* 1. 接口层
     1. 数据接入服务
        1. 数据采集接口实现
           1. 网管侧Telemetry静态订阅接口

Telemetry静态订阅是指设备作为客户端，采集器作为服务端，由设备主动发起到采集器的连接，进行数据采集上送，长时间、周期性地监控比如某个端口的端口流量趋势。

对于网管侧来说，Telemetry=收集系统+存储系统+应用分析系统

Telemetry网管侧和设备侧协同运作，完成整体的Telemetry静态订阅需要五个操作步骤顺序执行，每个步骤的内容和网管侧所需接口如下：

1. 静态配置：控制器通过命令行配置支持Telemetry的设备，订阅数据源，完成数据采集。

输入：命令行输入或配置文件

输出：下发给订阅网络设备的通信协议报文

1. 推送采样数据或自定义事件：网络设备依据控制器的配置要求，将采集完成的数据或自定义事件，上报给采集器进行接收和存储。

输入：接收到的来自网络设备的数据报文，报文内容编码格式为JSON或GPB

输出：采集到的原始数据

1. 读取数据：分析器读取采集器存储的采样数据或自定义事件。

输入：命令行输入

输出：已储存的网络设备原始数据

1. 分析数据：分析器分析读取到的采样数据或自定义事件，并将分析结果发给控制器，便于控制器对网络进行配置管理，及时调优网络。

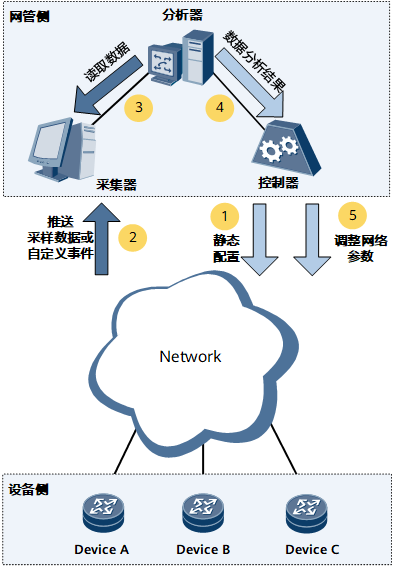
输入：命令行输入

输出：数据分析结果

1. 调整网络参数：控制器将网络需要调整的配置下发给网络设备；配置下发生效后，新的采样数据或自定义事件又会上报到采集器，此时Telemetry网管侧可以分析调优后的网络效果是否符合预期，直到调优完成后，整个业务流程形成闭环。

输入：数据分析结果

输出：下发给订阅网络设备的通信协议报文



图表 1 Telemetry静态订阅

* + - * 1. 网管侧Telemetry动态订阅接口

Telemetry动态订阅是指设备作为服务端，采集器作为客户端发起到设备的连接，由设备进行数据采集上送。

当用户想要监控某些接口，可以配置Telemetry动态订阅功能。在不感兴趣时，断开连接即可，订阅自动取消且不会配置恢复，从而避免对设备造成长期负载，也简化了用户和设备的交互。

Telemetry网管侧和设备侧协同运作，完成整体的Telemetry动态订阅需要五个操作步骤顺序执行，每个步骤的内容和网管侧所需接口如下：

1. 动态配置：支持Telemetry的设备在完成gRPC服务的相关配置后，由采集器下发动态配置到设备，完成数据采集。

输入：命令行输入或配置文件

输出：下发给订阅网络设备的通信协议报文

1. 推送采样数据：网络设备依据采集器的配置要求，将采集完成的数据，上报给采集器进行接收和存储。

输入：接收到的来自网络设备的数据报文，报文内容编码格式为JSON或GPB

输出：采集到的原始数据

1. 读取数据：分析器读取采集器存储的采样数据。

输入：命令行输入

输出：已储存的网络设备原始数据

1. 分析数据：分析器分析读取到的采样数据，并将分析结果发给控制器，便于控制器对网络进行配置管理，及时调优网络。

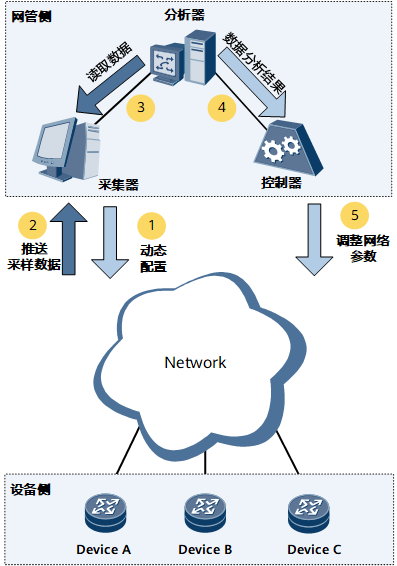
输入：命令行输入

输出：数据分析结果

1. 调整网络参数：控制器将网络需要调整的配置下发给网络设备；配置下发生效后，新的采样数据又会上报到采集器，此时Telemetry网管侧可以分析调优后的网络效果是否符合预期，直到调优完成后，整个业务流程形成闭环。

