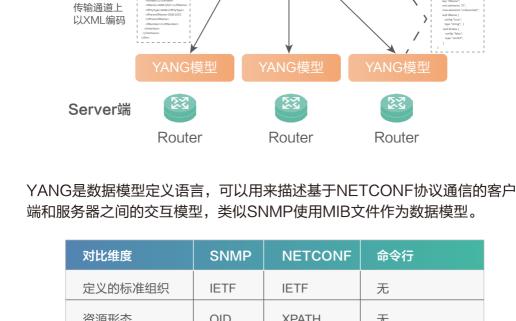


# NETCONF协议是一种网络设备管理协议,类似SNMP,提供一套新增、修

管理应用 Client端

改、删除网络设备配置,查询配置、状态和统计信息的框架机制。



资源形态 **XPATH** 无 数据模型文件 .mib 无模型 .yang 建模语言 SMI YANG 无

**NETCONF** 

XML

无

纯文本

**SNMP** 

BER

管理操作协议

数码方式

	传输协议栈	UDP	SSH	TELNET/SSH		
为什么需要NETCONF&YANG?						
云时代对网络的关键诉求之一是网络自动化,包括业务快速按需自动发放,自动化运维等。传统的命令行和SNMP已经不适应云化网络的诉求。主要表现在:						
	传统命令行是人机接口		<b>夏杂,厂商差异大,</b> 同设备,表现不同	人工学习成本高		
		缺	乏数据 模型			

### 厂商设备集成, 维护困难 缺少版本管理 互通困难

解析复杂

SNMP配置效率低,不支持事务机制,更多被用来做监控类协议

数字索引(1.3.6.1.2.1.2.2.1.4)

语法语义解析

复杂,不利于 网络编程

无法实现互通 面向人类界面,

安全隐患

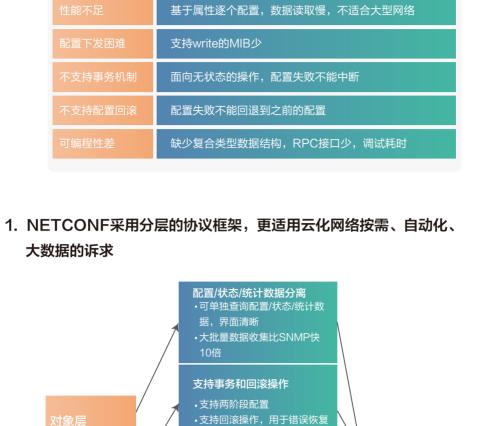
Telnet连接

按需

自动化

大数据

٨



## ·调用RPC方式实现操作和控 ・制定义了更丰富的操作接口 面向连接

可靠性更高

隔离。

存的配置文件。

可扩展能力

RFC6241:

Validate能力 Startup能力 URL能力

XPath能力

优点:

・可扩展:

Writable-Running能力

Confirmed Commit能力 Rollback-on-Error能力

Candidate Configuration能力

基于RPC操作

可扩展性好

基于XML编码

• 数据类型/结构可扩展

•可复用传统的XML工具

• 文本文件能表示灵活复杂的层

次化数据,模型驱动,存储效

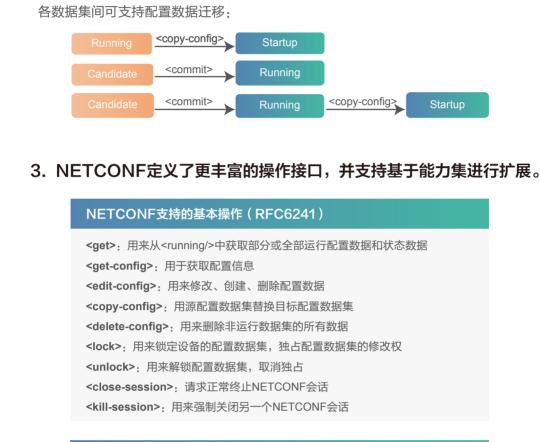
• 采用面向连接的传输层协议承

2. NETCONF支持对数据的分类存储和迁移,支持分阶段提交和配置

<running/>:运行配置数据集,保存网络设备上当前处于活动状态的完整配置 <candidate/>: 备选配置数据集,存放设备将要提交到<running>的各项配置数

