

一种 ForCES 结构网络设备资源管理模型

贾凤根¹ 郭云飞¹ 丁心毅²

(信息工程大学信息工程学院 郑州 450002)¹ (浙江工商大学信息与电子工程学院 杭州 310018)²

摘 要 在 ForCES 体系结构和 FE 模型的基础上,提出了 ForCES 资源的概念,将 CE 对 FE 的操作抽象成对 ForCES 资源的操作,提出了 ForCES 资源操作管理模型,研究了实现该模型的关键技术,包括 FE 端资源库管理机制、CE 端资源存储机制。论述了该模型的实现方法,并给出了其运用的实例,构建了基于 Web 管理的 IPv6 路由器。最后,对实现的系统的功能和性能进行了测试,并给出了比较分析。

关键词 ForCES, LFB, 资源管理, 资源存储

中图法分类号 TP393-08 **文献标识码** A

Model of Network Device Resource Management Based on ForCES

JIA Feng-gen¹ GUO Yun-fei¹ DING Xin-yi²

(Institute of Information Engineering, Information Engineering University, Zhengzhou 450002, China)¹

(College of Information & Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China)²

Abstract In this paper, the concept of ForCES (Forwarding and Control Element Separation) resources was proposed based on the ForCES architecture and FE model. The operation of CE to the FE was abstracted to the operation of ForCES resources, and a model of ForCES resources management was proposed. The key technologies of the model were realized, including the mechanism of FE resources library management and CE resources storage. The implementation method of the model was discussed, which gives example of its running to build a Web-based management of IPv6 routers. Finally, the functional and performance tests were carried out, and a comparative analysis was put forward.

Keywords ForCES, LFB, Resources manage, Resources storage

1 前言

传统的互联网技术起源于美国国防部,属于非盈利网络,具有较强的自我管理能力和管控性比较弱,而现代化商用的互联网必须具备必要的管理和控制能力。互联网的重要组成部分即网络设备的资源管理的重要性日趋增强,其主要技术瓶颈在于协议及管理对象的标准化模型。ForCES (Forwarding and Control Element Separation, 转发件和控制件分离)^[1-4]工作组是 IETF 路由领域 (Routing Area) 的一个工作组,它专门研究开放可编程^[5]的 IP 路由器的体系结构和协议问题,是当前开放可编程网络研究最受关注的研究组织之一。ForCES 的基本思想是把 IP 路由器分成转发件 (Forwarding Elements, FE) 和控制件 (Control Elements, CE), 认为 IP 路由器可由多个 (可达几百个) FE、CE 和连接它们的 ForCES 协议标准接口构成, CE 通过该协议接口控制和管理 FE 的资源。按照 ForCES 的基本思想,一个 ForCES 路由器就是一个 NE。它由一个主要的 CE (如图 1 中的 CE1) 和多个冗余的 CE (如图中的 CE2), 以及可多达几百个的 FE 组成, 而 CE 与 FE 之间的通信就是通过 ForCES 协议消息的形式来实现的。

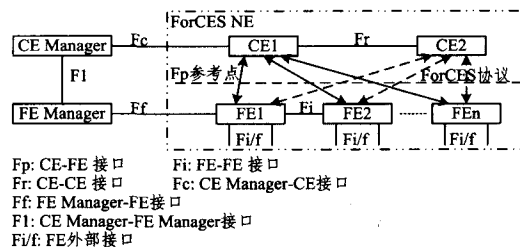


图 1 ForCES 体系结构

ForCES 网关 (ForCEG)^[7] 是 FlexiNET IST 计划研究项目的部分内容,它是控制与转发分离协议的设备,通过 ForCES 网关,可完成对现有软硬件的集成管理。可配置软件边缘路由器的架构^[8]也有了新的发展,它融合了 Click 软件路由器的框架和转发控制分离的思想。