输入：数据分析结果

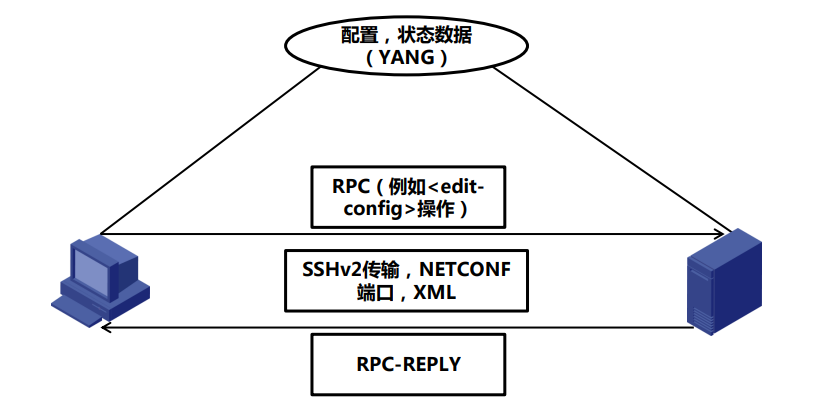
输出：下发给网络设备的通信协议报文



图表 2 Telemetry动态订阅

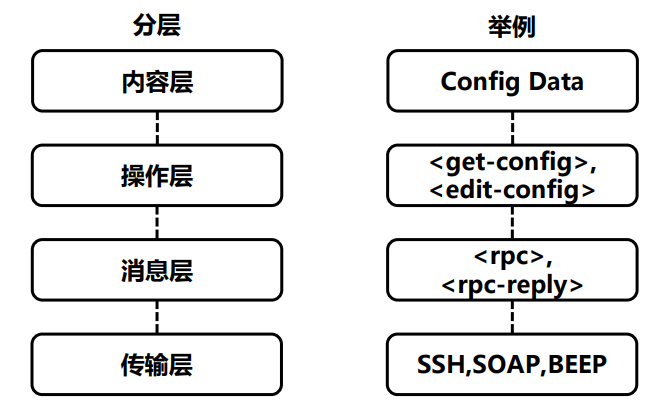
* + - 1. NETCONF接口实现

NETCONF（Network Configuration Protocol）是一种基于XML的网络管理协议，它提供了一种可编程的、对网络设备进行配置和管理的方法，采用基于TCP的SSHv2进行传送，以RPC的方式实现操作和控制。



图表 3 NETCONFIG系统架构

NETCONF协议分为四层：内容层、操作层、消息层、传输层



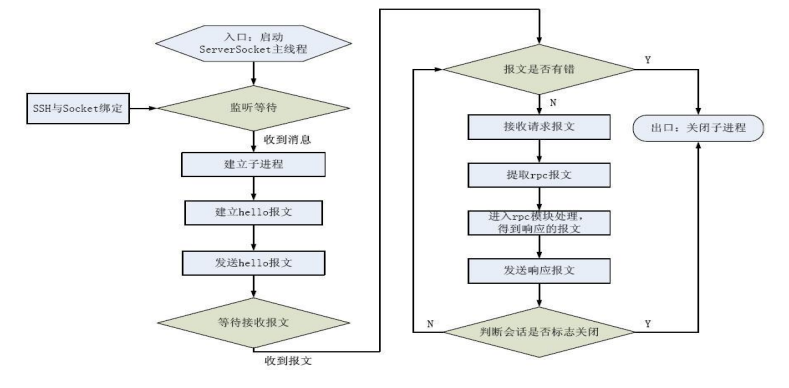
图表 4 NETCONF分层

NETCONFIG网络架构的主要元素：

* 客户端(Client)，主要作用如下
  + 利用NETCONFIG协议对网络设备进行系统管理‘。
  + 向向NETCONF Server发送请求，查询或修改一个或多个具体的参数值。
  + 向NETCONF Server发送请求，查询或修改一个或多个具体的参数值。
* 服务器(Server)，主要用于维护被管理设备的信息数据并响应客户端的请求
  + 服务器收到客户端的请求后会进行数据解析，然后给客户端返回响应。
  + 当设备发生故障或其他事件时，服务器利用Notification机制主动将设备的告警和事件通知给客户端，向客户端报告设备的当前状态变化。
* 配置数据集(Datastores)
* NETCONF定义了一个或多个配置数据集的存在，并允许对它们进行配置操作。配置数据集被定义为使设备从其初始默认状态进入期望的操作状态所需的完整配置数据集合。
* 运行的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。 在设备上只存在一个此类型的配置数据集，并且始终存在。
* 基本模型中只存在配置数据集。 其他配置数据集可以由能力集定义，且只在宣称支持该能力集的设备上可用。