<startup/>: 启动配置数据集,存放设备启动时所加载的配置数据,相当于已保

据的集合。对<candidate>的任何改变不会直接影响网络设备。



# 4. NETCONF协议基于YANG模型对设备进行操作。

YANG模型定义了设备功能的配置模板,和命令行相比,YANG模型具有如下

支持grouping重用、refine重定义和augment扩展,且支持基于typedef

人工输入:

**ip address** 1.1.1.1 24

客户端发送xml请求报文:

<ip xmlns="http://www.huawei.com/</pre>

IPv4和IPv6命令采用相同的

ip address ip-address { mask

厂商B设备

格式:

| mask-length }

设备运行数据

· 机器语言: 结构化定义,支持定义约束条件,机器直接识别,不需要人工干预

· 易集成: IETF定义了多个标准类型定义和YANG模型,便于各厂商参考和引用

・ 定义丰富: 支持丰富的基础数据类型定义和数据属性定义

扩展数据类型定义

step1: 人工阅读手册 Step2: 人工理解差异

Step3:人工输入命令行

设备

配置。

功能实现是否有差异? 命令行格式是否有差异? 配置是否有约束和依赖?

IPv4配置命令:

IPv6配置命令:

ipv4 address ipv4-address {

ipv6 address { ipv6-address prefix-length | ipv6address/prefix-length }

厂商A设备

mask | mask-length }

ip address ip-address { mask | mask-length }

RFC 5277:

Notification能力

Interleave能力

RFC 6243: with-defaults能力

RFC 6022:

letf-netconf-monitoring能力

```
when "not(../../addrCfgType='negotiation')";
                                               netconf/vrp/huawei-ip">
      key ip-address
        { leaf ip-address
                                               <mask>24</mask>
            type inet:ipv4-address-no-zone
                                              </ip>
         leaf mask
            type ifmlpv4Mask}
      }
当不同厂商的功能实现有差异时:
传统命令行需要工程师人工学习不同厂商的命令行差异,人工做映射。
                                          服务
                                用户
```

工程师

功能

基于YANG模型时,配置工程师不需要关注YANG模型的具体定义和YANG模 型差异,由APP自动适配解析,从而将人的关注重点从设备本身和功能差异转 移到了用户需求上。工程师只要操作图形化界面的APP,就可以实现自动化的

用户

工程师

工程师

YANG模型A

自动生成XML报文

{ ipv4-address

{ ipv6-address

mask }

{ ipv6

YANG模型A:

{ ip { ipv4 功能

YANG模型B

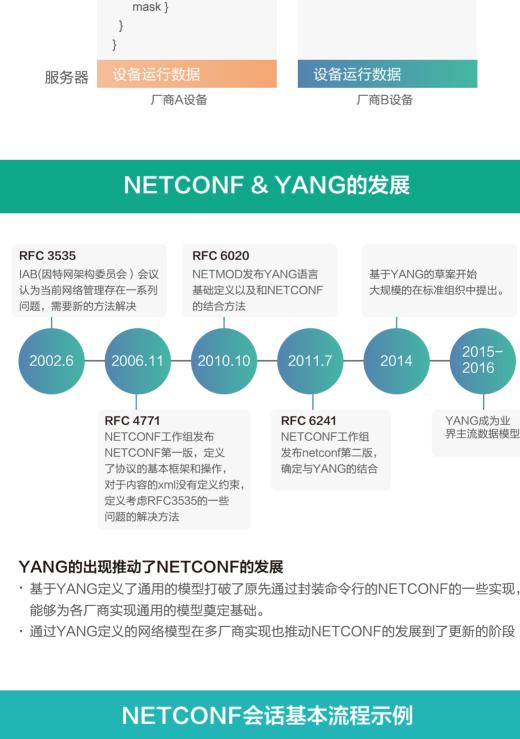
YANG模型B:

