基于 OpenRouter 模型的 ForCES 协议实现技术研究

关 中
(广州城市职业学院 广州 510230)

摘 要 本文在对基于 OpenRouter 模型的 ForCES 协议进行研究的基础上,重点介绍了基于 OpenRouter 模型的原型系统实现的相关技术,并对 ForCES 协议实现的关键技术进行了详细分析。

关键词 OpenRouter 模型,ForCES 协议,网络单元,平稳重启,高可用性

Research on Implementation Technology of the ForCES Protocol Based on OpenRouter Model

GUAN Zhong
(Guangzhou City Polytechnic, Guangzhou 510230)

Abstract Firstly, this paper makes a study of the ForCES protocol based on OpenRouter model. Then we present the related techniques of the implementation of a prototype system based on OpenRouter model. Finally, we analyse key techniques of the implementation of ForCES protocol in detail.

Keywords OpenRouter model, ForCES protocol, Network element, Graceful restart, High availability

1 引言

在短短的数十年时间里,互联网从一个简单的试验性网络演变成一个巨大的商业网络,并成为全球最大的数据通信网络,从根本上影响了整个人类社会的生活方式。特别是近年来,互联网规模以接近每年翻一倍的高速率持续增长。随着网络技术的日益发展和信息技术的突飞猛进,互联网呈现出许多新的特征,并对网络中的路由器提出了诸多挑战。

新的网络应用需求和应用模式的转变,要求路由器提供更强的协议处理能力的同时,还要能够快速方便地创建、部署和管理新的服务,以满足日新月异的用户需求。但目前的路由器是基于一体化的、不灵活的和不通用的传统体系结构,因此在开发新的技术和协议时,带来标准化过程复杂、部署周期长、维护困难等问题。

作为信息保存和传播的主要载体,人们不断向互联网络提出新的需求,希望能够把网络扩展到信息社会的方方面面,希望在网络上实现更多的功能,提供更好的服务,保证更安全的信息交互,互联网络不断向着"更大、更快、更及时、更方便、更安全、更可管理和更有效"的方向发展。但传统的路由器体系结构在硬件的可伸缩性、软件的可扩展性、系统的开放性和可用性等方面都很难满足新一代互联网的发展需要。因此,新一代网络单元体系结构模型成为目前互联网的研究热点之一。

针对当前路由器结构不能满足灵活多变的网络控制需求,我们提出了一个多维开放可扩展的路由器体系结构——OpenRouter[1],OpenRouter 具有开放、高度灵活和高可扩展的特点,支持控制与转发分离、控制和转发平面的开放可编程、多机集群结构和用户控制选择等特性。使得路由器可以从规模、安全、性能、功能和服务等多个需求角度达到开放可扩展的目的。

ForCES (Forwarding and Control Element Separation)协议[2]是转发与控制分离协议,定义了转发和控制之间通信的体系结构及协议需求,用于基于 OpenRouter 模型的原型系统中。控制单元(CE)通过 ForCES 协议来对转发单元(FE)进行控制和管理操作。在一个 ForCES 网络单元中可以包含多个 CE 和多个 FE 的实例,每个 CE 可以控制多个 FE,同时每个 FE 也可以由多个 CE 控制。ForCES 的目标是定义一个框架和相关的协议,用来规范在控制平面和转发平面之间的信息交换。

本文首先介绍了控制分离模型、OpenRouter 模型以及 ForCES 协议方面的相关研究内容,然后重点论述了基于 OpenRouter 模型的原型系统实现的相关技术,最后对 ForCES 协议实现的关键技术进行深入的研究和分析。

2 相关工作

2.1 控制分离模型

目前国内外对控制分离的思想引起了广泛的关注,提出了许多典型的分离控制模型。IETF 组织提出的通用交换管理协议(GSMP)[3]、网络处理机论坛提出的 NPF Software Model[4]、IEEE 基于电信模型提出的 P1520 分层参考模型[5]以及 IETF 的 ForCES 工作组提出的转发与控制单元分离的思想都对分离控制领域的研究有着巨大的推动和促进作用。

