

目 录

1 NETCONF	1-1
1.1 NETCONF简介	1-1
1.1.1 NETCONF协议结构	1-1
1.1.2 NETCONF报文格式	1-2
1.1.3 如何使用NETCONF	1-3
1.1.4 协议规范	1-3
1.2 FIPS相关说明	1-3
1.3 NETCONF配置任务简介	1-3
1.4 建立NETCONF会话	1-4
1.4.1 配置限制和指导	1-4
1.4.2 配置NETCONF会话公共属性	1-5
1.4.3 建立NETCONF over SOAP会话	1-6
1.4.4 建立NETCONF over SSH会话	1-7
1.4.5 建立NETCONF over Telnet或NETCONF over Console会话	1-8
1.4.6 交换能力集	1-8
1.5 获取设备配置信息	1-9
1.5.1 配置限制和指导	1-9
1.5.2 获取运行配置和运行状态	1-9
1.5.3 获取可配置的功能的变量值	1-11
1.5.4 获取NETCONF信息	1-12
1.5.5 获取YANG文件的内容	1-13
1.5.6 获取NETCONF会话信息	1-13
1.5.7 获取运行配置举例	1-14
1.5.8 获取可配置的功能的变量值配置举例	1-15
1.5.9 获取Syslog功能的变量值配置举例	1-16
1.5.10 获取NETCONF会话信息配置举例	1-17
1.6 过滤表和列的信息	1-18
1.6.1 功能简介	1-18
1.6.2 配置限制和指导	1-19
1.6.3 基于表的过滤	1-19
1.6.4 基于列的过滤	1-19
1.6.5 正则表达式过滤配置举例	1-22
1.6.6 条件匹配过滤配置举例	1-23

1.7 配置加锁和解锁.....	1-24
1.7.1 功能简介.....	1-24
1.7.2 配置限制和指导.....	1-24
1.7.3 配置加锁.....	1-25
1.7.4 配置解锁.....	1-25
1.7.5 加锁配置举例.....	1-25
1.8 下发配置.....	1-26
1.8.1 功能简介.....	1-26
1.8.2 配置步骤.....	1-27
1.8.3 下发配置举例.....	1-27
1.9 保存配置.....	1-28
1.9.1 功能简介.....	1-28
1.9.2 配置限制和指导.....	1-28
1.9.3 配置步骤.....	1-28
1.9.4 保存配置举例.....	1-29
1.10 加载配置文件.....	1-30
1.10.1 功能简介.....	1-30
1.10.2 配置限制和指导.....	1-30
1.10.3 配置步骤.....	1-30
1.11 回滚配置.....	1-31
1.11.1 配置限制和指导.....	1-31
1.11.2 回滚配置文件.....	1-31
1.11.3 回滚运行配置.....	1-31
1.12 下发命令行.....	1-35
1.12.1 功能简介.....	1-35
1.12.2 配置限制和指导.....	1-35
1.12.3 命令行操作.....	1-35
1.12.4 命令行操作配置举例.....	1-36
1.13 订阅事件.....	1-37
1.13.1 功能简介.....	1-37
1.13.2 配置限制和指导.....	1-37
1.13.3 订阅Syslog事件.....	1-37
1.13.4 订阅监控事件.....	1-38
1.13.5 订阅模块上报事件.....	1-39
1.13.6 订阅事件配置举例.....	1-41
1.14 关闭NETCONF会话.....	1-42

1.14.1 功能简介	1-42
1.14.2 Kill-session操作	1-42
1.14.3 Kill-session操作配置举例	1-42
1.15 退出XML视图.....	1-43
2 Comware V7 支持的NETCONF操作	2-1
2.1.1 action.....	2-1
2.1.2 CLI	2-1
2.1.3 close-session.....	2-2
2.1.4 edit-config: create	2-2
2.1.5 edit-config: delete	2-3
2.1.6 edit-config: merge	2-3
2.1.7 edit-config: remove	2-3
2.1.8 edit-config: replace.....	2-4
2.1.9 edit-config测试处理选项	2-4
2.1.10 edit-config缺省操作选项	2-5
2.1.11 edit-config缺省错误处理选项.....	2-5
2.1.12 edit-config增量下发	2-6
2.1.13 get.....	2-7
2.1.14 get-bulk	2-7
2.1.15 get-bulk-config.....	2-8
2.1.16 get-config	2-8
2.1.17 get-sessions	2-9
2.1.18 kill-session	2-9
2.1.19 load	2-9
2.1.20 lock.....	2-10
2.1.21 rollback.....	2-10
2.1.22 save.....	2-10
2.1.23 unlock	2-11

1 NETCONF

1.1 NETCONF简介

NETCONF (Network Configuration Protocol, 网络配置协议) 是一种基于 XML 的网络管理协议, 他提供了一种可编程的、对网络设备进行配置和管理的方法。用户可以通过该协议设置属性、获取属性值、获取统计信息等。这使得他在第三方软件的开发上非常便利, 很容易开发出在混合不同厂商、不同设备的环境下的特殊定制的网管软件。

1.1.1 NETCONF协议结构

NETCONF 协议采用分层结构, 分为内容层 (Content)、操作层 (Operations)、RPC (Remote Procedure Call, 远程调用) 层和通信协议层 (Transport Protocol) 等。

表1-1 XML 分层与 NETCONF 分层模型对应关系

NETCONF 分层	XML 分层	说明
内容层	配置数据、状态数据、统计信息等	被管理对象的集合, 可以是配置数据、状态数据、统计信息等 NETCONF协议具体可读写的的数据请参见《NETCONF XML API 手册》
操作层	<get>,<get-config>,<edit-config>...	在RPC中应用的基本的原语操作集, 这些操作组成NETCONF的基本能力 NETCONF全面地定义了对被管理设备的各种基础操作, 设备支持的操作请参见“ 2 Comware V7支持的NETCONF操作 ”
RPC层	<rpc>,<rpc-reply>	为RPC模块的编码提供了简单的、传输协议无关的机制。通过使用<rpc>和<rpc-reply>元素分别对NETCONF请求和响应数据 (即操作层和内容层的内容) 进行封装
通信协议层	非FIPS模式下: Console/Telnet/SSH/HTTP/HTTPS/TLS FIPS模式下: Console/SSH/HTTP S/TLS	为NETCONF提供面向连接的、可靠的、顺序的数据链路。 非FIPS模式下: <ul style="list-style-type: none">NETCONF 支持 Telnet、SSH 和 Console 等 CLI 登录方式/协议, 即 NETCONF over SSH、NETCONF over Telnet 和 NETCONF over ConsoleNETCONF 支持封装成 SOAP (Simple Object Access Protocol, 简单对象访问协议) 报文后通过 HTTP 或 HTTPS 协议传输, 即 NETCONF over SOAP over HTTP 和 NETCONF over SOAP over HTTPS FIPS模式下: <ul style="list-style-type: none">NETCONF 支持 SSH 和 Console 等 CLI 方式/协议, 即 NETCONF over SSH 和 NETCONF over ConsoleNETCONF 支持封装成 SOAP 报文后通过 HTTPS 协议传输, 即 NETCONF over SOAP over HTTPS

1.1.2 NETCONF报文格式

1. NETCONF

NETCONF 命令必须符合 XML 语言的基本格式，格式遵循 RFC 4741。

NETCONF 操作以及可操作的数据项，请参见《NETCONF XML API 手册》。NETCONF 报文的数据合法性都将经过校验才会下发，如果校验失败则会向客户端报错。其中，数据合法性校验通过 XML Schema 的方式完成。

如下为一个 NETCONF 报文示例，用于获取设备上所有接口的所有参数：

```
<?xml version="1.0" encoding="utf-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface/>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get-bulk>
</rpc>
```

2. NETCONF over SOAP

NETCONF over SOAP 之后，NETCONF 报文会放在 SOAP 报文的 BODY 元素里，这些报文除了需要遵循纯 NETCONF 报文的规则外，还需要遵循以下规则：

- SOAP 消息必须用 XML 来编码。
- SOAP 消息必须使用 SOAP Envelope 命名空间。
- SOAP 消息必须使用 SOAP Encoding 命名空间。
- SOAP 消息不能包含 DTD（Document Type Definition，文件类型定义）引用。
- SOAP 消息不能包含 XML 处理指令。

如下为一个 NETCONF over SOAP 报文示例，用于获取设备上所有接口的所有参数：

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <auth:Authentication env:mustUnderstand="1"
      xmlns:auth="http://www.h3c.com/netconf/base:1.0">
      <auth:AuthInfo>800207F0120020C</auth:AuthInfo>
    </auth:Authentication>
  </env:Header>
  <env:Body>
    <rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <get-bulk>
        <filter type="subtree">
          <top xmlns="http://www.h3c.com/netconf/data:1.0">
            <Ifmgr>
```

