

## 基于意图的网络(IBN)研究综述

李福亮<sup>1</sup>, 范广宇<sup>1</sup>, 王兴伟<sup>1</sup>, 刘树成<sup>2</sup>, 谢坤<sup>3</sup>, 孙琼<sup>4</sup>

<sup>1</sup>(东北大学 计算机科学与工程学院, 辽宁省 沈阳 110819)

<sup>2</sup>(华为技术有限公司, 深圳 518129)

<sup>3</sup>(北京邮电大学 软件学院, 北京 100876)

<sup>4</sup>(中国电信 北京研究院, 北京 102209)

通讯作者:王兴伟, E-mail: wangxw@mail.neu.edu.cn



**摘要:** 随着互联网规模不断增大,网络管理和运维变得极其复杂,网络自治成为未来网络发展的趋势,基于意图的网络(Intent-Based Networking,IBN)应运而生.本文首先从IBN的定义入手,介绍学术界及产业界对IBN范畴及体系结构的描述,并概述IBN实现的闭环,包括意图获取、意图转译、策略验证、意图下发与执行、实时反馈及优化;其次,按照IBN闭环,详细阐述IBN关键技术的研究现状.随后,举例说明IBN在网络测量和网络业务编排中的应用.最后,展望未来研究工作并总结全文.

**关键词:** 互联网; 基于意图的网络; 意图获取; 意图转译; 策略验证

**中图分类号:** TP311

### State-of-the-Art Survey of Intent-Based Networking

LI Fu-Liang<sup>1</sup>, FAN Guang-Yu<sup>1</sup>, WANG Xing-Wei<sup>1</sup>, LIU Shu-Cheng<sup>2</sup>, XIE Kun<sup>3</sup>, SUN Qiong<sup>4</sup>

<sup>1</sup>(School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China)

<sup>2</sup>(Huawei Technologies Co.,Ltd, Shenzhen 518129, China)

<sup>3</sup>(School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)

<sup>4</sup>(China Telecom Beijing Research Institute, Beijing 102209, China)

**Abstract:** With the increasing scale of the Internet, network management and operations become extremely complex. Intent-Based Networking (IBN) emerges when network autonomy becomes a major trend of future network. First of all, this paper gives the definition of IBN, and describes the IBN category and architecture in academia and industry. The implementation closed-loop of IBN is also summarized, including intention acquisition, intention translation, policy verification, intention distribution and execution, real-time feedback and optimization. Then, key technologies of IBN implementation are elaborated in detail according to the IBN closed-loop. In addition, examples are given to illustrate the IBN-based applications from the aspects of network measurement and network service orchestration. Finally, the future research work is prospected and the whole paper is concluded.

**Key words:** Internet; intent-based networking; intent acquisition; intent translation; policy verification

网络规模不断扩大,网络流量迅速增长,得网络管理变得极其复杂且容易出错.微软云每小时有超过 10000 个内核受到配置错误的影响<sup>[1]</sup>.Veriflow 邀请了全球 315 名在 IT 公司工作的网络技术专家,其中 97%的人认为人为操作错误会导致网络故障发生,74%的人所在的公司一年中会多次因网络变动影响公司业务<sup>[2]</sup>.然而,现在大多数的网络管理员仍然使用 Ping<sup>[3]</sup>、TraceRoute<sup>[4]</sup>等管理工具排查网络故障,极大降低了网络故障恢复效率.为应对这些问题,新的网络架构、组网方式以及网络管理方法已经成为了研究的热点.

\*基金项目: 十三五国家重点研发计划课题(2018YFB1800201, 2019YFB1802600), 国家自然科学基金面上项目(61872073)

收稿时间: 2020-01-06; 修改时间: 2020-03-09; 采用时间: 2020-05-20; jos 在线出版时间: 2020-06-08

软件定义网络 (software-defined networking, SDN) [5] 为网络全局感知和智能管理带来了新的机遇。SDN 创新性地将控制平面与数据平面解耦 [6], 允许用户在应用平面对网络进行编程, 提高网络管理运维效率。然而, 管理员在对网络编程时仍然需要了解相关的底层实现细节 [7], 这极大地限制了非专业人员对于网络行为的感知和控制。为了支持业务的敏捷性, 网络需从一个静态资源系统演变成一个能一如既往地满足商业目标的动态系统, 一种新兴的网络范式 IBN (Intent-Based Networking) [8] 应运而生。IBN 是一种全新的网络模型, 该模型中定义了意图, IBN 通过分析用户意图将意图转译为相应的网络策略, 最终实现网络感知和控制策略的自动化部署 [9]。意图是 IBN 的核心, IBN 的运行过程都与意图紧密相关。用户只需要描述想要的结果, 而不用描述如何实现 [10], IBN 就可自动地实现用户意图, 并能够持续监控网络状态信息, 判断用户意图是否实现。如果检测到意图没有实现, 则系统将通过人工智能和自动化技术对用户意图进行重新转译和优化, 最大程度满足用户的意图。在架构上 IBN 可以看成是一种更高级的 SDN 或者是 SDN 向前演变的一种形式, 因为一些 SDN 控制器 (ONOS、OpenDaylight 等) 已经开放了北向用户意图接口; 然而, SDN 的本质还是使用集中式控制器来驱动整个网络, 使用控制器与网络交互、控制网络, 意图接口只是它的一个应用, 而 IBN 始终由“意图”驱动, 网络的运行都是围绕着“意图”进行, 网络运转完全自动化。因此, IBN 可以看作是高级、智能的 SDN, 但是 IBN 不限于 SDN, 它是一种更加灵活、智能、自动化的网络模型。