mask }

{ ip-address

{ ip

自动生成XML报文



场景描述:介绍最基本的通过客户端修改设备的IP地址的过程,并采用两阶段

前提条件:客户端触发NETCONF会话建立,完成SSH连接、完成认证和授权。

Step 1: 建立NETCONF会话,并通过<hello>消息进行能力通告

客户端

netconf:base:1.0">

base:1.1</capability>

<session-id>1</session-id>

<rpc-reply message-id="2" xmlns="</pre>

urn:ietf:params:xml:ns:netconf:

<capabilities>

</capabilities>

好的

base:1.0"> <ok/>

</rpc-reply>

hello,我支持的能力如下:

<hello xmlns="urn:ietf:params:xml:ns:</pre>

<capability>urn:ietf:params:netconf:

生效模式。

<hello xmlns="urn:ietf:params:xml:ns:</pre>

<capability>urn:ietf:params:netconf:

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

<running></running>

Step 2: 锁定运行数据集,避免和其他客户端冲突

netconf:base:1.0">

base:1.1</capability>

<capabilities>

</capabilities>

<target>

</target> </lock> </rpc>

<target>

<source>

</source> </copy-config>

</rpc>

<running></running>

Step 4:编辑备选数据集的配置

<rpc xmlns="urn:ietf:params:xml:ns:</pre>

netconf:base:1.0" message-id="4">

<edit-config> <target>

> <candidate/> </target>

<default-operation>merge </default-operation>

<interfaces>

<error-option>rollback-on-error </error-option> <config>

<ifm xmlns="http://www.huawei. com/netconf/vrp/huawei-ifm">

<interface nc:operation= "merge" xmlns:nc="urn:ietf:params: xml:ns:netconf:base:1.0">

<ifName>Gigabitethernet1/0/0</ifName> <ifAdminStatus>up</ifAdminStatus> <ipv4Config> <addrCfgType>config</addrCfgType>

</hello>

#### <rpc xmlns="urn:ietf:params:xml:ns:</pre> <rpc-reply message-id="3" xmlns=</pre> netconf:base:1.0" message-id="3"> "urn:ietf:params:xml:ns:netconf: <copy-config> base:1.0"> <candidate></candidate> </rpc-reply> </target>

Step 3: 复制运行数据集配置到备选数据集,保证修改前配置同步到最新状态

好的

好的

<ok/>

</rpc-reply>

<rpc-reply message-id="4" xmlns="urn:</pre>

ietf:params:xml:ns:netconf:base:1.0">

	<am4cfgaddrs></am4cfgaddrs>			
	<am4cfgaddr></am4cfgaddr>			
	<iflpaddr>1.1.1.1</iflpaddr>			
	<pre><subnetmask>255.255.255.252</subnetmask></pre> /			
	subnetMask>			
	<addrtype>main</addrtype>			
Class C. 相交名类数记住到空间与气物记住				
Step 5:提交备选数据集配置到运行数据集				

OK,已提交

</rpc-reply>

<rpc-reply message-id="6" xmlns=</pre>

"urn:ietf:params:xml:ns:netconf:base:1.0">

<rpc xmlns="urn:ietf:params:xml:ns:</pre>

netconf:base:1.0" message-id="6">

<commit></commit>

</rpc>

</rpc>

缩略语

**XML** 

**RPC** 

YANG

**NETCONF** 

### Step 6: 解锁运行数据集 好的 <rpc xmlns="urn:ietf:params:xml:ns:</pre> <rpc-reply message-id="7" xmlns="urn:ietf:</pre> netconf:base:1.0" message-id="7"> params:xml:ns:netconf:base:1.0"> <ok/> <target> </rpc-reply> <running></running> </target> </unlock>

缩略语

**Network Configuration Protocol** 

Extensible Markup Language

Yet Another Next Generation

Remote Procedure Call

英文

后续处理:关闭NETCONF会话,关闭SSH连接。

网络资料部