在 ForCES 路由器的 FE 端,存在各种不同的硬件组件,这些硬件组件所实现的功能各不同,具有不同的操作接口及不同的属性特征,这些硬件根据不同的功能被抽象成相应的 LFB。ForCES 软件系统应该只为硬件组件提供统一的操作接口,因此当 CE 模块要对各被控模块硬件组件进行操作时,可不必关心具体硬件组件的实现细节。

到稿日期:2009-10-30 返修日期:2010-01-07 本文受国家自然科学基金 (No. 60603072), 863 计划 (No. 2007AA01Z201), 浙江省自然科学基金 (No. Z106829, Y106747, Y1080078), 浙江科学技术项目 (No. 2008C21093) 资助。

贾凤根 (1979-), 男, 博士生, 主要研究方向为高性能网络体系结构, E-mail: jfg6688@hotmail.com; 郭云飞 (1962-), 男, 博士生导师, 主要研究方向为宽带信息网络; 丁心毅 (1983-), 硕士生, 主要研究方向为网络协议。

同样地,CE 端在支持开发新业务组件的同时,应能实现对各种现有业务软件的复用。如何对不同的业务的操作进行抽象,是本文研究的主要内容。当不同的服务功能软件具有不同的处理及操作接口方式时,如何将这种具体的业务操作转换为 ForCES 操作的内容及格式,本文提出的 ForCES 资源操作管理模型很好地解决了这个问题,并提高了软件的复用程度。

2 ForCES 架构网络设备分层模型

图 2 给出了 ForCES 架构网络设备分层模型,各层功能规定如图 2 所示。

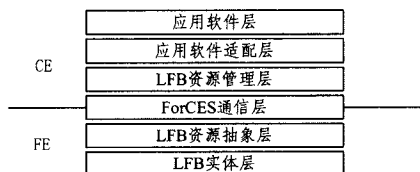


图 2 ForCES 网络设备分层模型

应用软件层:包含运行于各种网络设备控制平面上的各种设备应用,比如路由协议软件、网络管理软件、防火墙管理以及 QoS 管理等应用;

应用软件适配层:其主要功能是将应用软件层的管理命令转化为标准的对 LFB 资源的调用。比如针对路由表下发应用,就需要把路由表的操作转化为对转发 LFB 的属性操作;

LFB 资源管理层:完成对网络设备中众多 LFB 资源的统一管理,并对应用软件适配层提供标准 LFB 资源操作接口;

ForCES 通信层:完成 ForCES 协议的众多功能,提供网络设备内控制平面和转发平面的标准化通信功能;

LFB 资源抽象层:完成对数据平面上各种数据处理功能的标准化抽象,为控制平面的标准化控制提供支撑;

LFB 实体层:实现数据平面上数据处理功能的模块化。

本文提出了 ForCES 资源操作管理模型。该模型主要考虑接口复用和模块复用,模型的具体需求为:该模型能屏蔽应用软件开发商及 LFB 组件开发商对 ForCES 协议内部规范的认识;能与 LFB 的接口标准化(或规范化),LFB 组件开发商能很方便地添加或删除一个具体的 LFB;在 CE 中,LFB 资源管理层与应用操作层的接口标准化,该接口不应体现具体的某个 LFB;能适应于不同的操作系统平台,如常用的 Windows, Linux 及 VxWorks,还应能支持其它专用操作系统;能根据 LFB 类定义描述,自动生成代码,并允许用户添加相关的操作函数。

3 模型框架

本文提出的 ForCES 资源操作管理模型可分成 CE 端和 FE 端两部分,两端都主要由开发工具和 ForCES 资源操作管理模块组成。

3.1 CE 端模型

CE 端模型主要考虑的是如何对服务组件进行复用,以及如何提高 CE 端软件的开发效率。图 3 给出 CE 端 ForCES 资源操作管理模型。以下对 CE 端模型的各个部分进行介绍。

