The **Network Configuration Protocol** (**NETCONF**) is a [network management](https://en.wikipedia.org/wiki/Network_management) protocol developed and standardized by the [IETF](https://en.wikipedia.org/wiki/Internet_Engineering_Task_Force). It was developed in the NETCONF working group[[1]](https://en.wikipedia.org/wiki/NETCONF#cite_note-1) and published in December 2006 as [RFC 4741](https://tools.ietf.org/html/rfc4741)[[2]](https://en.wikipedia.org/wiki/NETCONF#cite_note-2) and later revised in June 2011 and published as [RFC 6241](https://tools.ietf.org/html/rfc6241).[[3]](https://en.wikipedia.org/wiki/NETCONF#cite_note-3) The NETCONF protocol specification is an Internet Standards Track document.

NETCONF provides mechanisms to install, manipulate, and delete the configuration of network devices. Its operations are realized on top of a simple [Remote Procedure Call](https://en.wikipedia.org/wiki/Remote_Procedure_Call) (RPC) layer. The NETCONF protocol uses an [Extensible Markup Language](https://en.wikipedia.org/wiki/Extensible_Markup_Language) (XML) based data encoding for the configuration data as well as the protocol messages. The protocol messages are exchanged on top of a secure transport protocol.

The NETCONF protocol can be conceptually partitioned into four layers:

1. The Content layer consists of configuration data and notification data.
2. The Operations layer defines a set of base protocol operations to retrieve and edit the configuration data.
3. The Messages layer provides a mechanism for encoding remote procedure calls (RPCs) and notifications.
4. The Secure Transport layer provides a secure and reliable transport of messages between a client and a server.

The NETCONF protocol has been implemented in network devices such as routers and switches by some major equipment vendors. One particular strength of NETCONF is its support for robust configuration change using transactions involving a number of devices.

History[[edit](https://en.wikipedia.org/w/index.php?title=NETCONF&action=edit&section=1)]

The IETF developed the [Simple Network Management Protocol](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol) (SNMP) in the late 1980s and it proved to be a very popular [network management](https://en.wikipedia.org/wiki/Network_management) [protocol](https://en.wikipedia.org/wiki/Communications_protocol). In the early part of the 21st century it became apparent that in spite of what was originally intended, SNMP was not being used to configure network equipment, but was mainly being used for [network monitoring](https://en.wikipedia.org/wiki/Network_monitoring). In June 2002, the [Internet Architecture Board](https://en.wikipedia.org/wiki/Internet_Architecture_Board) and key members of the IETF's network management community got together with network operators to discuss the situation. The results of this meeting are documented in [RFC 3535](https://tools.ietf.org/html/rfc3535). It turned out that operators were primarily using proprietary [Command Line Interfaces](https://en.wikipedia.org/wiki/Command_Line_Interface) (CLI) to configure their devices. This had a number of features that the operators liked, including the fact that it was text-based, as opposed to the [BER-encoded](https://en.wikipedia.org/wiki/Basic_Encoding_Rules) SNMP. In addition, many equipment vendors did not provide the option to completely configure their devices via SNMP. As operators generally liked to write scripts to help manage their boxes, they found the SNMP CLI lacking in a number of ways. Most notably was the unpredictable nature of the output. The content and formatting of output was prone to change in unpredictable ways.

Around this same time, [Juniper Networks](https://en.wikipedia.org/wiki/Juniper_Networks) had been using an XML-based network management approach. This was brought to the IETF and shared with the broader community. Collectively, these two events led the IETF in May 2003 to the creation of the NETCONF working group. This working group was chartered to work on a network configuration protocol, which would better align with the needs of network operators and equipment vendors. The first version of the base NETCONF protocol was published as [RFC 4741](https://tools.ietf.org/html/rfc4741) in December 2006. Several extensions were published in subsequent years (notifications in [RFC 5277](https://tools.ietf.org/html/rfc5277) in July 2008, partial locks in [RFC 5717](https://tools.ietf.org/html/rfc5717) in December 2009, with-defaults in [RFC 6243](https://tools.ietf.org/html/rfc6243) in June 2011, system notifications in [RFC 6470](https://tools.ietf.org/html/rfc6470) in February 2012, access control in [RFC 6536](https://tools.ietf.org/html/rfc6536) in March 2012). A revised version of the base NETCONF protocol was published as [RFC 6241](https://tools.ietf.org/html/rfc6241) in June 2011.

## Protocol layers[[edit](https://en.wikipedia.org/w/index.php?title=NETCONF&action=edit&section=2)]

### Content**[**[**edit**](https://en.wikipedia.org/w/index.php?title=NETCONF&action=edit&section=3)**]**

The content of NETCONF operations is well-formed XML. Most content is related to [network management](https://en.wikipedia.org/wiki/Network_management). Subsequently, support for encoding in [JavaScript Object Notation](https://en.wikipedia.org/wiki/JavaScript_Object_Notation) (JSON) was also added.

The NETMOD working group has completed work to define a "human-friendly" modeling language for defining the semantics of operational data, configuration data, notifications, and operations, called [YANG](https://en.wikipedia.org/wiki/YANG). YANG is defined in [RFC 6020](https://tools.ietf.org/html/rfc6020) (version 1) and [RFC 7950](https://tools.ietf.org/html/rfc7950) (version 1.1), and is accompanied by the "Common YANG Data Types" found in [RFC 6991](https://tools.ietf.org/html/rfc6991).