2.2 OpenRouter 模型

OpenRouter 模型是一个开放可扩展的通用路由器模型,其体系结构模型图如图 1 所示,从图中可以看出,OpenRouter 模型中在垂直方向存在三个访问接口:U-API、P-API 和 C-API。

用户接口 U-API:是一组用于应用服务与网络控制进行交互的协议或协议扩展,它是应用感知网络的接口,方便新型网络应用的部署;

编程接口 P-API:是进行控制服务开发的软件编程接口,其作用是为模块化的路由器开发提供统一的模块间调用接口;

控制接口 C-API:是一个控制转发分离的交互协议接口,

关 中 讲师,研究方向:网络系统、数据库系统、嵌入式系统、软件工程等。

它使不同的控制单元和转发单元间可以进行互操作级访问，类似于 IETF 的 ForCES 协议。

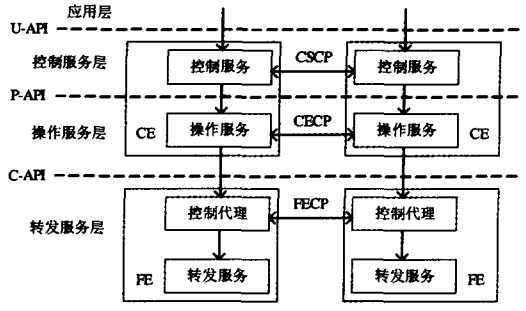

图 1　OpenRouter 体系结构模型图

同时，OpenRouter 模型中的水平方向还存在三个协同协议：控制服务协同协议（CSCP，Control Service Cooperation Protocol）、控制单元协同协议（CECP，Control Element Cooperation Protocol）和转发单元协同协议（FECP，Forward Element Cooperation Protocol）。这些协同协议向上层提供统一的虚拟视图和服务。

2.3　ForCES 协议

ForCES 协议是 IETF 正在开发的控制单元和转发单元的接口协议。IETF 组织成立了 ForCES 工作小组专门对 ForCES 协议进行研究。目前已经提出了关于 ForCES 协议整体框架的 RFC3746[6]，定义了 ForCES 网络单元的体系结构框架，规定了相关联的实体标识和关联关系。RFC3654[7] 提出了 IP 转发和控制分离的需求。同时 ForCES 工作组还提出了大量的草案，具体地规定了协议的报文格式、状态转换和安全性考虑等问题[8~10]。

图 2 显示了 ForCES 协议的体系结构框架以及各逻辑部件之间的相互关系。

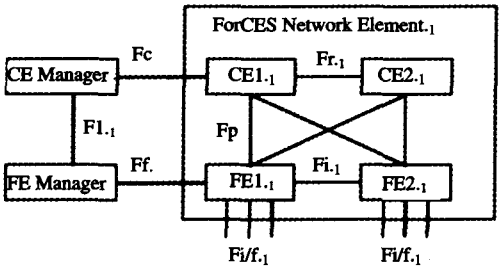

图 2　ForCES 体系结构框架

一个 ForCES 的网络单元（NE）可以包含多个 CE 和多个 FE 的实例，每个 CE 可以控制多个 FE，同时每个 FE 也可以由多个 CE 控制。每个 FE 包含一个或多个物理介质接口 Fi/f，该接口用来接收从该网络单元外部来的报文或者将报文传输到其他的网络单元，这些 FE 接口的集合就是 NE 的外部接口。除了这些外部接口，NE 内部还有一些内部互连用以 CE 和 FE 之间通信以及 FE 之间转发报文的接口。在网络单元外部还有两个辅助的实体：CE 管理者和 FE 管理者，它们用来在前关联阶段对相应的 CE 和 FE 进行配置。为了方便起见，在 ForCES 体系结构中的组成成分之间使用参考点来标记其逻辑交互，参考点分别为：Fc、Ff、Fr、Fi、Fl、Fi/f 和 Fp。完成控制单元与转发单元之间交互的 ForCES 协议是在 Fp 参考点上定义的。