服务组件管理器:我们按照不同的服务功能划分组件,目

前主流的路由器向用户提供的服务主要有:路由服务、QoS 服务和管理服务等,相应地,就有路由组件、QoS 组件和管理组件实现它们的功能;而该模块的功能就是为这些组件提供服务,提供统一的接口函数,协调组件的运行。

控制操作处理器:大部分服务组件要完成特定的服务功能,需要向 FE 进行控制操作,配置或者查询 FE 上的资源;该模块的主要功能就是对这些控制操作进行管理,包括对操作的数据格式进行验证,缓存实例数据,解析操作的路径等。

LFB 类定义库:保存 LFB 类定义信息,为数据格式验证提供类型信息,为管理服务提供描述信息。

LFB 实例库:缓存从 FE 获得的实例数据,从而减少直接从 FE 查询的次数,减轻网络的负担。

资源存储管理器:负责管理 LFB 类定义库和 LFB 实例库,并保证实例库中存储的数据是符合类定义要求的。

LFB 类定义解析器:将以 XML 文件描述的 LFB 类定义信息,转化为以模型内部格式存储的信息;以 XML 文件的形式描述 LFB 类定义,方便用户了解 LFB 结构,而使用内部格式存储方便了机器识别。

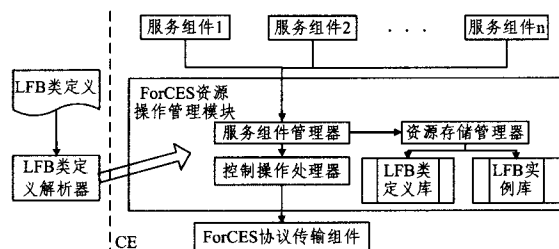


图 3 CE 端 ForCES 资源操作管理模型示意

另外,由于模型提供了统一的操作接口,因此用户可以充分利用现有的系统来构建相应的服务组件。

为了实现 ForCES 数据存储,我们将 ForCES 存储模块分为如下子模块:ForCES 树管理模块、类库管理模块、实例缓存模块和树操作抽象,如图 4 所示。ForCES 树管理将类定义树和实例数据树抽象成一棵 ForCES 树,该树既有 LFB 类定义信息也有实例数据值。该模块提供 ForCES 树的访问接口。类库管理负责 LFB 类定义数据的存储,主要功能是产生和管理 LFB 类定义树,并提供类定义数据访问接口。实例缓存管理负责实例数据的存储,主要功能是产生和管理 LFB 实例树,当新的 FE 建链的时候,将查询到的数据保存到 LFB 实例树中,并提供实例数据访问接口。树操作抽象层为上层提供通用的树操作接口,屏蔽下层的具体存储方式。

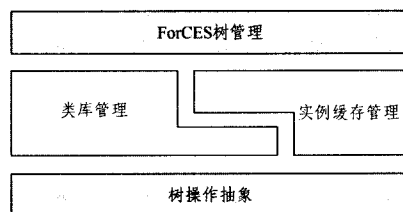


图 4 CE 端资源存储模块划分示意图

3.2 FE 端模型

FE 端模型主要考虑的是如何对 LFB 组件进行复用,并提供了代码生成器来方便用户开发新的 LFB 组件。图 5 给出 FE 端 ForCES 资源操作管理模型。以下是 FE 端模型的各个组成部分。

LFB 组件管理器:主要负责管理各个 LFB 组件。实现 FE 功能的 LFB,即 ForCES 资源,在本模型表现为各个 LFB 组件。本模块提供注册机制,允许动态地添加新的 LFB 组件。对于 LFB 组件的管理,体现以下几个方面:(1)LFB 组件注册认证:对 LFB 组件合法性进行验证,即该组件是否实现了规定的接口,否则不能添加;对 LFB 组件的版本进行验证,不同的厂商可以提供不同版本 LFB,不允许同一个 LFB 组件多次添加;对 LFB 进行依赖性验证,只有添加的 LFB 组件所依赖的 LFB 组件在软件系统中存在,才能完成添加工作。(2)LFB 组件协调:协调多个组件一起工作,包括完成 LFB 类的实例化,以及连接 LFB 数据通路,提供模块间的消息传递通道。

