SDN广域网下应用信息的获取与呈现

# 摘要

[摘要 2](#_Toc498027342)

[1 绪论 1](#_Toc498027343)

[1.1 研究背景 1](#_Toc498027344)

[1.2 研究目的和意义 1](#_Toc498027345)

[1.3 国内外相关研究情况 1](#_Toc498027346)

[1.4 本文的主要研究内容 1](#_Toc498027347)

[2 ADWAN应用信息分析相关技术分析 1](#_Toc498027348)

[2.1 ADWAN（Application-driven Wide Area Network） 1](#_Toc498027349)

[2.2 SDN控制器：ODL（OpenDaylight） 2](#_Toc498027350)

[2.3 报文统计技术 4](#_Toc498027351)

[2.4 数据存储 10](#_Toc498027352)

[2.5 应用信息报表的呈现 13](#_Toc498027353)

[2.6 本章小结 13](#_Toc498027354)

[3 ADWAN应用信息获取与呈现模块系统分析与设计 14](#_Toc498027355)

[3.1 ADWAN系统设计 14](#_Toc498027356)

[3.2 应用信息获取与呈现模块结构设计 15](#_Toc498027357)

[3.3 基础管理 16](#_Toc498027358)

[3.4 应用管理 19](#_Toc498027359)

[3.5 报文解析 20](#_Toc498027360)

[3.6 数据存储 20](#_Toc498027361)

[3.7 呈现层 21](#_Toc498027362)

[3.8 本章小结 22](#_Toc498027363)

[4 ADWAN应用信息分析的系统实现 22](#_Toc498027364)

[4.1 基础管理 22](#_Toc498027365)

[4.2 报文解析 24](#_Toc498027366)

[4.3 数据存储 24](#_Toc498027367)

[4.4 报表呈现 24](#_Toc498027368)

[4.5 本章小结 24](#_Toc498027369)

[5 SDN总结与展望 24](#_Toc498027370)

[5.1 全文总结 24](#_Toc498027371)

[5.2 展望 24](#_Toc498027372)

[致谢 24](#_Toc498027373)

[参考文献 24](#_Toc498027374)

# 绪论

近几年随着计算机技术的快速发展，大数据、云计算、移动互联网等业务的蓬勃兴起，这些服务在对人们生活提供了极大的便利的同时，也极大的增加了现有的计算机网络的负担，对目前的计算机网络的性能提出了更高的要求。基于这样的状况，SDN（Software Defined Network，软件定义网络）技术愈发被人们重视研究了起来，其热度持续升温。

## 研究背景

在2012年时，

## 研究目的和意义

## 国内外相关研究情况

## 本文的主要研究内容

# ADWAN应用信息分析相关技术分析

## ADWAN（Application-driven Wide Area Network）

ADWAN是H3C公司（新华三技术有限公司）的一款基于SDN框架ODL开发的新一代广域网解决方案。

ADWAN方案，基于全局视角，通过统一整合全网资源、多维观测网络状态、智能分析运行数据，使整个网络实现多层次、全方位的可视化，并根据用户策略和应用需求进行集中控制、全局调度、及实时调优，实现应用驱动的广域网服务，它具有以下特点：

* 全开放：首先是架构开放，网络的各个层次、不同组件是解耦的，做到网络可以很容易的扩展，也能很好的兼容多个厂商的设备；其次是接口开放，网络各组件之间通过开放、标准的接口来通信，从设备到控制器、APP提供多层次的、不同抽象度的API接口，赋予网络灵活的可编程可定义能力，能够让应用很容易的使用网络服务。
* 场景化：基于场景化的开发思路，提供面向业务的可定制APP应用，满足不同场景用户需求。
* 全流程：全流程重构广域网，整网视角管理和控制网络，简化运维管理。
* 端到端：业务驱动网络，基于业务应用的不同需求动态部署安全、WAN优化、CDN缓存等，提供端到端的网络服务。
* 可迁移：兼容传统网络，支持平滑迁移到SDN方案。

本项目既是基于ADWAN开发一个子模块功能，向用户提供应用信息查询的功能。通过符合ADWAN的编码规则，利用Maven、YANG模型使其嵌入进ADWAN中。

## SDN控制器：ODL（OpenDaylight）

### ODL概览

2013年，思科、微软、戴尔、惠普、IBM、英特尔、VMware等18个国际知名厂商联合建立了OpenDaylight项目。他们的共同目标是以透明、开放、公平、协作为原则建立一个供应商、客户、合作伙伴和开发人员可以共同使用的SDN开源平台，从而推动SDN的产品化和商业化。

ODL项目于2013年2月启动，2014年2月发布第一个正式版本，每隔8个月发布一个正式版本，截止目前为止已发布了氢（Hy）、氦（He）、锂（Li）、铍（Be）四个正式版本，能够为服务提供商和企业提供自动化服务交付、网络资源优化、云计算、NFV、以及区域网络的自动化、可见性和可控制相关的解决方案。