总的来说, 可以将基于意图的网络概括为一种在掌握网络全局状态的前提下, 基于人类业务意图去自动搭建和操作的闭环网络架构。根据基于意图的网络的定义, 基于意图的网络包含几个关键步骤, 首先是能够获取到用户所提出的网络需求即意图, 将接收到的意图转译成网络策略, 并能根据当前网络的状态验证策略的可执行性, 之后将通过验证的策略下发到实际网络中, 此外, 系统还要实时地监控网络状态, 确保用户意图正确实现, 并将结果反馈给用户。IBN 能够使得网络管理的效率得到极大的提升 [11], 降低企业网络运维成本。因此, IBN 可以更好地协助网络运营商和企业去控制基础网络设施, 被认为是实现网络自治的关键技术手段。自 2013 年起, 工业界和学术界先后制定了多项 IBN 相关标准, 发表了多篇学术论文, 推动了 IBN 相关研究的快速发展, 使其逐渐成为近年来的研究热点。

本文第 1 节首先论述 IBN 的定义及体系结构, 并概述 IBN 的实现闭环, 第 2 节详细描述 IBN 实现过程中的关键技术及研究现状, 第 3 节探讨 IBN 在不同场景下的应用, 第 4 节对 IBN 的未来研究工作进行展望。最后对本文进行总结。

## 1 IBN 的定义及体系结构

IBN 可以分析用户的意图, 自动化地完成网络管理工作, 减少网络管理人员的工作负担, 是未来网络发展的趋势。本节首先介绍 IBN 的定义及其诞生背景, 说明 IBN 成为未来网络发展方向的必然性。接下来论述了 IBN 的体系结构, 概述 IBN 实现的优化闭环。

### 1.1 IBN 的定义

意图是 IBN 的核心概念, IBN 的实现过程都围绕着意图进行, 所以要弄清 IBN, 首先要了解意图 (intent) 的具体含义。《现代汉语词典》中这样定义意图: “希望达到某种目的的想法”, 从这个角度出发可以把意图理解为一种描述用户目的主观想法。因而把意图引入到网络中, 就可以理解为用户希望达到某种网络目的, 进一步说就是用户希望网络达到某种状态。在 2013 年, IFIP 会议提出了意图驱动的网络管理抽象, 为网络连接配置方式的改变提供了范式转变的基础 [12]。2015 年, ONF 发布了一篇名为《Intent: Don't Tell Me What to Do! Tell Me What You Want》[13] 的标准草案, 其中提出: “在基于意图的网络中, 智能软件将决定如何把意图转化为针对特定基础设施的配置, 从而使网络以期望的方式运行”。2017 年, 思科在《向基于意图的网络迈进》白皮书 [14] 中也给出 IBN 的定义: 网络团队可以用简明的语言描述想要完成的工作 (意图), 然后网络就能够将此意图转化为众多策略, 这些策略将利用自动化功能在复杂的异类环境中建立适当的配置和设置变更。此外, 相关文献中也提出了 IBN 的定义 [9][11][15], 虽然形式不同, 但是本质却一致, “为用户提供一个简单的意图接口, 用户通过这个接口描述他们期盼网络达到的状态而不用描述如何实现这个状态, 网络将自动完成配置, 满足用户的需求”。

IBN 的相关标准逐步制定.互联网工程任务组(Internet Engineering Task Force, 简称 IETF)在 2014-2015 年期间成立了 SUPA (Simplified Use of Policy Abstractions)工作组,在此期间 ECA (Event-Condition-Action) 和 intent(当时叫 declarative)策略都作为 SUPA 的一部分,不过后来讨论认为 intent 较为偏远,暂时从 SUPA 范围中删除 intent,但保留了未来扩展的计划.2018 年,SUPA 架构发布为 rfc8328<sup>[16]</sup>,除了策略架构,还探讨了策略-业务-资源的关系,为未来网络对意图的支持做了铺垫.此外,互联网研究专门工作组(Internet Research Task Force,简称 IRTF)的 NMRG (Network Management Research Group)工作组也在努力制定相关标准,NMRG 在 RFC7575<sup>[17]</sup>和 RFC7576<sup>[18]</sup>着手于网络自动化的研究,并将意图定义为“一种用来运维网络的抽象的、高层次的策略”.此外,在 RFC8316<sup>[19]</sup>中描述了一个网络自动化的应用场景,即对于服务级别协议 (Service Level Agreement,SLA) 的监控.在国内,IBN 的相关标准制定与研究工作也在如火如荼地进行,中国通信标准化协会 (China Communications Standards Association,CCSA)在 2019 年也开始了 IBN 的相关标准制定,并逐步展开各项工作,提出 2025 年基本实现 IBN 的目标.综上所述,本文给出 IBN 的具体定义:基于意图的网络以“意图”作为核心驱动网络运行,是一种实现自动搭建和操作的闭环网络架构.IBN 通过将用户需求抽象为意图,而不用描述如何实现,网络将自动地分析用户意图、验证策略、自优化策略及下发部署策略,并且在意图实现后持续监测网络状态,在出现异常后自动调整网络策略使网络能够一如既往地满足用户意图.

从技术层面看,IBN 并不是一种新技术,而是一种利用已有技术的创新组合而产生的一种新的理念,通过意图的概念为网络管理提供一个更高级的抽象.IBN 也并不是凭空出现,随着 SDN 的提出与发展,以及人工智能技术的不断成熟,网络自动化程度逐步提高,IBN 也应运而生.IBN 的实现步骤分为:意图的获取、意图的转译、策略的验证、意图的下发与执行以及网络状态信息的实时反馈.其中,意图的获取和转译需要用到自然语言解析识别以及 AI 智能策略选取技术,策略验证需要用到网络自动化验证<sup>[1]</sup>和形式化验证<sup>[20]</sup>的方法,策略的下发可采用 OpenFlow<sup>[21]</sup>协议、P4<sup>[22]</sup>数据平面编程语言等技术,最后还需要实时感知网络状态并分析意图是否成功实现.