LFB 资源库:LFB 资源库中保存了各个 LFB 组件的资源,包括 LFB 的能力、属性以及可以订阅的事件,还有相应的处理函数。每个 LFB 组件都要向资源库中注册相应的处理函数;当 CE 端对相应的资源进行操作时,LFB 组件管理器从资源库中查找相应的处理函数完成操作。

控制操作处理器:该模块的主要功能是对 CE 端下发控制操作进行管理,查找对应的 LFB 完成操作,并对 CE 端进行回应。

LFB 组件代码生成器:该部件的主要功能是,按照开发商提供的 LFB 类定义文件和用户函数文件,自动生成 LFB 组件代码。其中,LFB 类定义文件是对该 LFB 属性进行描述的文件,在 FE 模型中使用 XML 语言进行表达;用户代码文件提供了与具体模块相关的函数的实际执行代码,这些函数包括模块接口函数和操作处理函数。

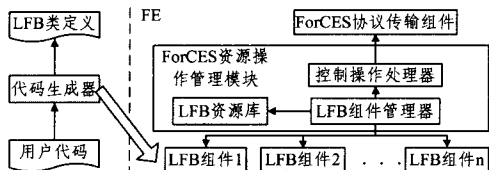


图5 FE 端 ForCES 资源操作管理模型示意

如果用户要对 ForCES 路由器的功能进行扩展,只须向 FE 端添加相应的 LFB 组件即可。添加的 LFB 组件的代码需要与路由器的整个代码一起编译后再执行。对于 CE 端对新添加的 LFB 组件的识别,主要是通过 FE 端和 CE 端共享 LFB 类定义文件的方式实现,这在现有的 ForCES 草案中有所描述,本文不对其进行进一步的讨论。另外,对于组件运行时的动态添加,不同的操作系统的实现方式不同,如 Linux 系统下可采用动态链接库技术;对于涉及到硬件功能的扩展,还需要硬件设备的支持;因此,本文中的模型不涉及运行时添加 LFB 组件。

在模型的 FE 端,LFB 资源库是模型中的重要模块,它将所有的 LFB 组件提供的资源进行抽象,为提高软件的复用度提供了基础。因此,如何有效地组织资源库,并对其进行高效的查找,就显得尤为重要。

由上面的分析可知,ForCES 资源呈树状分布。根据这一特点,可采用树状结构进行组织。图 6 表示的是资源库中的资源树,图中的方框表示树的节点,方框中的文字表示节点的名称和 ID 号。树上的节点分 3 种,即类节点、实例节点以及资源节点。树的顶层是 LFB 类节点,所有 FE 支持的类构成一个队列。树的第二层是类的实例节点。当类实例化时,系

统向该类节点下添加新的实例节点。树的第三层以及以下的几层是 LFB 资源节点。树的每个资源节点保存资源操作的指针,这些操作函数包括设置 (SET)、获取 (GET) 和删除 (DELETE) 操作。

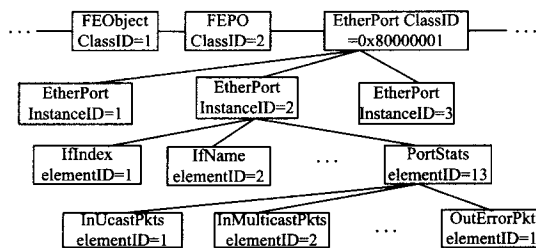


图6 资源树示意

为了提高资源操作函数的查找速度,这里使用了辅助散列表。这里使用的技术被称为“桶散列”(bucket hashing)。如图 7 所示,散列表本身由一个指针数组组成,其中每个指针给出一个链表的地址,链表中的节点代表可进行散列寻址的 LFB 资源。要找出某个 LFB 资源的信息,系统可利用该资源路径的数字表示形式计算散列值,从散列表中选出一个链表,然后搜索该链表。链表中的每个表项不仅包含一个指向它所代表的 LFB 资源节点的指针,还包括一个指向链表中下一个表项的指针。