```

        <Interfaces>
          <Interface/>
        </Interfaces>
      </Ifmgr>
    </top>
  </filter>
</get-bulk>
</rpc>
</env:Body>
</env:Envelope>

```

1.1.3 如何使用NETCONF

用户可通过以下方式使用 NETCONF 配置和管理设备：

- 通过 Telnet、SSH 和 Console 登录到设备并进入 XML 视图，将合法的 NETCONF 报文直接拷贝、粘贴到命令行提示符处执行，即可实现对设备的配置和管理。
- 使用配置工具给设备下发 NETCONF 指令来实现对设备的配置和管理。该方式需要将 NETCONF 指令用 SOAP 封装成通用的格式，以便设备能够正常转换。因此，使用该方式前必须开启 NETCONF over SOAP 功能。

1.1.4 协议规范

与 NETCONF、SOAP 协议相关的协议规范有：

- RFC 3339: Date and Time on the Internet: Timestamps
- RFC 4741: NETCONF Configuration Protocol
- RFC 4742: Using the NETCONF Configuration Protocol over Secure SHell (SSH)
- RFC 4743: Using NETCONF over the Simple Object Access Protocol (SOAP)
- RFC 5277: NETCONF Event Notifications
- RFC 5381: Experience of Implementing NETCONF over SOAP
- RFC 5539: NETCONF over Transport Layer Security (TLS)
- RFC 6241: Network Configuration Protocol

1.2 FIPS相关说明

设备运行于 FIPS 模式时，本特性部分配置相对于非 FIPS 模式有所变化，具体差异请见本文相关描述。有关 FIPS 模式的详细介绍请参见“安全配置指导”中的“FIPS”。

1.3 NETCONF配置任务简介

NETCONF 配置任务如下：

- (1) [建立NETCONF会话](#)
 - a. （可选）[配置NETCONF会话公共属性](#)
 - b. [建立NETCONF over SOAP会话](#)
 - c. [建立NETCONF over SSH会话](#)

- d. [建立NETCONF over Telnet或NETCONF over Console会话](#)
- e. [交换能力集](#)
- (2) (可选) [获取设备配置信息](#)
 - o [获取运行配置和运行状态](#)
 - o [获取可配置的功能的变量值](#)
 - o [获取NETCONF信息](#)
 - o [获取YANG文件的内容](#)
 - o [获取NETCONF会话信息](#)
- (3) (可选) [过滤表和列的信息](#)
 - o [基于表的过滤](#)
 - o [基于列的过滤](#)
- (4) (可选) [配置加锁和解锁](#)
 - a. [配置加锁](#)
 - b. [配置解锁](#)
- (5) (可选) [下发配置](#)
- (6) (可选) 配置文件管理
 - o [保存配置](#)
 - o [加载配置文件](#)
 - o [回滚配置](#)
- (7) (可选) [下发命令行](#)
- (8) (可选) [订阅事件](#)
 - o [订阅Syslog事件](#)
 - o [订阅监控事件](#)
 - o [订阅模块上报事件](#)
- (9) (可选) [关闭NETCONF会话](#)
- (10) (可选) [退出XML视图](#)

1.4 建立NETCONF会话

1.4.1 配置限制和指导

建立 NETCONF 会话后，客户端先与设备进行能力集交互，完成能力集的交互后，设备才会处理客户端发送的其他请求。

多个用户同时配置设备时，可能会导致用户配置与配置结果不一致，因此，请避免多个用户同时配置设备。

设备同一时间内允许建立的最大会话数可以通过 `aaa session-limit` 命令配置，关于该命令的详细描述，请参见“安全配置指导”中的“AAA”。用户数超过上限后，新登录的用户将登录失败。

1.4.2 配置NETCONF会话公共属性

1. 功能简介

NETCONF 支持共用命名空间和专用命名空间，两种类型的命名空间互不兼容。客户端与设备必须使用相同的命名空间才能建立会话。如果客户端不支持共用命名空间，则需要配置本功能，使设备和客户端建立会话时使用专用命名空间。

- 共用命名空间：各模块共用一个命名空间。缺省情况下，设备使用共用命名空间。使用共用命名空间的报文中，命名空间位于<top>元素中，模块位于<top>元素下，报文示例如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get-bulk>
</rpc>
```

- 专用命名空间：每个模块单独使用一个命名空间。使用专用命名空间的报文中无<top>元素，命名空间位于模块名后，报文示例如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-bulk>
    <filter type="subtree">
      <Ifmgr xmlns="http://www.h3c.com/netconf/data:1.0-Ifmgr">
        <Interfaces>
        </Interfaces>
      </Ifmgr>
    </filter>
  </get-bulk>
</rpc>
```

2. 配置步骤

- (1) 进入系统视图。

system-view

- (2) 配置 NETCONF 会话超时时间。

netconf { agent | soap } idle-timeout minute

属性	说明
agent	agent 表示如下NETCONF会话类型： <ul style="list-style-type: none">NETCONF over SSHNETCONF over TelnetNETCONF over Console 缺省超时时间为0（不超时）

属性	说明
soap	soap 表示如下NETCONF会话类型： <ul style="list-style-type: none"> NETCONF over SOAP over HTTP NETCONF over SOAP over HTTPS 缺省超时时间为10分钟

- (3) 开启 NETCONF 日志功能。

```
netconf log source { all | { agent | soap } * } { protocol-operation { all
| { action | config | get | set | session | syntax | others } * } |
row-operation | verbose }
```

缺省情况下，NETCONF 日志功能处于关闭状态。

- (4) 配置设备使用专用命名空间。

```
netconf capability specific-namespace
```

缺省情况下，设备使用共用命名空间。

配置 NETCONF 使用专用命名空间后，需要重新建立 NETCONF 会话，使客户端和设备使用相同的命名空间。

1.4.3 建立NETCONF over SOAP会话

1. 功能简介

NETCONF 支持封装成 SOAP 报文后通过 HTTP 或 HTTPS 协议传输，即 NETCONF over SOAP over HTTP 和 NETCONF over SOAP over HTTPS，使用该功能后，用户可以通过配置工具给设备下发 NETCONF 指令来实现对设备的访问。

2. 配置限制和指导

用户可以配置在 SOAP 请求的<UserName>中携带认证域信息，该信息仅对当前请求生效。通过 **netconf soap domain** 命令配置强制认证域后，SOAP 请求中携带的认证域不生效。

3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 开启 NETCONF over SOAP 功能。

（非 FIPS 模式）

```
netconf soap { http | https } enable
```

（FIPS 模式）

```
netconf soap https enable
```

缺省情况下，NETCONF over SOAP 处于关闭状态。

- (3) 配置设备发送的 SOAP 报文的 DSCP 优先级。

（非 FIPS 模式）

```
netconf soap { http | https } dscp dscp-value
```

（FIPS 模式）

```
netconf soap https dscp dscp-value
```

缺省情况下，设备发送的 SOAP 报文的 DSCP 优先级为 0。

- (4) 配置 NETCONF over SOAP 关联 IPv4 ACL，只有 IPv4 ACL 允许通过的客户端可以与设备建立 NETCONF over SOAP 会话。

（非 FIPS 模式）

```
netconf soap { http | https } acl { ipv4-acl-number | name ipv4-acl-name }
```

（FIPS 模式）

```
netconf soap https acl { ipv4-acl-number | name ipv4-acl-name }
```

缺省情况下，未配置 NETCONF over SOAP 关联 IPv4 ACL。

- (5) 配置 NETCONF 用户的强制认证域。

```
netconf soap domain domain-name
```

缺省情况下，未配置 NETCONF 用户的强制认证域。

配置该功能后，所有用户都会使用强制认证域进行认证。关于认证域的详细介绍请参见“安全配置指导”中的“AAA”。

- (6) 通过配置工具与设备建立 NETCONF over SOAP 会话。关于配置工具的使用方法，具体参见配置工具的配置指导。

1.4.4 建立NETCONF over SSH会话

1. 配置限制和指导

进入 XML 视图后，用户需要严格按照 NETCONF 报文格式将报文拷贝、粘贴到 XML 视图中，请勿手工输入 NETCONF 报文。

在一次 NETCONF 操作完成前，请勿进行其他任何操作（例如粘贴 NETCONF 报文或输入回车），否则可能导致 NETCONF 配置失败。

在 XML 视图下进行 NETCONF 配置时，NETCONF 报文最后需要添加“]]>]]>”结束符，否则设备无法识别。本手册举例中，为方便识别 XML 格式，均未添加该结束符，实际操作中请自行添加。

2. 配置准备

通过命令行建立 NETCONF over SSH 会话前，请先通过 SSH 登录到设备。

3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 开启 NETCONF over SSH。

```
netconf ssh server enable
```

缺省情况下，NETCONF over SSH 处于关闭状态。

- (3) 配置 NETCONF over SSH 的监听端口。

```
netconf ssh server port port-number
```

缺省情况下，NETCONF over SSH 的监听端口为 830。

- (4) 建立 NETCONF over SSH 会话。请选择其中一项进行配置。

- 通过配置工具与设备建立 NETCONF over SSH 会话。关于配置工具的使用方法，具体参见配置工具的配置指导。
- 请依次执行以下命令建立 NETCONF over SSH 会话。

```
quit
```

```
xml
```

进入 XML 视图，即表示成功建立 NETCONF over SSH 会话。

1.4.5 建立NETCONF over Telnet或NETCONF over Console会话

1. 配置限制和指导

进入 XML 视图后，用户需要严格按照 NETCONF 报文格式将报文拷贝、粘贴到 XML 视图中，请勿手工输入 NETCONF 报文。

在一次 NETCONF 操作完成前，请勿进行其他任何操作（例如粘贴 NETCONF 报文或输入回车），否则可能导致 NETCONF 配置失败。

在 XML 视图下进行 NETCONF 配置时，NETCONF 报文最后需要添加“]]>]]>”结束符，否则设备无法识别。本手册举例中，为方便识别 XML 格式，均未添加该结束符，实际操作中请自行添加。

2. 配置准备

建立 NETCONF over Telnet 或 NETCONF over Console 会话前，请先通过 Telnet 或 Console 口登录到设备。

3. 配置步骤

请在用户视图下执行本命令，进入 XML 视图。

```
xml
```

进入 XML 视图，即表示成功建立 NETCONF over Telnet 或 NETCONF over Console 会话。

1.4.6 交换能力集

1. 功能简介

建立 NETCONF 会话后，客户端和设备必须交换各自支持的能力集，双方收到对方的能力集后才可以进行下一步操作。

2. 设备发送给客户端的报文

建立 NETCONF 会话后，设备会发送如下报文自动告知客户端支持的 NETCONF 能力集：

```
<?xml version="1.0" encoding="UTF-8"?><hello
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"><capabilities><capability>urn:ietf:params:netconf:base:1.1</capability><capability>urn:ietf:params:netconf:writable-running</capability><capability>urn:ietf:params:netconf:capability:notification:1.0</capability><capability>urn:ietf:params:netconf:capability:validate:1.1</capability><capability>urn:ietf:params:netconf:capability:interleave:1.0</capability><capability>urn:h3c:params:netconf:capability:h3c-netconf-ext:1.0</capability></capabilities><session-id>1</session-id></hello>]]>]]>
```

<capabilities>和</capabilities>之间的内容表示设备支持的能力集，以具体设备实际情况为准。

<session-id>和</session-id>之间的内容表示为本次会话分配的会话 ID，用来唯一标识本次会话。

3. 客户端发送给设备的报文

客户端收到设备发送的能力集协商报文后，需要给设备发送如下格式的报文，告知设备客户端支持哪些 NETCONF 能力集。

Hello 协商报文格式如下：

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
      capability-set
    </capability>
  </capabilities>
</hello>
```

