NETCONF&YANG 什么是NETCONF&YANG? NETCONF协议是一种网络设备管理协议,类似SNMP,提供一套新增、修 改、删除网络设备配置,<u>查询配置、状态和统计信息的框架机制。</u> YANG 模型 Client端 管理应用 传输通道上 以XML编码 中 Server端 YANG是数据模型定义语言,可以用来描述基于NETCONF协议通信的客户 端和服务器之间的交互模型,类似SNMP使用MIB文件作为数据模型。 SNMP 对比维度 **NETCONF** 命令行 **IETF** 定义的标准组织 **IETF** 无 资源形态 OID **XPATH** 无 无模型 数据模型文件 .mib .yang 建模语言 SMI YANG 无 管理操作协议 SNMP **NETCONF** 无 数码方式 XML 纯文本 BER UDP SSH TELNET/SSH 传输协议栈 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,配置过程复杂,厂商差异大,人工学习成本高 相同命令在不同 | 设备,表现个同 维护困难 据模型 命令行 全隐患 互通困难 厂商设备集成, Telnet连接 无法实现互通 面向人类界面,语法语义 解析复杂,不利于网络编程 SNMP配置效率低,不支持事务机制,更多被用来做监控类协议 可读性差 性能不足 配置下发困难 不支持事务机制 不支持配置回滚 可编程性差 NETCONF采用分层的协议框架,更适用云化网络按需、自动化、大数 据的诉求。 配置/状态/统计数据分离 •可单独查询配置/状态/统计数 据,界面清晰 ·大批量数据收集比SNMP快 支持事务和回滚操作 . 支持两阶段配置 • 支持回滚操作,用于错误恢复 对象层 可扩展性好 操作层 ・厂商可自定义新能力,实现特 定义RPC调用的方 定功能 •数据类型/结构可扩展 基于XML编码 RPC层 ・文本文件能表示灵活复杂的层 大数据 次化数据,模型驱动,存储效 消息请求/回复 ·可复用传统的XML工具 传输层 负责连接的建立和 数据的传输安全 基于RPC操作 ·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 • 采用面向连接的传输层协议承 载(SSH/SSL),安全性高, 可靠性更高 NETCONF支持对数据的分类存储和迁移,支持分阶段提交和配置 隔离。 <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 存的配置文件。 各数据集间可支持配置数据迁移: <copy-config> Running Running commit <copy-config> <commit> 3 NETCONF定义了更丰富的操作接口,并支持基于能力集进行扩展。 NETCONF支持的基本操作(RFC6241) <get>: 用来从<running/>中获取部分或全部运行配置数据和状态数据 <get-config>: 用于获取配置信息 <edit-config>: 用来修改、创建、删除配置数据 <copy-config>: 用源配置数据集替换目标配置数据集 <delete-config>: 用来删除非运行数据集的所有数据 <lock>: 用来锁定设备的配置数据集,独占配置数据集的修改权 <unlock>: 用来解锁配置数据集,取消独占 <close-session> : 请求正常终止NETCONF会话 <kill-session> : 用来强制关闭另一个NETCONF会话 可扩展能力 RFC6241: RFC 5277: Writable-Running 能力 Notification 能力 Candidate Configuration能力 Interleave 能力 Confirmed Commit 能力 Rollback-on-Error能力 RFC 6243: with-defaults 能力 Validate能力 Startup能力 URL能力 RFC 6022: XPath能力 letf-netconf-monitoring 能力 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模型具有如下优点: ◆ 定义丰富: 支持丰富的基础数据类型定义和数据属性定义 ◆ 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 ◆ 可扩展:支持grouping重用、refine重定义和augment扩展,且支持基于 typedef 扩展数据类型定义 ◆ 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 人工输入: ip address ip-address { mask | mask-length } ip address 1.1.1.1 24 客户端发送xml请求报文: { list ip <ip xmlns="http://www.huawei.com/ when "not(../../addrCfgType='negotiation')"; netconf/vrp/huawei-ip"> key ip-address <ip-address>1.1.1.1</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 用户 服务 工程师 功能 (Step1) 人工阅读手册 (Step2) 人工理解差异 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? (Step3) 人工输入命令行 设备运行数据 设备运行数据 IPv4配置命令: IPv4和IPv6命令采用相同的 ipv4 address ipv4-address { 格式: mask | mask-length } ip address ip-address { mask IPv6配置命令: | mask-length } ipv6 address { ipv6-address prefix-length | ipv6address/prefix-length } 设备 厂商A设备 厂商B设备 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 配置。 用户 服务 工程师 工程师 YANG模型B ANG 模型 自动生成XML报文 自动生成XML报文 设备运行数据 设备运行数据 YANG 模型B: YANG 模型A: { ip { ip { ip-address { ipv4 mask } { ipv4-address mask } } } { ipv6 { ipv6-address mask } } } 服务器 厂商A设备 厂商B设备 NETCONF & YANG的发展 REC 3535 RFC 6020 IAB(因特网架构委员会)会议 认为当前网络管理存在一系列 问题,需要新的方法解决 规模的在标准组织中提出。 基础定义以及和NETCONF 2015· 2006.11 2010.10 2011.7 2014 2016 YANG成为业 界主流数据模型 RFC 4771 RFC 6241 NETCONF工作组发布NETCONF第一版 定义了协议的基本框架和操作,对于内容的 xml没有定义约束,定义考虑RFC3535的 一些问题的解决方法 NETCONF工作组发布netconf 第二版,确定与YANG的结合 -YANG的出现推动了NETCONF的发展 ◆ 基于YANG定义了通用的模型打破了原先通过封装命令行的NETCONF的一些实现, 能够为各厂商实现通用的模型奠定基础 通过YANG定义的网络模型在多厂商实现也推动NETCONF的发展到了更新的阶段 NETCONF会话基本流程示例 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