## 1.2 IBN的体系结构

目前业界还没有统一的 IBN 标准体系结构,但是不同的 IBN 参考体系结构的核心思想是一致的,因此总结各个组织的工作给出 IBN 的体系结构,如图 1 所示.IBN 的体系结构参照 SDN 的分层结构从上到下(或称由北向南)分为应用层、意图层和网络层.应用层与意图层通用北向意图输入接口进行通信,北向意图输入接口主要负责用户意图的输入以及意图引擎向上反馈相关信息.意图层与网络层通过南向策略下发接口进行通信,南向策略下发接口主要负责将经过验证后的网络策略下发到实际的网络基础设施中.

IBN 的应用层主要负责收集用户以各种形式输入的意图,并把各种形式的意图统一为标准的形式,例如把各种形式(语音输入、GUI 输入等)都统一为自然语言的形式.应用层面向的用户包括但不限于普通用户、网络管理员等.意图层是 IBN 的核心,是驱动 IBN 运行最为关键的因素.意图层的核心部件是意图引擎,主要负责用户意图的解析与转译,然后根据当前的网络状态信息对网络策略进行验证,并对策略进行优化进而下发到实际网络设施中.IBN 的网络层是负责各种数据包转发的网络元素的集合.理想情况下,IBN 可以在不同的网络中同时进行意图的部署,然而由于各种因素的限制,目前大多数的研究工作仅在单一的网络环境下实现了 IBN.

IBN 可以为网络的管理与维护带来极大的方便,并降低各个公司或者网络运营商的运营成本.然而,由于 IBN 从根本上改变了传统网络的设计和运营方式,它带来的风险有可能成为运营商发展的障碍,因此 GARTNER 认为 IBN 在 2020 前不会成为主流.不过,目前已经有许多公司和团体开始了 IBN 落地的实践.OpenDayLight<sup>[23]</sup>提供开放的北向接口,并使用 YANG 数据建模语言为服务和数据建模,提供了一些与北向接口有关的组件,目的是提供一种通用的抽象的策略语法,让用户方便地描述自己意图.NEMO<sup>[24]</sup> (Network Model),是华为在推出的一种基于意图的北向接口语言;NEMO 基于 DSL<sup>[25]</sup> (Domain Specific Language) 的抽象网络模型以及结论性的操作模式,使得它只提供很少的关键词和表达语句就可以描述出用户或者应用对于网络的需求.思科的 DNA 项目<sup>[26]</sup>也一直致力于意图驱动网络的落地,满足企业对全数字化转变的需求,为企业更安全地提供更好的体验.此外,欧洲发起的“5G EVE 5G European Validation platform for Extensive trials”

项目明确提出了要使用基于意图或者其他高级别抽象的接口。

总体来说,虽然 IBN 的体系结构目前还没有统一的形式,但是从目前的研究到实现工作均围绕着意图进行,意图可以消除高级别管理需求与低级别配置语言之间的鸿沟,提升网络管理的灵活性和智能化程度。

应用层

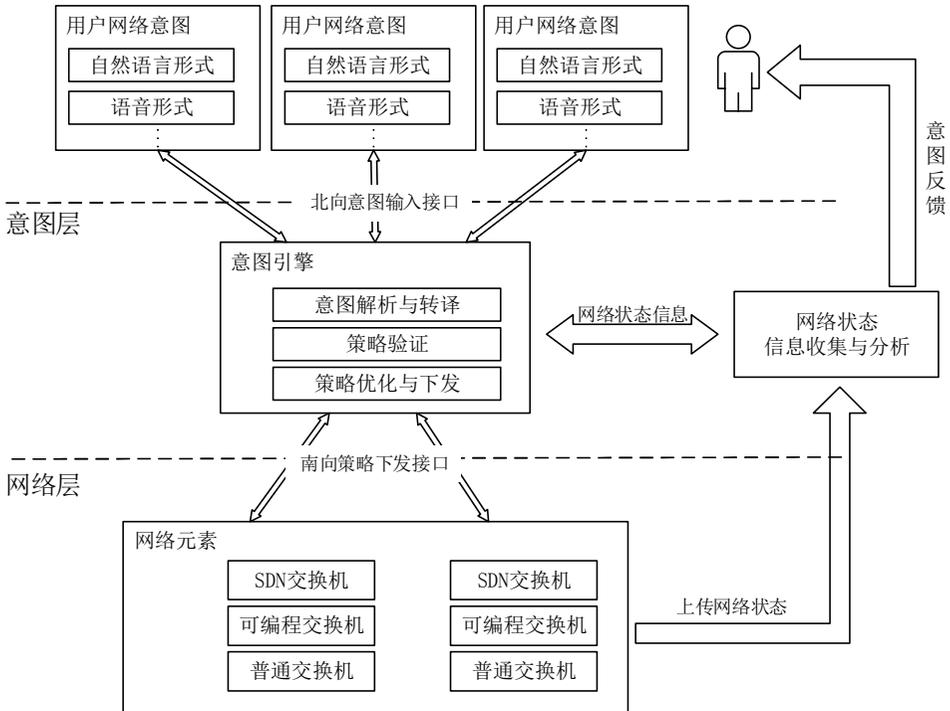


Fig.1 IBN architecture

图 1 IBN 体系结构

## 2 IBN 关键技术研究

IBN 的实现按照意图的获取、意图的分析转译、策略的验证、策略的下发与执行以及实时反馈的顺序执行,形成一个优化闭环,如图 2 所示.在 IBN 的实现过程中,对于意图的处理、分析等工作需要相关的技术支持并对接口进行设计.本节根据 IBN 实现过程的全生命周期,阐述每一步所用到的技术以及已取得的相关研究成果.