属性	说明
capability-set	客户端支持的能力集，由用户定义 一个<capability>和</capability>选项对中只能填写一个能力集，可以使用多个选项对，配置多个能力集

1.5 获取设备配置信息

1.5.1 配置限制和指导

用户执行<get>、<get-bulk>、<get-config>或<get-bulk-config>操作获取设备数据时，如果所获取数据中包含 NETCONF 不能识别的字符，则 NETCONF 将该字符转换为“?”输出到客户端。

<get><netconf-state/></get>操作不支持数据过滤。

关于获取设备配置信息操作的详细规则，请参见《NETCONF XML API 手册》。

1.5.2 获取运行配置和运行状态

设备支持通过<get>操作获取设备运行状态和运行配置。

设备支持通过<get-bulk>操作从指定索引的下一条开始获取 N 条运行状态和运行配置（索引行数据不返回）。

<get>操作会返回所有符合条件的数据，在某些情况下，会导致获取数据效率不高。<get-bulk>允许用户从固定数据项开始，向后获取指定条目的数据记录。

<get>和<get-bulk>报文的通用格式如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <getoperation>
    <filter>
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        指定模块，子模块，表名，列名
      </top>
    </filter>
  </getoperation>
</rpc>
```

属性	说明
getoperation	可以为get或者get-bulk
filter	<p>用于过滤信息，<filter>中可包括模块名、子模块名、表名和列名：</p> <ul style="list-style-type: none"> 如果不指定模块（子模块），则表示全部模块（子模块）。指定模块（子模块）时，则返回数据只包含指定模块（子模块）。 如果模块下不指定表，则表示全部表。指定表时，则返回数据只包含指定表 如果只指定索引列，则返回的数据包括全部的列。如果同时指定索引列之外的其他列，则返回的数据仅包含索引列和指定的列

- <get-bulk>操作报文中可以携带 count 和 index 属性。

属性	说明
index	指定索引。如未指定索引，则以第一条操作为索引
count	<p>获取指定数量的运行状态和运行配置信息。count属性遵循如下约定：</p> <ul style="list-style-type: none"> count属性的位置在可以从top下的节点开始，到表节点为止，即：模块节点，表节点这几个位置都能放置 count 属性，其他位置的 count 将不被解释 如果 count 放在模块节点上，则报文中指定的子孙节点（表）中没有 count 的都默认 count 属性的值和模块一致 如果不指定 count，或者数据表中符合条件的数据记录小于 count，则获取从指定索引开始的所有数据

如下为一个携带了 count 和 index 属性的<get-bulk>操作报文示例：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:xc="http://www.h3c.com/netconf/base:1.0">
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0"
xmlns:base="http://www.h3c.com/netconf/base:1.0">
        <Syslog>
          <Logs xc:count="5">
            <Log>
              <Index>10</Index>
            </Log>
          </Logs>
        </Syslog>
      </top>
    </filter>
  </get-bulk>
</rpc>
```

- 获取接口数据时，如果 IfIndex 和 vrfindex 元素的值为数字，设备无法识别该值为名称类型还是索引类型。此时，用户可以使用 valuetype 指定该值的类型。valuetype 取值为：

属性	说明
name	值为名称类型
index	值为索引类型
auto	设备先按名称类型进行匹配，如果没有匹配到任何信息，再按照索引类型进行匹配 如果不指定valuetype的属性，缺省使用 auto

下面以 **IfIndex** 元素值为 **index** 类型，值为 1 为例：

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <getoperation>
    <filter>
      <top xmlns="http://www.h3c.com/netconf/config:1.0"
xmlns:base="http://www.h3c.com/netconf/base:1.0">
        <VLAN>
          <TrunkInterfaces>
            <Interface>
              <IfIndex base:valuetype="index">1</IfIndex>
            </Interface>
          </TrunkInterfaces>
        </VLAN>
      </top>
    </filter>
  </getoperation>
</rpc>
```

设备收到配置获取请求报文后会将相应属性的值通过如下报文反馈给客户端：

```
<?xml version="1.0"?>
<rpc-reply message-id="100"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    全部配置数据和状态数据
  </data>
</rpc-reply>
```

1.5.3 获取可配置的功能的变量值

设备支持通过<get-config>和<get-bulk-config>操作获取设备所有可配置的功能的变量值。

<get-config>和<get-bulk-config>操作报文中可以包含子标签<filter>，用来对要获取的信息进行过滤。

<get-config>和<get-bulk-config>的通用报文格式为：

```
<?xml version="1.0"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
  <filter>
```

```
<top xmlns="http://www.h3c.com/netconf/config:1.0">
    指定模块，子模块，表名，列名
</top>
</filter>
</get-config>
</rpc>
设备收到配置获取请求报文后会将相应配置通过如下报文反馈给客户端：
<?xml version="1.0"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <data>
        所有指定 filter 内的数据
    </data>
</rpc-reply>
```

1.5.4 获取NETCONF信息

<get><netconf-state/></get>报文的通用格式为：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="m-641" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<get>
<filter type='subtree'>
<netconf-state xmlns='urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring'>
    <getType/>
</netconf-state>
</filter>
</get>
</rpc>
```

其中，*getType* 可以为 **capabilities**、**datastores**、**schemas**、**sessions** 或者 **statistics**：

- 不指定 *getType* 时，该操作则获取 **NETCONF** 全部信息。
- 当指定 *getType* 时，该操作仅返回相应类型的应答数据。其中，*getType* 取值为：

属性	说明
capabilities	该操作用来获取设备能力集
datastores	该操作用来获取设备中的数据库
schemas	该操作用来获取设备中的YANG文件名称列表
sessions	该操作用来获取设备中的会话信息
statistics	该操作用来获取NETCONF的统计信息

设备收到 **NETCONF** 信息获取请求报文后，将相应属性的值通过如下报文反馈给客户端：

```
<?xml version="1.0"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <data>
        全部 NETCONF 全部相关信息
    </data>
</rpc-reply>
```

1.5.5 获取YANG文件的内容

YANG 文件中保存了设备支持的 NETCONF 操作，用户通过获取、并分析 YANG 文件的内容，从而获知设备支持哪些 NETCONF 操作。

YANG文件集成在设备软件中，命名格式为`yang_identifier@yang_version.yang`，用户不能通过`dir`命令显示YANG文件名，关于如何获取设备上YANG文件名称，详细请参见“[1.5.4 获取NETCONF信息](#)”。

如下报文要求返回名称为 `syslog-data@2017-01-01.yang` 的 YANG 文件中的数据：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-schema xmlns='urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring'>
    <identifier>syslog-data</identifier>
    <version>2017-01-01</version>
    <format>yang</format>
  </get-schema>
</rpc>
```

设备收到 YANG 文件获取请求报文后，将相应属性的值通过如下报文反馈给客户端：

```
<?xml version="1.0"?>
<rpc-reply message-id="100"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    指定 YANG 文件的内容
  </data>
</rpc-reply>
```

1.5.6 获取NETCONF会话信息

使用该功能用户可以获取当前设备的所有 NETCONF 会话信息。

获取 NETCONF 会话信息的格式为：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-sessions/>
</rpc>
```

设备收到命令行指令后会回应客户端，当客户端收到如下报文时，表示命令行执行成功：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-sessions>
    <Session>
      <SessionID>用户会话 ID 信息</SessionID>
      <Line>line 信息</Line>
      <UserName>用户登录名称</UserName>
      <Since>用户登录时间</Since>
      <LockHeld>用户是否持有锁</LockHeld>
    </Session>
  </get-sessions>
</rpc-reply>
```


1.5.7 获取运行配置举例

1. 组网需求

获取接口表的一条数据

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.0</capability>
  </capabilities>
</hello>
```

获取接口表的一条数据。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0"
xmlns:web="http://www.h3c.com/netconf/base:1.0">
        <Ifmgr>
          <Interfaces web:count="1">
        </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get-bulk>
</rpc>
```

3. 验证配置

如果客户端收到类似如下的报文，则表示操作成功：

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="100">
  <data>
    <top xmlns="http://www.h3c.com/netconf/data:1.0">
      <Ifmgr>
        <Interfaces>
          <Interface>
            <IfIndex>3</IfIndex>
            <Name>GigabitEthernet1/0/2</Name>
            <AbbreviatedName>GE1/0/2</AbbreviatedName>
            <PortIndex>3</PortIndex>
            <ifTypeExt>22</ifTypeExt>
            <ifType>6</ifType>
```

```

        <Description>GigabitEthernet1/0/2 Interface</Description>
        <AdminStatus>2</AdminStatus>
        <OperStatus>2</OperStatus>
        <ConfigSpeed>0</ConfigSpeed>
        <ActualSpeed>100000</ActualSpeed>
        <ConfigDuplex>3</ConfigDuplex>
        <ActualDuplex>1</ActualDuplex>
    </Interface>
</Interfaces>
</Ifmgr>
</top>
</data>
</rpc-reply>

```

1.5.8 获取可配置的功能的变量值配置举例

1. 组网需求

获取所有可配置的功能的变量值。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

```

<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
      urn:ietf:params:netconf:base:1.0
    </capability>
  </capabilities>
</hello>

```

获取模块所有配置数据。

请将以下报文拷贝、粘贴到客户端：

```

<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
  </get-config>
</rpc>

```

3. 验证配置

如果客户端收到类似如下的报文，则表示操作成功：

```

<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="100">
  <data>
    <top xmlns="http://www.h3c.com/netconf/config:1.0">

