-string	
uuid	
dotted-quad	

ipv6-flow-label	IPv6FlowLabel (IPv6-FLOW-LABEL-MIB)
port-number	InetPortNumber (INET-ADDRESS-MIB)
as-number	InetAutonomousSystemNumber
ip-address	
ipv4-address	
ipv6-address	
ip-address-no-zone	
ipv4-address-no-zone	
ipv6-address-no-zone	
ip-prefix	
ipv4-prefix	
ipv6-prefix	
domain-name	
host	
uri	Uri (URI-TC-MIB)

References/Links

[RFC 3535](#) Overview of the 2002 IAB Network Management Workshop (see chapter 3 – Requirements by Network Operators)

[RFC 6241](#) Network Configuration Protocol (NETCONF)

[RFC 6020](#) YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)

[RFC 6991](#) Common YANG Data Types

[RFC 7950](#) The YANG 1.1 Data Modeling Language

[Overview](#) NETCONF and YANG Overview

[Tutorial](#) YANG Tutorial

[Tutorial](#) YANG Boot Camp

[pyang](#) An extensible YANG validator and converter in python

[pyang](#) pyang documentation