ODL控制器是一款基于Java开发的SDN控制器。其是SDN概念的一种具体实现，主要使用了以下工具：

**Maven**：ODL通过使用Maven来达到更加便捷的自动化构建项目的目的。

**OSGi**：是Java动态化模块化系统的一系列规范。ODL使用该OSGi框架以达到动态化的加载bundle和JAR包以及不同bundle之间的交换信息的目的。

**JAVA interfaces**：是同Java接口以达到事件监听、格式规范等目的。是一些bundle实现特定事件的回调函数的主要实现形式。

**REST APIs**：是ODL北向接口（northbound APIs）的接口形式。诸如拓扑管理、静态路由、主机跟踪等北向接口的接口形式。

ODL控制器通过向应用程序提供开放的基于web服务的北向接口来提供服务。

在南向（southbound）接口中，通过插件的形式提供诸如OpenFlow、BGP-LS等多种协议的支持。还有其他ODL捐赠者提供的一些特定的接口代码等模块。这些不同的模块功能都被动态的连接进了ODL的SAL（业务抽象层，Service Abstraction Layer）中。

SAL向北向接口提供服务。不管控制器底层使用的是什么协议、是什么网络设备，SAL都会清楚如何满足从北向传来的请求服务。对于控制器而言，它需要知道设备的功能、可达性等信息来控制其控制域中的设备。这些信息在拓扑管理（Topology Manager）模块中被存储和管理。

### ODL组织架构

图2-1-1 为ODL架构。

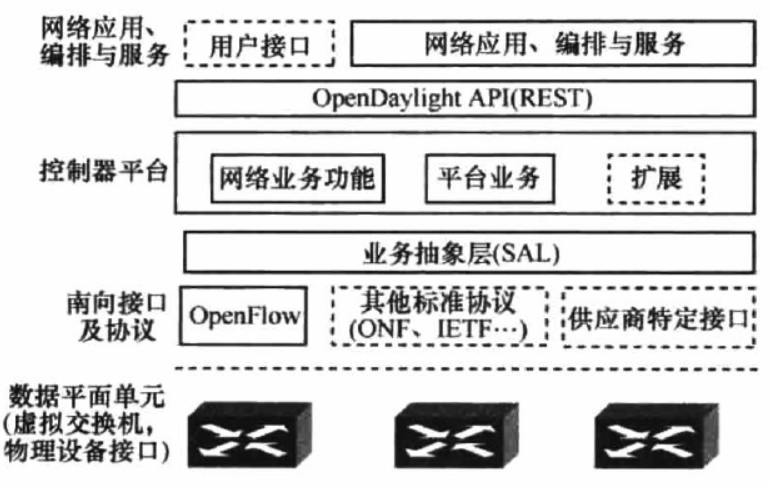


图2-2-1 ODL架构分析

可以看出，ODL有以下几大特点：

* 南向接口不仅支持OpenFlow，还支持SNMP、PCEP、BGP-LS等其他标准化协议，甚至允许出现私有化接口。
* 引入了SAL（业务抽象层），这使得上下层模块之间的调用可以相互隔离，屏蔽多种南向协议的差异，为上层的功能模块提供一致性服务。
* 北向提供开发可扩展的API，用户根据具体需要通过调用函数或者REST接口开发应用。
* ODL具有模块化、可扩展的控制器核心。采用OSGi（开放服务网关，Open Service Gateway Initiative）体系结构，解决功能组件之间的隔离问题，实现代码与功能的灵活加载，并可支持运行业务或应用的安装、更新、删除等插拔操作。
* 使用YANG工具可直接生成业务管理的“骨架”，开发者只需专注于具体业务，根据业务驱动模型工具来设计接口，实现业务功能。
* ODL拥有一个开源的分布式网络平台，该平台不仅能实现数据的存储、查找和监听，更重要的是它使得ODL支持控制器集群。

## 报文统计技术

本项目的主要功能：应用信息的获取与呈现，其第一步即为分析从设备上上报的流量信息。

目前行业内主流的报文统计协议有netstream、netflow和sFlow等技术。

Netstream技术是由华三公司提出的一种基于网络流信息的统计和发布技术。是一种利用网络设备（主要是交换机和路由器）进行网络流量统计、分析、打包的技术。因此在华为设备、华三设备中广泛使用了netstream技术。

Netflow技术是由思科公司在1996年设计出的，最开始时主要用于加速数据流的交换兵同步计量和统计正在高速转发中的IP数据流。随着技术的不断发展和迭代，netflow主要功能变为了对网络设备（主要是交换机和路由器）中的IP信息流进行测量和统计。思科设备主要使用的是netflow技术。

sFlow技术是InMon、Foundry和HP共同提出的一种网络流量监控技术。它可以提供完成整的数据链路层到传输层、甚至全网络覆盖范围内的流量统计信息。

由于笔者实习阶段是在华三公司，且参与的SDN项目开发都是基于华三设备，因此应用信息的获取与呈现这里使用的为netstream技术。因此下面着重介绍一下netstream技术。

