软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(1):118−134 [doi: 10.13328/j.cnki.jos.004458]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

互联网自动配置研究*

李福亮[1,2], 杨家海[1,2], 吴建平[1,2], 安常青[1,2], 姜 宁[1,2]

[1](清华大学 信息网络工程研究中心,北京 100084)

[2](清华信息科学与技术国家实验室(清华大学),北京 100084)

通讯作者: 杨家海, E-mail: yang@cernet.edu.cn

摘 要: 互联网越来越复杂,网络设备支持的功能和服务越来越多,导致配置错误多发.配置错误已成为网络中断和异常产生的主要原因之一.互联网配置问题引起众多研究者的兴趣和重视,成为网络管理领域的一个重要研究课题.自 2002 年以来,研究者先后从不同角度对互联网配置问题进行了大量的研究,这些研究极大地促进了网络自动配置技术的发展.首先对互联网自动配置及配置案例进行概述;然后,按照配置自动生成、配置验证、配置自动实现这 3 个方面对互联网自动配置研究进行分类总结和分析评价;最后,总结了当前研究中存在的问题,并对未来研究发展趋势进行展望,希望能为该领域的研究者提供一些有益的启示.

关键词: 网络管理;自动配置;配置生成;配置验证;配置实现

中图法分类号: TP393 **文献标识码**: A

中文引用格式: 李福亮,杨家海,吴建平,安常青,姜宁.互联网自动配置研究.软件学报,2014,25(1):118−134. http://www.jos.org.cn/1000-9825/4458.htm

英文引用格式: Li FL, Yang JH, Wu JP, An CQ, Jiang N. Research on Internet automatic configuration. Ruan Jian Xue Bao/ Journal of Software, 2014,25(1):118−134 (in Chinese). http://www.jos.org.cn/1000-9825/4458.htm

Research on Internet Automatic Configuration

LI Fu-Liang[1,2], YANG Jia-Hai[1,2], WU Jian-Ping[1,2], AN Chang-Qing[1,2], JIANG Ning[1,2]

[1](The Network Research Center, Tsinghua University, Beijing 100084, China)

[2](Tsinghua National Laboratory for Information Science and Technology (Tsinghua University), Beijing 100084, China)

Corresponding author: YANG Jia-Hai, E-mail: yang@cernet.edu.cn

Abstract: The Internet is becoming extremely complex. Meanwhile, the network devices have been supporting more functions and services, which result in much more misconfigurations. Such misconfigurations, however, have become the main reason for network interruption as well as network anomalies. This issue has drawn many researchers' interest and attention, thus becomes a significant research topic in the field of network management. Since 2002, researchers have devoted themselves to solve configuration problems from different perspectives, and these studies greatly contribute to the development of the Internet automatic configuration. This paper firstly presents Internet automatic configuration and some configuration cases; then categorizes and evaluates Internet automatic configuration from the aspects of automatic configuration generation, configuration validation and automatic configuration realization; last but not least, summarizes the defects in the current research and then prospect the development of future research. The purpose of this paper is to provide some available information and beneficial enlightenment for researchers of this field.

Key words: network management; automatic configuration; configuration generation; configuration validation; configuration realization

配置管理是网络管理中的基础功能模块,在网络运行中占有非常重要的地位.然而,由于网络越来越复杂,

* 基金项目: 国家重点基础研究发展计划(973)(2009CB320505); 国家科技支撑计划(2008BAH37B05); 国家高技术研究发展计划(863)(2008AA01A303, 2009AA01Z251); 国家自然科学基金(61170211); 教育部博士点基金(20110002110056)

收稿时间: 2012-07-05; 修改时间: 2013-04-09; 定稿时间: 2013-07-09; jos 在线出版时间: 2013-11-01

CNKI 网络优先出版: 2013-11-01 13:49, http://www.cnki.net/kcms/detail/11.2560.TP.20131101.1349.003.html

© 中国科学院软件研究所 http://www.jos.org.cn

网络设备所能支持的功能越来越多,导致网络配置错误经常出现.很多文献致力于对 IP 骨干网[1,2]、互联网服务[3]、BGP 路由[4,5]等领域发生网络中断和异常的原因进行分析,结果显示,配置错误是导致网络中断和异常的最重要的原因.Yankee Group 公司在 2002 年对北美多家不同行业的公司企业网进行调研[6],调查显示,62%的网络中断事件与配置错误有关.错误配置不仅会引发网络故障,还会影响到网络的安全运行.因为安全策略大都基于高层的抽象,但是为了实现这些安全策略,往往需要管理员手工地将其转化成底层的配置命令,由于网络的复杂性以及配置命令的多样性,不可避免地会出现一些非预期的配置状态,从而造成安全漏洞,成为网络运行的隐患.

Benson 等人引入关系图和模板作为复杂度量化模型,以说明网络设备和网络服务在配置上的复杂性[7,8],结果表明,网络配置的复杂度仍然处于一个上升的趋势,这从侧面说明,配置出错的可能性也会随之增加.导致网络配置错误的原因主要分为如下 4 个方面[9]:

(1) 网络配置参数繁多.网络设备需要配置各种协议和服务,包括路由协议(RIP,OSPF,IS-IS,BGP,MPLS 等)、端口、ACL、QoS、SNMP、NTP、NAT 等,每种协议或者服务都包含很多配置参数;

(2) 复杂低级的配置语言.设备的配置都是基于低级的配置命令,每一个配置命令只代表一个简单的功能,一个复杂的配置任务可能需要上千行的配置命令;

(3) 网络设备多样性.搭建一个网络可能采用不同厂商的设备,然而不同厂商的设备在配置命令集上不尽相同;

(4) 网络服务需求增多.为满足不同用户的需求,ISP 需要提供大规模的基于网络的服务,如 VPN,VPLS,VoIP,Virtual-Wire,DDoS protection 等.

以上原因都导致了互联网在配置的过程中可能会出现错误.

然而在实际的网络管理中,需要进行配置的场景很多:搭建一个全新的网络,如企业网、校园网、政府办公网等;在现有网络的基础上增加新的设备(路由器、交换机等)、新的策略(ACL、QoS 等)、新的服务(VPN、VPLS 等)、新的客户(如 IP 段、接入网络等);在运行的网络中调节一些配置项,如调节 OSPF 的 *cost* 值、调节 BGP 的 *preference* 值等.然而面对大量的配置需求,多数管理员依旧采用传统的配置方法,该方法大致分为两个步骤:首先是确定顶层配置策略;然后,根据网络拓扑信息和网络运行状态,将顶层配置策略转换成底层的配置操作.这是一个从顶层直接到底层的配置过程,管理员充当了整个配置过程中的所有角色,管理员在任何一个环节出现错误,都会造成最终配置的失败.传统配置方法对管理员的要求很高,不仅需要掌握大量的配置命令,还需要注意设备配置之间的依赖关系,对于复杂的配置任务,配置工作量很大,出错的几率也会增加.为了应对互联网大量的配置需求以及配置复杂度日益增长的趋势,很多学者致力于互联网自动配置的研究.自动配置是互联网配置管理研究领域的一部分,按照流程,可以将自动配置分为配置自动生成、配置验证和配置自动实现,目标就是最大程度地实现互联网配置的自动化执行,减少管理员的手动操作,降低配置出错的可能性.

清华大学依托对 CERNET(中国教育和科研计算机网)和 CERNET2(中国下一代教育和科研计算机网)的管理,开发了一系列网络管理软件,如 iNetBoss 等,这些软件被应用于 CNRNET 和 CERNET2 主干网以及清华大学和全国其他多所高校的校园网管理上.近年来,配置管理在网络管理中的基础性地位日益突显,所以我们将部分力量转移到网络配置管理的研究中来,尤其是互联网自动配置研究.但是,目前研究仍处于探索和认知的阶段,所以本文除了希望能给该领域的研究者一些启示外,也是对我们今后研究方向的一个探讨.

本文首先对互联网自动配置以及互联网配置案例进行概述;然后按照配置自动生成、配置验证和配置自动实现这 3 个方面对互联网自动配置研究进行分类总结和分析评价;在文章最后,总结当前互联网自动配置研究中存在的问题,并对未来研究发展趋势进行展望.

1 概 述

在 ISO/IEC 7498-4 中,互联网管理被划分为 5 个功能域,即配置管理、故障管理、性能管理、计费管理和安全管理[10].配置管理是网络管理的基础功能模块,然而,由于网络设备的多样性以及网络设备所需要支持的功

© 中国科学院软件研究所 http://www.jos.org.cn

能越来越多,配置工作量大,配置错误多发,所以,互联网自动配置越来越引起工业界和学术界的重视.