```

```

        <Ifmgr>
        <Interfaces>
        <Interface>
        <IfIndex>1307</IfIndex>
        <Shutdown>1</Shutdown>
        </Interface>
        <Interface>
        <IfIndex>1308</IfIndex>
        <Shutdown>1</Shutdown>
        </Interface>
        <Interface>
        <IfIndex>1309</IfIndex>
        <Shutdown>1</Shutdown>
        </Interface>
        <Interface>
        <IfIndex>1311</IfIndex>
        <VlanType>2</VlanType>
        </Interface>
        <Interface>
        <IfIndex>1313</IfIndex>
        <VlanType>2</VlanType>
        </Interface>
        </Interfaces>
    </Ifmgr>
    <Syslog>
    <LogBuffer>
    <BufferSize>120</BufferSize>
    </LogBuffer>
    </Syslog>
    <System>
    <Device>
    <SysName>Sysname</SysName>
    <TimeZone>
    <Zone>+11:44</Zone>
    <ZoneName>ABC</ZoneName>
    </TimeZone>
    </Device>
    </System>
</top>
</data>
</rpc-reply>

```

1.5.9 获取Syslog功能的变量值配置举例

1. 组网需求

获取 Syslog 模块的所有配置数据

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
      urn:ietf:params:netconf:base:1.0
    </capability>
  </capabilities>
</hello>
```

获取 Syslog 模块的所有配置数据。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="100"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/config:1.0">
        <Syslog/>
      </top>
    </filter>
  </get-config>
</rpc>
```

3. 验证配置

如果客户端收到类似如下的报文，则表示操作成功：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="100">
  <data>
    <top xmlns="http://www.h3c.com/netconf/config:1.0">
      <Syslog>
        <LogBuffer>
          <BufferSize>120</BufferSize>
        </LogBuffer>
      </Syslog>
    </top>
  </data>
</rpc-reply>
```

1.5.10 获取NETCONF会话信息配置举例

1. 配置需求

获取会话信息。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
      urn:ietf:params:netconf:base:1.0
    </capability>
  </capabilities>
</hello>
```

获取设备上当前存在的 NETCONF 会话的信息。

请将以下报文拷贝、粘贴到客户端：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-sessions/>
</rpc>
```

3. 验证配置

如果客户端收到类似如下的报文，则表示操作成功。

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="100">
  <get-sessions>
    <Session>
      <SessionID>1</SessionID>
      <Line>vty0</Line>
      <UserName></UserName>
      <Since>2017-01-07T00:24:57</Since>
      <LockHeld>>false</LockHeld>
    </Session>
  </get-sessions>
</rpc-reply>
```

以上信息表明：目前有一个 NETCONF 连接，SessionID 是 1，登录用户类型 vty0，登录时间是 2011-01-05T00:24:57，此用户不持有锁。

1.6 过滤表和列的信息

1.6.1 功能简介

当用户执行<get>、<get-bulk>、<get-config>或者<get-bulk-config>操作时，在 XML 语言中增加过滤条件，可以使用户只看到自己关心的数据。数据过滤包括基于表的过滤和基于列的过滤：

- 基于表的过滤用于过滤表的信息。
- 基于列的过滤用于过滤单个列的信息。

1.6.2 配置限制和指导

基于表的过滤需要在基于列的过滤之前配置。

1.6.3 基于表的过滤

1. 功能简介

基于表的过滤中，过滤条件类似命令行，命令空间为 `http://www.h3c.com/netconf/base:1.0`，属性名为 `filter`。关于支持该过滤方式表的详细信息，具体请参见模块的 XML API 手册。

如下要求返回 IPv4 路由表中匹配 IP 地址为 1.1.1.0 和 24 位掩码最长的数据：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:h3c="http://www.h3c.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Route>
          <Ipv4Routes>
            <RouteEntry h3c:filter="IP 1.1.1.0 MaskLen 24 longer"/>
          </Ipv4Routes>
        </Route>
      </top>
    </filter>
  </get>
</rpc>
```

2. 配置限制和指导

配置基于表的过滤时，查询条件必须写在行的属性上。

1.6.4 基于列的过滤

1. 功能简介

基于列的过滤包括严格匹配过滤、正则表达式匹配过滤和条件匹配过滤。

同时指定多种过滤时，只有一个生效，其优先级从高到低依次为：严格匹配过滤、正则表达式匹配过滤和条件匹配过滤。

2. 严格匹配过滤

严格匹配包括两种匹配方式，一种是元素值方式，还有一种是属性名方式。

用户在 XML 语言中直接指定对应的元素值，设备将对这些值进行严格匹配。如果指定了多个元素的值，则返回同时符合这几个条件的数据。只有完全匹配条件才返回成功。

如下为一个 NETCONF 报文示例，用于获取所有状态为 UP 的接口的配置信息：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
```

```

        <Interface>
            <AdminStatus>1</AdminStatus>
        </Interface>
    </Interfaces>
</Ifmgr>
</top>
</filter>
</get>
</rpc>

```

当用户在行的位置上放置的某个属性的名称和当前表的某个列的名称一致，则这个属性的值将与用户下发配置中同名称的列的值进行严格匹配，例如，上面例子用属性名方式的等价 XML 请求为：

```

<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <get>
        <filter type="subtree">
            <top xmlns="http://www.h3c.com/netconf/data:1.0"
xmlns:data="http://www.h3c.com/netconf/data:1.0">
                <Ifmgr>
                    <Interfaces>
                        <Interface data:AdminStatus="1"/>
                    </Interfaces>
                </Ifmgr>
            </top>
        </filter>
    </get>
</rpc>

```

以上两个 NETCONF 报文示例说明通过不同的严格匹配方式都可以获取所有状态为 UP 的接口的索引和指定列信息。

3. 正则表达式匹配过滤

当过滤条件比较复杂时，可以在指定元素上设置 **regExp** 属性为一个正则表达式，以完成过滤的目的。

支持正则表达式匹配的数据类型有整数、日期和时间、字符串、IPv4 地址、IPv4 掩码、IPv6 地址、MAC 地址、OID 和时区。

如下为一个 NETCONF 报文示例，用于获取接口的描述信息，并要求这些描述信息全部为大写字母，不能有其他字符。

```

<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:h3c="http://www.h3c.com/netconf/base:1.0">
    <get-config>
        <source>
            <running/>
        </source>
        <filter type="subtree">
            <top xmlns="http://www.h3c.com/netconf/config:1.0">
                <Ifmgr>
                    <Interfaces>
                        <Interface>
                            <Description h3c:regExp="^[A-Z]*$"/>
                        </Interface>
                    </Interfaces>
                </Ifmgr>
            </top>
        </filter>
    </get-config>
</rpc>

```

```

        </Interface>
    </Interfaces>
</Ifmgr>
</top>
</filter>
</get-config>
</rpc>

```

4. 条件匹配过滤

由于正则表达仅能够完成字符匹配，对于数值逻辑的判断过滤实现起来比较麻烦，此时，可使用条件匹配过滤功能。

条件匹配通过在元素中增加 **match** 属性完成，属性的值（即过滤条件）可以为数字、字符串。

表1-2 条件匹配命令

操作	命令	说明
大于	match="more:value"	值大于value，支持的数据类型为：日期、数字、字符串
小于	match="less:value"	值小于value，支持的数据类型为：日期、数字、字符串
不小于	match="notLess:value"	值不小于value，支持的数据类型为：日期、数字、字符串
不大于	match="notMore:value"	值不大于value，支持的数据类型为：日期、数字、字符串
等于	match="equal:value"	值等于value，支持的数据类型为：日期、数字、字符串、OID、BOOL
不等于	match="notEqual:value"	值不等于value，支持的数据类型为：日期、数字、字符串、OID、BOOL
包含	match="include:string"	包含字符串string，支持的数据类型为：字符串
不包含	match="exclude:string"	不能包含字符串string，支持的数据类型为：字符串
开始于	match="startWith:string"	以字符串string开头，支持的数据类型为：字符串、OID
结束于	match="endWith:string"	以字符串string结束，支持的数据类型为：字符串

如下为一个 NETCONF 报文示例，用于获取实体扩展信息中 CPU 利用率大于 50%的实体。

```

<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:h3c="http://www.h3c.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Device>
          <ExtPhysicalEntities>
            <Entity>
              <CpuUsage h3c:match="more:50"></CpuUsage>
            </Entity>
          </ExtPhysicalEntities>
        </Device>
      </top>
    </filter>
  </get>
</rpc>

```



```

    </top>
  </filter>
</get>
</rpc>

```

1.6.5 正则表达式过滤配置举例

1. 组网需求

取 Ifmgr 模块下 Interfaces 表下 Description 列包含 Gigabit 的所有数据。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

```

<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
      urn:ietf:params:netconf:base:1.0
    </capability>
  </capabilities>
</hello>

```

取 Ifmgr 模块 Interfaces 表下 Description 列包含 Gigabit 的所有数据。

请将以下报文拷贝、粘贴到客户端：

```

<?xml version="1.0"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:h3c="http://www.h3c.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
              <Description h3c:regExp="(Gigabit)+" />
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get>
</rpc>

```

3. 验证配置

如果客户端收到类似如下的报文，则表示操作成功。

```

<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:h3c="http://www.h3c.com/netconf/base:1.0" message-id="100">
  <data>

```

```

<top xmlns="http://www.h3c.com/netconf/data:1.0">
  <Ifmgr>
    <Interfaces>
      <Interface>
        <IfIndex>2681</IfIndex>
        <Description>GigabitEthernet1/0/1 Interface</Description>
      </Interface>
      <Interface>
        <IfIndex>2685</IfIndex>
        <Description>GigabitEthernet1/0/2 Interface</Description>
      </Interface>
      <Interface>
        <IfIndex>2689</IfIndex>
        <Description>GigabitEthernet1/0/3 Interface</Description>
      </Interface>
    </Interfaces>
  </Ifmgr>
</top>
</data>
</rpc-reply>

```

1.6.6 条件匹配过滤配置举例

1. 组网需求

取 Ifmgr 模块 Interfaces 表下 ifindex 值大于等于 5000 的 Name 列信息。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

```

<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
      urn:ietf:params:netconf:base:1.0
    </capability>
  </capabilities>
</hello>

```

取 Ifmgr 模块 Interfaces 表下 ifindex 值大于等于 5000 的 Name 列信息。

请将以下报文拷贝、粘贴到客户端：

```

<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:h3c="http://www.h3c.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>