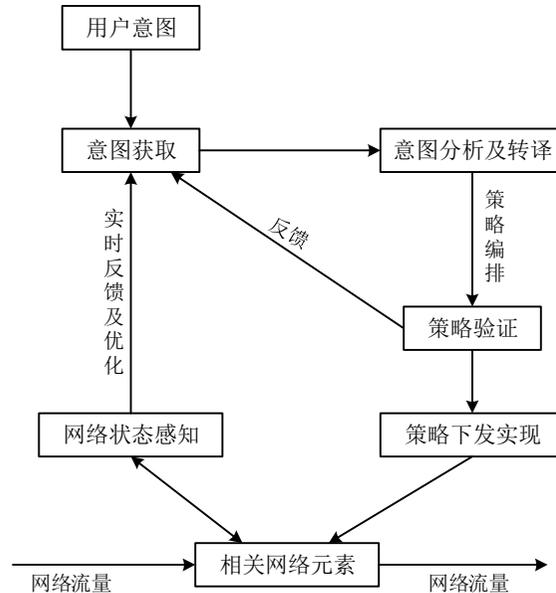


Fig.2 Implementation process of IBN

图 2 IBN 的实现流程

## 2.1 意图获取

在 IBN 中意图是核心概念,意图以一种声明形式的描述用户想要网络所达到的状态,而不用描述如何实现意图<sup>[9]</sup>,因此用户的意图需要作为 IBN 的输入.目前,用户意图实现的自动化程度与意图实现的场景成反比关系,也就是说一个意图要想完全自动化实现,那么它可能就只适用于一个特定的场景.IBN 的自动化程度根据意图抽象的级别从低到高分为 4 个阶段,a.) 全面自动化网络:能够实现绝大部分业务发放、网络部署和维护的自动化,并实现较全面地网络状态感知与局部地机器决策;b.) 自优化网络:能够实现深度地网络状态感知,自动网络控制,满足用户网络意图;c.) 部分自治网络:在特定的环境,不一定需要人参与决策,能够实现自主调整网络状态;d.) 全面自治网络:在不同网络环境、网络条件,网络均能自动适应、自主调整,满足用户的意图.IBN 可按照这 4 个阶段的顺序逐步向前演进.

用户意图的表达形式有很多,最普遍的是自然语言的形式,此外还有用户的语音输入或者相关的 GUI 输入等.目前,针对意图获取的工作还没有一个统一的标准,对于自然语言形式的意图输入,目前大多研究使用语义学<主语,谓语,宾语,修饰语>形式<sup>[9]</sup>的表示意图,其中的每一项都可进行嵌套表示以丰富语义,修饰语可以省略.例如当描述意图“I want to connect A to B.”时,可以使用<A,connect,B>表示意图并作为输入发送到网络.此外,在 SDN 中基于意图的北向接口也受到了广泛的关注,基于意图的北向接口<sup>[27]</sup>致力于统一意图的表达形式,并提供统一的接口供应用调用.ONOS<sup>[28]</sup>控制器已经率先提出基于意图的北向接口<sup>[29]</sup>,用于屏蔽服务运行的复杂性,允许应用向网络请求服务而不需要了解服务运行的具体细节.不过目前有很多研究使用高级别描述性编程语言来表示意图,例如,Frenetic<sup>[30]</sup>是一种针对 SDN 北向接口的函数型编程语言,它可以隐藏 SDN 中网络元素的各种底层细节,提供类似 SQL 的语法结构对 SDN 应用进行开发,它的重点在于使用高级别抽象的形式在描述报文的转发规则.NetKAT<sup>[31]</sup>同样也是一种面向 SDN 的函数型编程语言,它提供简单的原语来描述下层交换机的转发行为以及全网的结构,并且它基于严格的数学证明从而为网络的验证工作提供方便<sup>[32]</sup>.另外,JINJING<sup>[33]</sup>设计了一种名为“LAI”的意图语言用来生成并维护 ACL (Access Control List),它提供了 scope、modify、check 等几种原语描述用户对于内网 ACL 的意图需求,例如当需要在 ACL 上添加一条“A 可以连通 B”的意图,便可以使用“scope A\*,B\* allow A\*,B\* control A->B check fix”来描述该意图.上 3 种语言都是基于 DSL 设计的,其中 Frenetic 和 NetKAT 更偏向为一种编程语言,需要一定的专业知识,而“LAI”不要太多的专业知识就可以

描述 ACL 意图,虽然相比于使用自然语言描述意图,这 3 种语言的抽象级别较低,但是在特定的环境下它们一样可以具有丰富的语义,完整地描述用户意图.在今后的研究中可以采用由自然语言向 DSL 转译的方法实现用户的意图.总的来说,随着自然语言解析技术的不断发展,用户输入意图的抽象级别可以越来越高,从而实现全面自治网络.