1.1 互联网自动配置

基于新的网络配置管理架构[11],我们将互联网自动配置分为配置自动生成、配置验证和配置自动实现这 3 个部分:

(1) 配置自动生成.传统的配置方法是管理员逐个对设备进行配置,这种方法无法应对大规模网络设备的配置,并且很难兼顾设备之间的逻辑关系,自动配置方法应该能够对配置行为进行抽象描述,这样管理员面对的是抽象的配置行为,而不是每台设备具体的配置操作,抽象的配置行为能够被机器识别,为自动地生成多个设备的配置奠定基础.对配置行为进行抽象描述之后,需要将抽象的配置行为描述转换成具体的配置信息,这需要有一个知识库能够识别抽象的配置行为描述,或者有一个智能的工具能够根据抽象的配置行为描述进行智能的推理计算,最终得出具体的配置信息;

(2) 配置验证.生成的配置信息并不能保证是正确的,如果直接将其配置在设备上,存在配置出错的可能性较大,所以应该对配置信息的正确性进行验证;

(3) 配置自动实现.经过验证的配置信息依然是抽象的描述,它涵盖了配置的逻辑关系以及具体配置内容,需要构建通用的处理器,能够解读自动生成的配置信息,生成具体的配置命令.转换后的配置命令能够自动地下发到每一台设备上,需要对配置命令进行抽象的封装处理、隐藏配置命令实现的细节以及不同厂商设备在配置命令上的差异.

1.2 互联网配置案例

我们以配置 VPN 为例,说明一个网络服务的配置过程[8].搭建 VPN 服务,从 ISP 边界路由器的角度,需要满足以下 3 个具体要求:(1) ISP 能够区分不同用户的流量;(2) ISP 允许每一个用户选择自己的控制机制转发数据包; (3) 允许每一个用户拥有与其他用户相同的地址.如图 1 所示,红色用户与蓝色用户可能会给主机分配相同的地址.为了满足上述 3 个要求,ISP 需要为每一个用户创建虚拟空间,如图 1 所示,ISP 需要创建两个虚拟空间,一个为红色的用户群体,一个为蓝色的用户群体,那么 ISP 的关键作用是能够将这两个虚拟空间隔离,这种隔离需要 4 种能力:(1) ISP 能够在连接相同用户的 PE(provider edge)设备之间建立隧道,需要配置 MPLS(multiprotocol label switching)和 LSP(label switched path);(2) ISP 能够为同一个虚拟空间的多个用户交换路由信息,需要配置 MP-BGP,可以携带路由信息;(3) ISP 能够为不同的虚拟空间维持各自的路径信息,需要配置 VRF(virtual routing and forwarding),可以为每一个虚拟空间构建虚拟路由表.在图 1 中,PE1 需要配置两个 VRF,PE2 需要配置一个 VRF;(4) ISP 能够进行 CoS(class of service)路由,需要配置 MPLS,可以为报文加标签以及分类处理.

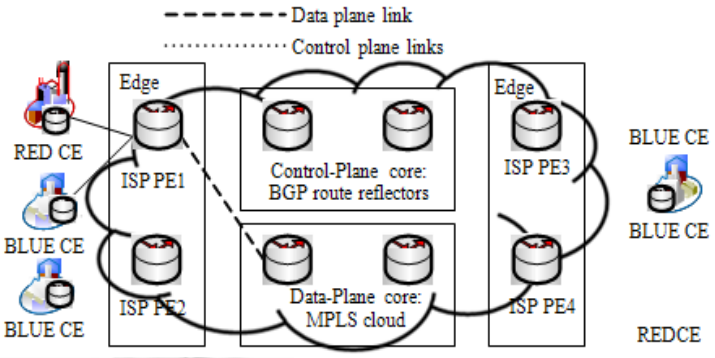

Fig.1 Architecture for a VPN service in an ISP[8]

图 1 ISP 的一个 VPN 服务结构[8]

结合我们在网络配置管理中的经验,介绍几项我们在自动配置领域的实际工作.

© 中国科学院软件研究所 http://www.jos.org.cn

1. SAVI 配置策略生成及自动配置.

CERNET2 的一项关键技术是源地址验证,该技术包括接入网、自治域内和自治域间这 3 个层次的验证机制,目标是确定一个源地址的合法性.SAVI(source address validation improvement)协议提供了在接入网进行源地址验证的方案.以清华校园网为例,管理员部署了 1 000 台支持 SAVI 协议的交换机.因为 SAVI 协议在不断地改进和提升,所以需要经常升级这些交换机的配置.每次升级,管理员都要对这 1 000 交换机逐一配置.为了提高配置的效率和准确性,我们设计了一个 SAVI 交换机配置管理系统,该系统不仅能自动生成 SAVI 配置策略,还能自动地将 SAVI 配置策略下发到每一台设备上[12].

2. VLAN 划分策略生成及自动配置.

为了减少广播流量并增强网络的安全性,管理员常常需要根据用户的角色将网络划分成多个 VLAN,每一个 VLAN 里的用户属于一个用户组,如学生用户组、教师用户组、工作人员用户组等.VLAN 划分需要遵守如下 3 个准则:

(1) 每一个 VLAN 下的用户数目不能多于子网前缀允许的最大主机数;

(2) 不同的用户组的用户不能划分在同一个 VLAN 里;

(3) 每一个 VLAN 的广播流量要控制在一定的范围之内.

以清华校园网为例,管理员管理着 2 500 台交换机,VLAN 划分及配置是一项繁琐的任务,我们设计了一个 VLAN 配置管理系统,根据网络拓扑及用户组的信息,该系统可以生成 VLAN 划分策略并且能够将策略自动配置到网络设备上[13].

3. 网络设备参数的自适应调整.

网络设备在运行的过程中有很多参数需要根据网络的运行状态进行适当的调整,这样可以优化网络性能,提高网络吞吐率.常见的网络设备参数如路由器中 OSPF 的 *cost* 属性、BGP 的 *preference* 属性等.基于 CERNET 主干网的实验环境,我们正在开展关于 OSPF 和 BGP 参数自动调整的工作.我们已有的一项工作是根据 SAVI 交换机中过滤表项数的变化对过滤表项生存期参数进行动态调整,在保证交换机过滤表在任何时间都不会被填满的前提下,尽可能地减少交换机对过滤表项的重复确认次数[14].

2 配置自动生成

当前,互联网配置自动生成方法主要分为两类:基于任务封装的方法和基于模型查找(model finder)思想的方法.我们首先对这两种方法分别进行总结评述,然后对这两种方法进行比较.

2.1 任务封装

任务封装是将一个个配置任务封装成配置模板,一个模板能够完成一个功能的配置,高层的配置策略触发配置模板中的变量,生成具体的配置方案.Boehm 等人[15]设计了一个基于任务封装的自动配置系统,该系统把单个 BGP 声明的路由策略转化为网络范围的路由策略,能够对网络中所有的路由器生成合适的配置方案.早期的基于任务封装的配置方法大多集中在路由方面[16],使用该方法在其他配置领域的代表是 PRESTO 系统[17],该系统首先定义了数据模型,即使用关系型数据库对被管对象的基本信息进行归类、存储,并且将配置命令写成模板的形式,模板不仅包含了设备某个功能所需的配置命令,还包含控制结构,如迭代、条件等.PRESTO 系统使用了类似 SQL 查询语句的方式对需要配置的目标设备的基本信息进行获取,并将查询到的设备基本信息以及相应的配置信息传递给模板中的变量;然后,模板会将其中的变量替换成获取到的设备信息以及配置信息,并根据其中的控制逻辑生成相应的配置方案.PRESTO 系统完成了对 VPN 的配置和 VoIP 的配置.Chen 等人提出的 PACMAN 系统[18]使用高级语言来增强配置任务的控制逻辑.MOP(method of procedure)文件描述了一系列的网络配置操作的流程,PACMAN 系统从 MOP 文件入手,将其转换成有利于机器执行的 AD(active document)文件.这里的 AD 文件类似于模板,一个 AD 文件完成一个具体的配置任务,输入正确的参数就会触发 AD 文件生成具体的配置方案.PACMAN 采用 Petri 网[19]的思想创建 AD,在如图 2 所示的 AD 设计范式[20]的基础上,通过任务组合的机制,可以实现复杂的任务配置方案.PACMAN 系统实现了 IGP 迁移任务的模拟实验.基于任务封装的方

© 中国科学院软件研究所 http://www.jos.org.cn

法是一种非常实用的配置自动生成方法,管理员根据配置任务的需求添加一些控制流程,输入一些参数,就可以批量地生成多个设备的配置.这种方法简单、易懂,而且易于操作.

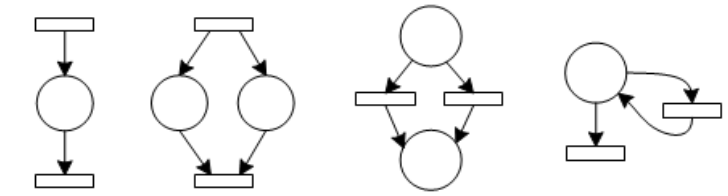

Fig.2 Active document design paradigms[20]

图 2 AD 的设计范式[20]

2.2 Model Finder思想

复杂的端到端网络服务的建立,需要通过配置实现,每个功能组件与一系列有限的配置参数相对应,而每个配置参数可以取若干可枚举的配置值.端到端的网络服务在连通性、安全性、性能以及故障恢复方面都存在着需求.然而,这些高层的需求和具体的配置参数之间仍存在很大的鸿沟.基于 Model Finder 的核心思想是:对网络配置对象和网络功能需求进行形式化描述,经过解析,将需求描述转换成布尔逻辑表达式,每一个布尔表达式代表了一个配置参数,找出满足所有布尔逻辑表达式为真的解,也就是确定了每一个配置参数的值,这个解的集合将作为可行的配置方案.