During the summer of 2010, the NETMOD working group was re-chartered to work on core configuration models (system, interface, and routing) as well as work on compatibility with the [SNMP](https://en.wikipedia.org/wiki/SNMP) modeling language.

### Operations**[**[**edit**](https://en.wikipedia.org/w/index.php?title=NETCONF&action=edit&section=4)**]**

The base protocol defines the following protocol operations:

|  |  |
| --- | --- |
| **Operation** | **Description** |
| <get> | Retrieve running configuration and device state information |
| <get-config> | Retrieve all or part of a specified configuration datastore |
| <edit-config> | Edit a configuration datastore by creating, deleting, merging or replacing content |
| <copy-config> | Copy an entire configuration datastore to another configuration datastore |
| <delete-config> | Delete a configuration datastore |
| <lock> | Lock an entire configuration datastore of a device |
| <unlock> | Release a configuration datastore lock previously obtained with the <lock> operation |
| <close-session> | Request graceful termination of a NETCONF session |
| <kill-session> | Force the termination of a NETCONF session |

Basic NETCONF functionality can be extended by the definition of NETCONF capabilities. The set of additional protocol features that an implementation supports is communicated between the server and the client during the capability exchange portion of session setup. Mandatory protocol features are not included in the capability exchange since they are assumed. [RFC 4741](https://tools.ietf.org/html/rfc4741) defines a number of optional capabilities including :xpath and :validate. Note that [RFC 6241](http://tools.ietf.org/html/rfc6241) obsoletes [RFC 4741](https://tools.ietf.org/html/rfc4741).

A capability to support subscribing and receiving asynchronous event notifications is published in [RFC 5277](https://tools.ietf.org/html/rfc5277). This document defines the <create-subscription> operation, which enables creating real-time and replay subscriptions. Notifications are then sent asynchronously using the <notification> construct. It also defines the :interleave capability, which when supported with the basic :notification capability facilitates the processing of other NETCONF operations while the subscription is active.

A capability to support partial locking of the running configuration is defined in [RFC 5717](https://tools.ietf.org/html/rfc5717). This allows multiple sessions to edit non-overlapping sub-trees within the running configuration. Without this capability, the only lock available is for the entire configuration.

A capability to monitor the NETCONF protocol is defined in [RFC 6022](https://tools.ietf.org/html/rfc6022). This document contains a data model including information about NETCONF datastores, sessions, locks, and statistics that facilitates the management of a NETCONF server. It also defines methods for NETCONF clients to discover data models supported by a NETCONF server and defines the <get-schema> operation to retrieve them.

### Messages**[**[**edit**](https://en.wikipedia.org/w/index.php?title=NETCONF&action=edit&section=5)**]**

The NETCONF messages layer provides a simple, transport-independent framing mechanism for encoding

* RPC invocations (<rpc> messages),
* RPC results (<rpc-reply> messages), and
* event notifications (<notification> messages).

Every NETCONF message is a well-formed XML document. An RPC result is linked to an RPC invocation by a message-id attribute. NETCONF messages can be pipelined, i.e., a client can invoke multiple RPCs without having to wait for RPC result messages first. RPC messages are defined in [RFC 6241](https://tools.ietf.org/html/rfc6241) and notification messages are defined in [RFC 5277](https://tools.ietf.org/html/rfc5277).

### Transport**[**[**edit**](https://en.wikipedia.org/w/index.php?title=NETCONF&action=edit&section=6)**]**

* NETCONF Protocol over Secure Shell (SSH): rfc:6242
* NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication: rfc:7589

## See also[[edit](https://en.wikipedia.org/w/index.php?title=NETCONF&action=edit&section=7)]

* [YANG](https://en.wikipedia.org/wiki/YANG)
* Stefan Wallin (2014-10-18). [*NETCONF Tutorial*](https://www.youtube.com/watch?v=xoPZO1N-x38) (YouTube). Stockholm: tail-f.
* [Network management](https://en.wikipedia.org/wiki/Network_management)
* [Configuration management](https://en.wikipedia.org/wiki/Configuration_management)
* [Network monitoring](https://en.wikipedia.org/wiki/Network_monitoring)
* [XML Schema](https://en.wikipedia.org/wiki/XML_Schema_(W3C))

## References[[edit](https://en.wikipedia.org/w/index.php?title=NETCONF&action=edit&section=8)]

* 1. [**^**](https://en.wikipedia.org/wiki/NETCONF#cite_ref-1) [*"Network Configuration Working Group"*](https://tools.ietf.org/wg/netconf/). IETF.
  2. [**^**](https://en.wikipedia.org/wiki/NETCONF#cite_ref-2) Enns, Rob (2006). [*NETCONF Configuration Protocol*](https://tools.ietf.org/html/rfc4741) (Technical report). IETF. [*doi*](https://en.wikipedia.org/wiki/Doi_(identifier)):[*10.17487/RFC4741*](https://doi.org/10.17487%2FRFC4741). RFC4741.
  3. [**^**](https://en.wikipedia.org/wiki/NETCONF#cite_ref-3) Enns, Rob; Björklund, Martin; Schönwälder, Jürgen; Bierman, Andy (2011). [*Network Configuration Protocol (NETCONF)*](https://tools.ietf.org/html/rfc6241) (Technical report). IETF. [*doi*](https://en.wikipedia.org/wiki/Doi_(identifier)):[*10.17487/RFC6241*](https://doi.org/10.17487%2FRFC6241). RFC6241.