```

```

        <Interface>
          <IfIndex h3c:match="notLess:5000" />
          <Name />
        </Interface>
      </Interfaces>
    </Ifmgr>
  </top>
</filter>
</get>
</rpc>

```

3. 验证配置

如果客户端收到类似如下的报文，则表示操作成功。

```

<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:h3c="http://www.h3c.com/netconf/base:1.0" message-id="100">
  <data>
    <top xmlns="http://www.h3c.com/netconf/data:1.0">
      <Ifmgr>
        <Interfaces>
          <Interface>
            <IfIndex>7241</IfIndex>
            <Name>NULL0</Name>
          </Interface>
        </Interfaces>
      </Ifmgr>
    </top>
  </data>
</rpc-reply>

```

1.7 配置加锁和解锁

1.7.1 功能简介

设备支持通过 **NETCONF**、**CLI** 和 **SNMP** 等多种方式配置，当用户管理、维护设备或者定位网络问题时，为防止其他用户修改当前配置、引入干扰，可以使用本特性给当前配置加锁。给当前配置加锁后，只有持有锁的用户可以修改设备的当前配置，其他用户只能执行读取操作，不能进行配置。只有持有锁的用户可以解锁，解锁后其他用户才可以修改设备的当前配置或另外加锁。如果持有锁的用户的当前连接断开，系统会自动解锁。

1.7.2 配置限制和指导

NETCONF 加锁操作将锁定配置数据，即<edit-config>操作中指定的配置数据，其他操作（例如<get>操作）不受锁的限制。

目前设备只支持对当前配置加锁，不能对具体的功能模块进行加锁。

只有持有锁的用户才能解锁，其他用户不能解锁。

1.7.3 配置加锁

请将以下报文拷贝、粘贴到客户端，用户即能完成加锁操作：

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <lock>
      <target>
        <running/>
      </target>
    </lock>
  </rpc>
```

设备收到加锁报文后会回应客户端，当客户端收到如下报文时，表示加锁成功：

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc-reply message-id="100"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
  </rpc-reply>
```

1.7.4 配置解锁

请将以下报文拷贝、粘贴到客户端，用户即能完成解锁操作：

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <unlock>
      <target>
        <running/>
      </target>
    </unlock>
  </rpc>
```

设备收到解锁报文后会回应客户端，当客户端收到如下报文时，表示解锁成功：

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc-reply message-id="100"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
  </rpc-reply>
```

1.7.5 加锁配置举例

1. 组网需求

给设备加锁，以免其他用户使用 XML 语言修改设备的当前配置。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
```

```

    <capabilities>
      <capability>
        urn:ietf:params:netconf:base:1.0
      </capability>
    </capabilities>
  </hello>

```

对当前配置加锁。

请将以下报文拷贝、粘贴到客户端：

```

<?xml version="1.0" encoding="UTF-8"?>
  <rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <lock>
      <target>
        <running/>
      </target>
    </lock>
  </rpc>

```

3. 验证配置

如果客户端收到如下报文，则表示加锁成功：

```

<?xml version="1.0" encoding="UTF-8"?>
  <rpc-reply message-id="100"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
  </rpc-reply>

```

用户加锁成功后，另一客户端发送加锁报文，设备会返回如下报文：

```

<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<rpc-error>
  <error-type>protocol</error-type>
  <error-tag>lock-denied</error-tag>
  <error-severity>error</error-severity>
  <error-message xml:lang="en">Lock failed because the NETCONF lock is held by another
session.</error-message>
  <error-info>
    <session-id>1</session-id>
  </error-info>
</rpc-error>
</rpc-reply>

```

以上报文表明加锁失败，session-id 是 1 的用户已经持有锁。

1.8 下发配置

1.8.1 功能简介

设备支持通过<edit-config>操作下发配置，该操作支持如下配置选项：merge、create、replace、remove、delete、缺省操作选项、缺省错误处理选项、测试处理、增量下发，关于这些选项的详细描述请参见“[2 Comware V7 支持的NETCONF操作](#)”。

1.8.2 配置步骤

当需要向设备下发配置时，请使用如下格式的 NETCONF 报文：

```
<?xml version="1.0"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target><running></running></target>
  <error-option>
    error-option
  </error-option>
  <config>
    <top xmlns="http://www.h3c.com/netconf/config:1.0">
      指定模块名，子模块名，列名，表名
    </top>
  </config>
</edit-config>
</rpc>
```

error-option 为下发配置过程中遇到错误时的处理方式，取值为：

属性	说明
stop-on-error	停止下发配置
continue-on-error	继续下发配置
rollback-on-error	回滚到该<edit-config>操作前的配置 配置回滚时缺省不能下发<edit-config>操作。因为回滚到该<edit-config>操作前的配置时，如果回滚时间超过客户端等待时间，客户端会认为<edit-config>操作下发失败，此时客户端将重新下发<edit-config>操作，设备再次触发回滚，如此反复，则会导致CPU/内存资源耗尽，设备重启 如需回滚配置的同时执行<edit-config>操作，请先执行<action>操作将DisableEditConfigWhenRollback的属性修改为false

设备收到 edit-config 请求后会回应客户端，当客户端收到如下报文时，表示设置成功：

```
<?xml version="1.0">
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

用户还可以通过<get>操作可以查看属性的当前值是否和 edit-config 操作设置的值一致。

1.8.3 下发配置举例

1. 组网需求

修改 syslog 模块中的日志缓冲区可存储的信息条数为 512。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.0</capability>
  </capabilities>
</hello>
```

把 Syslog 模块 LogBuffer 表中的 BufferSize 列，改成 512。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <top xmlns="http://www.h3c.com/netconf/config:1.0" web:operation="merge">
        <Syslog>
          <LogBuffer>
            <BufferSize>512</BufferSize>
          </LogBuffer>
        </Syslog>
      </top>
    </config>
  </edit-config>
</rpc>
```

3. 验证配置

如果客户端收到类似如下的报文，则表示操作成功：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

1.9 保存配置

1.9.1 功能简介

用户可以执行保存配置操作将当前配置保存到配置文件中，同时该文件将自动作为下次启动配置文件。

1.9.2 配置限制和指导

保存配置操作消耗系统资源较多，因此，请避免在系统资源占用较大的情况下执行这些操作。

1.9.3 配置步骤

请将以下报文拷贝、粘贴到客户端：

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save OverWrite="false">
    <file>配置文件的名称</file>
  </save>
</rpc>
```

属性	说明
file	文本配置文件的名称，该名称必须以存储介质的名称开头，后缀为.cfg。当报文中存在file属性时，必须输入文本配置文件的名称，不能为空 设备会将当前配置同时保存到文本配置文件（后缀为“.cfg”）和对应的二进制配置文件（后缀为“.mdb”，仅软件能解析） <ul style="list-style-type: none">如果file属性指定的文件不存在，则自动创建该文本配置文件和对应的二进制配置文件未指定file属性时，设备自动将当前配置保存到下次启动配置文件（包括文本配置文件和对应的二进制配置文件）
OverWrite	指定的配置文件名与原有配置文件名相同时，是否覆盖原配置文件。取值为： <ul style="list-style-type: none">true: 当指定的配置文件名与原有配置文件名相同时，覆盖原配置文件false: 当指定的配置文件名与原有配置文件名相同时，当前配置保存失败，同时返回错误提示信息 缺省情况下，OverWrite属性取值为true

设备收到配置保存请求后会回应客户端，当客户端收到如下报文时，表示保存成功：

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc-reply message-id="100"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
  </rpc-reply>
```

1.9.4 保存配置举例

1. 组网需求

将设备的当前配置保存到配置文件 my_config.cfg。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
      urn:ietf:params:netconf:base:1.0
    </capability>
  </capabilities>
</hello>
```

将设备的当前配置保存到配置文件 config.cfg。

请将以下报文拷贝、粘贴到客户端：


```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save>
    <file>config.cfg</file>
  </save>
</rpc>
```

3. 验证配置

如果客户端收到类似如下的报文，则表示操作成功。

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc-reply message-id="100"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
  </rpc-reply>
```

1.10 加载配置文件

1.10.1 功能简介

用户可以执行<load>操作，将指定配置文件中的配置会被合并到设备的当前配置中。

设备将配置文件中的配置中的配置与当前配置进行比较：

- 对于配置文件中，有，但当前配置中没有的配置，则直接运行。
- 对于配置文件与当前配置中不一致的配置，则用配置文件中的配置替换当前配置中的对应配置。

1.10.2 配置限制和指导

加载配置文件操作消耗系统资源较多，因此，请避免在系统资源占用较大的情况下执行这些操作。

如果只有删除设备中现有的配置才能使配置文件中的某些配置生效，则需要先删除当前配置后再加载配置文件，否则这些配置无法生效。

1.10.3 配置步骤

请将以下报文拷贝、粘贴到客户端，用户即可将配置文件中的配置追加到当前配置中：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <load>
    <file>配置文件的名称</file>
  </load>
</rpc>
```

其中，“配置文件的名称”必须以存储介质的名称开头，后缀为.cfg。

设备收到配置加载请求后会回应客户端，当客户端收到如下报文时，表示配置加载成功：

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc-reply message-id="100"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
  </rpc-reply>
```

1.11 回滚配置

1.11.1 配置限制和指导

回滚配置操作消耗系统资源较多，因此，请避免在系统资源占用较大的情况下执行这些操作。

缺省情况下，通过 NETCONF 回滚配置时，不能执行<edit-config>操作。如果需要在回滚配置的同时执行<edit-config>操作，请先执行<action>操作，将 **DisableEditConfigWhenRollback** 的属性修改为 **false**。

1.11.2 回滚配置文件

请将以下报文拷贝、粘贴到客户端，用户即可将设备的当前配置恢复到指定配置文件中的配置：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <rollback>
    <file>配置文件的名称</file>
  </rollback>
</rpc>
```

设备收到配置回滚请求后会回应客户端，当客户端收到如下报文时，表示配置回滚成功：

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc-reply message-id="100"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
  </rpc-reply>
```