COOLAID[21]是一个基于模型查找思想的配置自动生成系统,该系统将所有的网络配置信息以及状态信息都建模成关系型数据,并在关系数据的基础上构建视图,使用声明式的语言 Datalog[22]来对网络服务的依赖关系以及规则进行描述.图 3 显示的是 OSPF 路由学习规则的 Datalog 描述.COOLAID 系统设计了针对 Datalog 描述语言的规则处理器,它可以进行递归式的信息查询,最终找到可行的配置方案.但是,Datalog 的描述能力较弱,它不支持否定、所有、存在等量词的描述,所以对于复杂的配置任务,Datalog 的抽象描述能力是不够的.

```
R0 EnabledIntf (ifId,rId): –TRouterIntf (ifId,rId),
      Cinterface(ifId,"enable");
R1 OspfRoute(rId,prefix): –EnabledIntf(ifId,rId),
      CIntPrefix(ifId,prefix), CIntfOspf(ifId);
R2 OspfRoute(rId1,prefix): –OspfRoute(RId2,prefix),
      TIntfConnection(ifId1,rId1,ifId2,rId2),
      EnabledIntf(ifId1,rId1), CIntfOspf(ifId1),
      EnabledIntf(ifId2,rId2), CIntfOspf(ifId2);
```

Fig.3 Data description of rules for OSPF route leaning[21]

图 3 OSPF 路由学习规则的 Datalog 描述[21]

Sanjai 等人[23]采用 Alloy 语言[24,25]对网络服务的依赖关系和规则进行抽象描述.Alloy 是一阶逻辑语言,相比 Datalog,其描述需求的能力更加丰富.Sanjai 等人设计了一个需求解析模型,如图 4 所示:首先,输入网络组件的集合和基于这些组件的配置需求;然后,分析器进行解析和计算;最终得出满足这些需求的每个网络组件的正确配置.基于这个模型,Sanjai 等人搭建了一个 VPN 服务,实现了配置生成、需求增加、组件增加和需求验证这 4 个功能,验证了 Alloy 的需求描述能力和规则解析能力.但是,将一阶逻辑描述转换成布尔逻辑的过程中会产生大量的中间约束,求解时间会随着中间约束数量的增加呈指数增长,所以,该方法的求解效率是无法保证的.

在对配置规则及约束进行形式化描述时,除了以上介绍的 Alloy 语言、Datalog 语言,ConfigAssure 系统[26]采用了量词无关的形式(quantifier-free form)——QFF,它增加了简单的谓词运算,比布尔逻辑有着更为丰富的表述能力,QFF 采用 Prolog[27,28]语言来描述.PoDIM[29]则是一种面向对象的语言,但是它不适合描述基于策略的配置任务,如 ACL、防火墙等.

© 中国科学院软件研究所 http://www.jos.org.cn

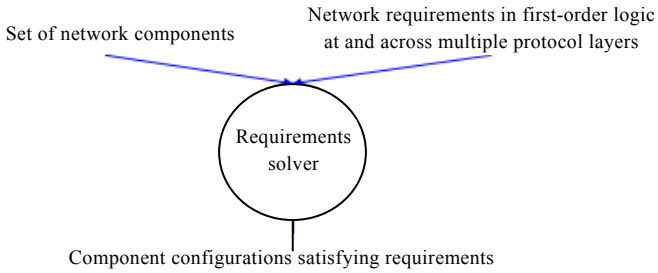

Fig.4 Requirement solver[23]

图 4 需求解析模型[23]

基于模型查找思想的配置自动生成方法是一种真正意义上的配置自动生成方法,总结该方法的研究现状,需要注意以下几个问题:

(1) 合理地选择描述语言

常见的形式化描述语言包括 Prolog,Datalog,Alloy 等,Prolog 和 Datalog 对于否定、所有、存在等量词不能正确表述,较弱的表述能力决定了它们不适用于复杂的网络配置任务的描述;而 Alloy 有着丰富的表述能力,能够充分地表达复杂配置任务的需求,所以相对其他描述语言,Alloy 是比较适合对配置需求进行描述的语言.另外,需要选择合适的规则解析方法计算可行的配置方案.不同的描述语言可能会选择不同的解析方法,如 Datalog 语言会根据本身具有的特色,通过递归查询的方式获取可行解;而 Alloy 首先将形式化描述转换成 3-SAT 范式,然后通过 SAT 解析工具寻求可行的配置方案.根据描述语言的特色选择合适的解析方法,这不仅是能够获取解的前提,也是保证解的正确性的前提.

(2) 充分地了解配置需求

要实现充分的需求描述,除了选择一种丰富的形式化描述语言以外,还需要管理员充分了解配置任务的需求,需求可分为连接需求、安全需求、可靠性需求、性能需求等几个方面.其中,连接需求是基础的需求,也可以叫作可达性需求,它要求两个节点之间存在路径,对于 IP 网络,节点表示的是子网或者路由器,链路表示节点之间有路径相连,然而现实网络中不仅仅是简单的 IP 网络,还需要充分考虑 VLANs,GRE,IPSec,BGP 和 MPLS 等,以及一些策略,如 ACL,firewall 等,使得可达性需求的描述更加复杂;安全需求主要是保证数据的可信性、整体性、可认证以及访问控制等;可靠性需求主要是保证节点或者链路在失效的情况下,网络能够正常运行等;性能需求主要是优先级处理报文以及控制某些协议的报文的传输等.

(3) 有效地提高求解效率

需求描述要充分,但并不是为了涵盖各个方面的需求而不考虑需求的规模,要做到尽可能地精简需求,这样,在规则推理过程中才能尽快地找出可行的配置方案.以 Alloy 描述方法为例,它首先将需求描述转换成 3-SAT 范式,但是求解 3-SAT 是一个 NP 完全问题,不可能在多项式时间内找到解.有一种高近似的随机算法能够求出解,时间复杂度为$(4/3)^n$,其中,n 为 3-SAT 从句的个数.那么,减少 n 的数量就可以有效地缩短求解时间.但是,在由一阶谓词逻辑转换成布尔逻辑的过程中会产生很多中间约束变量,这会大大增加 n 的数量,可以采用将网络划分成多个子部分,并对每个子部分分别进行描述、推理的方式,还可以通过优化网络设计,以减少网络组件的数量[23].另外, kodkod 采用局部最优的思想,已经有确定值的配置参数不再参与运算,一定程度上缩小了 3-SAT 的规模[30].

2.3 配置自动生成方法比较

前面根据任务封装和Model Finder思想这两个方面对配置自动生成方法进行了总结,每种方法都有一定的优势,但也都存在一定的不足.基于任务封装的方法适用于基础配置,因为这些配置在技术上相对成熟,需求变化较少,如 IP 地址分配、端口配置等;基于 Model Finder 思想的方法适合于基于策略和依赖关系的配置行为,如配置 VPN,VPLS 等服务以及 ACL,QoS 等策略.理想的配置自动生成方法应具备方法实施简单、执行效率高

© 中国科学院软件研究所 http://www.jos.org.cn

效、可扩展性良好等特点.两种配置自动生成方法在方法实施、执行效率和可扩展性上的对比结果见表 1.

Table 1 Comparison on methods of automatic configuration generation

表 1 配置自动生成方法比较

方法名称	代表研究	实施难度	执行效率	可扩展性
任务封装	PRESTO 系统[17],PACMAN 系统[18]	较容易	较高	不好
Model Finder 思想	COOLAID 系统[21],ConfigAssure 系统[26]	较难	较低	好

- 方法实施

任务封装方法的本质是构造配置模板,管理员输入参数,触发模板中的变量,便可生成相应的配置.配置模板的构造大都是直接封装配置命令,每一个配置模板完成一个特定功能的配置.管理员可以将配置模板进行组合,完成复杂的配置任务.那么,任务封装方法的难度在于配置模板的构建需要管理员熟悉不同设备的配置命令,因为不同厂商的设备、相同厂商的不同型号设备、相同型号设备的不同操作系统,在配置命令集上都会存在一些不同,管理员在构建配置模板时需要区别这些不同,并能保证模板逻辑上的正确性,这对管理员的要求很高.基于 Model Finder 思想的自动配置方法是一种真正意义上的配置自动生成方法:首先,对网络配置对象和实现网络配置功能的需求进行抽象的描述;然后,通过对描述语言的解析,将描述换成布尔逻辑表达式,找出满足所有布尔逻辑表达式为真的解,这个解的集合将被视为满足配置任务的模型,也就是可行的配置方案.该方法的难度在于如何有效地对需求进行描述,由于需求涉及到多个方面,如连接性、可靠性、安全性等,首先要做到需求描述能够涵盖各个方面,另外还要尽量做到需求描述的精简.除此之外,还要选择合适的描述语言和求解方法.所以,基于模型查找的方法在实施的过程中仍然存在一定难度.

- 执行效率

基于任务封装的方法虽然在构造配置模板上有一定的难度,但是该方法易于执行,对于简单的批量配置任务,管理员可以容易地构造配置模板,然后生成多个设备的配置.所以,基于任务封装的方法对于简单的配置任务执行效率很高.在实际的网络管理中,这种配置自动生成方法依旧最为实用.基于 Model Finder 思想的方法需要对配置任务进行准确的需求描述,但该方法的求解过程是一个 NP 完全问题,只能通过随机算法求得近似解,然而随机算法的求解效率却依然较低.因此,基于 Model Finder 思想的自动方法更适合于复杂配置任务的配置方案规划,对于简单配置以及实时参数优化等配置任务,该方法并不适合.