## 2.2 意图转译

意图给出用户期望网络应该达到某种状态的声明,并没有指出如何实现.然而网络在实际运行期间,仍然是按照网络策略进行转发决策、资源分配.因此,需要根据意图中的内容以及当前网络状态将用户意图转化为相应的网络配置策略<sup>[10]</sup>,这就是意图的转译过程.目前用户意图的转译主要采用自然语言处理的方法对意图进行处理,对用户意图进行关键字提取,词法分析,语义挖掘等操作,从而获得用户期望的网络运行状态,并使用智能化的方法生成网络策略.网络策略描述了网络为实现某一目标而执行的动作以及动作执行的顺序,可以把一条网络策略分解为多条子策略方便策略的实现,也可以把多条策略组合成一条复合策略使得网络策略的实现模块化<sup>[34]</sup>.INDIRA<sup>[11]</sup>系统利用自然语言处理和本体论<sup>[35]</sup>的方法在 SDN 控制平面和应用之间实现了意图转译模块,用户以自然语言的形式输入意图,该模块将自然语言处理为 RDF 图并自动生成网络策略或配置命令,从而实现意图的转译,该方法为意图的转译工作提供了指导性的方案.以意图 “I want to connect A to B with high speed now.” 为例,其具体实现过程为:①使用自然语言处理、机器推理等方法处理意图,去掉无效信息 (I、want、to 等),提取关键信息 (connect、A、B 等);②利用处理得到的信息将意图映射为 RDF (Resource Description Framework) 图 (如图 3 所示),并利用网络拓扑信息等将 RDF 图中的参数实例化;③编写程序对 RDF 图进行处理,生成网络策略或配置命令.Liao 等人<sup>[36]</sup>提出了基于模板的策略生成方法,系统预先定义网络策略库,将不同类型的网络策略存储在策略库中,当用户意图输入时,即可使用自然语言处理的方法获取关键信息,并根据策略库映射为相应的网络策略.另外 Riftadi 等人<sup>[37]</sup>创建了一个 P4 的代码模板库,将用户意图映射为 P4 代码.Beigi 等人<sup>[38]</sup>提出了基于案例学习的方法,通过用户意图的不断输入,系统将不断学习意图的特征,并生成相应的策略,但由于该方法需要大量的数据支撑,所以该方法在初始阶段的效率可能不理想.此外,在文献<sup>[7][15]</sup>中也都都在 SDN 环境下设计了一组封装好的意图接口,用户直接输入意图就可自动生成相应的 OpenFlow 流表规则,从而下发到实际网络.

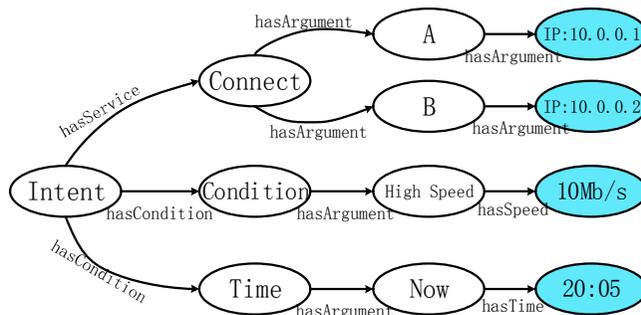


Fig.3 Using RDF Graph to describe intent

图 3 使用 RDF 图描述意图

意图的转译工作是 IBN 中核心的任务,它实现了用户意图到网络策略的转变,但是意图的转译工作并不是一种新的技术,它只是为简化网络管理工作而产生的一种新理念,并且随着自然语言处理技术以及其他智能化方法的进步,意图转译的智能程度也会逐步提升.

## 2.3 策略验证

通过意图的转译工作得到了相应的网络策略,然而这些策略还不能直接下发到实际的数据平面,为了保证

IBN 整体运行的流畅,必须在策略下发之前对策略进行可执行性验证.目前,对于策略的可执行性验证主要考虑资源的可用性、策略的冲突以及策略的正确性这三个方面.对于资源的可用性的验证工作,主要是对当前的网络状态进行感知,维护一个网络状态信息的数据库.在策略下发之前,查看当前策略所需要的网络资源是否可用、是否足够,这样就可以实现资源可用性的验证.对于策略的冲突的验证工作,根据策略匹配域相交关系以及策略执行的动作给出几种策略的冲突关系<sup>[15][39]</sup>: 冗余 (Redundancy)、覆盖 (Shadowing)、泛化 (Generalization)、相关 (Correlation)、重叠 (Overlap).如果检测到待下发策略与网络当前策略存在上述冲突,则要进行冲突的消解.目前冲突的消解方法主要采用设置优先级的方法来消除一些优先级低的策略;通过设置优先级的方法消解冲突可能存在一定的局限性,未来还需要进一步的研究.在 SDN 环境下,主要对 OpenFlow 流表规则进行验证; FlowChecker<sup>[40]</sup>将 OpenFlow 流表编码为二元决策图,在单一状态机内对网络行为建模,进而使用基于二元决策图的符号模型检测及时序逻辑来识别由冲突的流规则引发的配置错误. VeriFlow<sup>[41]</sup>通过对流表规则进行等价类划分以实时检查插入的流规则是否违反网络约束如访问控制规则、路由环路等.最后,对于策略的正确性的验证工作,策略的正确性验证是指策略在下发到实际网络中是否能够按照用户的预期实现;目前对于策略的正确性验证主要采用形式化验证<sup>[1]</sup>的方法;形式化方法是用于将复杂系统建模为数学实体的技术<sup>[42]</sup>.通过构建复杂系统的数学模型,设计人员不仅可以更彻底地验证系统的属性(而不是通过经验测试),还可以使用数学证明作为系统测试的补充,以确保系统行为的正确.形式化方法曾被广泛应用于硬件和软件的验证工作,近些年被引入到网络验证领域并成为研究热点<sup>[1]</sup>;目前的形式化验证的方法主要有模型检查<sup>[43]</sup>、定理证明<sup>[44]</sup>、符号执行<sup>[45]</sup>以及 SAT/SMT 求解器<sup>[46]</sup>这四种.可以把策略通过形式化建模语言(如 datalog<sup>[47]</sup>,alloy<sup>[48]</sup>等)建模为形式化模型,之后把模型和相关条件限制带入到相应的求解器(如 Z3 求解器等)中就可以判断策略是否可以正确实现,不过在形式化验证的过程中要注意状态空间爆炸的问题,防止无法求解.目前,已经有一些形式化验证工作应用于策略正确性的验证,Sooel Son 等人<sup>[49]</sup>提出了“FLOVER”系统,使用模型检查的方法验证 OpenFlow 流规则是否违反网络底层安全属性,从而确保策略可以正确实现.该系统使用 Yices2 作为求解器,将匹配域为 {Src IP:5, Dst IP:6, Src Port:0~19, Dst Port:0~19},动作为 {drop} 的 OpenFlow 流规则转化为如图 4 所示的 Yices2 输入形式,并把网络底层安全属性按照同样的方式进行建模,最后将流规则和安全属性的形式化描述带入到求解器中进行求解,求解器将自动分析它们的关系并返回验证结果.

```

1 (define Cond::(-> packet bool ) (lambda (x::packet)(and
2   (>= (select x Field0) 5) (<= (select x Field0) 5)
3   (>= (select x Field1) 0) (<= (select x Field1) 19)
4   (>= (select x Field2) 6) (<= (select x Field2) 6)
5   (>= (select x Field3) 0) (<= (select x Field3) 19) ) )
6 (define Rules::(->packet mixstate) (lambda (x::packet)
7   (if (Cond x)
8     (mk-record d0::x d1::drop)
9     (mk-record d0::x d1::nomatch) )))