1.11.3 回滚运行配置

1. 功能简介

设备支持对当前正在运行的配置进行回滚，配置回滚点后，设备可以在如下场景下进行配置回滚：

- NETCONF 客户端主动下发配置回滚指令。
- NETCONF 客户端在指定的时间内无任何指令，即 NETCONF 会话空闲时间超过配置的回滚空闲超时时间。
- NETCONF 客户端和设备间的连接异常断开。

2. 配置限制和指导

如果存在多个 NETCONF 会话同时配置设备，建议回滚配置前先用<lock>锁定系统，否则可能导致设备运行配置与回滚前配置不一致。

3. 配置步骤

- (1) 对当前系统加锁，具体请参见“[1.7 配置加锁和解锁](#)”。
- (2) 启动配置回滚功能，下发<save-point>/<begin>操作配置起始回滚点。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <begin>
      <confirm-timeout>100</confirm-timeout>
    </begin>
  </save-point>
</rpc>
```

```

        </begin>
      </save-point>
    </rpc>

```

属性	说明
confirm-timeout	回滚空闲超时时间，取值范围为1~65535，单位为秒，缺省值为600。该属性为可选

当客户端收到如下报文时，表示配置起始回滚点成功：

```

<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <save-point>
      <commit>
        <commit-id>1</commit-id>
      </commit>
    </save-point>
  </data>
</rpc-reply>

```

(3) 下发<edit-config>操作，对设备各功能进行配置。

(4) 下发<save-point>/<commit>操作配置回滚点。

配置<save-point>操作后，系统支持对最近 50 个回滚点的配置回滚。超过 50 个回滚点后，进行<commit>操作时需指定 **force** 属性强制覆盖最早的回滚点。

请将以下报文拷贝、粘贴到客户端：

```

<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <commit>
      <label>SUPPORT VLAN<label>
      <comment>vlan 1 to 100 and interfaces. Each vlan used for different custom as follows: .....</comment>
    </commit>
  </save-point>
</rpc>

```

<label/>和<comment/>为可选。

当客户端收到如下报文时，表示确认成功：

```

<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <save-point>
      <commit>
        <commit-id>2</commit-id>
      </commit>
    </save-point>
  </data>
</rpc-reply>

```

(5) 下发<save-point>/<get-commits>操作获取<commit>的操作记录。

```

<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <get-commits>
  </commit-id/>

```

```

<commit-index/>
<commit-label/>
  </get-commits>
  </save-point>
</rpc>

```

<commit-id>, <commit-index>, <commit-label>任选一个, 也可以都不选, 显示所有的 commit 记录。

请将以下报文拷贝、粘贴到客户端:

```

<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <get-commits>
      <commit-label>SUPPORT VLAN</commit-label>
    </get-commits>
  </save-point>
</rpc>

```

当客户端收到如下报文时, 返回对应的 commit 操作记录:

```

<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <save-point>
      <commit-information>
        <CommitID>2</CommitID>
        <TimeStamp>Sun Jan 1 11:30:28 2017</TimeStamp>
        <UserName>test</UserName>
        <Label>SUPPORT VLAN</Label>
      </commit-information>
    </save-point>
  </data>
</rpc-reply>

```

- (6) 下发<save-point>/<get-commit-information>操作获取对应的 commit 操作时的配置。

```

<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <get-commit-information>
      <commit-information>
        <commit-id/>
        <commit-index/>
        <commit-label/>
      </commit-information>
      <compare-information>
        <commit-id/>
        <commit-index/>
        <commit-label/>
      </compare-information>
    </get-commit-information>
  </save-point>
</rpc>

```

<commit-id/>, <commit-index/>, <commit-label/>任选一个。<compare-information/>可选。

属性	说明
commit-id	系统唯一标识的commit编号
commit-index	最近50次的commit, 0表示最近一次下发commit, 49表示最远一次commit
commit-label	回滚点的标签, 不同回滚点的标签不能相同
get-commit-information	显示最近的使用的commit配置

请将以下报文拷贝、粘贴到客户端:

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <get-commit-information>
      <commit-information>
        <commit-label>SUPPORT VLAN</commit-label>
      </commit-information>
    </get-commit-information>
  </save-point>
</rpc>
```

当客户端收到如下报文时, 显示对应的配置信息:

```
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <save-point>
      <commit-information>
        <content>
          ...
interface vlan 1
...
        </content>
      </commit-information>
    </save-point>
  </data>
</rpc-reply>
```

- (7) 下发<save-point>/<rollback>进行回滚配置操作。用户可选择对应的回滚点进行配置回滚, 或者等待 NETCONF 会话空闲时间超过配置的回滚空闲超时时间而自动进行配置回滚到最近回滚点的配置。

请将以下报文拷贝、粘贴到客户端:

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <rollback>
      <commit-id/>
      <commit-index/>
      <commit-label/>
    </rollback>
  </save-point>
</rpc>
```

<commit-id/>、<commit-index/>、<commit-label/>任选一个, 或者不选时回滚此前最近commit的配置。

属性	说明
commit-id	系统唯一标识的commit编号
commit-index	最近50次的commit，0表示最近一次下发commit，49表示最远一次commit
commit-label	回滚点的标签，不同回滚点的标签不能相同

当客户端收到如下报文时，表示主动下发配置回滚成功：

```
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok></ok>
</rpc-reply>
```

- (8) 下发<save-point>/<end>结束配置回滚或取消回滚配置。

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <end/>
  </save-point>
</rpc>
```

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <end/>
  </save-point>
</rpc>
```

当客户端收到如下报文时，返回对应的 **end** 结果成功。

```
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

- (9) 对当前系统解锁，具体请参见“[1.7 配置加锁和解锁](#)”。

1.12 下发命令行

1.12.1 功能简介

通过 NETCONF 功能，用户可以将命令行封装在 XML 报文中对设备进行操作。

1.12.2 配置限制和指导

通过 NETCONF 下发命令行操作消耗系统资源较多，因此，请避免一次下发多个命令行、或多个用户同时进行 NETCONF 下发命令行操作。

1.12.3 命令行操作

当需要给设备发送命令时，请使用格式如下的 NETCONF 报文：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <CLI>
  <Execution>
```

```
    命令行
  </Execution>
</CLI>
</rpc>
```

一对<Execution>子标签中可以包含多个命令行，一条命令输入完毕，换行，再输入下一条命令即可。

设备收到命令行指令后会回应客户端，当客户端收到如下报文时，表示命令行执行成功（注意命令响应被 CDATA 节点包含）：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <CLI>
    <Execution>
      <![CDATA[对应命令行响应]]>
    </Execution>
  </CLI>
</rpc-reply>
```

1.12.4 命令行操作配置举例

1. 配置需求

向设备发送显示 VLAN 相关信息的命令。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
      urn:ietf:params:netconf:base:1.0
    </capability>
  </capabilities>
</hello>
```

向设备发送显示 VLAN 相关信息的命令。

请将以下报文拷贝、粘贴到客户端：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <CLI>
    <Execution>
      display vlan
    </Execution>
  </CLI>
</rpc>
```

3. 验证配置

如果客户端收到类似如下的报文，则表示操作成功。

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <CLI>
    <Execution><![CDATA[
<Sysname>display vlan
Total VLANs: 1
The VLANs include:
1(default)
]]>
    </Execution>
  </CLI>
</rpc-reply>
```

1.13 订阅事件

1.13.1 功能简介

用户向设备订阅事件后，设备上发生用户订阅的事件时，设备会自动向订阅的客户端发送事件的日志。

1.13.2 配置限制和指导

订阅只对当前连接生效。如果连接断开，订阅会自动取消。
如果不配置订阅的事件流<stream></stream>，则缺省订阅 Syslog 事件。

1.13.3 订阅Syslog事件

事件订阅报文格式如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <stream>NETCONF</stream>
    <filter>
      <event xmlns="http://www.h3c.com/netconf/event:1.0">
        <Code>code</Code>
        <Group>group</Group>
        <Severity>severity</Severity>
      </event>
    </filter>
    <startTime>start-time</startTime>
    <stopTime>stop-time</stopTime>
  </create-subscription>
</rpc>
```

属性	说明
stream	订阅的事件流，Syslog事件流的名称为NETCONF
event	订阅的事件，此处可订阅的事件为系统支持的日志。具体可以订阅哪些模块的事件，请参见《日志信息参考手册》

属性	说明
code	日志信息中的助记符
group	日志信息中的模块名
severity	日志信息中的安全级别
start-time	订阅的开始时间
stop-time	订阅的结束时间

设备收到订阅报文后会回应客户端，当客户端收到如下报文时，表示订阅成功：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply >
```

当订阅出错时设备会返回错误信息，例如：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<rpc-error>
  <error-type>error-type</error-type>
  <error-tag>error-tag</error-tag>
  <error-severity>error-severity</error-severity>
  <error-message xml:lang="en">error-message</error-message>
</rpc-error>
</rpc-reply>
```

错误报文的详细定义请参见 RFC 4741。

1.13.4 订阅监控事件

配置该功能后，NETCONF 每隔指定时间获取一次所订阅事件，并将符合订阅条件的信息发送给用户。

事件订阅报文格式如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<create-subscription xmlns='urn:ietf:params:xml:ns:netconf:notification:1.0'>
  <stream>NETCONF_MONITOR_EXTENSION</stream>
  <filter>
    <NetconfMonitor xmlns='http://www.h3c.com/netconf/monitor:1.0'>
      <XPath>XPath</XPath>
      <Interval>interval</Interval>
      <ColumnConditions>
        <ColumnCondition>
          <ColumnName>ColumnName</ColumnName>
          <ColumnValue>ColumnValue</ColumnValue>
          <ColumnCondition>ColumnCondition</ColumnCondition>
        </ColumnCondition>
      </ColumnConditions>
    </NetconfMonitor>
  </filter>
</create-subscription>
</rpc>
```