NETCONFIG会话流程：

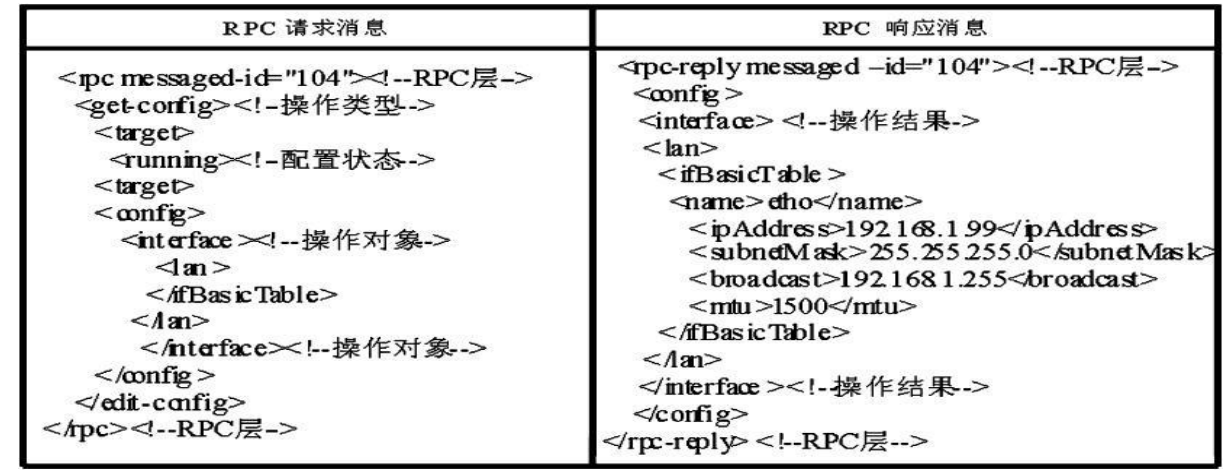
* 客户端触发NETCONF会话建立，完成SSH连接建立，并进行认证与授权。
* 客户端和服务器完成NETCONF会话建立和能力协商。
* 客户端发送一个或多个请求给服务器，进行RPC交互（鉴权）。例如：
* 修改并提交配置。
* 查询配置数据或状态。
* 对设备进行维护操作。
* 客户端关闭NETCONF会话。
* SSH连接关闭。
  + - * 1. 传输层通信接口



图表 5传输层处理机制

* + - * 1. 消息层RPC报文封装/解封装接口

RPC 层主要作用是接收经过NETCONF 传输层传递过来的XML格式的请求报文和回传响应报文，进行对应的解析和封装。



图表6 RPC请求和RPC响应各部分内容

* + - * 1. 操作层接口

NETCONF定义对被管理设备的各种基础操作。

|  |  |  |
| --- | --- | --- |
| 操作分类 | 操作名称 | 功能 |
| 取值操作 | get/get-bulk | 获取状态数据和running数据库的配置数据 |
| get-config/<get-bulk-config> | 获取某数据库的配置数据 |
| 配置操作 | edit-config | 修改、添加、删除配置数据 |
| copy-config | 替代某数据库中的所有数据 |
| delete-config | 删除某数据库中的所有数据 |
| 锁操作 | lock | 对某数据库上锁 |
| unlock | 对某数据库解锁 |
| 会话操作 | close-session | 结束自身的NETCONF会话 |
| kill-session | 强行结束其他的NETCONF会话 |

<get>和<get-bulk>接口

<get>操作用来获取数据，包括运行状态数据和配置数据。

<get-bulk>操作用来从指定索引的下一条开始批量获取后续N条数据（索引行数据不返回），包括运行状态数据和配置数据。用户通过index属性由指定索引，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。

参数：

filter：此参数指定要检索的系统配置和状态数据部分。如果此参数不存在，则返回所有设备配置和状态信息

positive response：如果设备能够满足请求，则发送 <rpc-reply>。<data> 部分包含适当的子集。

negative response：如果由于任何原因无法完成请求则 <rpc-reply> 中会包含 <rpc-error> 元素

1. 客户端发送报文

<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>

其中，getoperation可以为get或者get-bulk。<filter>选项用于过滤信息，<filter>中可包括模块名、子模块名、表名和列名：

·     如果不指定模块（子模块），则表示全部模块（子模块）。一旦指定模块（子模块），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。一旦指定表，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。如果同时指定了索引列之外的其他列，则意味着返回的数据仅仅包含索引列和指定的列。

1. 结果验证

设备收到配置获取请求报文后会将相应参数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>

<get-config>和<get-bulk-config>接口

<get-config>和<get-bulk-config>用来获取系统中所有可配置的变量的值，配置的方式包括CLI、MIB、Web等。

参数：

source：被查询的配置数据存储名称，如<running/>

filter：此参数标识要检索的设备配置数据存储的部分。如果此参数不存在，则返回整个配置。

positive response：如果设备可以满足请求，服务器会发送一个 <rpc-reply> 元素，其中包含一个带有查询结果的 <data> 元素。

negative response：如果由于任何原因无法完成请求，则 <rpc-reply> 中会包含 <rpc-error> 元素。

1. 客户端发送报文

<rpc message-id="101"

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

<get-config>

<source>

<running/>

</source>

<filter type="subtree">

<top xmlns="http://example.com/schema/1.2/config">

<users/>

</top>

</filter>

</get-config>

</rpc>

source: 被查询的配置数据存储的名称，例如<running/>。

filter: 此参数标识要检索的设备配置数据存储部分。 如果此参数不存在，则返回整个配置

1. 结果验证

设备收到配置获取请求报文后会将相应参数的值通过如下报文反馈给客户端：

<rpc-reply message-id="101"