### Netstream技术

Netstream提供报文统计功能，它根据IPv4报文的目的IP地址、目的端口号、源IP地址、源端口号、协议号、ToS（Type of Service，服务类型）和输入接口或输出接口来区分不同的流信息并针对不同的流信息进行专门的数据统计功能。

Netstream技术主要有三部分组成：NDE、NSC和NDA：

NDE（Netstream Data Exporter，网络流量采样）：主要功能是对网络流进行采集与发送采集到的流量信息。提取符合条件的流并进行统计，然后将统计到流信息发送至网络流量采集设备。一般网络设备如交换机、路由器等充当NDE的角色。

NSC（Netstream Collector，网络流量采集）：一般为运行在电脑上一个应用程序，负责收集和存储从NDE发送过来的netstream流信息。存储的netstream流信息可以作为网络流量分析程序的数据来源供其进行流量解析。NSC可以从多个NDE设备上接收数据并对数据进行统一的过滤、聚合和存储，供NDA进行网络流量解析。

NDA（Netstream Data Analyzer，网络流量分析）：即网络流量分析程序，它从NSC存储的流量信息中获取数据，然后对数据进行解析、归并、加工处理、生成报表，为其他业务提供支持。

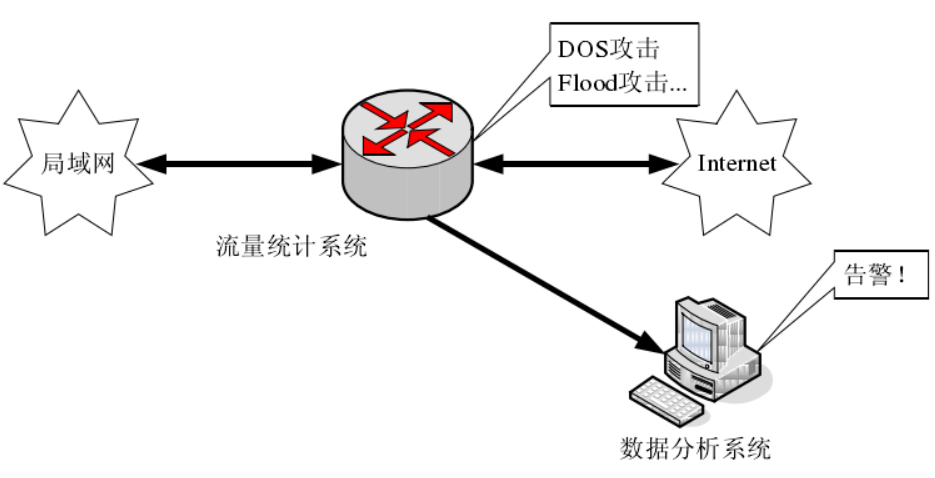


图2-3-1 netstream技术的典型应用

Netstream网络流老化是NSC输出流统计信息的前提。当设备上netstream缓存区达到符合输出条件时，NDE才会将流信息发送至NSC让其进行进一步的数据分析操作。

流老化主要有以下五种方式：

* 按活跃时间老化：当一条流的活跃时间超过所配置的时间限制时，流信息将会被老化。
* 按不活跃时间老化：当一条流的活跃时间超过所配置的时间限制时，流信息将会被老化。
* 按TCP连接FIN和RST报文触发老化：对于TCP连接，当有FIN或者RST报文发送时，说明一次会话的结束。因此，当一条已经存在TCP协议netstream流中流过一条有FIN或者RST标志的报文时，会立即该流老化。
* 按统计字节超过限制时老化：Netstream缓存区中的流需要记录流过的报文字节数，当字节数超过配置的大小时，就会造成流溢出，所以系统在检测到某条流的字节数超过限制时，为了避免发生计数错误，会立即对该流进行老化。
* 强制老化：管理员可以通过手工输入命令的方式对流进行老化处理。该功能主要是对未满足老化条件的流进行老化，比如未到老化时间但又需要最新的老化流来统计信息，或者netstream业务出现故障，需要对缓存区中的流进行全部老化处理。

当设备上的netstream缓冲区中达到流老化的条件时，NDE便会向NSC发送流信息，方便NSC对网络流量进行过滤、存储操作。

其中netstream流输出，分为三种方式：

* 原始流输出：设备上的所有的流信息都会被统计，然后在到达流老化的条件后，netstream缓冲区中的流全部输出到NSC。这样的输出方式可以详细的知道每条流的统计信息，但也会显著的消耗大量的计算资源和网络资源，同时在许多情况下，网络管理员并不一定需要这么全面详细的流统计信息。
* 聚合流输出：在原始流输出的基础上，以特定的方式对流进行分类和聚合。NDE向NSC发送经过聚合后的流统计信息。这种方式可以很大程度上减少网络设备的内存消耗、计算消耗和网络带宽的消耗。