```
<MustIncludeResultColumns>
  <ColumnName>columnName</ColumnName>
</MustIncludeResultColumns>
</NetconfMonitor>
</filter>
<startTime>start-time</startTime>
<stopTime>stop-time</stopTime>
</create-subscription>
</rpc>
```

属性	说明
stream	订阅的事件流，监控事件流的名称为NETCONF_MONITOR_EXTENSION
NetconfMonitor	监控事件的过滤信息
XPath	监控事件的路径，格式为：模块名[/子模块名]/表名
interval	监控的时间间隔，取值范围为1~4294967，缺省值为300，单位为秒，即每隔300秒获取一次符合订阅条件的信息
ColumnName	监控列的名称，格式为：[组名称.]列名称
ColumnValue	监控列的过滤值
ColumnCondition	监控列的过滤条件： <ul style="list-style-type: none">• more: 大于• less: 小于• notLess: 不小于• notMore: 不大于• equal: 等于• notEqual: 不等于• include: 包含• exclude: 不包含• startWith: 开始于• endWith: 结束于 请根据监控列的过滤值的类型填写过滤条件
start-time	订阅的开始时间
stop-time	订阅的结束时间

设备收到订阅报文后会回应客户端，当客户端收到如下报文时，表示订阅成功：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

1.13.5 订阅模块上报事件

配置该功能后，所订阅模块发生用户所订阅事件时，将主动上报 NETCONF，NETCONF 将这样信息发送给用户。

事件订阅报文格式如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:xs="http://www.h3c.com/netconf/base:1.0">
<create-subscription xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<stream>XXX_STREAM</stream>
    <filter type="subtree">
<event xmlns="http://www.h3c.com/netconf/event:1.0/xxx-features-list-name:1.0">
    <ColumnName xs:condition="Condition">value</ColumnName>
</event>
</filter>
<startTime>start-time</startTime>
<stopTime>stop-time</stopTime>
</create-subscription>
</rpc>
```

属性	说明
stream	订阅的事件流，请根据设备实际支持情况填写模块上报事件流的名称
event	订阅的事件的名称，一个事件流中包括多个事件，请根据事件流下的事件填写，命名空间为事件流的命名空间
ColumnName	当前事件下需要过滤的列的名称
Condition	事件列的过滤条件： <ul style="list-style-type: none">• more: 大于• less: 小于• notLess: 不小于• notMore: 不大于• equal: 等于• notEqual: 不等于• include: 包含• exclude: 不包含• startWith: 开始于• endWith: 结束于 请根据事件列的过滤值的类型填写过滤条件
value	当前事件下需要过滤的列的值
start-time	订阅的开始时间
stop-time	订阅的结束时间

设备收到订阅报文后会回应客户端，当客户端收到如下报文时，表示订阅成功：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
</rpc-reply>
```

1.13.6 订阅事件配置举例

1. 组网需求

客户端订阅没有时间限制的全部事件。订阅后在断开连接之前，设备发生的所有事件都会发送给客户端。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
      urn:ietf:params:netconf:base:1.0
    </capability>
  </capabilities>
</hello>
```

订阅全部事件，不限制订阅时间。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <stream>NETCONF</stream>
  </create-subscription>
</rpc>
```

3. 验证配置

如果客户端收到如下报文，则表示订阅成功：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="100">
  <ok/>
</rpc-reply>
```

当用户（IP 地址为 192.168.100.130）登录设备时，设备会发送如下报文来通知订阅客户端：

```
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2011-01-04T12:30:52</eventTime>
  <event xmlns="http://www.h3c.com/netconf/event:1.0">
    <Group>SHELL</Group>
    <Code>SHELL_LOGIN</Code>
    <Slot>1</Slot>
    <Severity>Notification</Severity>
    <context>VTY logged in from 192.168.100.130.</context>
  </event>
</notification>
```

1.14 关闭NETCONF会话

1.14.1 功能简介

NETCONF 支持用户关闭除自己外的 NETCONF 会话，被关闭会话的用户退回到用户视图。

1.14.2 Kill-session操作

请将以下报文拷贝、粘贴到客户端，用户即能关闭指定的会话：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <kill-session>
    <session-id>
      指定 session-ID
    </session-id>
  </kill-session>
</rpc>
```

设备收到会话关闭请求后会回应客户端，当客户端收到如下报文时，表示指定会话已经被关闭：

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc-reply message-id="100"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
  </rpc-reply>
```

1.14.3 Kill-session操作配置举例

1. 配置需求

当前有两个 NETCONF 登录用户，session ID 分别是 1 和 2，session ID 为 1 的用户要关闭 session ID 为 2 用户的会话。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
      urn:ietf:params:netconf:base:1.0
    </capability>
  </capabilities>
</hello>
```

关闭 session ID 为 2 的用户会话。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <kill-session>
    <session-id>2</session-id>
```

```
</kill-session>
</rpc>
```

3. 验证配置

如果客户端收到如下报文，则表明 session ID 为 2 的 NETCONF 会话已经被关闭。建立该会话的用户会从 XML 视图退回到用户视图。

```
<?xml version="1.0" encoding="UTF-8"?>
  <rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
  </rpc-reply>
```

1.15 退出XML视图

1. 配置限制和指导

进入 XML 视图后，必须先完成能力交换，再执行退出 XML 视图操作。

2. 配置步骤

处于 XML 视图的用户退回到用户视图时，需要将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <close-session/>
</rpc>
```

设备收到以上请求后会返回如下报文并退回到用户视图：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

2 Comware V7 支持的NETCONF操作

Comware V7 平台对 NETCONF 标准协议做了一些修订，删除了不常用的操作，增加部分新的操作

2.1.1 action

【使用指导】

下发可配置的功能的动作，例如 **reset** 操作。

【XML 格式样例】

清除全部接口的统计信息，XML 请求如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <action>
    <top xmlns="http://www.h3c.com/netconf/action:1.0">
      <Ifmgr>
        <ClearAllIfStatistics>
          <Clear>
          </Clear>
        </ClearAllIfStatistics>
      </Ifmgr>
    </top>
  </action>
</rpc>
```

2.1.2 CLI

【使用指导】

执行命令行的命令。

请求消息将命令行语句封装在<CLI>标签中，命令行输出信息被封装在<CLI>标签中返回。

CLI 支持 Execution 和 Configuration 方式执行命令行：

- **Execution**：在用户视图下执行命令行。
- **Configuration**：在系统视图下执行命令行。使用该方式时需要指定 **exec-use-channel** 的属性：
 - **false**：不使用 **channel** 方式执行命令行。
 - **true**：使用临时 **channel** 执行命令行，执行完成后，自动关闭该 **channel**。
 - **persist**：使用保留 **channel** 执行命令行。使用该方式时，则需要执行 **Open-channel** 操作打开当前对话的 **channel**；使用完成后，执行 **Close-channel** 关闭 **Channel**。
- 对于其他视图下命令，则需要在 **Configuration** 下先指定进入子视图的命令，再指定配置命令。

一个 NETCONF 会话只能打开一个保留 **channel**，可以打开多个临时 **channel**。

如果没有执行 **Open-channel** 操作，则系统将自动打开 **channel**。

如果没有执行关闭保留 **channel** 操作，再次使用保留 **channel** 时，系统在上次最后执行的命令行所在视图再次执行命令行。

不支持执行交互式命令。

使用 **channel** 方式执行命令行时，不支持执行 **quit** 命令退出用户视图。

【XML 格式样例】

在系统视图下不使用 **channel** 方式执行 **display this** 命令：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <CLI>
    <Configuration exec-use-channel="false">display this</Configuration>
  </CLI>
</rpc>
```

2.1.3 close-session

【使用指导】

关闭当前 **NETCONF** 会话，并释放锁和这个 **session** 用到的内部资源（如内存等），退出 **XML** 视图。

【XML 格式样例】

关闭当前 **NETCONF** 会话，并释放锁和这个 **session** 用到的内部资源（如内存等），退出 **XML** 视图

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <close-session/>
</rpc>
```

2.1.4 edit-config: create

【使用指导】

创建配置。

create 操作必须指定配置对象。

- 如果当前配置表支持创建对象，且当前对象不存在，则先创建配置对象，再创建指定的配置。
- 如果配置对象下对应的配置项已经存在，则返回 **data-exist** 错误。

【XML 格式样例】

将 **BufferSize** 设置为 120 的 **xml** 请求如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <top xmlns="http://www.h3c.com/netconf/config:1.0">
        <Syslog xmlns="http://www.h3c.com/netconf/config:1.0" xc:operation="create">
          <LogBuffer>
            <BufferSize>120</BufferSize>
          </LogBuffer>
        </Syslog>
      </top>
    </config>
  </edit-config>
</rpc>
```



```
        </Syslog>
    </top>
</config>
</edit-config>
</rpc>
```

2.1.5 edit-config: delete

【使用指导】

删除配置。

- 当指定的删除对象中只有表索引时，则删除此配置指定对象的所有配置，同时删除指定对象。
- 当指定的删除对象中不仅仅有表索引还存在配置项时，则删除此对象下面的指定配置。
- 如果系统中指定对象不存在，则直接返回不存在的错误消息。

【XML 格式样例】

同 edit-config: create，把 create 修改为 delete 即可。

2.1.6 edit-config: merge

【使用指导】

在当前运行配置的基础上直接运行指定配置。

merge 操作的 XML 数据格式和 create 类似，只是 operation 属性需要指定为 “merge”。

merge 操作必须指定具体的操作对象（行）：

- 如果指定的对象存在，则直接配置该对象。
- 如果指定的对象不存在，但允许创建，则先创建再配置该对象。
- 如果指定的对象不存在且不允许创建，则返回 merge 失败。

【XML 格式样例】

同 edit-config: create，把 create 修改为 merge 即可。

2.1.7 edit-config: remove

【使用指导】

删除配置。

- 当指定的删除对象中只有表索引时，则删除此配置指定对象的所有配置，同时删除指定对象。
- 当指定的删除对象中不仅仅有表索引还存在配置项时，则删除此对象下面的指定配置。
- 如果系统中指定对象不存在，或者 XML 消息未指定对象，则直接返回成功。

【XML 格式样例】

同 edit-config: create，把 create 修改为 remove 即可。

2.1.8 edit-config: replace

【使用指导】

替换配置。

- 如果指定的对象存在，则替换指定对象的配置为当前配置。
- 如果指定的对象不存在，但允许创建，则先创建再配置该对象为当前配置。
- 如果指定对象不存在且不允许创建，则不进行 **replace** 操作，返回 **invalid-value** 错误，提示用户配置对象不存在。

【XML 格式样例】

同 **edit-config: create**，把 **create** 修改为 **replace** 即可。

2.1.9 edit-config测试处理选项

【使用指导】

执行 **edit-config** 操作时，可指定一个测试选项，使用 **<test-option>** 节点来决定当前配置是否下发。该节点的缺省值为 **test-then-set**，全部取值为：

- **test-then-set**：如果没有错误则将配置设置到系统。
- **set**：将配置设置到系统。
- **test-only**：只测试，并不下发配置到系统。语法检查通过，就返回 **ok** 成功，否则失败。

【XML 格式样例】

下发一个接口的配置，仅测试，XML 请求如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <test-option>test-only</test-option>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <top xmlns="http://www.h3c.com/netconf/config:1.0">
        <Ifmgr xc:operation="merge">
          <Interfaces>
            <Interface>
              <IfIndex>262</IfIndex>
              <Description>222</Description>
              <ConfigSpeed>2</ConfigSpeed>
              <ConfigDuplex>1</ConfigDuplex>
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </config>
  </edit-config>
</rpc>
```