- 可扩展性

基于任务封装的方法对于全新的配置任务或者改进的配置任务,需要管理员构建新的配置模板.另外,当设备更换或者系统升级时,配置模板中封装的配置命令也需要重新设计.很明显,可扩展性较差.基于 Model Finder 思想的自动配置方法需要针对全新的或者改进的配置任务进行新的需求描述,然后通过推理计算,寻求新的配置方案.因为有很多抽象的需求描述可以被重用,所以相对于重新构造配置模板,该方法的可扩展性相对好一些.另外,对于复杂的配置任务,由于配置模型可以自动生成,所以该方法在扩展性方面表现出明显的优势.

3 配置验证

配置验证是一个非常重要但又容易被忽略的问题.端到端的需求转化为具体的配置需要经过很多步骤,转换的每一个阶段都有可能出现问题,最终反映为在配置上的错误.对于一个新搭建的网络,我们可以尝试着将配置方案运用于实践,在运行过程中去检验配置方案是否正确;但是对于实际运行的网络,如果我们同样直接将配置方案运用于实践,那么存在的风险是很大的,因为配置上细小的变化就可能引起整个网络异常或者中断,所以需要对配置方案进行正确性的验证.对于任务封装的方法,管理员可以根据自己丰富的领域知识对构造的模板进行错误排查,这不仅对管理员的要求很高,而且依然存在出错的可能性;基于 Model Finder 思想的方法可以进行迭代式的需求验证[23,26]:首先,需要充分考虑网络在运行过程中可能出现的问题;然后,将问题转换成需求并进行描述,与之前已经得到配置方案的需求一起再一次进行规则解析,如果能够找到可行的配置方案,那么就说明之前的需求描述存在缺陷,需要重新修改需求,以避免不希望出现的问题.这种方法需要充分考虑网络运行过

© 中国科学院软件研究所 http://www.jos.org.cn

程中可能存在的各种问题,但这一点很难做到.前面说的一些验证方法或多或少存在一些不足,所以并不能保证生成的配置方案是正确的.那么,最可信的验证方法是将配置方案应用在设备上,用实际的配置效果检验方案的正确性.根据配置验证场景的不同,配置验证可分为模拟验证和实际验证两种方法.

3.1 模拟验证

模拟验证指的是配置方案生成以后,借助于虚拟或者实验性的网络环境,对产生的配置操作进行验证,如果方案正确,再进行实际运行网络的配置.模拟可以分为 3 类.

(1) 路由模拟软件(simulator)

这是由软件实现的路由模拟器,如 Router-Sim,CIM(Cisco interactive mentor),OPNET,Boson NetSim 等,其中,NetSim 是一款推荐的路由器模拟软件,它可以模拟路由器和部分交换机,而且最先提供了自定义网络拓扑的功能.与真实的实验网络相比,使用 Boson NetSim 省去了制作网线连接设备、频繁变换 CONSOLE 线、不停地往返于设备之间的环节.同时,Boson NetSim 的命令也和最新的 Cisco 的 IOS 保持一致,它可以模拟出 Cisco 部分中端级别的产品,如 35 系列交换机和 45 系列路由器等,因此,我们可以根据模拟的网络环境对配置方案的有效性进行验证.

(2) 路由仿真软件(emulator)

仿真软件与模拟软件的区别是模拟软件使用编程语言重现设备的 IOS,而仿真软件则是提供真实的设备 IOS 代码,在此基础上仿真硬件,所以在命令集上更加丰富,它的一个代表软件是 Dynamips,辅助支持的软件包括 Dynagen 和 GNS(graphical network simulator).

(3) 真实实验网络

不论是模拟还是仿真,或缺少丰富的命令集合,或缺少丰富的设备型号支持,有时不能满足配置验证的需求.那么,最好的验证环境是真实的实验网络,但是实验网络涉及到设备贵、布线难、灵活性差等缺点.针对以上问题,Huan 等人提出了一种致力于配置测试的真实实验网络架构[31].

3.2 实际验证

实际验证指的是当配置已经运行在实际的网络环境中,通过观察网络运行状况或者分析控制平面和数据平面的数据(配置文件和路由转发表)对配置进行验证,如果发现配置错误,能够迅速恢复设备的配置到最近的有效状态.实际验证可分为 3 类.

(1) 观察网络运行状况

可以通过链路的传输速率、利用率、吞吐率、延时、丢包率等来观察设备配置的效果,如果发现异常,可以将设备的配置文件恢复到最近的有效状态.当然,网络出现异常不一定是配置错误导致的,所以这种观测方法一般适用于新配置刚刚下发的情况.对比下发前后的网络运行状况,如果下发之后没有达到预期效果或者出现异常,那么可以认为是配置不当;然而其他情况下,观测的网络异常需要对其产生的原因进行深入的分析.

(2) 分析控制平面数据

可以周期性地备份设备的配置文件,提取其中的配置数据进行配置一致性的验证.配置一致性的验证不仅可以发现最近下发的配置错误,对历史配置可能存在的错误也可以进行验证,文献[3,32−35]通过分析控制平面数据对网络进行验证.总结验证方法大致分为如下 4 个模块:

- 获取配置信息模块.负责提取设备配置文件的信息,并保存为与设备无关的数据格式;
- 建立需求数据库模块.负责记录最优化情况下的端到端的需求;
- 形式化描述模块.负责选择描述语言,简化需求的规范化描述;
- 需求验证模块.负责验证需求,给出错误配置的修改意见,并且能够展示设备的逻辑连接关系.

这个过程类似于基于 Model Finder 思想的自动配置研究,其原理和方法有很多相似之处.

(3) 分析数据平面数据

数据平面中的数据主要指的是路由转发表的数据,文献[36−39]通过分析数据平面的数据对网络进行验证,

© 中国科学院软件研究所 http://www.jos.org.cn

这种验证方法不能直接反映配置上的错误,但它可以推断网络中存在的一些错误,如路由环路、丢包等,管理员根据这些反馈可以定位一些配置上的错误.文献[40]给出了通用的验证方法,该方法分为如下 4 个模块:

- 获取转发信息模块.负责获取路由设备的转发表信息(FIBs),这些信息可能是简单的最长匹配规则,也可能是复杂的访问控制和报文封装策略等;
- 建立不变量库模块.不变量表示转发行为的正确条件,违背这些条件,通常预示着网络可能存在问题,使用脚本语言或者形式化语言描述这些不变量;
- 验证模块.将 FIBs 信息及不变量转换成 SAT 实例,并提交给 SAT 解析器处理,处理结果如果违背了不变量,则预示着网络中可能存在错误;
- 与此同时,验证模块给出一个反例指导该错误的修改.

4 配置自动实现

配置自动实现的目标是将正确的配置方案自动地下发到具体的设备上.当前比较主流的研究主要分为两类:基于脚本封装的方法和基于 NETCONF 协议的方法.我们首先对这两种方法分别进行总结、评述,然后对这两种方法进行比较.

4.1 脚本封装

脚本是批处理文件的延伸,是一种纯文本保存的程序,包含了一系列控制机器操作的组合.基于脚本封装的方法是将设备能够识别的配置命令封装到脚本里,这里的配置命令指的是 CLI 命令,有时也会涉及 SNMP 命令,然后借助可以与设备交互的工具,如 expect 等,便可直接将配置方案下发到具体的设备上.配置脚本与配置模板密切关联,配置模板可以依据脚本的形式构建,并结合 expect 等工具,便可实现对多个设备的批量配置.基于任务封装的配置自动生成方法大都结合了脚本封装[15−17]来构建自动配置系统,一个任务被封装成一个脚本.然而,基于 Model Finder 思想的配置自动生成方法生成的配置信息仍然是抽象的描述,需要构建通用的处理器,将抽象的配置信息先转换成具体的配置命令,然后再构建相应的配置脚本.在实际的网路管理运行中,基于脚本封装的配置自动实现方法常常被管理员认为是最为实用的方法,因为它对管理员的要求较低,管理员只要有了配置方案,就可以容易地将其转换成可以在机器上自动执行的脚本.基于脚本封装的方法适合在多台设备上配置相同的配置命令,从而极大地减轻了管理员的工作负担.2012 年 6 月,清华大学校园网 1 000 台 SAVI 交换机系统升级,如果由人工手动升级,那么工作量是很大的.我们采用脚本封装的方法,将要升级的配置方案封装到脚本里,依据 1 000 台交换机的管理 IP 地址,逐一对设备进行了配置,整个过程持续了 6 个小时.可想而知,如果是人工手动配置,那么耗费的时间会远不止 6 个小时,而且由于疲劳或疏忽导致配置出错的可能性很大.我们已有的一些工作也都基于脚本封装的方法来实现自动配置[11−13].另外,为了提高配置实现的效率,我们还采用了多线程机制同时对多个设备进行配置.