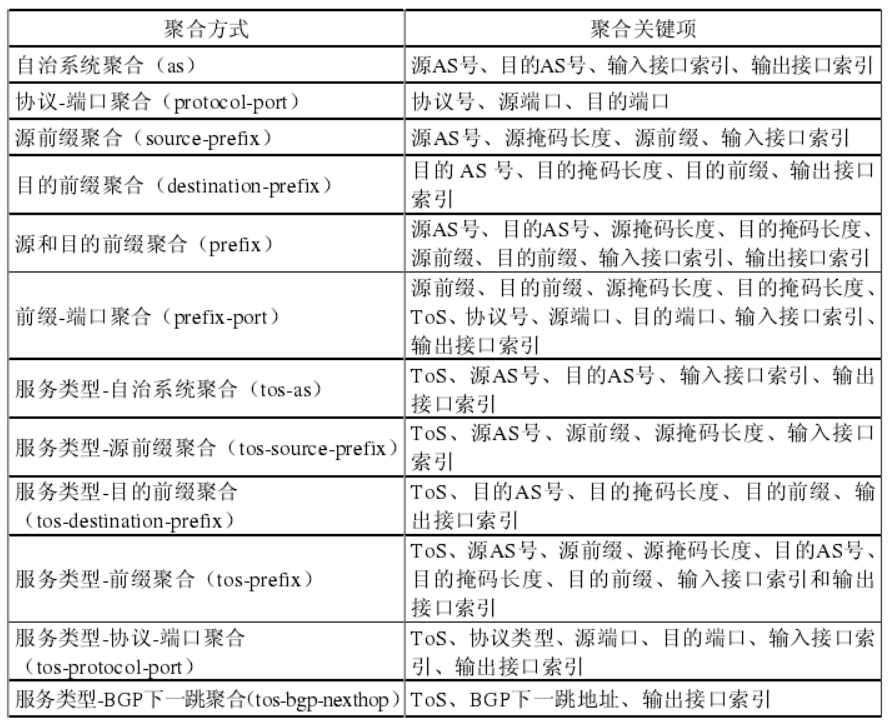


表2-3-1 聚合流输出的12种方式

设备根据选择的聚合方式和聚合关键项，可以将符合聚合条件的多条流聚合为一条聚合流，对应一条聚合记录。这12个聚合方式各自相互独立，互不影响，可以同时配置。

* 灵活流输出：是一种可以自定义形成流条件的流输出方式。这些条件可以是七元组中的任意组合。通过这种方式进行的流统计，可以得到网管人员最想要的流统计信息，并且也会在一定程度上减少网络带宽的消耗。

在NDE向NSC发送流信息报文时，会将流信息以用户指定的报文格式进行封装并以UDP的形式发送至NSC。NSC在接收到网络设备发来的报文后，以对应格式进行解析。下面是netstream的报文信息格式。

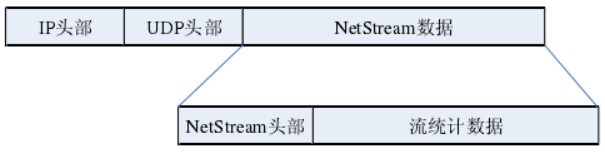


图2-3-2 Netstream报文格式

Netstream输出报文目前主要有V5、V8、V9几种版本：

|  |  |  |
| --- | --- | --- |
| 版本 | 优点 | 缺点 |
| V5 | * 输出的字段比较丰富，可以把聚合前的流记录的所有字段都输出给NSC * NDE设备负荷较小 | * 报文格式固定且不可扩展 * 数据量大，NSC无法长期保存，NSC和NDA分析压力大 |
| V8 | * 按一定规则进行聚合，数据量相对较小 * 承载内容略为简单，适合特定分析 * 可以增加新的聚合方式 | * 报文格式固定且不可扩展 * NDE设备完成聚合工作，负荷较重 * 如果增加新的聚合方式，需要NDE和NSC同时升级版本 |
| V9 | 基于模版   * 允许单独输出需要的域统计信息，减少了输出流的数据量进而减少了NDE、NSC可能的内存以及带宽消耗 * 不需要改变输出报文格式即可在输出记录中增加新的域，因为是以模版的形式输出所以即使NSC无法理解新增的域的真正语意，它仍然可以解释流记录 * 输出方式灵活，既可输出聚合前的流记录，又可输出聚合后的流记录 |  |

## 数据存储

通过netstream技术，控制器可以得到某台设备上的具体的应用流量信息。针对设备上报的netstream报文进行分析。通过分析我们可以获取到实时的流量信息。我们的系统支持历史应用信息的查询，因此还必须将实时的这种流量信息存储起来以便用户以后进行查询。