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

<data>

<top xmlns="http://example.com/schema/1.2/config">

<users>

<user>

<name>root</name>

<type>superuser</type>

<full-name>Charlie Root</full-name>

<company-info>

<dept>1</dept>

<id>1</id>

</company-info>

</user>

<!-- additional <user> elements appear here... -->

</users>

</top>

</data>

</rpc-reply>

<edit-config>接口

此操作允许以多种方式表示新配置，例如使用本地文件、远程文件或内联。如果目标配置数据存储不存在，则会创建它。

参数：

target：正在编辑的配置数据存储的名称，例如<running/>或<candidate/>。

default-operation：为此 <edit-config> 请求选择默认操作。<default-operation> 参数的

默认值为“merge”。<edit-config>支持如下operation选项：merge、create、replace、remove、delete

merge：将包含该属性的元素标识的配置数据与<target>参数标识的配置数据存储中相应级别的配置合并。这是默认行为。

create: 当且仅当配置数据不存在于配置数据存储中时，包含此属性的元素标识的配置数据才会添加到配置中。如果配置数据存在，则返回一个 <rpc-error> 元素，其 <error-tag> 值为“data-exists”。

replace: 由包含此属性的元素标识的配置数据替换由 <target> 参数标识的配置数据存储中的任何相关配置。如果配置数据存储中不存在此类配置数据，则会创建它。与替换整个目标配置的 <copy-config> 操作不同，只有 <config> 参数中实际存在的配置受到影响。

remove: 如果配置数据当前存在于配置数据存储, 删除包含该属性的元素所标识的配置数据.如果配置数据不存在，“删除”操作将被服务器静默忽略。

delete: 当且仅当配置数据当前存在于配置数据存储中时，从配置中删除由包含该属性的元素标识的配置数据。如果配置数据不存在，则返回一个 <rpc-error> 元素，其中的 <error-tag> 值为“data-missing”。

1. 客户端发送报文

<?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>

   失败时默认操作

</error-option>

    <config>

      <top xmlns="http://www.xxx.com/netconf/config:1.0">

        指定模块名，子模块名，列名，表名

      </top>

    </config>

  </edit-config>

</rpc>

1. 结果验证

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>

<copy-config>接口

使用另一个完整配置数据存储的内容创建或替换整个配置数据存储。如果目标数据存储存在，则将其覆盖。否则将创建一个新的。

参数：

target：配置数据存储的名称，用作<copy-config> 操作的目标。

source：配置数据存储的名称，用作<copy-config> 操作或 <config> 元素的源，包含要复制的完整配置。

positive response：如果设备能够满足请求，则发送包含 <ok> 元素的 <rpc-reply>。

negative response：如果由于任何原因无法完成请求，则 <rpc-reply> 中包含 <rpc-error> 元素。

<delete-config>接口

删除配置数据存储，但<running>配置数据存储无法删除。

参数：

target：配置数据存储的名称，用作<delete-config> 操作的目标。

positive response：如果设备能够满足请求，则发送包含 <ok> 元素的 <rpc-reply>。

negative response：如果由于任何原因无法完成请求，则 <rpc-reply> 中包含 <rpc-error> 元素。

<lock>接口

<lock> 操作允许客户端锁定设备的整个配置数据存储系统。此类锁定旨在短暂存在，并允许客户端进行更改而不必担心与其他 NETCONF 客户端、非ETCONF 客户端（例如，SNMP 和命令行界面 (CLI)脚本）发生冲突。

现有会话或其他实体对锁定目标的任何部分持有锁定，锁定配置数据存储的尝试必须失败。当获得锁时，服务器必须阻止对锁定资源的任何更改，而不是该会话请求的更改。修改资源的 SNMP 和 CLI 请求必须失败并出现适当的错误。

参数：

target：要锁定的配置数据存储的名称

positive response：如果设备能够满足请求，则发送包含 <ok> 元素的 <rpc-reply>。

negative response：如果由于任何原因无法完成请求，则 <rpc-reply> 中包含 <rpc-error> 元素。

以下示例显示了成功获取锁：

请求：

<rpc message-id="101"

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

<lock>

<target>

<running/>

</target>

</lock>

</rpc>

响应：

<rpc-reply message-id="101"

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

<ok/> <!-- lock succeeded -->

</rpc-reply>

以下示例显示获取锁失败：

请求：

<rpc message-id="101"

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

<lock>

<target>

<running/>

</target>

</lock>

</rpc>

响应：

<rpc-reply message-id="101"

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

<rpc-error> <!-- lock failed -->

<error-type>protocol</error-type>

<error-tag>lock-denied</error-tag>

<error-severity>error</error-severity>

<error-message>

Lock failed, lock is already held

</error-message>

<error-info>

<session-id>454</session-id>

<!-- lock is held by NETCONF session 454 -->

</error-info>

</rpc-error>

</rpc-reply>

<unlock>接口

<unlock> 操作用于释放先前通过 <lock> 操作获得的配置锁。

如果以下任一条件为真，则 <unlock> 操作将不会成功：

\* 指定的锁当前未激活。

\* 发出 <unlock> 操作的会话与获得锁的会话不同。

服务器必须以 <ok> 元素或 <rpc-error> 响应。

参数：

target：要解锁的配置数据存储的名称。不允许 NETCONF 客户端解锁它未锁定的配置数据存储。

positive response：如果设备能够满足请求，则发送包含 <ok> 元素的 <rpc-reply>。

negative response：如果由于任何原因无法完成请求，则 <rpc-reply> 中包含 <rpc-error> 元素。

<close-session>接口

请求正常终止 NETCONF 会话。当 NETCONF 服务器收到 <close-session> 请求时，它会

优雅地关闭会话。服务器将释放与会话关联的所有锁和资源，并优雅地关闭任何关联的连接。在 <close-session> 请求之后收到的任何 NETCONF 请求都将被忽略。

positive response：如果设备能够满足请求，则发送包含 <ok> 元素的 <rpc-reply>。

negative response：如果由于任何原因无法完成请求，则 <rpc-reply> 中会包含 <rpc-error> 元素

示例：

<rpc message-id="101"