4.2 NETCONF协议

虽然 SNMP 被广泛应用到网络管理领域,但它在配置管理方面相对薄弱:首先,SNMP 报文受到 UDP 大小的限制,配置效率较低;其次,SNMP 协议为原子操作,它不能将网络作为一个整体,而是针对每一台设备进行配置,很难应对复杂的网络服务配置的需求[41].CLI 是最简单、最常用的配置方法,但是不同厂商的设备、同一厂商不同型号设备、同一型号设备不同版本的操作系统,其 CLI 命令都可能存在差异,所以需要管理员掌握大量的配置命令、协议以及设备的结构等,这对管理员的要求很高.另外,由于 CLI 命令本身的细粒度性,管理员在配置的过程中容易出错.基于配置管理存在的诸多问题,IETF 于 2003 年 5 月成立了 NETCONF 工作组,基于 XML 强大的描述能力和可读性以及 XML 相关技术的发展,IETF 于 2006 年 12 月正式将 XML 作为 NETCONF 协议的标准[42].NETCONF 协议采用 C/S 架构,概念模型如图 5 所示,分为 4 个层次:内容层、操作层、RPC 层和传输协议层.其中,

- 传输层主要负责为服务器与客户端(管理系统与底层设备)之间的信息交互提供通路,保证管理信息有

© 中国科学院软件研究所 http://www.jos.org.cn

效、可靠、安全的传输,基于 SSH,SOAP,BEEP 的方法相继被提了出来[43−45];

- RPC 层定义了简单且与协议无关的 NETCONF 协议通信模型,该通信模型不受操作系统、实现环境的限制,规范了远程过程调用的信息编码格式;
- 操作层用以执行各种基本配置操作,如增加、删除、修改、备份、存储配置数据等;
- 内容层类似于 MIB,使用 XML 保证数据信息描述的一致性,但它并不涉及数据建模,所以可以使用其他高级建模语言对数据进行建模,使得 NETCONF 协议有较好的可扩展性.

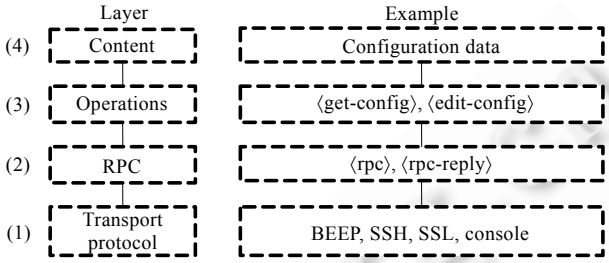

Fig.5 NETCONF conceptual model[42]

图 5 NETCONF 协议概念模型[42]

如上所述,NETCONF 协议不涉及数据建模,而为了弥补高层网络策略与低层网络配置数据之间的差距,一般是通过复杂的抽象语言进行数据建模,在这种情况下,NETMOD 工作组针对 NETCONF 协议提出了标准的数据建模语言 YANG[46].YANG 语言不仅支持强大的语法和语义验证规则,而且书写简单、易于理解,能够对配置信息的数据结构、数据关系以及约束的完整性等进行建模.基于数据建模语言 YANG,Elbadawi 等人[47]设计了一个语义层模型,定义为 CSM(configuration semantic model).基于 CSM 的配置自动实现架构如图 6 所示.这个语义层能够正确、有效、合理地描述网络配置,使得自动生成的配置信息能够被语义层识别和解析,并找到合适的配置模型,配置模型被代理转换成物理设备识别的配置命令,然后下发到具体的设备上.NETMOD 工作组在 YANG 语言的基础上提出了一系列草案[48],这些草案定义了多种配置模型,每种配置模型定义了一种具体的配置任务,如端口、路由协议、IP 地址、访问控制等,这些配置模型为配置自动实现奠定了基础.

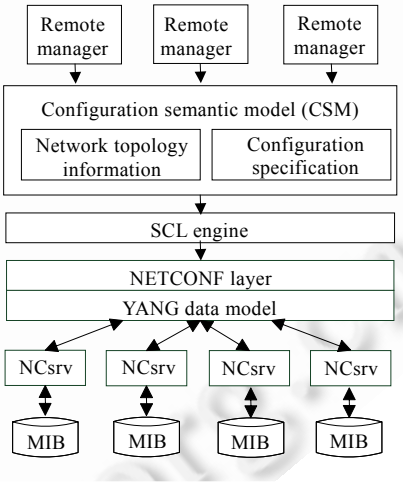

Fig.6 A framework for network automatic configuration based on NETCONF[47]

图 6 基于 NETCONF 协议的网络自动配置架构[47]

工业界成为了 NETCONF 协议的主要推动者:Juniper 最早提出了 NETCONF 协议规范[42];而瑞典的 Tail-f 公司一直致力于研究基于 XML 技术的网络管理软件,不仅提出了一系列支持 NETCONF 的协议和标准[49,50],还完成了基于这些协议的代表产品[51];Cisco 公司提出了基于 BEEP 的 NETCONF 协议以及 NETCONF 事件通知机制[45,52].NETCONF 协议为网络管理及配置管理引入了新的思路,这也引起了学术界的大力关注,比较有代表性的如法国的 LORIA,INRIA 研究院、韩国的 POSTTECH 大学以及中国的华中师范大学[53−57]等.

4.3 配置自动实现方法比较

前面按照脚本封装和 NETCONF 协议两个方面对配置自动实现方法研究进行了总结.基于脚本封装的方

© 中国科学院软件研究所 http://www.jos.org.cn

法在实际的网络管理中较为常用,而基于 NETCONF 协议的方法因为需要额外部署代理,目前还没有普遍使用.一种理想的配置自动实现方法同样应该具备方法实施简单、执行效率高效、可扩展性良好等特点.配置自动实现方法在方法实施、执行效率和可扩展性上的对比结果见表 2.

Table 2 Comparison on methods of automatic configuration realization

表 2 配置自动实现方法比较

方法名称	代表研究	实施难度	执行效率	可扩展性
脚本封装	PRESTO 系统[17]	较易	高	不好
NETCONF 协议	CSM 模型[47]	较难	较高	好

- 方法实施

基于脚本封装的方法是将 SNMP 或 CLI 命令写成脚本的形式,然后让机器自动运行脚本.构建脚本对管理员要求较低,熟悉脚本的基本知识,弄清楚具体的配置方案,就可以编写一个自动执行配置脚本,所以基于脚本封装的方法实施起来相对简单.基于 NETCONF 协议的方法需要使用 XML 对配置数据进行描述,使用 RPC 模型进行通信编码,使用 YANG 语言对底层配置数据进行建模,使得配置管理系统能够通过代理与物理设备进行交互,执行各种基本配置操作,如增加、删除、修改、查询、存储配置数据等.基于 NETCNOF 协议方法的难点在于,对于传统的网络架构,需要额外的设计和部署外部代理.

- 执行效率

基于脚本封装的方法只要将脚本直接运行在网管服务器上,便可对设备进行配置.另外,可以采用多线程机制,实现同时对多台设备进行配置,提高执行效率.脚本与模板关系密切,所以,基于脚本封装的配置自动实现方法适合与基于任务封装的配置自动生成方法结合起来使用,对于一些简单的批量配置任务,二者的结合执行实用并且高效.基于 NETCONF 协议的方法需要相关研究,尤其是 YANG 语言及基于 YANG 语言定义的一系列配置模型的配合,需要通过代理与设备进行交互.相比于脚本直接对设备进行操作,基于 NETCONF 的方法需要几个步骤的结合才能最终将配置下发到设备上.然而,基于 NETCONF 协议的自动实现方法可以隐藏配置命令的实现细节,所以适合与基于 Model Finder 思想的配置自动生成方法结合起来使用,我们只需要为自动生成的抽象配置信息找到合适的配置模型,无需关心具体的配置命令.

- 可扩展性

基于脚本封装方法的核心是针对每一种配置功能都要构建一个脚本,当设备的型号、厂商、操作系统版本等不同时,配置命令就都有可能存在差异,所以同一种功能的配置脚本也不尽相同.基于 NETCONF 协议方法的核心是通过代理与设备进行交互.基于 YANG 语言可以构建一系列配置模型,一种模型实现了一种功能的配置,这种配置模型隐藏了配置命令的细节,所以面对不同型号、不同厂商、不同操作系统版本的设备,配置模型可以被复用.另外,还可对配置模型进行组合,加以一定的控制逻辑,以满足复杂配置功能的需求.

5 研究展望

前面按照配置自动生成、配置验证和配置自动实现这 3 个方面对互联网自动配置研究进行了总结和评述.配置自动生成与配置自动实现之间的关系非常密切,当前的研究大都将两部分放在一起讨论.本文按照先后顺序的方式来叙述,主要是为了方便研究者理解自动配置的层次关系,突出配置验证的重要性.总结当前的研究方法,主要存在以下 5 个方面的不足:

(1) 缺少层次化的数据抽象体系

我们可以用计算机编程语言的发展作类比:开始,编程人员直接面对机器语言,之后,出现汇编语言、高级语言,逐渐地,很多人会实现一些代码库,而应用开发人员则利用这些代码库来编写更高层次的应用程序.不同层次的人员使用不同的抽象,专注于不同的层面,而其他层则对其透明.传统的网络配置相当于配置人员直接使用“机器语言”来完成“高层应用的开发”,这之间显然缺少多个方面的抽象.自动配置研究的目标仅从自动化的角度讲可以定义为配置任务的自动解析、配置方案的自动生成和配置命令的自动实现.如果要做到自动地实现每

© 中国科学院软件研究所 http://www.jos.org.cn

一个目标,那么每一个层面都需要进行数据抽象,机器识别抽象的数据描述,借助“代码库”,自动地去完成每一个目标.然而,当前的自动配置研究方法更多地专注于配置的自动生成和配置命令的自动实现,对于配置任务的自动解析没有系统化的研究,也缺少配置自动生成研究与配置自动实现研究之间的关联.如果将配置自动生成定义为中间层,配置自动实现定义为底层,那么配置任务的自动解析可定义为顶层.如果对顶层进行合理的数据抽象,那么管理员面对的将是高层的配置任务;如果建立有效的机制,实现配置自动生成与配置自动实现之间的有效关联,那么中间层面对的是配置策略,而不是具体的配置命令.如果一个自动配置方法包含了完整的数据抽象体系,那么管理员的工作只是制定相应的配置任务.然而,当前的自动配置研究缺少层次化的数据抽象体系,管理员在配置的过程中仍需参与很多的操作流程,这不符合自动配置的目标.