```

Fig.4 Formal description of flow rules

图 4 流规则的形式化描述

策略的验证是 IBN 中一个十分重要的步骤,它关系到 IBN 的稳定程度,确保用户意图可以在不破坏网络正常运行的情况下正确地在数据平面实现.然而,对于策略的验证工作,还没有统一的模块,对于上面所述的三个验证方面,需要将它们统一为单独的模块,并进行优化工作,否则会导致验证的时间过长导致 IBN 的灵活性太低.

#### 2.4 意图下发与执行

在网络策略进行验证过后,IBN 会将网络策略自动地下发到实际的网络基础设施,并对转发设备进行配置,此过程需要对网络进行全局的控制,以从一个单点集中式的意图需求到分布式全局网络配置的转换.目前 IBN 大多在 SDN 环境下实现,这是因为 SDN 的控制器可以收集网络状态信息为 IBN 的验证工作提供方便.因此,在 SDN 环境下,需要把网络策略转换为相应的 OpenFlow 流表规则从而实现用户意图.Lewis 等人<sup>[50]</sup>和 Riftadi 等人<sup>[37]</sup>提出使用可编程交换机来完成策略的下发与执行,使用 P4 编程语言来对可编程交换机进行配置.P4 是一种数据平面编程语言,可以对交换机内部的转发逻辑进行编程;同时 P4 也是一种协议无关的编程语言,也就是

说 P4 可以对交换机进行任意一种协议的编程及下发,可以将网络策略编译为相应的 P4 代码从而实现意图的下发.Sonata<sup>[51]</sup>利用这种思想将用户的网络遥测任务编译为相应的 P4 代码,之后对可编程交换机进行配置从而实现了网络数据的查询.MAFIA<sup>[52]</sup>在 Sonata 的基础上进一步的优化 P4 代码,在可编程交换机上实现网络测量任务.在传统网络的环境下,网络策略可以使用配置综合的方法生成设备的配置文件,从而使网络设备按照预期对数据进行处理.例如,Propane<sup>[53]</sup>和 Propane/AT<sup>[54]</sup>允许用户表明对于流量的高级约束,并使用正则表达式和有限状态自动机表达流量的传输路径,之后利用系统的编译器将这些约束综合为相应路由器的 BGP 配置,不过该方法只支持 BGP 协议.Synet<sup>[55]</sup>将网络配置问题描述为分层次的 Datalog 问题,并使用 SMT 求解器对用户需求进行配置综合从而生成相应的配置文件,它支持静态路由、OSPF、BGP 三种路由协议.NetComplete<sup>[56]</sup>也可以对用户需求进行配置综合并同样支持静态路由、OSPF、BGP 三种路由协议,但是它的求解速度要优于 Synet 并且更易于拓展到大型网络上.

目前,IBN 的实现大多数局限于特定的环境,因此在每种环境下都可能有一些不同的策略下发技术,在未来的工作中可能需要统一相关的技术,为 IBN 的落地提供基础.

## 2.5 意图的实时反馈

在策略下发到实际网络后,需要对网络的状态信息进行实时监控,确保网络的转发行为符合用户意图.此外,网络的状态是会不断变化的,执行之初的网络状态与运行过程中的网络状态可能存在不一致,IBN 需要自动地根据期望达到的状态以及当前的网络状态对策略进行适当的优化与调整,保证网络始终满足意图需求.网络状态监控是确保网络始终满足用户意图的基础,传统网络中网络状态监控主要使用 NetFlow 和 SFlow.NetFlow<sup>[57]</sup>需要交换机 CPU 上的额外内存和负载才能提取和处理流信息,并且为了获取报告数据需要对交换机进行轮询,导致导出 NetFlow 数据的时间很长(例如 15 秒),因此 NetFlow 无法满足 IBN 对网络状态实时监控的需求;SFlow<sup>[58]</sup>对通过交换机的数据包流进行采样,进而分析网络状态,虽然 SFlow 在一定程度上可以满足实时性并且降低了交换机资源的使用,但是它却牺牲了一定的准确性,因此无法确保 IBN 可以正确满足用户意图.随着可编程交换机的不断发展,一种基于 P4 语言的技术——带内网络遥测(In-band Network Telemetry, INT)为解决网络状态的实时监控提供了一种可行方案.INT 是一个网络监控的框架,允许数据包查询交换机内部状态,可以直接在数据平面收集并报告网络设备状态及网络当前状态,从而可以实现实时、细粒度、端到端的网络监控.由于 INT 只能监控一条指定链路上网络状态,无法实现整个网络域内的状态感知,Tian Pan 等人<sup>[59]</sup>提出了“INT-path”进行全网范围内的网络遥测,使用基于源路由的方法向网络域内发送 INT 数据包,获取所有交换机的状态,进而可以感知整个网络域内的网络状态.INT 技术可以为 IBN 提供实时、准确的网络状态感知;并且它利用了可编程交换机,这也与 IBN 灵活的特点相契合,因此 INT 可以作为实现 IBN 网络状态感知的重要技术.另外,对于网络策略转发行为的监控,Zhang P 等人<sup>[60]</sup>提出采用“探针”的方法,通过向数据平面注入探针,根据探针的转发行为来判断数据平面是否按照相应的策略转发数据包,判断用户意图是否正确实现.Symnet<sup>[61]</sup>是一个基于符号执行的数据平面验证工具,通过向网络注入符号数据包来对网络进行建模,并静态地分析网络的行为是否满足用户的意图.此外,Minesweeper<sup>[62]</sup>可以将稳定的网络状态编码为可满足约束形式,实现了对于数据平面的全覆盖验证,同时保证足够的可伸缩性,从而对实际运行网络进行验证.如果用户意图没有正确实现或者在网络运行期间被意外改变,那么就需要及时向上层反馈信息,根据当前的网络状态对用户意图进行重新转译、编排.