数据存储，主要是有文件存储和数据库存储两种形式。本文中使用的为数据库存储。

目前业内主流的数据库主要有两类，一类是关系型数据库，另一类是NoSQL型数据库。它们各有特点。

传统的关系型数据库是建立在关系模型基础上的数据库，借助于集合代数等数据概念和方法来处理数据库中的数据。通过SQL语句可以实现复杂查询以及事务支持。常见的MySQL、Oracle、SQLite等都属于关系型数据库。

NoSQL型数据库泛指非关系型数据库。主要分为四大类：键值对存储数据库、列存储数据库、文档型数据库和图形数据库。其数据模型相对于关系型数据库来讲比较简单，性能比较高，易于水平扩展。常见的有Redis、HBase、MongoDB等。

本文中的应用信息模型相对简单，没有复杂的耦合性。最终选择使用的数据库为MongoDB，文档型数据库。MongoDB是一个基于分布式文件存储的数据库，它支持的数据结构是一种类似于json形的bson数据格式。具有高性能、可扩展、易部署、易使用的特点。将netstream上报的报文解析后存储在对应的数据库集合内，以供用户查询历史应用信息的情况。

### MongoDB

MongoDB是一个旨在提供高性能、高可用性和自动扩展的开源文档型数据库（document database）。MongoDB中的每一条记录称之为一个文档（document），这是由键值对（field and value pairs）组成的数据结构。它和JSON数据很相似。Field的value可以是另一个document、可以是一个数组（arrays），可以是一个由document构成的arrays，还可以是一些基本数据类型。一个或多个文档构成一个集合（collection），一个或多个集合构成一个数据库。



图2-4-1 document示意

document具有以下几个优点：

* document（即对象，object）可以对应很多编程语言中的原生数据类型；
* 通过嵌入式的document可以大幅度的减少关系型数据库表连接的消耗；
* Document动态的数据结构可以流畅的的支持数据的多态性。

MongoDB的关键特性：高性能（high performance）、功能强大的查询语言（rich query language）、高可用性（high availability）、水平扩展（horizontal scalability）、支持多种存储引擎（support for multiple storage engines）。下面分别对这几点做介绍。

**高性能**

MongoDB提供高性能的数据持久化操作。尤其是：

* 对嵌入式数据模型的支持减少了数据库系统层上的I/O活动。
* 通过索引，可以支持更快的查询。

**功能强大的查询语言**

MongoDB提供一个功能强大的查询语言使数据聚合查询、文本搜索和地理位置查询像读写操作那样便捷。

**高可用性**

MongoDB具有备份功能，这称为副本集（replica set），它可以提供自动故障转移和数据的冗余备份功能。

副本集是一组维护着一份同样的数据的MongoDB数据库服务器，可以提供数据的冗余备份和提高数据的可用性。

**水平扩展**

MongoDB的水平扩展性是作为它的核心功能提供的：

通过分片（sharding）将数据分配至一组服务器集群上；

MongoDB自3.4版本后，便支持通过创建基于shard key的数据区域（zones of data）

**支持多种存储引擎**

MongoDB支持多种存储引擎，例如WiredTiger存储引擎和MMAPv1存储引擎。另外，MongoDB还提供了一种可插拔的存储引擎接口，允许第三方为MongoDB开发存储引擎。

### WiredTiger存储引擎

WiredTiger存储引擎是MongoDB在3.0版本及后续版本中提供的新一代的64位存储引擎。它具有以下几个特点：

* Doucment Level Concurrency，文档级的并发控制。WiredTiger引擎针对写操作使用了文档级的并发控制。因此在多客户端可以同时更新同一个集合下的不同文档。
* Snapshots and Checkpoints，快照和检查点。WiredTiger使用了多重版本并发控制（MVCC，MultiVersion Concurrency Control）技术。在一个操作开始之前，WiredTiger会提供一个涉及数据的时间点快照。快照呈现了内存数据的一致性。
* Journal，日志系统。WiredTiger通过提前写入事物日志与checkpoints结合来保证数据的持久性。
* Compression，数据压缩。WiredTiger提供集合、索引的压缩功能。此功能可以减小数据的存储占用空间，但会消耗额外的CPU计算资源。
* Memory Use，内存占用。MongoDB可以同时使用WiredTiger引擎的内部缓存和文件系统的缓存。并可指定其内存的使用量。

### MMAPv1存储引擎

MMAPv1存储引擎是MongoDB最开始时使用的一个基于映射文件的存储引擎。其擅长处理大量的插入、读取、更新的操作。

本项目采用的引擎为WiredTiger，MMAPv1不做过多的介绍。

## 应用信息报表的呈现

设备的具体应用信息存储在数据库中，其原始数据是不易于用户直接查看，晦涩难懂。因此需要对其进行专门的处理来提供给用户一个简单、高效的查询接口。同时，通过此接口返回的数据应进行响应的处理，使其在页面上以图文报表的形式显示，方便用户直观的查看信息。