(2) 缺少对配置任务描述完备性的深入探讨

完备的配置任务描述是自动配置顺利进行的前提,完备性不仅要保证配置任务的描述是正确的,而且要保证描述是全面的.要做到配置任务描述的完备性,首先要有合适的抽象描述语言,其次要对配置任务有充分的认识,最后要有合适的方法验证描述是完备的.目前的自动配置研究方法更多地关注抽象描述方法的选择,很少关心完备性的探讨.但是完备性在配置自动的执行过程中起着至关重要的作用,因为它是自动配置的起始,如果在开始阶段没有好的规划,那么在自动执行的过程中可能会出现各种情况,最严重的后果就是生成的配置方案并不能满足配置任务的本质需求.但是它并不存在错误,只是不够全面,如果这样的配置方案在设备上进行配置,导致的后果也许不仅仅是功能上无法实现,更有可能导致网络异常或中断.

(3) 缺乏健全的动态反馈机制

要想让管理人员彻底地从琐碎的配置中解脱出来,最理想的方式是网管系统可以收集网络设备当前的运行状态信息,然后对这些信息进行分析,根据分析结果指导配置工作的自动完成.在使用传统的配置方法时,通常管理员在配置某个参数时都会采用各种 show 命令来查看设备当前的运行状况,然后结合自己的领域知识,完成具体的配置工作.采用这种方式对管理员能力要求很高,而且当需要考虑的运行状况比较复杂时,管理员通常无法通过自身的能力进行准确的判断,往往采用一种主观臆断的方法来进行配置.显然,这种方式并不能保证配置的正确性以及配置后网络设备运行的高效性.因此,配置管理系统需要网络动态信息反馈的支持,具备将运行监控与配置自动关联的能力,实现完全的自动配置.

(4) 缺少有效的配置验证方法

由于不能保证配置方案的正确性,所以在配置具体实施之前,需要对其进行正确性的验证.验证的方法有很多种,最为有效的方法是将配置方案运用到实际的网络环境中,但是这种方法可能会影响正常的网络运行,代价较大.理想的方法是模拟相同的网络环境,在虚拟或者实验性质的网络进行配置方案的验证.但是很多时候,我们无法模拟完整的实际网络,因为一个配置任务不仅仅是简单的在几台设备上执行配置命令就可以了,还要考虑配置对当前无关设备的影响.所以,模拟验证很大程度上能够对配置进行验证,但是依然不能保证配置是完全正确的.所以,当配置部署到实际的网络环境中时,仍需作进一步的验证.当前的自动配置研究大都与配置验证分离,虽然一些方法能够基于建模语言进行语法和语义的检查,但仍然不能保证配置方案的正确性,所以需要将当前的自动配置研究与配置验证研究结合起来.也就是说,在配置自动生成与配置自动实现之间,有必要加入配置正确性验证环节.

(5) 缺少一个通用的自动配置体系架构

当前的研究方法或者系统架构大都局限于某些配置场景,虽然解决了配置管理领域的部分需求,但并不具有通用性,所以需要设计一个通用的自动配置体系架构,结合层次化的数据抽象体系,加以配置正确性的验证,实现完整的自动配置流程.由于配置问题的多样性,并不是一种自动配置流程可以满足所有的配置场景.当前的配置研究大都针对一种或者几种配置任务,然而这种方法对于其他的配置任务并不一定适用.所以,可以将配置任务分类,综合考虑方法实施简单、执行效率高效、可扩展性良好等因素,针对每一类配置问题选择最佳的自动配置流程.那么,需要该架构了解整个网络的详细拓扑结构,采集各种设备的运行数据,根据网络实际配置需求,智能地推算出符合条件的自动配置流程.自动解析配置任务,自动生成配置方案,加以正确性的验证,最终将

© 中国科学院软件研究所 http://www.jos.org.cn

配置方案自动地下发到相应的设备上.

5.1 构建层次化的数据抽象体系

互联网自动配置的目标就自动化而言包括配置任务的自动解析、配置方案的自动生成和配置命令的自动实现,那么可以针对每一个目标进行相应的数据抽象,构建一个层次化的数据抽象体系.

(1) 配置数据的抽象

目前,对于网络中设备的配置方法主要还是基于 CLI 命令,由于不同厂商设备的 CLI 命令集不尽相同,即使配置相同的功能项或参数,使用的命令也有细微差别.因此,我们需要对不同设备中的配置项进行数据抽象,为自动配置系统提供统一的配置项模型,配置项模型能够隐藏相同配置项在配置命令上的差异.在此基础上,构建通用功能的配置模型,配置模型能够自动地关联相应的配置项,每一个配置项模型也可算作是一个简单的配置模型.通过配置项模型和通用配置模型的建立,实现底层配置命令对上层的透明.

(2) 配置规则的抽象

互联网配置的功能越来越多,配置工作量越来越大.对配置规则进行抽象,可以应对复杂的配置任务,实现自动生成多个设备的配置方案的能力.对于一种全新或者改进的配置任务,采用当前的任何一种自动配置方法,管理员在配置设备前均需仔细学习设备厂商提供的配置手册,学习其中的规则后对每台设备进行配置,这些规则代表了设备的行为.那么可以对规则进行抽象描述,为系统提供统一的规则模型.这种描述不是描述规则该如何实现,而是描述规则是什么.计算机能够对规则进行解析,智能地选择最佳的配置方法,自动生成相应的配置方案.这种方式实现了配置规则(行为)对上层的透明,上层只需要清晰配置任务的规则,而无需关心规则如何实现,就可以自动地生成相应的配置方案.

(3) 配置任务的抽象

对于每一个配置任务,管理员都需要制定相应的配置策略,如果配置任务较多或者配置任务较为复杂,管理员在将配置任务转换成配置策略时会很麻烦.那么,可以对高层的配置策略进行抽象描述,为系统提供统一的策略模型.这样,首先对配置任务进行抽象的描述,抽象的描述可以被解析,找到合适的策略模型.如果策略模型并不包含我们需要的模型,那么需要扩展现有的策略模型.与此同时,为了降低配置任务的复杂度,配置任务可能通过解析被分解成一个个原子任务.每一个原子任务完成整个配置任务的一部分功能,每一个原子任务会对应一个策略模型.另外,应该能够按照优先级调度原子任务,并支持原子任务之间有值的传递.采用这种方式,管理员面对的是高层的配置任务,无需关心具体的配置策略.

5.2 配置任务描述的完备性

配置描述的完备性表现为描述的正确性和全面性.首先,要选择一种合适的描述语言,因为配置任务可能涉及到复杂的逻辑关系,所以描述语言要有丰富的表述能力,同时还应具备良好的语法和语义验证机制.为了使得任务描述之后能够被机器解析,描述语言还应支持一些简单的运算,这样可以将复杂的配置任务拆分成一个个原子任务,降低配置任务的实现难度.其次,要对配置任务有充分的了解,实现一个配置任务不仅要考虑设备如何实现这个功能,还要考虑这个功能在连通性、安全性、性能及可靠性上的需求,以及潜在的可能产生的对无关设备的影响等,所以,需要管理员在进行配置任务分析的时候要做好细致的调研和规划工作.最后,要选择合适的方法验证描述是完备的,需要建立完备性验证模型,该模型应该分为两个层次:一是对配置对象建模,用以描述配置对象的组成及拓扑连接关系,保证配置对象描述的一致性;二是对设备行为进行建模,用以描述配置对象的行为规范,保证设备行为描述的一致性.其中,配置对象可能是一个子网或者一个服务,包含了一个或者多个配置实体(交换机或路由器),每一个实体包含了多个配置端口,端口之间存在连接关系.另外,还需要定义一些函数,用以描述配置实体的输入和输出等.设备行为表示配置实体的约束规范,一个实体实现形式可能会有多种方法,那么可以表示为一个实现空间,配置对象包含很多配置实体,每一个配置实体都对应一个实现空间,最终的配置方案不仅仅是在每一个实现空间内取一种方法,而是要考虑各实体方法之间的约束关系.也就是说,能够找到一个满足该实体的方法,也能够找到满足其他所有实体的方法,并且作为整体来看,方法是有效的.