## 3 IBN 应用探索

IBN 作为一种新兴的网络范式,为网络管理带来很多益处,但目前技术尚不成熟,各环节间缺乏统一的接口规范,目前尚无 IBN 的落地项目.不过随着相关研究的不断深入,IBN 可以应用到多种场景下.下面以网络测量和网络业务编排为例,介绍 IBN 的应用探索.

### 3.1 基于IBN的网络测量

网络测量可以对网络行为进行特征化,量化各种指标,对网络性能进行评价,网络测量有很多分类标准,按测量方式的不同分为主动测量和被动测量;按测量对象的不同分为拓扑测量、网络性能测量和网络流量测量等<sup>[63]</sup>.网络测量是网络性能管理、网络安全防御以及网络日常维护的基础,网络测量本应该随着互联网规模的扩大而不断发展,但目前网络测量仍然主要依靠于 SNMP、Ping、TraceRoute 等工具<sup>[52]</sup>.而且随着互联网规模的不断扩大,网络测量工作也变得更加复杂,而手工部署测量任务很容易出现错误,因此将 IBN 中意图的思想引入到网络测量可解决上述问题.通过将网络测量任务抽象为测量意图,系统可以自动解析和执行用户的测量需求,这种测量的方式称为“意图驱动网络测量”,如图 5 所示.

意图驱动网络测量的执行过程如下:

- 测量意图的获取: 当用户由于某种需求需要对网络进行测量时,将想要得到的网络性能指标抽象为测量意图输入到系统中;
- 测量意图的转译: 意图引擎会分析进入测量控制平面的测量意图并获取用户想要的网络的性能指标,然后自动生成相应的测量方法策略;
- 测量策略的验证: 网络测量的任务不同于一般的网络配置任务,因为测量任务通常无法验证其结果是否正确,因此测量策略的验证只需关注测量策略本身是否可以正确实现即可;
- 测量意图的下发与实时反馈: 测量策略实时下发到数据平面,收集数据平面的信息进而计算得到网络测量的结果.由于无法验证测量结果的正确性,系统需要自动进行多次测量,自动过滤错误的测量结果,最后将测量的结果反馈给用户.

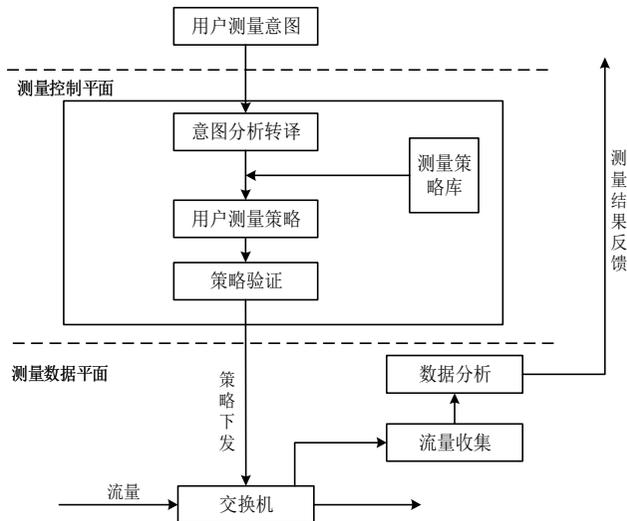


Fig.5 Implementation process of Intent Driven Network Measurement

图 5 意图驱动网络测量实现流程

意图驱动网络测量可以应用到诸多方面,例如网络运维,当网络出现故障的时候,需要对网络性能进行测量从而找到故障发生的位置,然而手工部署这些测量任务需要考虑很多因素,耗时较大.而意图驱动网络测量只需要用户输入测量意图,便可综合多方面因素得到最优的网络测量策略,得到网络性能指标,方便用户找到网络故障的发生位置.

### 3.2 基于IBN的网络业务编排

在运营商网络架构中,网络主要由骨干网、城域网以及数据中心组成.其中,骨干网实现多种网络的互联,同时与国际互联网对接,实现国内国际流量互通.骨干网为多种功能需求进行业务编排,例如为某些企业实现不同

区域之间通信的服务保障等.然而,目前大多数的网络业务编排都需要进行人工部署,并且由于骨干网的服务范围大、覆盖范围广、网络拓扑复杂,所以人工部署需要考虑多种问题并且可能需要在多地同时部署,因此运营商级别的网络业务编排通常需要很长的时间.通过将 IBN 中意图的思想引入到运营商网络中,可有效降低业务编排所需的时间.IBN 系统可感知网络的拓扑以及状态信息,因此在进行业务编排时可以最大程度减少人工的介入,从而提升业务编排的效率.网络管理员进行业务编排时,只需要将用户的需求输入到系统中,系统便可自动将用户需求转译为相关网络配置以及具体的网络性能指标,从而可实现在多地同时部署业务.