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

<close-session/>

</rpc>

<rpc-reply message-id=" 101"

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

<ok/>

</rpc-reply>

<kill-session>接口

强制终止 NETCONF 会话。当一个 NETCONF 实体收到一个打开会话的 <kill-session> 请求时，它将中止当前正在进行的所有操作，释放与会话相关的所有锁和资源，并关闭所有相关的连接。如果 NETCONF 服务器在处理确认提交时收到 <kill-session> 请求，它必须将配置恢复到发出确认提交之前的状态。否则，<kill-session> 操作不会回滚由持有锁实体所做的配置或其他设备状态修改。

参数：

session-id：要终止的NETCONF会话的会话标识。 如果此值等于当前会话 ID，则返回“无效值”错误。

positive response：如果设备能够满足请求，则发送包含 <ok> 元素的<rpc-reply>。

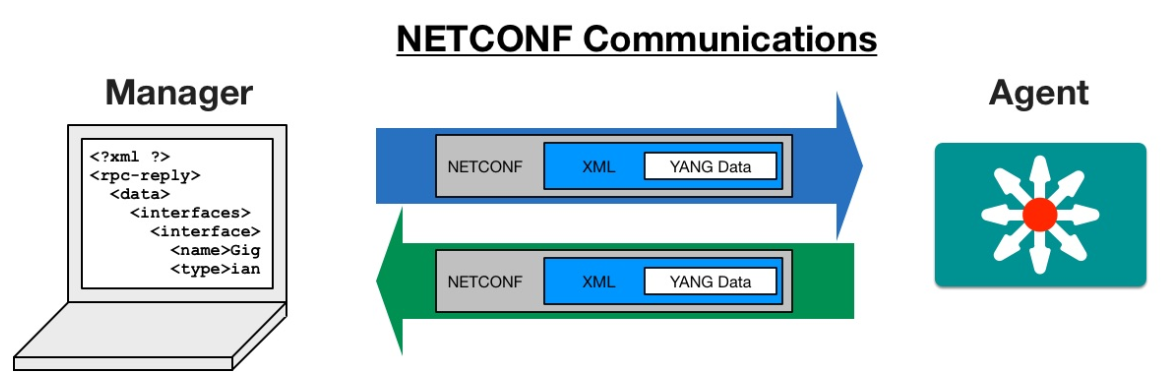
negative response：如果由于任何原因无法完成请求，则 <rpc-reply> 中会包含 <rpc-error> 元素。

* + - * 1. 内容层-YANG解析接口

NETCONF内容层是唯一没有被标准化的层。内容层表示的是被管对象的集合，可以是配置数据、状态数据、统计信息等。内容层的内容需要来自数据模型，数据模型需要建模语言描述，例如：YANG

YANG在网络设备数据中的使用：

* 使用带有XML（数据格式）的NETCONF（传输协议）通信。
* 发送到网络设备或从网络设备发送的YANG数据将根据协议格式化为XML或JSON（例如：NETCONF或RESTCONF）。