这里主要使用的ADWAN（Application-driven Wide Area Network, H3C新一代基于ODL开发的广域网解决方案产品）的Rest接口的设置，定义输入参数、输出参数形式以及前台web页面的具体显示。

前台页面通过Ajax请求，使用html5绘图eChart框架显示。应给予用户最直观易懂的操作与信息。

## 本章小结

应用信息的获取与呈现，是基于SDN产品ADWAN的基础上进行开发子模块。SDN产品ADWAN是华三公司的一款基于ODL框架的新一代广域网整体解决方案，目的是为了帮助用户构建一个就架构开放、灵活编程、易于运维的广域网来承载日益丰富应用流量，最终实现应用按需驱动、网络动态适应应用的目的。而应用信息的获取与呈现也是为了这个目的而产生的模块。通过ODL框架的规范、Rest接口的定义，YANG模型的定义以及前台Web页面的请求与显示作用以达到最终目的。

# ADWAN应用信息获取与呈现模块系统分析与设计

应用信息的获取与呈现是基于ODL框架的ADWAN项目的一个子模块，其主要功能是向用户提供近一年以内任意时段内的网络流量情况，以供用户进行其他相关的数据分析、处理等操作。下面主要讲述ADWAN项目以及子项目应用信息的获取与呈现的系统分析与设计。

## ADWAN系统设计

H3C凭借长期在广域网领域的技术积累和丰富经验，深入调研用户实际应用中存在的痛点和需求，采用SDN思想及其相关技术开发了新一代广域网整体解决方案，这个方案帮助用户构建了一个架构开放、灵活编程、易于运维的广域网，来承载日益丰富的应用流量，最终实现应用按需驱动网络、网络动态适应应用，即ADWAN（Application-driven Wide Area Network）。

### 方案架构



图3-1-1 ADWAN方案架构

如上图所示，ADWAN方案和其他场景下的SDN网络架构一样，也是一个分层、开放、灵活的网络架构，如下图所示，整个ADWAN方案分为网络设备、控制器+APP、管理编排三个层次：

* 网络设备层：网络设备接收SDN控制器的控制和管理，除了传统的SNMP、NETCONF、命令行等方式外，支持SDN架构下的BGP-LS、BGP Flowspec、PCEP、Openflow等协议，和控制器进行通信；同时在转发层面进行优化，支持Segment Routing、Openflow硬件转发，提供高性能的转发平面。
* 控制器+APP层：整个方案基于开源的ODL平台，支撑各种APP集成；根据广域网不同的场景，比如DCI网络、骨干网、分支接入等，开发定制化、场景化的APP，满足用户在不同场景下的网络需求；南向通过标准南向接口协议和设备互通；北向面向用户提供定制化的API接口，实现和编排系统集成，满足用户差异化的业务需求。
* 管理编排层：通过调用APP提供的API接口，实现业务流组定义（Qos优先级、五元组、VPN等）和业务的策略定义和管理编排，全网的实时监控、可视化呈现、及故障排查等，进而增强网络的可视化，简化网络的运维管理。

在基于ADWAN项目的基础上编码应用信息的获取与呈现具有很大优势与便捷性。可以是开发人员着重于聚焦业务逻辑问题，避免在底层协议、控制设备上面付出过多的精力。应用信息的获取与呈现主要位于“控制器+APP”和“管理编排”层上。

## 应用信息获取与呈现模块结构设计

应用信息的获取与呈现应主要分为基础管理、应用管理、报文解析、数据存储以及呈现层这五个部分。这五部分相互依存，共同构成了基于ADWAN系统下的应用信息的获取与呈现模块。

其中基础管理主要负责设备连接、下发配置。应用管理主要是集成了系统默认初始定义的应用信息和用户自定义的应用信息，供报文解析时匹配应用信息。报文解析主要负责解析设备侧上报的netstream报文。数据存储主要负责连接MongoDB服务并存储应用信息的数据。呈现层主要是从数据存储提供的接口中请求数据然后显示给用户。

## 基础管理

通过调用基础管理模块的接口，向设备下发流采集命令，以达到设备向控制器上报流量信息的目的。

众所周知，在互联网中的有许多通信厂商，同一家公司下的网络设备也有不同的型号，即使同厂商同型号的设备软件版本还有所不一致。不同的厂商、不同的型号、不同的软件版本其设备的流采集配置命令也不尽相同。例如华三设备MSR88和SR88的流采集命令一个为netstream，一个为Qos。因此解决不同型号、不同版本的命令配置是基础管理模块的一个核心需求点。