目前国际和国内对 ForCES 整体协议的研究还处于探索阶段，协议的很多细节问题和实现问题还有待于考虑。ForCES 未来将会对 CE 和 FE 间的接口提出公认的标准协议和规范，并陆续对 ForCES 体系结构中其他参考点的接口进行标准化定义和实现。

3　基于 OpenRouter 模型的原型系统实现相关技术

3.1　无中断转发技术

Internet 是由分布在各地的若干网络互联构成，这些网络之间和网络内部运行着 IP 路由协议。对于一个无中断的设备平台来说，要想有效地实现无中断转发功能，路由协议必须能够支持平稳重启（Graceful Restart）[11] 机制，实现无中断路由（hitless routing）。

在网络单元中，CE 可能因为产生某种错误或其它原因而需要重新启动，这将有可能导致整个网络控制平面的中断，给网络正常的路由和转发带来了延迟。若网络规模是 Internet，那么基于路由聚合所需时间的长短，就有可能产生瞬时黑洞和转发回路。如果网络中同时存在大量的路由更新报文或同时有多个路由器重启，那么网络中所有的路由器都将受到影响。

在网络中引入平稳重启技术可以将上述缺陷造成的负面影响最小化。虽然平稳重启技术的实现随着协议的不同而不同，但其基本思想是一样的。使用平稳重启技术后，重启的路由器在其重新启动的时候向其邻居发出通告，邻居检测到邻接丢失信息后既不重新计算新的路由，也不向自己的邻居发送路由更新报文。这些重启路由器的邻居在该路由器重启完成之前的一段有限的时间内继续保持从重启路由器中收到的路由信息。通过这种方式，重启路由器的转发平面可以在其重启过程中按照已有的路由表继续接收和转发报文。当重启路由器的重新启动完成后，重新与邻居建立路由邻接，从所有邻居处下载更新路由表，重新计算路由，并使用新的路由表更新替换原来的转发表。在重启路由器向其邻居发送已更新的路由信息后，平稳重启过程完成。

无中断转发是平稳重启的前提条件，无中断转发保证了重启后的路由器在接收新的路由信息和重新计算路由的同时持续转发报文。平稳重启机制与无中断转发的结合，使得在无中断转发所提供的转发平台稳定性的基础上进一步地增强了整个网络级控制平面的稳定性。

3.2　路由一致性机制

在实现基于 OpenRouter 模型的原型系统时，控制平面和转发平面是物理分离的，所以可能产生路由不一致的问题。在 OpenRouter 模型的体系结构框架下，有两种情况会导致控制平面和转发平面的路由信息表出现不一致：第一种情况是路由协议软件甚至整个控制软件发生故障，在平稳重启的过程前后，为保持转发平面的转发功能正常运行，会产生路由不一致；第二种情况是当控制平面与转发平面间的控制链路异常中断，在控制链路恢复前后存在路由不一致。

一致化过程是将待删除路由项集合中的路由项从 FIB 表中进行删除，将待更新路由项集合中的路由项进行更新，将待添加路由项集合中的路由项加入到 FIB 表中的过程。路由一致化过程是通过控制平面和转发平面之间的协同操作完成的。在实现路由一致化的过程中，可以对 RIB 表和 FIB 表分别定义一些操作原语，包括：添加表项、删除表项、查询表项以及遍历表等操作原语。我们把转发平面分为两类，第一类为可遍历的转发平面，提供上面四个路由项集合操作原语；第二类为不可遍历的转发平面，提供前三个路由项集合操作原语或者只提供添加和删除两个路由项集合操作原语。路由一致化过程是指对 RIB 使用查询和遍历两个原语，对 FIB 表使用上述的 4 个原语进行操作，达到 FIB 表和 RIB 表的一致化过