图表8 NETCONF通信

YANG模型对各功能模型的解析：

VLAN模型

VLAN模型包含VLAN管理和子接口管理

模型构建

VLAN 管理

|  |  |  |  |
| --- | --- | --- | --- |
| 节点 | 说明 | 取值范围 | 备注 |
| vlanid | 管理VLAN ID | 整数形式，取值范 围为1～4094。 | NA |
| id | VLAN ID | 整数形式，取值范 围为1～4094。 | NA |
| Vlan-pool/name | VLAN pool名称 | 字符串类型，可输 入的字符串长度为 1～31个字符。不 能包含“?”和空 格，双引号不能出 现在字符串的首 尾。 | NA |
| vlans | 添加到VLAN pool 中的VLAN范围 | 整数形式，取值范 围为1～4094，格式为startvlan[..end-vlan ]。其中， start-vlan必须小于 end-vlan。例如， 10..20表示将VLAN 10～20添加到 VLAN pool中。 | 没有创建的VLAN 也能够加入到 VLAN pool中，请 注意将VLAN加入 VLAN pool后，必 须创建该VLAN， 否则该VLAN不生 效。 |
| assignmentmethod | VLAN pool中的 VLAN分配算法 | 枚举值类型，取值 如下： ● even：配置 VLAN分配算法 为顺序分配。 ● hash：配置 VLAN分配算法 为哈希分配。 | NA |
| port-link-type | 接口类型 | 接口类型 ● access：接口类 型为Access。 ● hybrid：接口类 型为Hybrid。 ● trunk：接口类 型为Trunk。 ● dot1q-tunnel： 接口类型为 Dot1qtunnel。 | NA |
| default-vlan | 缺省VLAN ID | 整数形式，取值范 围为1～4094。 | NA |
| trunk | Trunk类型 |  | NA |
| trunk-vlan | Trunk类型接口允 许通过的VLAN | ● VLAN list：整 数形式，取值范 围为1～4094。 ● VLAN range： 仅支持输入 “1..4094”， 即仅支持配置 trunk类型接口 允许所有VLAN 通过。 | NA |
| hybrid | Hybrid类型 |  | NA |
| taggedvlan | Hybrid类型接口允 许通过的VLAN， 这些VLAN的帧以 Tagged方式通过接 口 | 整数形式，取值范 围为1～4094。 | NA |
| untagged vlan | Hybrid类型接口允 许通过的VLAN， 这些VLAN的帧以 Untagged方式通 过接口 | 整数形式，取值范 围为1～4094。 | NA |
| vlan/name | VLAN的名字 |  | NA |
| vlan/ description | VLAN的描述 |  | NA |
| ipv4- check-enable | 是否使能VLAN下 的IPv4报文检查功能 | 布尔型，具体取值 如下： ● true：使能 ● false：去使能 | NA |
| ipv6- check-enable | 是否使能VLAN下 的IPv6报文检查功 能 | 布尔型，具体取值 如下： ● true：使能 ● false：去使能 | NA |
| arp-check-enable | 是否使能VLAN下 的ARP报文检查功 能 | 布尔型，具体取值 如下： ● true：使能 ● false：去使能 | NA |
| arp-check-alarm-enable | 是否使能VLAN下 的ARP报文检查告 警功能 | 布尔型，具体取值 如下： ● true：使能 ● false：去使能 |  |
| arp-check-alarm-threshold | 配置VLAN下的ARP 报文检查功能的告 警阈值 | 整数形式，取值范 围1～1000。 | arp-check-alarm-enable节点取值为 true时，可配置该 节点。 |
| ip-check-alarm-enable | 是否使能VLAN下 的IP报文检查告警 功能 | 布尔型，具体取值 如下： ● true：使能 ● false：去使能 缺省值是false。 |  |
| ip-check-alarm-threshold | 配置VLAN下的IP报文检查功能的告警阈值 | 整数形式，取值范 围1～1000。 | ip-check-alarm-enable节点取值为true时， 可配置该节点。 |
| Ip-prefix | 基于IP子网划分 VLAN依据的源IP地 址或网络地址和掩 码长度 | 格式为ipv4- address/masklength： ● ipv4-address表 示源IP地址或网 络地址，采用点 分十进制形式。 ● mask-length表 示掩码长度，采 用整数形式，取 值范围是0～ 32。 | NA |
| priority | IP地址或网段对应 VLAN的802.1p优 先级 | 整数形式，取值范 围是0～7，值越大 优先级越高。缺省 值是0。 | NA |
| enable | 是否使能接口基于 IP子网划分VLAN的功能 | 布尔型，具体取值 如下： ● true：使能 ● false：去使能 缺省值是false。 | NA |
| Unknown-flow-drop | 是否使能VLAN内 未知报文隔离功能 |  |  |
| block-time | 指定VLAN pool中 VLAN的锁定时 长。 |  |  |

子接口管理

|  |  |  |  |
| --- | --- | --- | --- |
| 节点 | 说明 | 取值范围 | 备注 |
| termination-type | 终结子接口类型 | 枚举值类型，取值 如下： ● dot1q：配置子 接口为Dot1q终 结子接口。 ● qinq：配置子接 口为QinQ终结 子接口。 | NA |
| dot1q-termination-list | Dot1q终结子接口 对单层Tag报文的 终结功能 | 子接口对用户报文 配置规则的数量最 多为8条，每条规 则配置的VLAN范 围不超过128个。 | NA |
| qinq-termination-mod | QinQ终结子接口 对单层Tag报文的 终结功能 | 子接口对用户报文 配置规则的数量最 多为8条，每条规 则配置的VLAN范 围不超过128个。 | NA |
| statistics-enable | 子接口流量统计功 能 | 布尔型，具体取值 如下： ● true：使能。 ● false：未使 能。 | NA |
| statisticsdirection | 子接口流量统计功 能统计流量的方向 | 枚举值类型，取值 如下： ● both：配置子 接口流量统计功 能统计接口双向 流量。 ● inbound：配置 子接口流量统计 功能统计接口入 方向流量。 ● outbound：配 置子接口流量统 计功能统计接口 出方向流量。 | NA |

QoS模型

YANG模型应对不同业务或用户的时延、抖动、固定带宽、保证带宽和最大带宽等SLA参数进行建模，方便NETCONF协议针对不同的用户和业务进行不同的SLA参数配置方法。

QoS机制包括业务流分类（Traffic classification）、优先级标记（Marking）、排队及调度（Queuing and scheduling）、流量整形（Traffic shaping）和流量管制（Traffic policing）、拥塞避免（Congestion avoidance）、缓存管理（Buffer management）等。YANG模型应该对接收到的数据包进行建模，获取对应帧结构实现对应功能。

功能分析

* 业务流分类

功能：厂家设备采用NETCONF配置参数应实现对上行业务流进行分类的功能。

分析：YANG模型基于GEM Port-ID、User Priority（IEEE 802.1Q）、EtherType（例如IPoE、PPPoE等）、目的IP地址、源IP地址、IP协议类型（TCP、UDP、ICMP、IGMP等）、IP TOS/DSCP和L4协议端口号建模，实现对上行业务流的分类能力。