基础管理的核心需求主要为：

* 连接设备，保活设备。ADWAN是一个SDN产品，所有的才做基本上都是与设备息息相关的。因此基础管理模块主要任务就是和设备建立连接。其中与设备简历连接的方式多种多样，有openFlow发现、BGP-LS发现、SNMP等等协议方式。根据用户配置的不同的设备发现方式，基础管理应能够与合法的设备建立连接并可正常保活设备。
* 对外提供ADWAN系统中所有设备、链路的状态信息。作为一款SDN产品，ADWAN的所有功能模块基本上都是基于设备、链路的状态信息实现的。因此作为基础模块，其在连接设备的同时，也应向其他模块提供接口以暴露设备、链路的状态信息。例如拓扑选路，就需要根据网络拓扑中链路的状态信息来进行合理的选路的。
* 对其他模块提供统一的设备管理接口，屏蔽底层设备、协议的差异。主要是屏蔽不同厂商、不同型号、不同软件版本号、不同协议的差异，提供统一的北向接口，主要是配置下发接口。

基础管理可对设备、链路进行管理。设备、链路的所有状态信息都需要存储在数据库中。下表分别是设备、链路的字段信息。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 范围 | 默认值 |
| deviceId | 设备唯一标识 | UUID | -- |  |
| deviceName | 设备名称 | String | 1-255（长度） | -- |
| manageIp | 设备管理IP | String | -- | -- |
| manageMac | 设备管理口Mac地址 | String | -- | -- |
| deviceType | 设备类型：  1：通用设备  2：虚设备 | Integer | 0-1 | 0 |
| description | 描述 | String | 1-255（长度） | -- |
| deviceStatus | 节点状态：  1：active  2：inactive | Integer | 1-2 | 2 |
| inactiveReason | 节点下线原因 | Integer | 0-65535 |  |
| role | 通用设备角色 | Integer | 1-65535 |  |
| scene | 设备所属场景 | Integer | 1-5 |  |
| serialNum | 设备序列号 | String | 1-31（长度） |  |
| serialNums | 设备序列号列表 | List<String> |  |  |
| sysObjectOid | 设备OID | String |  |  |
| softVersion | 软件版本号 | String |  |  |
| company | 厂商信息：  0：H3C  1：HP  2：Cisco  3：HuaWei  4：Juniper  65535：Unknown | Integer | 0-65535 | 65535 |
| model | 设备型号，如MSR810 | String |  |  |
| regionId | 设备所属区域信息 | UUID |  |  |
| tpIds | 接口索引列表 | List<UUID> |  |  |
| comeFrom | 设备来源 | Integer | 0-4 |  |
| nodeOperStatus | 设备业务状态  1：up  2：down | Integer | 1-2 |  |

表3-3-1 设备字段信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 范围 | 默认值 |
| topoId | 拓扑索引标识 | UUID |  |  |
| linkId | 链路唯一标识 | UUID |  |  |
| linkName | 链路名称 | String | 1-255（长度） |  |
| owner | 链路所属的拓扑类型：  1：物理拓扑  2：业务拓扑 | Integer | 1-2 |  |
| linkType | 链路类型 | Integer | 1-65535 |  |
| srcNodeId | 源节点索引 | UUID |  |  |
| srcTpId | 源接口索引 | UUID |  |  |
| dstNodeId | 目的节点索引 | UUID |  |  |
| dstTpId | 目的接口索引 | UUID |  |  |
| linkStatus | 链路状态：  1：up  2：down | Integer | 1-2 |  |
| metric | 链路metric值 | Long | 0-4294967295 | 0 |
| bandwidth | 链路带宽 | Long | 0-4000000000（kbps） |  |
| reservable-Bandwidth | 可分配带宽 | Long | 0-4000000000（kbps） |  |
| tunnelId | 链路对应的隧道索引 | UUID |  |  |

表3-3-2 链路字段信息

## 应用管理

应用管理向应用信息的获取与呈现的提供基础服务支持。报文解析层解析从设备侧发来的netstream报文时，会依据应用管理模块内的应用信息进行流信息的过滤、聚合等操作。

互联网中有许许多多的应用流量，诸如QQ聊天、优酷视频等等。这里我们定义一条应用信息为一个或若干个流的组合。下表为应用管理字段示意。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 范围 | 默认值 |
| appId | 应用唯一标识 | UUID |  |  |
| appName | 应用名称 | String | 0-255（长度） |  |
| flows | 应用匹配的流 | List<Flow> |  |  |
| comment | 备注 | String |  |  |

表3-4-1 应用字段示意

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 范围 | 默认值 |
| protocol | 协议类型 | String | ip、tcp、udp |  |
| srcAddr | 源IP | String | 网段，或精确的IP。必须具有掩码 |  |
| srcPort | 源端口范围 | String | eq/lt/gt/neq/range + 端口号 |  |
| dstAddr | 目的IP | String | 网段，或精确的IP。必须具有掩码 |  |
| dstPort | 目的端口范围 | String | eq/lt/gt/neq/range + 端口号 |  |
| matchMode | 匹配模式 | String | permit/deny |  |