程。因为一致化过程中对 RIB 表并未进行添加和删除操作，因此 RIB 在一致化过程中保持不变。对于 FIB 表的操作，我们可以定义添加和删除原语操作之间的优先关系，这样能够有效地减少一致化过程中报文的错误投递行为或黑洞行为。

3.3 安全性机制

在实现原型系统中，安全性技术已经占据了十分重要的位置，并且随着网络规模的不断扩大，各种新技术的不断涌现，协议安全的重要性也会与日俱增。随着 Internet 的不断发展，它的脆弱性和安全问题已经逐渐的地暴露出来，网络的非授权访问、信息泄露、资源耗尽、资源被盗或破坏、网络病毒木马等行为都将威胁着网络的安全性。所以，在实现原型系统的过程中，应该针对系统可能出现的攻击类型进行有效地防范。系统可能遭受的攻击有："Join"或"Remove"报文泛洪攻击、角色欺骗、重播攻击、失败恢复攻击、数据完整性、数据私密性、安全性参数共享威胁、拒绝服务攻击等。

目前在网络安全研究领域，基本的方法和技术已经基本成熟。需要研究的是如何针对 ForCES 协议的特殊应用把已有的安全方法用于 ForCES 协议结构框架中，使整个协议系统更加安全。ForCES 协议的体系结构框架能够识别多种不同层次的安全性机制。ForCES 协议层在执行时可以使用传输映射层提供的安全服务，如：终端认证服务、消息认证服务以及加密服务等。只有采用适当有效的安全防护措施，才能有效地抵御恶意的攻击行为，才能真正保证 ForCES 协议安全、稳定、可靠地运行。

4 ForCES 协议实现的关键技术

4.1 控制、数据通道分离技术

ForCES 网络单元很容易受到外部的恶意攻击，如：拒绝服务(DoS)攻击，网络中的恶意系统可能向网络单元内部泛洪伪造的数据报文，使得网络单元内部 CE 与 FE 之间的通信遭到破坏。所以在 ForCES 协议实现的过程中，为了保护网络单元免受外部的攻击，采用控制通路和数据通路分离的方式来对这两种报文分别进行传输。

控制通路由于传输 ForCES 协议内部的控制报文，所以必须是安全可信的通道，这可以通过在关联建立时提供安全认证机制来保证。数据通道虽然不能保证完全的可靠性，但必须提供拥塞检测和控制功能，并且提供区分报文优先级以及限制传输速率等机制。这样，当有恶意的报文试图攻击网络单元时，只能进入数据通道，并赋予最低优先级权限，当检测到传输通道拥塞时会立即限定报文的传输速率，从而很好地防止和避免了外部攻击，提高了协议的安全性和可靠性。

4.2 协议体的动态机制

网络单元中包含有多个 CE 和多个 FE，这些 CE 和 FE 可以采用静态的方式或者动态的方式加入和退出网络单元，所以在协议实现时，必须支持相应的协议体动态机制。

在前关联阶段，可以预先为 CE 协议体静态配置与其相关联的 FE，同时也可以为 FE 静态配置控制它的多个 CE。当协议运行到后关联阶段，就需要通过相应静态配置来进行关联建立和交互通信。但与 CE 协议体相连的 FE 不仅限于静态配置的这些 FE，其他的 FE 同样可以在协议运行时动态地加入网络单元中，与 CE 动态地建立关联。这时，仅仅通过在前关联阶段使用静态配置文件来完成协议体的配置工作是不够的，必须在协议体实现时引入其他机制来支持动态关联。