* 优先级标记

功能：厂家设备根据NETCONF协议反馈的信息可以对上行业务优先级进行标识，同时能够实现上行流分类与上行业务优先级标识的映射。

分析：YANG模型应该支持以太网PRI字段作为优先级标识，同时可选用IP的TOS/DSCP域作为优先级标识。

* 队列调度

功能：厂家设备处理上、下行业务时，需要将业务映射到不同的优先级队列上，并进行队列调度。

分析：YANG语言对队列和调度策略进行建模，厂家根据需要在调度策略模块中添加策略，并且通过NETCONF进行调用。

* 限速功能

功能：厂家设备需要支持DBA机制，且具备限制下行业务流速率的能力，支持L2 Traffic Shaping或Policing机制。

分析：YANG语言对基于用户侧的以太网端口、VLAN ID和User Priority（IEEE 802.1D）建模，调用策略模块对上述模块进行配置管理。

模型构建

QoS模板配置节点

|  |  |  |  |
| --- | --- | --- | --- |
| 节点 | 说明 | 取值范围 | 备注 |
| name | 指定QoS模板的名称。 | 字符串形式，区分大小写，不支持空格，长度范围是1 ～31。当输入的字符串两端使用双引号时，可在字符串中输入空格。模板名称不能是--。 |  |
| direction | 指定流量监管的方向。 | 枚举类型，取值如下：   * + - inbound：入方向。 * outbound：出方向。 | NA |
| cir | 配置承诺信息速率 （Committed Information Rate），即保证能够通过的平均速率。 | 整数形式，取值范围是64～ 4294967295，单 位为kbit/s。 | 若在QoS模板中配 置流量监管，则必须配置本节点。 |
| cbs | 配置承诺突发尺寸 （Committed Burst Size），即瞬 间能够通过的承诺 突发流量。 | 整数形式，取值范围是4000～ 4294967295，单 位为byte。如果不指定峰值信息速率，承诺突发尺寸缺省值为承诺信息速率的188倍。如果指定峰值信息速率，承诺突发尺寸缺省值为承诺信息速率的125倍。 | 本节点需与pbs节点同时配置 |
| pir | 配置峰值信息速率 （Peak Information Rate），即最大能 够通过的速率。 | 整数形式，取值范围是64～ 4294967295，单位为kbit/s。峰值信息速率必须大于等于承诺信息速率.缺省值等于承诺信息速率。 | NA |
| pbs | 配置峰值突发尺寸 （Peak Burst Size），即瞬间能够通过的峰值突发流量。 | 整数形式，取值范围是4000～ 4294967295，单 位为byte。如果不 指定峰值信息速率，峰值突发尺寸缺省值为承诺信息速率的313倍。如果指定峰值信息速率，峰值突发尺寸缺省值为峰值信息速率的125倍。 | 本节点需与cbs节点同时配置 |
| priority-direction | 指定重标记报文的 方向。 | 枚举类型，取值如下：  ● inbound：入方向。  ● outbound：出方向。 | 无论配置重标记 VLAN报文的 802.1p优先级，还是配置重标记IP报文的DSCP优先级，都必须配置本节点。 在配置重标记 VLAN报文802.1p 优先级时，虽然指 定方向无意义，但 是必须指定本节点 的值为inbound或 者outbound，才 能保证其他节点配置成功。 如果一次性下发重 标记报文802.1p优先级的重复配置， 既有指定本节点的 值为inbound的一 组配置，又有指定本节点的值为 outbound的S另一组配置，那么这两 组配置中重标记报文的优先级的值必须相同，否则将无法下发配置。若需要删除重标记报文 优先级的有关配置，则必须分别删除两组配置。 |
| priority-type | 指定重标记报文的 优先级的类型。 | 枚举类型，取值如下：  ● 8021p：重标记 VLAN报文 802.1p优先级。  ● dscp：重标记IP 报文的DSCP优先级。 | NA |
| priority-value | 配置重标记报文的 优先级的值。 | 整数形式，值越大优先级越高。若指 定IP报文的DSCP优先级值，取值范围 是0～63；若指定 VLAN报文的 802.1p优先级，取 值范围为0～7。 | NA |

队列调度和流量整形管理

|  |  |  |  |
| --- | --- | --- | --- |
| 节点 | 说明 | 取值范围 | 备注 |
| queue-profile/name | 队列调度模板名称。 | 字符串形式，不支持空格，不区分大小写。 | N/A |
| queue-profile/id | 队列索引。 | 整数形式，取值范围是0～ 7。 | 配置队列时， 必须8个队列同时配置。 |
| Schedule-mode | 队列调度模式。 | 枚举类型，取值如下：   * 绝对优先级（SP）调度策略 * 基于权重的相对优先级（WRR）调度策略 * 混合调度策略（SP+WRR） |  |
| weight | 参与WRR调度的队列的权重 |  | N/A |
| cir | 队列整形的承 诺信息速率。 | 整数形式，不 同的接口类型 取值范围不 同，取值范围 如下： ● 端口组：0 ～ 10000000  ● 其它接口： 0～X，X为接口实际可支持的最大速率单位为 kbit/s。缺省值为接口的最大带宽。 | N/A |
| pir | 整形峰值信息速率。 | 整数形式，不同的接口类型取值范围不同，取值范围如下：  ● 端口组：0～ 10000000  ● 其它接口： 0～X，X为接口实际可支持的最大速率单位为 kbit/s。缺省值为接口的最大带宽。 | 整形峰值信息速率必须大于等于队列整形的承诺信息速 率，缺省等于队列整形的承诺信息速率。 |
| cbs | 队列整形的承诺突发尺寸。 | 整数形式，取值范围是 10000～ 4294967295 ，单位是字节。 | NA |
| pbs | 指定峰值突发 尺寸，即瞬间 能够通过的峰 值突发流量。 | 整数形式，取值范围是10000～ 4294967295 ，单位是字 节。 | NA |
| trust | 对报文进行映 射的优先级类 型。 | 字符串类型， 取值如下：  ● cos： 802.1p优先 级  ● dscp： DSCP优先 级 | NA |
| queue-profile | 队列调度模板 名称。 | 已经存在的队 列调度模板名 称。 | NA |