表3-4-2 flow字段示意

## 报文解析

通过基础管理向设备下发流采集配置，然后控制器可以获取到设备侧传来的应用流量信息。通过对netstream报文的解析，可以获取此刻设备上具体的流量信息。将这些信息通过一定的规则编码组织，即可存入数据库中。

其主要需求功能为：

* 监听设备侧发来的netstream报文；
* 根据netstream报文格式解析报文，对报文进行过滤、聚合和格式化；
* 调用数据存储模块的接口，存储应用信息原始数据。

## 数据存储

数据库层对报文分析层提供接口供报文分析层调用以存储解析的原始数据。本项目可提供用户查询近一年的任意时间段的应用信息的详情。因此使用MongoDB存储时需要考虑到数据的存储方式以达到用户无论请求什么格式的数据都可以迅速的作出响应。这里主要应用到数据分层、分集合技术对数据进行存储。

其需求主要为：

* 拉起并连接MongoDB数据库服务。；
* 对报文解析层提供存储接口；
* 存储报文解析层的数据，使其原始数据以季度为分隔按集合存储；
* 定时对不同数据段的数据取均值，建立数据的分层模型；
* 对外提供符合ODL框架的Rest接口，供用户界面调用拉取数据；
* 自动删除过期数据，防止数据的无限制膨胀。

本项目使用MongoDB存储流量信息，下面介绍一下存储流量信息的主要字段。

## 呈现层

在数据库中存储的是解析后的设备上报到控制器的报文的原始数据，其不易于用户查看和操作。因此，在面向客户的版本中，需要增加适当的用户界面以方便用户操作和查看信息。

Web页面主要有两部分，一部分是数据的请求，一部分是数据的展示。其中数据的请求应根据用户的选择时间段通过Ajax向ADWAN后台请求数据。而数据的展示则是利用了html5新的绘图API绘制出流量曲线走势已经应用表格。

因此本层的主要需求为：

* 转化用户操作，将用户的需求转化为ADWAN后台接口对应的入参；
* 通过Ajax调用ADWAN后台接口，并获取后台响应的数据；
* 通过Highcharts绘图框架，使响应数据转化为用户直观可见的图文报表的形式。

## 本章小结

应用信息的获取与呈现是基于华三公司的SDN产品ADWAN上开发的一个子模块项目。其利用ADWAN的ODL框架与设备进行交互并向外提供Rest接口。前台Web页面通过这些Rest接口向后台ADWAN发送用户请求并且在相应数据返回后对数据进行格式化处理，供用户查看。

这其中主要的层次结构可以理解为和MVC相似，即数据库模型，控制层APP与Web视图层。

# ADWAN应用信息分析的系统实现

经过前面几章对ADWAN的架构分析、应用信息的获取与呈现项目的需求、系统设计的分析，可以逐步明确出项目的具体实现过程。

该项目的具体实现大致可分为信息的获取、信息的存储与信息的呈现，具体可分为如下几个部分：

## 基础管理

控制器需要知道设备上都跑了哪些具体的流量才能实现相应的功能。这些信息的获取则需要对设备进行专门的netstream配置。

应用信息的获取与呈现模块通过调用基础管理管理的下发配置接口向设备下发配置。其中下发设备配置根据设备不同的厂商与型号，配置内容有所区别。这里主要是采用了配置xml文件与设备型号、版本进行匹配的操作来达到下发正确配置的目的的。

ADWAN支持华三公司、HP公司的网络设备。华三公司与HP公司的设备均支持netstream配置。

在设备上配置netstream命令后，设备会通过UDP报文向指定的控制器IP地址发送netstream信息。

已知netstream支持的流输出版本有V5、V8、V9和V10四种版本。在我们的系统实现中采用的是V9版本的流输出格式。V9版本是一种基于模板方式的流输出，具有输出信息灵活的特点，且支持BGP下一跳、MPLS等统计输出。在本项目中，采用的即是netstream V9版本。

而针对同厂商的不同型号设备，基础管理主要采用了XML配置文件、小版本识别技术来进行配置下发。

### XML配置文件

对不同厂商的网络设备，下发不同的命令配置，使用的是XML配置文件。不同的配置命令具有不同的语句参数，通过XML文件，将不同的命令直接配置在XML文件中，这样在下发某个命令的配置时，便可以直接寻找到该命令的XML文件，然后根据文件内容下发配置。



图4-1-1 获取VCPE设备的虚拟机状态信息的命令

在其他模块向设备下发配置时，基础管理会根据已经配置好的XML文件寻找对应的配置命令参数与语句，并对传来的入参对应填到命令中去。然后通过netConf报文下发至设备上。对应返回设备响应的netConf报文。

### 小版本识别：

小版本识别技术简单来讲即是根据设备的软件版本号，

## 报文解析

## 数据存储

## 报表呈现

## 本章小结

# SDN总结与展望

## 全文总结

## 展望

# 致谢

# 参考文献