(Step 1) 建立NETCONF会话,并通过<hello>消息进行能力通告 <hello xmlns="urn:ietf:params:xml:ns:</pre> <hello xmlns="urn:ietf:params:xml:ns: netconf:base:1.0"> netconf:base:1.0"> <capabilities> <capabilities> <capability>urn:ietf:params:netconf: <capability>urn:ietf:params:netconf: base:1.1</capability> base:1.1</capability> </capabilities> </capabilities> <session-id>1</session-id> </hello> </hello> Step 2)锁定运行数据集,避免和其他客户端冲突 申请锁定运行数据集 确保别人不影响我的操作 <rpc-reply message-id="2" xmlns="</pre> <rpc xmlns="urn:ietf:params:xml:ns:</pre> netconf:base:1.0" message-id="2"> urn:ietf:params:xml:ns:netconf: <lock> base:1.0"> <target> <nk/> <running></running> </rpc-reply> </target> </lock> </rpc> 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 申请复制 <rpc-reply message-id="3" xmlns= <rpc xmlns="urn:ietf:params:xml:ns:</pre> netconf:base:1.0" message-id="3"> "urn:ietf:params:xml:ns:netconf: hase:1.0"> <copy-config> <ok/> <target> <candidate></candidate> </rpc-reply> </target> <source> <running></running> </source> </copy-config> </rpc> Step 4)编辑备选数据集的配置

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

<rpc-reply message-id="7" xmlns="urn:ietf:</pre>

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

</rpc-reply>

Yet Another Next Generation

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

<ok/>

</rpc-reply>

请求配置接口GE1/0/0的

<rpc xmlns="urn:ietf:params:xml:ns:
netconf:base:1.0" message-id="4">

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

<interfaces>

xml:ns:netconf:base:1.0">

Step 6 解锁运行数据集

<unlock> <target>

</target>

YANG

我的配置完成了,可以解锁了 <pr

netconf:base:1.0" message-id="7">

<running></running>

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

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

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

IP地址为1.1.1.1

<edit-config>

<target> <candidate/> </target>

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。

生效模式。

<addrCfgType>config</addrCfgType> <am4CfgAddrs> <am4CfgAddr> <iflpAddr>1.1.1.1</iflpAddr> <subnetMask>255.255.255.252</ subnetMask> <addrType>main</addrType> </am4CfgAddr> </am4CfgAddrs> </ipv4Config> </interface> </interfaces> </ifm> </config> </edit-config> </rpc> Step 5)提交备选数据集配置到运行数据集 申请提交配置 <rpc xmlns="urn:ietf:params:xml:ns:</pre> <rpc-reply message-id="6" xmlns= netconf:base:1.0" message-id="6"> "urn:ietf:params:xml:ns:netconf:base:1.0"> <commit></commit> <ok/> </rpc-reply>

后续处理 关闭NETCONF会话,关闭SSH连接

缩略语

 英文

NETCONF Network Configuration Protocol

XML Extensible Markup Language

RPC Remote Procedure Call