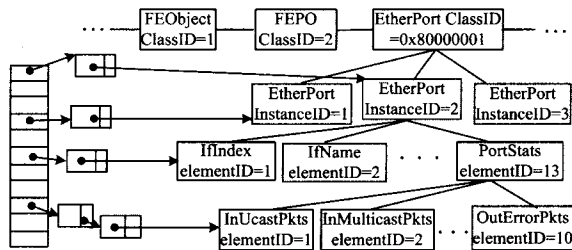


图7 Hash 查找优化后的资源树示意

在实际中,采用一种简单的散列技术就可以得到很好的效果。另外,只有添加了资源处理函数的资源节点,它才会被添加到散列表中;这样大大减少了散列表中的元素。对于目前定义的 LFB 资源来讲,散列化的结果是链表的平均长度比 1 稍大一些。因此,大多数查找能立即找到所需的值,并不需要系统搜索资源树。

4 测试

本文采用网络处理器实现 ForCES IP 路由器的 FE;在普通 PC 机上实现 CE;采用 Zebra 作为路由器的路由协议软件。

ForCES 路由器实现的硬件模型如图 8 所示,整个路由器在物理上由一个 CE 和多个 FE 通过以太网交换机(或以太网交换板)连接组成。每个 FE 对应一块网络处理器开发板。这里采用了两块 Intel IXDP 2401 开发板和一块 Intel IX-DP2851 开发板,分别作为 3 个 FE,开发板 Xscale 层上采用 Monta Vista Linux(Monta Vista Linux 3.1)操作系统。CE 的硬件平台为通用处理器,其软件部分目前是在 Linux 系统下实现的。

我们重点对资源操作管理模块进行了性能测试,尤其是属性的查询和配置功能。本次测试分为单组操作和多组操作,单组操作是将多个操作调用操作函数一次完成;多组操作是指,多个操作通过多次调用操作函数完成。对于不同的操

作方式,又以阻塞模式和非阻塞模式进行比较。

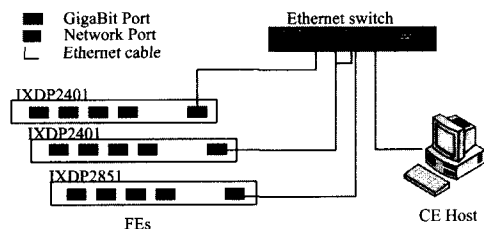


图8 ForCES路由器实现的硬件模型

单组操作测试是取多个路径,调用 LFBAttriQuery 接口一次完成查询操作,调用 LFBAttriConfig 接口一次完成配置操作,并分别使用阻塞模式和非阻塞模式进行操作;记录操作所耗的时间,观察所得结果,分析操作路径个数和所耗时间之间的关系,比较不同模式之间的差别。选用 FE ID 为 1 的 Object LFB 下的 FE Name 属性为操作对象,这样被操作的资源的路径为“1.1.1.3”。使用对同一个路径的多次操作来模拟对多个路径的操作。下面以查询操作为例,记录数据如表 1 所列。其中,平均耗时是对 5 组数据平均取值所得到的结果。

表 1 单组查询操作记录

序号	1	2	3	4	5	6	7	8	9	10
路径数(个数)	100	200	300	400	500	600	700	800	900	1000
平均 阻塞模式	533	804	849	1005	1097	1228	1452	1524	1533	1620
耗时 非阻塞模式	854	1019	1170	1180	1282	1327	1506	1635	1660	1880
(微秒)										