© 中国科学院软件研究所 http://www.jos.org.cn

5.3 基于运行反馈的动态配置能力

如果想要对网络中某台设备进行升级,则需要将其从网络中断开.如果域内路由协议是 OSPF,首先需要将该设备接口的 OSPF *cost* 提高,让路由重新收敛,以便所有的路径都不会使用该设备.但是,这个操作可能会导致某些设备的链路因超负载而丢包,影响网络的实际运行.因此,如果自动配置系统拥有运行监测能力,则可以通过对链路流量的实时采集,进行统计并推算出该配置对其他链路造成的影响,从而提前向网络运行人员反馈.另外,自动配置系统根据反馈信息自动地执行一些动态调优算法,调优算法的选择要根据具体的需求而定,调优结果会在相应的网络配置项上执行.动态反馈机制将调整后的运行信息反馈给自动配置系统,系统会根据反馈结果判断调整是否有效果、是否需要继续调整、是否有误、是否需要恢复到之前的状态等,经过反复的调整,使得网络性能达到最优.对于某些设备中的关键配置参数,如某些机制需要使用表,如 MAC 转发表、某些安全机制使用的过滤表等,为了提高查表速度,这些表通常采用硬件方式来实现.但是,硬件资源是有限的,对于表中每一项如何配置合理的生存时间,经常困扰着运行人员.如果自动配置框架可以基于运行监测数据进行合理的估算并动态地进行调节,将大大减少运行人员的工作量,并使设备运行在较好的状态上.

5.4 有效的配置验证

当配置生成以后,如果直接运用到实际运行的网络,造成网络中断或异常的可能性很大,一个好的自动配置系统应该具备配置验证的能力.配置验证方法分为模拟验证和实际验证(详见第 3.1 节和第 3.2 节).配置方案生成以后,优先选择的方法是模拟验证,因为该方法只是借助虚拟或者实验性的网络,并不会对实际网络产生影响,模拟验证相对于实际验证代价较小.但是模拟验证还不能保证配置方案是完全正确的,因为模拟验证并不能模拟完整的实际网络,它更多地关注配置功能是否已经实现,对于该功能对现有网络环境的影响以及潜在的设备之间、参数之间的依赖关系,模拟验证很难做到相应的验证,所以还要结合实际网络的验证.在模拟验证的基础上,实际验证可以更多地关注网络在运行过程中,由于设备之间的依赖关系以及网络环境的变化,配置是否会造成网络异常或中断.实际验证适合长期地对网络进行验证,以保证网络产生尽可能少的配置错误.模拟验证与实际验证结合起来,可以对自动生成的配置方案的正确性进行很好的验证,使得自动配置系统更加可靠.

5.5 搭建通用的互联网自动配置体系架构

为满足CERNET和CERNET2主干网以及清华大学校园网配置管理的需要,我们正在搭建一个相对通用的互联网自动配置体系架构,该架构分为 3 个层次:配置信息管理层、配置自动生成层和配置任务管理层.

(1) 配置信息管理层负责底层配置数据的抽象和管理,主要目标是配置的自动实现.配置信息管理层能够对不同设备中的配置项进行抽象,构建配置项模型和配置模型,实现底层配置命令对上层透明.配置信息管理层还能够对设备的配置信息进行提取、备份、分析和恢复,同时具备网络动态信息反馈的能力;

(2) 配置自动生成层负责中间层配置规则的抽象,主要目标是配置的自动生成.配置自动生成层能够构建配置规则模型,实现配置规则对上层透明.配置自动生成层能够结合实际的网络拓扑信息及动态反馈信息以及配置任务的规则描述,推算出符合条件的配置规则模型,并能够实现与配置管理层之间的数据转换.配置自动生成层还支持配置正确性的验证;

(3) 配置任务管理层负责顶层配置任务的抽象,主要目标是配置任务的自动解析.配置任务管理层能够构建配置策略模型,实现配置策略对管理员透明.配置任务管理层还能够进行配置任务描述完备性的验证以及根据配置任务的解析结果智能地推荐一套完整的自动配置流程.

6 总 结

互联网配置管理问题是网络管理的 5 大问题之一,但是传统的配置方法不仅容易出现错误配置,导致网络中断或异常,还会给网络安全带来隐患.所以,很多学者致力于研究互联网自动配置,以解决传统配置方法带来的问题.本文对互联网自动配置问题作了较为深入的探讨:(1) 对互联网自动配置进行了概述.以配置 VPN 服务

© 中国科学院软件研究所 http://www.jos.org.cn

为例,直观地展示了配置的过程,阐述了我们在该领域的一些实际工作;(2) 对当前的互联网自动配置研究按照配置自动生成、配置验证和配置自动实现的顺序分别进行了分类总结和分析评价;(3) 总结了当前互联网自动配置研究存在的几个问题,并针对每一个问题提出了解决思路.

目前,关于互联网自动配置问题的研究缺少理论性和系统性的指导,学术界关注的是基于 Model Finder 思想的配置自动生成,工业界关注的是基于 NETCONF 协议配置的自动实现.如果能将这两部分有效地结合起来,将会大大增加配置的自动化程度.综上,希望我们对互联网自动配置的分析和评述能为该领域的研究者提供一些有益的启示.

致谢 感谢清华大学信息网络运行和管理中心的老师和同学的关心与帮助,与诸位的讨论使我们受益匪浅.

References:

[1] Labovitz C, Ahuja A, Jahanian F. Experimental study of Internet stability and backbone failures. In: Proc. of the 29th Annual Int'l Symp. on Fault-Tolerant Computing (FTCS). Washington, 1999. 278−285. [doi: 10.1109/FTCS.1999.781062]

[2] Markopoulou A, Iannaccone G, Bhattacharyya S, Chuah C-N, Diot C. Characterization of failures in an IP backbone. In: Proc. of the 23rd Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM). Hong Kong, 2004. 2307−2317. [doi: 10.1109/INFCOM.2004.1354653]

[3] Oppenheimer D, Ganapathi A, Patterson DA. Why do Internet services fail, and what can be done about it. In: Proc. of the 4th on USENIX Symp. on Internet Technologies and Systems (USITS). Seattle, 2003. 1−15. http://dl.acm.org/citation.cfm?id=1251461

[4] Feamster N, Balakrishnan H. Detecting BGP configuration faults with static analysis. In: Proc. of the 2nd Conf. on Symp. Networked Systems Design & Implementation (NSDI). Boston, 2005. 43−56. http://dl.acm.org/citation.cfm?id=1251207

[5] Mahajan R, Wetherall D, Anderson T. Understanding BGP misconfiguration. In: Proc. of the ACM Special Interest Group on Data Communication (SIGCOMM) on Applications, Technologies, Architectures, and Protocols for Computer Communications. Pittsburgh, 2002. 3−16. [doi: 10.1145/633025.633027]

[6] Kerravala Z. As the value of enterprise networks escalates, so does the need for configuration management. The Yankee Group, 2004. http://www.cs.princeton.edu/courses/archive/spr12/cos461/papers/Yankee04.pdf

[7] Theophilus B, Aditya A, David M. Unraveling the complexity of network management. In: Proc. of the 6th USENIX Symp. on Networked Systems Design and Implementation (NSDI). Boston, 2009. 335−348. http://dl.acm.org/citation.cfm?id=1559000

[8] Theophilus B, Aditya A, Aman S. Demystifying configuration challenges and trade-offs in network-based ISP services. In: Proc. of the ACM Special Interest Group on Data Communication (SIGCOMM) on Applications, Technologies, Architectures, and Protocols for Computer Communications. Toronto, 2011. 302−313. [doi: 10.1145/2018436.2018471]

[9] Caldwell D, Gilbert A, Gottlieb J, Greenberg A, Hjalmtysson G, Rexford J. The cutting EDGE of IP router configuration. ACM SIGCOMM Computer Communication Review, 2004,34(1):21−26. [doi: 10.1145/972374.972379]

[10] ISO/IEC 7498-4 1989. 2006. http://www.iso.org/iso/catalogue_detail.htm?csnumber=14258

[11] Sanchez L, McCloghrie K, Saperia J. Requirements for configuration management of IP-based networks. RFC 3139, 2001.

[12] Yang JH, Jiang N, An CQ, Li FL. A formal approach to the design and implementation of configuration strategy automation for switch network. Journal of Tsinghua University, 2012,53(8):1041−1048 (in Chinese with English abstract).

[13] Li FL, Yang JH, An CQ, Wu JP, Wang SY, Jiang N. CSS-VM: A centralized and semi-automatic system for VLAN management. In: Proc. of the IFIP/ IEEE Int'l Symp. on Integrated Network Management (IM). Ghent, 2013. 623−629. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6573042

[14] Jiang N, An CQ, Yang JH. Adaptive tuning of operation parameters for automatically learned filter table. In: Proc. of the 13th Asia-Pacific Network Operations and Management Symp. (APNOMS). Taipei, 2011. 1−8. [doi: 10.1109/APNOMS.2011.6077039]

[15] Böehm H, Feldmann A, Maennel O, Reiser C, Volk R. Network wide inter-domain routing policies: Design and realization. In: Proc. of the 34th Conf. on North American Network Operators' Group Meeting. Seattle, 2005.