ForCES 协议在处于关联建立状态时，出于动态机制和高可用性机制的考虑，不应该规定 CE 和 FE 的启动顺序，即：无论一个网络单元中的 CE 和 FE 谁先启动，协议都应该能够正确地运行。正常情况下 CE 先启动，然后作为服务器方侦听端口。此后当多个 FE 启动时，通过主动向该 CE 的侦听端口发出连接请求来建立关联。但如果 FE 先于 CE 启动时，就需要提供一种动态机制来保证协议的正确性和可用性。此时可以通过静态配置文件来予以解决。当 CE 端启动时，无论是否有其他的 FE 已经启动，都应该先根据静态配置文件来向相应的 FE 发出连接，若相应的 FE 已经启动，则建立连接，完成与其的关联建立过程；若相应的 FE 尚未启动，则不做任何处理。此后，CE 再作为服务器端进行端口侦听，等待其他 FE 启动后与其进行连接。这样，无论 CE 与 FE 的启动顺序如何，均可以在 CE 与 FE 之间正确地建立关联。

4.3 协议高可用性机制

ForCES 协议实体 CE 和 FE 在物理上是分离的，通过不可靠的 IP 网络连接在一起，各种意外的错误都将会导致连接的中断和错误的产生，所以在协议实现时高可用性就显得非常的重要。为了支持 ForCES 协议的高可用性，引入 CE 冗余和失败恢复机制。CE 冗余机制是在网络单元中包含多个 CE，其中一个是主 CE，其余的是备份 CE，当主 CE 发生错误或者链路发生问题使关联断开时，备份 CE 能立即代替主 CE 继续地工作，从而使失败得到了快速地恢复[12]。为了支持快速的主备切换，FE 必须首先和主备 CE 均建立连接。失败恢复有两种实现模型，即强一致性模型和弱一致性模型[8]。在协议实现中，为了简化主备 CE 之间同步性和一致性的实现，采用强一致性模型，使 FE 与主备 CE 之间均建立数据控制通道，传递协议控制报文和数据报文，如：RIP、OSPF 路由报文等。这样就可以保证 CE 间的同步，使得失败时能够迅速地切换，而不需要使用 CE 之间的协议来进行一致性维护。

总结和下一步工作 我们已经在基于 OpenRouter 模型的原型系统中实现了 ForCES 协议，并且能够正确稳定地运行，这为进一步开展路由器控制技术的研究提供了一个良好的试验验证平台。目前国际上对 ForCES 协议的研究还处于探索阶段，对于协议体系结构中除 Fp 参考点外的其他参考点的细节问题和实现问题还有待于考虑。随着新一代互联网体系结构理论研究的不断深入，ForCES 协议将会不断地发展和完善。

参 考 文 献

1 Zhao Feng, Su Jinshu, Cheng Xiaomei. OpenRouter: A TCP-based Lightweight Protocol for Control Plane and Forwarding Plane Communication. ICCNMC, 2004

2 Yang L, Dantu R, Anderson T, et al. Forwarding and Control Element Separation (ForCES) Framework. RFC 3746, April 2004

3 Doria A. GSMPv3 Base Specification. draft-ietf-gsmp-v3-base-spec-00, 2002

4 Putzolu D. NPF Software Architecture. http://www.npforum.org

5 P1520 Reference Model. http://www.ieee-pin.org

6 Yang L, Dantu R, Anderson I, et al. Forwarding and Control Element Separation (ForCES) Framework. RFC 3746, April 2004

7 Khosravi H, Anderson T. Requirements for Separation of IP Control and Forwarding. RFC 3654, November 2003

8 Audu A, Gopal R, Khosravi H, et al. ForwArding and Control ElemenT protocol (FACT). draft-gopal-forces-fact-06.txt, November 2003

9 Salim J H, Haas R, Blake S. Netlink2 as ForCES Protocol. draft-jhsrha-forces-netlink2-02.txt, 2003

10 Dong L, Doria A, Gopal R, et al. ForCES Protocol Specification. draft-doria-forces-protocol-01.txt, June 2004

11 Moy J, Pillay-Esnault P, Lindem A. Graceful OSPF Restart. RFC 3623, Nov. 2003

12 Dong Ligang, Doria A, Gopal R, et al. ForCES Protocol Specification. draft-ietf-forces-protocol-02.txt, 2005