2.1.10 edit-config缺省操作选项

【使用指导】

edit-config 操作用于修改当前系统配置。

NETCONF 定制了 merge、create、delete 和 replace 等修改配置的方式。当 XML 消息中未指定修改配置方式时，则使用缺省操作作为当前指令的操作方式，不会修改缺省操作的缺省值。

缺省操作的缺省值为 merge，可以在 XML 消息中通过<default-operation>节点设置，取值为：

- merge: 当配置方式和缺省操作方式均未指定时，使用该方式。
- replace: 当配置方式未指定，缺省操作指定为 replace 时，edit-config 操作缺省为 replace 操作。
- none: 当配置方式未指定，缺省操作指定为 none 的时候，edit-config 操作缺省为 none 操作。none 操作主要用来检查，下发为 none 操作的配置仅仅做 Schema 校验，不进行真正的配置下发。语法检查通过，返回 ok 成功，否则失败。

【XML 格式样例】

下发一个空的操作，该操作仅仅验证格式，并不真正下发给系统，xml 请求如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <default-operation>none</default-operation>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <top xmlns="http://www.h3c.com/netconf/config:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
              <IfIndex>262</IfIndex>
              <Description>222222</Description>
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </config>
  </edit-config>
</rpc>
```

2.1.11 edit-config缺省错误处理选项

【使用指导】

edit-config 将指定的配置设置到系统上，完成配置设置的操作。

在执行 edit-config 的过程中，如果遇到一个实例配置出错，缺省情况下会直接返回错误，并提供错误选项，通过错误选项取值的不同，在发生错误的时候进行不同的处理操作。

<error-option> 节点用于设置一个实例配置出错后，后续实例配置的处理方式，缺省值为 stop-on-error，全部取值为：

- **stop-on-error:** 停止处理，返回错误。此选项为缺省选项。
- **continue-on-error:** 继续处理，但是报告错误。
- **rollback-on-error:** 停止并回滚配置。

【XML 格式样例】

下发两个接口的配置，当下发第一个接口的配置发生错误时，继续进行下一个接口配置的下发，XML 请求如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>    <error-option>continue-on-error</error-option>
  <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
    <top xmlns="http://www.h3c.com/netconf/config:1.0">
      <Ifmgr xc:operation="merge">
        <Interfaces>
          <Interface>
            <IfIndex>262</IfIndex>
            <Description>222</Description>
            <ConfigSpeed>1024</ConfigSpeed>
            <ConfigDuplex>1</ConfigDuplex>
          </Interface>
          <Interface>
            <IfIndex>263</IfIndex>
            <Description>333</Description>
            <ConfigSpeed>1024</ConfigSpeed>
            <ConfigDuplex>1</ConfigDuplex>
          </Interface>
        </Interfaces>
      </Ifmgr>
    </top>
  </config>
</edit-config>
</rpc>
```

2.1.12 edit-config增量下发

【使用指导】

增量下发选项 **incremental** 放置在列上，对于类似 **vlan permitlist** 列表集合性质的列，可能支持增量下发，用户请求 XML 中有增量下发选项时，最终执行结果不影响本列原有的数据。

增量下发只支持 **edit-config**，但不支持 **edit-config** 中的 **replace**。

关于支持增量下发的模块，请参见模块对应的 **NETCONF XML API**。

【XML 格式样例】

下发一个接口的 VLAN 配置，使用增量下发，262 接口原有 **untagvlanlist** 为 12~15，下发后为 1~10，12~15。XML 请求如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
```

```

xmlns:h3c="http://www.h3c.com/netconf/base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <top xmlns="http://www.h3c.com/netconf/config:1.0">
        <VLAN xc:operation="merge">
          <HybridInterfaces>
            <Interface>
              <IfIndex>262</IfIndex>
              <UntaggedVlanList h3c: incremental="true">1-10</UntaggedVlanList>
            </Interface>
          </HybridInterfaces>
        </VLAN>
      </top>
    </config>
  </edit-config>
</rpc>

```

2.1.13 get

【使用指导】

获取设备运行状态和运行配置。

【XML 格式样例】

获取 Syslog 模块的全部数据的 XML 请求如下：

```

<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:xc="http://www.h3c.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Syslog>
        </Syslog>
      </top>
    </filter>
  </get>
</rpc>

```

2.1.14 get-bulk

【使用指导】

从指定索引的下一条开始获取 N 条运行状态和运行配置（索引行数据不返回）。

用户通过 **index** 属性指定索引，通过 **count** 属性指定 N。如未指定索引，则以第一条为索引；如未指定 N，或者数据表中符合条件的数据记录不足 N 条，则返回表中所有剩下的数据条目。

get 操作会返回所有符合条件的数据，在某些情况下会导致效率问题。**get-bulk** 允许用户从固定数据项开始，向后获取指定条目的数据记录。

【XML 格式样例】

获取全部接口的数据的 xml 请求如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces xc:count="5" xmlns:xc="http://www.h3c.com/netconf/base:1.0">
            <Interface/>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get-bulk>
</rpc>
```

2.1.15 get-bulk-config

【使用指导】

从指定索引的下一条批量获取可配置的功能的变量值。

配置限制与 **get-bulk** 操作相同。

【XML 格式样例】

获取全部接口配置信息的 xml 请求如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-bulk-config>
    <source>
      <running/>
    </source>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/config:1.0">
        <Ifmgr>
          </Ifmgr>
        </top>
      </filter>
    </get-bulk-config>
  </rpc>
```

2.1.16 get-config

【使用指导】

获取可配置的功能的变量值。如果对应功能未配置，则返回一个空的<data>。

【XML 格式样例】

获取接口表内所有配置的 XML 请求如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:xc="http://www.h3c.com/netconf/base:1.0">
```

```

<get-config>
  <source>
    <running/>
  </source>
  <filter type="subtree">
    <top xmlns="http://www.h3c.com/netconf/config:1.0">
      <Ifmgr>
        <Interfaces>
          <Interface/>
        </Interfaces>
      </Ifmgr>
    </top>
  </filter>
</get-config>
</rpc>

```

2.1.17 get-sessions

【使用指导】

获取当前系统中所有 NETCONF 会话的信息（不能指定 sessions-ID）。

【XML 格式样例】

获取当前系统中所有 NETCONF 会话的信息：

```

<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-sessions/>
</rpc>

```

2.1.18 kill-session

【使用指导】

关闭其他 NETCONF 会话。

该操作不支持关闭用户自己的 NETCONF 会话。

【XML 格式样例】

关闭 session-id 为 1 的 NETCONF 会话：

```

<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <kill-session>
    <session-id>1</session-id>
  </kill-session>
</rpc>

```

2.1.19 load

【使用指导】

加载配置。

执行该操作后，指定配置文件中的配置被合并到当前运行配置中。

【XML 格式样例】

将文件 **a1.cfg** 中的配置合并到设备的当前配置中：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <load>
    <file>a1.cfg</file>
  </load>
</rpc>
```

2.1.20 lock

【使用指导】

NETCONF 加锁操作将锁定配置数据，即<edit-config>操作中指定配置数据，其他操作（例如<get>操作）不受锁的限制。

用户通过 NETCONF 加锁操作锁定 NETCONF 连接后，其他用户既不能使用 NETCONF 执行下发配置操作，也不能通过 CLI 或 SNMP 等其他配置方式对设备进行配置。

【XML 格式样例】

禁止 NETCONF 会话修改设备的当前配置，XML 请求如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <lock>
    <target>
      <running/>
    </target>
  </lock>
</rpc>
```

2.1.21 rollback

【使用指导】

回滚配置。

该操作必须使用子元素<file>指定需要回滚的配置文件的名称。

执行该操作后，当前运行配置回滚为指定配置文件中的配置。

【XML 格式样例】

将设备当前配置回退到文件 **1A.cfg** 中配置的状态：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <rollback>
    <file>1A.cfg</file>
  </rollback>
</rpc>
```

2.1.22 save

【使用指导】

保存运行配置。

该操作可以使用子元素<file>来指定保存的配置文件名称。当 save 操作中不存在子元素<file>列时，则设备会自动将当前运行配置保存到主用下次启动配置文件中。

OverWrite 属性用来判断当指定的配置文件名存在时，当前配置是否覆盖原配置文件并保存成功。

Binary-only 属性用来只保存当前使用的二进制配置文件。

【XML 格式样例】

将设备当前配置保存到文件 test.cfg 中：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save OverWrite="false" Binary-only="true">
    <file>test.cfg</file>
  </save>
</rpc>
```

2.1.23 unlock

【使用指导】

解锁。

当会话结束时锁也会被自动释放。

【XML 格式样例】

取消锁保护，允许 NETCONF 会话修改设备的当前配置：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <unlock>
    <target>
      <running/>
    </target>
  </unlock>
</rpc>
```