[16] Gottlieb J, Greenberg A, Rexford J, Wang J. Automated provisioning of BGP customers. IEEE Network Magazine, 2003,17(6): 44−55. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=1248660

[17] Enck W, McDaniel P, Sen S, Sebos P, Spoerel S, Greenberg A, Rao S, Aiello W. Configuration management at massive scale: System design and experience. In: Proc. of the USENIX Annual Technical Conf. (USENIX). Santa Clara, 2007. 73−86. http://dl.acm.org/citation.cfm?id=1364391

© 中国科学院软件研究所 http://www.jos.org.cn

[18] Chen X, Mao ZM, Van der Merwe J. PACMAN: A platform for automated and controlled network operations and configuration management. In: Proc. of the 5th Int'l Conf. on Emerging Networking Experiments and Technologies (CoNext). Rome, 2009. 277−288. http://dl.acm.org/citation.cfm?id=1658971

[19] Murata T. Petri nets: Properties, analysis and applications. Proc. of the IEEE, 1989,77(4):541−580. [doi: 10.1109/5.24143]

[20] Van der Aalst WM. The application of petri nets to workflow management. The Journal of Circuits, Systems and Computers, 1998, 8(1):21−66. [doi: 10.1142/S0218126698000043]

[21] Chen X, Mao Y, Mao ZM, van der Merwe J. Declarative configuration management for complex and dynamic networks. In: Proc. of the 6th Int'l Conf. on Emerging Networking Experiments and Technologies (CoNext). Philadelphia, 2010. 61−72. [doi: 10.1145/1921168.1921176]

[22] Ramakrishnan R, Ullman JD. A survey of research on deductive database systems. Journal of Logic Programming, 1993,23(2): 125−149.

[23] Narain S. Network configuration management via model finding. In: Proc. of the 19th Conf. on Large Installation Systems Administration (LISA). San Diego, 2005. 155−168. http://dl.acm.org/citation.cfm?id=1251165

[24] Alloy. http://alloy.mit.edu/

[25] Jackson D. Software Abstractions: Logic, Language, and Analysis. MIT Press, 2006.

[26] Narain S, Levin G, Kaul V, Malik S. Declarative infrastructure configuration synthesis and debugging. Journal of Network and Systems Management, 2008,16(3):235−258. [doi: 10.1007/s10922-008-9108-y]

[27] SWI-Prolog. http://www.swi-prolog.org/

[28] Bratko. Prolog Programming for Artificial Intelligence. Addison-Wesley Longman Publishing Co., Inc., 1990.

[29] Thomas D, Wouter J. PoDIM: A language for high-level configuration management. In: Proc. of the 21st Large Installation System Administration Conf. (LISA). 2007. 261−273. http://dl.acm.org/citation.cfm?id=1349447

[30] Kodkod. http://alloy.mit.edu/kodkod/

[31] Huan L, Dan O. Remote network labs: An on-demand network cloud for configuration testing. SIGCOMM Computer Communication Review, 2010,40(1):83−91. [doi: 10.1145/1672308.1672324]

[32] Goodell G, Aiello W, Griffin T, Ioannidis J, McDaniel P, Rubin A. Working around BGP: An incremental approach to improving security and accuracy of interdomain routing. In: Proc. of the Network and Distributed System Security Symp. (NDSS). San Diego, 2003. 47−61. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.3884

[33] Al-Shaer ES, Hamed HH. Discovery of policy anomalies in distributed firewalls. In: Proc. of the 23rd Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM). Hong Kong, 2004. 2605−2616. [doi: 10.1109/INFCOM.2004.1354680]

[34] Yuan LH, Mai JN, Su ZD, Chen H, Chuah C-N, Mohapatra P. FIREMAN: A toolkit for FIREwall modeling and analysis. In: Proc. of the Conf. on 2006 IEEE Symp. on Security and Privacy (S&P). Oakland, 2006. 199−213. [doi: 10.1109/SP.2006.16]

[35] Caldwell D, Lee S, Mandelbaum Y. Adaptive parsingof router configuration languages. In: Proc. of the Internet Network Management Workshop (INM). Orlando, 2008. 1−6.

[36] Hamed HH, Al-Shaer ES, Marrero W. Modeling and verification of IPSec and VPN security policies. In: Proc. of the 13th IEEE Int'l Conf. on Network Protocols (ICNP). Boston, 2005. 259−278. [doi: 10.1109/ICNP.2005.25]

[37] Roscoe T, Hand S, Isaacs R, Mortier R, Jardetzky P. Predicate routing: Enabling controlled networking. ACM SIGCOMM Computer Communication Review, 2003,33(1):65−70. [doi: 10.1145/774763.774773]

[38] Xie GG, Zhan J, Maltz D. On static reach ability analysis of IP networks. In: Proc. of the 24th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM). Miami, 2005. 2170−2183. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1498492

[39] Mai H, Khurshid A, Agarwal R, Caesar M, Godfrey PB, King ST. Debugging the data plane with anteater. In: Proc. of the ACM Special Interest Group on Data Communication (SIGCOMM) on Applications, Technologies, Architectures, and Protocols for Computer Communications. 2011. 290−301. [doi: 10.1145/2018436.2018470]

[40] Gogineni H, Greenberg A, Maltz DA, Ng TSE, Yan H, Zhang H. MMS: An autonomic network-layer foundation for network management. IEEE Journal on Selected Areas in Communications, 2008,28(1):15−27. [doi: 10.1109/JSAC.2010.100103]

[41] Strassner J. How policy empowers business-driven device management. In: Proc. of the 3rd Int'l Workshop on Policies for Distributed Systems and Networks. Monterey, 2002. 214−217. [doi: 10.1109/POLICY.2002.1011311]

[42] Enns R. NETCONF configuration protocol. RFC 4741, 2006.

[43] Wasserman M, Goddard T. Using the NETCONF configuration protocol over secure shell (SSH). RFC 4742, 2006.

© 中国科学院软件研究所 http://www.jos.org.cn

[44] Goddard T. Using NETCONF over the simple object access protocol (SOAP). RFC 4743, 2006.
[45] Lear E, Crozier K. Using the NETCONF protocol over blocks extensible exchange protocol (BEEP), RFC 4744t, 2006.
[46] Bjorklund M. YANG: A data modeling language for the network configuration protocol (NETCONF). RFC 6020, 2010.
[47] Elbadawi K, Yu J. Improving network services configuration management. In: Proc. of the 20th Int'l Conf. on Computer Communications and Networks (ICCCN). Maui, 2011. 1−6. [doi: 10.1109/ICCCN.2011.6006050]
[48] NETCONF data modeling language (netmod). http://datatracker.ietf.org/wg/netmod/
[49] Lengyel B, Bjorklund M. Partial lock remote procedure call (RPC) for NETCONF. RFC5717, 2009.
[50] Seottand M, Bjorklund M. YANG module for NETCONF monitoring. RFC6022, 2010.
[51] Tail-f systems ConfD. http://www.tail-f.com/products/confd/
[52] Chisholm S, Trevino H. NETCONF event notifications. RFC, 2008.
[53] Xu H, Ai X, Xiao DB. New generation network management based on the NETCONF protocol. Journal of Beijing University of Posts and Telecommunications, 2009,32(S1):10−14 (in Chinese with English abstract).
[54] Xiao DB, Chen LM, Ai X. Research and implement on next generation network configuration protocol NETCONF. Journal of Huazhong Normal University, 2008,42(4):530−534 (in Chinese with English abstract).
[55] Liang WM. Research and implementation of an ITIL-based NETCONF network configuration management system [MS. Thesis]. Wuhan: Huazhong Normal University, 2011 (in Chinese with English abstract).
[56] Chang YN. Research and implementation of YANG-based NETCONF data modeling [MS. Thesis]. Wuhan: Huazhong Normal University, 2009 (in Chinese with English abstract).
[57] Chen LM. Research of NETCONF-based network configuration management agent [MS. Thesis]. Wuhan: Huazhong Normal University, 2009 (in Chinese with English abstract).

附中文参考文献:

[12] 杨家海,姜宁,安常青,李福亮.基于形式化描述的交换机网络自动配置策略的设计与实现.清华大学学报,2012,53(8):1041−1048.
[53] 徐慧,艾翔,肖德宝.基于 NETCONF 协议的新一代网络管理.北京邮电大学学报,2009,32(S1):10−14.
[54] 肖德宝,陈历淼,艾翔.下一代网络配置管理协议 NETCONF 的研究与实现.华中师范大学学报,2008,42(4):530−534.
[55] 梁伟民.基于 ITIL 的 NETCONF 网络配置管理系统的研究与实现[硕士学位论文].武汉:华中师范大学,2011.
[56] 常亚楠.基于 YANG 语言的 NETCONF 网络管理数据建模的研究与实现[硕士学位论文].武汉:华中师范大学,2009.
[57] 陈历淼.基于 NETCONF 的网络配置管理代理的研究[硕士学位论文].武汉:华中师范法大学,2009.

李福亮(1986－),男,辽宁葫芦岛人,学士,CCF 学生会员,主要研究领域为网络测量,网络配置管理,网络配置验证.
E-mail: lfl09@mails.tsinghua.edu.cn

安常青(1970－),女,副研究员,CCF 会员,主要研究领域为网络管理与测量,下一代互联网关键技术.
E-mail: acq@cernet.edu.cn

杨家海(1966－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机网络,网络管理与测量,协议工程学.
E-mail: yang@cernet.edu.cn

姜宁(1985－),男,硕士,主要研究领域为网络测量,网络配置管理.
E-mail: jiangning85@126.com

吴建平(1953－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机网络体系结构,网络协议测试.
E-mail: jianping@cernet.edu.cn

© 中国科学院软件研究所 http://www.jos.org.cn