多组操作测试是指,多次调用 LFBAttriQuery 接口或者 LFBAttriConfig 接口完成操作;记录操作所耗的时间,观察所得结果,分析操作路径个数和所耗时间之间的关系,比较不同模式之间的差别,以及与单组操作间的差别。同样地,使用资源路径“1.1.1.3”,来模拟对多个路径的操作。下面以查询操作为例,并设置每次操作的路径条数为 500,记录数据如表 2 所列。其中,平均耗时是对 5 组数据平均取值所得到的结果。

表 2 多组查询操作记录

序号	1	2	3	4	5
操作次数(次)	1	2	4	8	16
平均耗时 阻塞模式	988	1579	2621	5200	10821
(微秒) 非阻塞模式	930	1547	1950	2565	3588

首先,分析阻塞模式下单组操作耗时情况。从图 9 中可以得到如下结论:操作的耗时和路径的个数成正比,在测试的路径条数范围内不会出现急剧的变化;当路径操作条数到达 1000 时,耗时控制在 2000 毫秒内,满足基本的控制操作对时间消耗的要求。

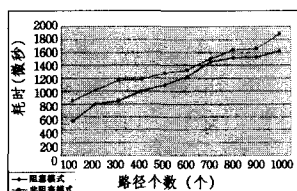


图9 单组查询操作耗时比较

然后,分析不同的操作模式对性能的影响。从图 9 可以看出,对于单组操作而言,非阻塞模式的耗时要大于阻塞模式,这是由于非阻塞模式下要进行回调函数处理等额外操作。从图 10 可以看出,阻塞模式的耗时大于非阻塞模式的耗时,这是因为非阻塞模式下并行处理优化了性能。最后得到如下结论:在单组操作的前提下,非阻塞模式降低了系统的效率;在多组操作的前提下,非阻塞模式极大地提高了系统的效率。

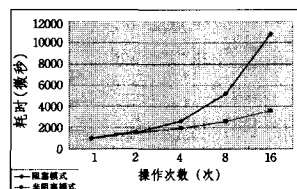


图10 多组查询操作耗时比较

结束语 ForCES 架构路由器采用的是一种全新的体系结构。本文提出的 ForCES 资源操作管理系统还有很多需要完善的地方,今后进一步工作的目标和方向为:1)补充 CE 端资源操作接口。目前的设计和实现的资源操作接口只是资源操作的子集,是最常用的部分,考虑到系统的易用性,还要添加新的接口。2)FE 端资源注册接口提供的回调函数还需要丰富。目前只提供了基本的操作函数的操作,但对其他的操作如:LFB 类实例化和 LFB 拓扑关系操作,都没有涉及,仍需要完善。

参考文献

- [1] Forwarding and Control Element Separation(forces)[DB/OL]. <http://www.ietf.org/html.charters/forces-charter.html>
- [2] Forwarding and Control Element Separation(ForCES) Working Group Supplemental[DB/OL]. <http://www.sstanamera.com/~forces/>
- [3] Doria A, et al. ForCES Protocol Specification[DB/OL]. Internet-Draft, March 2009. <http://www.ietf.org/id/draft-ietf-forces-protocol-22.txt>
- [4] Halpern J, Deiganes E. ForCES Forwarding Element Model [DB/OL]. IETF Draft, October 2007. <http://www.ietf.org/internet-drafts/draft-ietf-forces-model-08.txt>
- [5] IETF ForCES Working Group[DB/OL]. <http://www.ietf.org/html.charters/forces-charter.html>
- [6] Biswas W J, et al. The IEEE P1520 Standards Initiative for Programmable Network Interfaces[J]. IEEE Communications Magazine, IEEE, 1998, 36: 64-72
- [7] Louati W, Jouaber B, Zeghlache D. Configurable software-based edge router architecture[J]. Computer Communications, 2005, 28(14)
- [8] Haleplidis E, Haas R, Denazis S, et al. A Web Service and ForCES-based Programmable Router Architecture[C]//International Working Conference on Active and Programmable Networks (IWAN). Sophia Antipolis, France, November 2005