告警模型

YANG模型对采集到的故障信息例如设备CPU、板卡温度、风扇及中继带宽占用率等不同模块进行数据建模，对各模块的信息设置告警值，超过告警值自动告警。

YANG模型还要针对不同的告警功能基于采集到的故障信息进行建模，故障管理功能包括网元故障告警、性能故障告警、业务故障告警、故障告警级别管理、故障定位、故障处理等功能。

YANG模型针对出现的典型告警，建立告警信息处理的数据模型，方便在告警出现时由NETCONF协议直接调用。

功能分析

* 故障信息采集

功能：系统应能通过厂家网管告警接口或直联网元补充方式等对所管理的网元、板卡、端口、链路等告警信息进行采集。

分析：使用NETCONFIG协议进行rpc调用，yang对告警数据模型进行建模，网元作为NETCONFIG服务器集成yang模块，该yang模块包括警报列表管理、警报搁置和通知管理系统的功能。还有一些操作来管理警报的操作员状态。该模块作为标准化的告警接口，需要仔细映射到相关的告警标准。

* 告警信息处理

功能：实现在本系统内对告警信息的确认/反确认，包括对告警信息的清除、过滤、屏蔽及故障的处理建议等功能，系统应具备告警的集中实时监控功能。

分析：收到告警信息并解析后，存入数据库，显示至主界面告警信息汇总页，由管理员进行处理。

* 告警过滤

功能：根据以下条目实现告警过滤：告警网元、告警级别条目、告警网元所属区域、告警网元设备型号、告警类型/描述等。过滤规则可定制。

分析：告警的数据模型中需要包括网元标识、告警级别、告警描述。

* 告警归并

功能：（1）支持通过以下规则进行冗余告警消息归并压缩：事件关联、拓扑关联、设备关联、时间关联。

（2）网管系统应具备重复告警的合并功能，在告警列表中显示告警次数、首次发生时间、最近发生时间，对于恢复了的告警，需要提供告警恢复时间等相关信息；

（3）对于非重要告警，可配置规则，使相关告警不在活动告警中显示，但相关告警需要能够在历史库进行查询查询；

（4）对于频繁 发生/恢复 的告警，能够按照单位时间，配置告警的翻转次数后，派生出独立的告警条目，该条目不得自行删除；

分析：告警本身需要和事件关联，网管侧能对告警进行拓扑关联并实现按设备分类的功能，告警的数据模型需要包含时间戳。

* 告警查询

功能：系统可实现对告警网元、设备类型、所属业务系统等多种组合条件的告警过滤查询，并可以通过设置告警过滤器方式进行查询条件的保存和分类展现。

分析：提供告警查询接口配置，要能够实现多种组合条件的告警过滤查询

* 告警监控
* 告警确认
* 告警清除与删除
* 告警前传
* 告警统计
* 告警事件
* 告警信息同步
* 告警重定义
* 告警压缩
* 告警升级
* 告警关联分析
* 告警知识经验库
* 告警显示控制
* 故障恢复

和数据模型无关的功能描述与分析略去。



模型构建

报警数据模型的设计目标：

o易于使用。如果一个系统支持这个模块，它应该可以直接将其集成到基于 YANG 的警报中

o 警报被视为资源状态，而不是离散的通知。

o 提供了“警报”的精确定义，以便排除不应作为警报转发的一般事件通知。

o 准确识别告警类型和告警实例。

o 管理系统应该能够拉出所有可用的警报，即从系统中读取警报清单。

告警查询模型

告警查询配置节点

|  |  |  |  |
| --- | --- | --- | --- |
| 节点 | 说明 | 取值范围 | 备注 |
| Id | 网元设备编号 | - |  |
| alarm-status | 告警状态 | * 未确认和未清除 * 已确认和未清除 * 未确认和清除 * 确认并清除 |  |
| start-time | 告警时间戳 | 格式为YYYY-MM-DDTHH:MM:SSZ（年-月-日T时:分:秒Z），表示查看指定时间之后产生的告警。 |  |
| severity | 告警级别 | 枚举值类型，按照严重程度由高至低的取值如下：   * critical：紧急 * major：重要 * minor：次要 * warning：提示 * indeterminate：不确定 | Critical：表示全局发生、业务中断或日志源故障的告警。  Major：表示在给定范围内发生并在一定程度上影响日志源或影响系统性能的告警。  Minir：表示不影响日志源的常见告警，或未来可能影响业务的潜在或即将发生的故障。  Warning：表示不影响系统性能或用户服务，但可能影响日志源或资源服务质量的告警。 |
| alarm-type-id | 告警类型 | * 通讯报警 * 日志源告警 * 服务质量告警 * 加工错误报警 * 安全警报 * 阈值报警 |  |

告警上报模型

|  |  |
| --- | --- |
| 节点 | 说明 |
| Alarm-type-id | 告警类型 |
| Alarm-type-qualifier | 告警标识符 |
| **severity** | 告警级别 |
| **Event-time** | 告警时间戳 |
| Alarm-text | 告警详细信息 |