以企业 VPN 业务为例,如图 6 所示,基于意图的业务编排的执行过程如下:

- 业务编排意图的获取: 当企业用户向运营商提出 VPN 业务开通申请后,运营商需要了解用户的相关需求,例如开通的地点、开通时间的限制以及相关服务质量的要求;之后运营商即可把这些需求抽象为业务编排意图输入到系统中;
- 业务编排意图的转译: 业务编排意图将用户的需求转译为相关的网络配置策略,并将各种服务质量需求转译为具体的网络性能指标,例如带宽、时延、抖动等,之后分析采用的 VPN 接入方式,如 MPLS 或 SD-WAN 等,最后形成一条完备的 VPN 部署策略;
- 业务编排策略的验证: 业务编排策略的验证是基于意图的业务编排中最重要的一步,它可以事先验证业务是否符合用户预期,如果验证出现错误可能会带来巨大的损失;可以采用形式化的验证方法来验证策略是否可以正确实现,除此以外还需要考虑资源可用性问题;

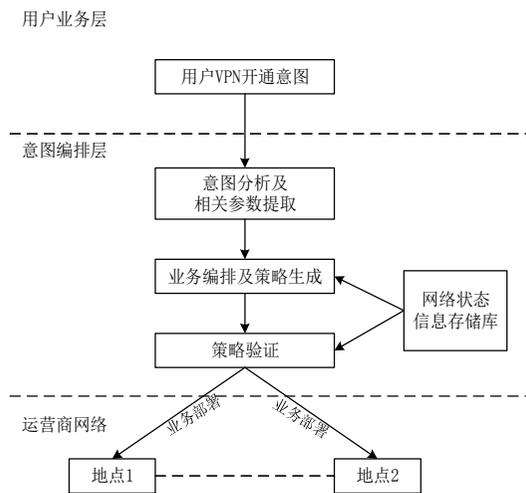


Fig.6 Implementation process of Intent Driven VPN deployment

图 6 意图驱动 VPN 部署实现流程

- 业务编排意图的下发与实时反馈: 由于 VPN 业务的编排涉及多地同时进行,因此在进行相关网络配置、资源分配的时候还要实时监测网络的状态,确保业务的编排满足用户的意图需求;并且实时向上反馈运行情况,当出现与意图不一致的情况需要及时对业务编排策略进行调整,保证意图实现的一致性.

在运营商网络环境下进行基于意图的业务编排可极大提升工作效率、减少业务开通的时间.但在运营层次实现 IBN 的落地还需要大量深入的研究,才能确保运营商免受相关风险的影响.

IBN 将不断为未来网络的发展助力.IBN 可在工业互联网领域大有作为,IBN 可以实现工厂的智能组网,极大减少人工操作可能带来的风险,另外,IBN 可以实时监测工厂网络状态,当网络出现异常时,可以迅速做出响应,减少工业生产损失.IBN 也可应用于数据中心网络,数据中心网络链路数量多、网络拓扑复杂,IBN 可实现数据中心网络的智能管理和监测,当数据中心网络出现故障时,IBN 可以迅速定位网络故障,降低网络维护成

本. IBN 凭借其灵活、智能、自动化的特点,在 5G、车联网、卫星网络等多个领域都将发挥重要作用。

## 4 研究展望

IBN 是一种新的理念,并不是一种新的技术,IBN 的实现综合了多种现有的技术,并基于 IBN 的性质开发了許多新技术.目前,IBN 已经得到了广泛的关注,学术界和产业界都已经开始对 IBN 进行深入研究与应用探索.然而 IBN 在迅速发展的同时也面临着一些挑战.

### (1)意图转译问题的研究

IBN 能实现网络任务自动化部署的一个关键步骤就是从意图到网络策略的转译工作,用户意图显示地或者隐式地表明了用户的需求,因此意图的转译工作是确保 IBN 正确实现的关键一步.然而,目前意图的转译工作还没有统一的实现方法,且现有方法大多停留在实验阶段.如 2.2 所述,对于意图的转译工作,不同的方法有各自的优缺点,缺少一个较为完善的解决方案.此外,对于用户意图的分析也是一个亟待解决的问题,目前的研究往往局限于一些简单的、特定环境下的意图描述或者策略描述,这使得一些研究工作往往忽略了意图的分析工作.意图的分析工作主要使用自然语言处理的方法,但是目前自然语言处理方法对于语义挖掘等任务的处理能力有限,因此要想完美实现意图的转译工作,对于意图的语义挖掘等问题还需要进一步的研究与探索.

### (2)策略验证问题的研究

IBN 实现网络任务自动化部署的另一个关键步骤就是策略的验证问题,策略的验证关系到网络的正常运行.在 IBN 的整个运行闭环中,如果一条策略出现错误,可能会导致一连串其他错误的出现,甚至导致网络停止运行.策略的验证工作是 IBN 实现不可缺少的一步,对于策略的验证的内容要有一个全面的考虑,除了上面提到过的资源的可用性验证、策略的冲突验证以及策略的正确性验证外,还要考虑其他需要验证的方面,例如对于网络策略与用户意图一致性的验证等,这是今后需要认真探究的内容.此外,策略的验证方法也是另一个需要继续深入研究的内容,例如上面提到过的形式化验证的方法,虽然它可以很好地对策略的正确性进行验证,但是它存在着状态空间爆炸这一严重的问题,这会导致系统无法正确求解或者无法求解.此外,其他的验证方法也有各自的局限性,因此需要继续对验证方法进行研究.并且,还要考虑验证的效率问题,策略的验证工作一定要高效并且全面,需要把所有的验证方法集成到一个模块之中.因此,验证方法的集成也是一个亟待解决的问题.

### (3)IBN 部署问题的研究

虽然,目前大多数对于 IBN 的研究都处于理论与实验阶段,但是鉴于 IBN 能极大地提高网络管理与维护的效率,IBN 的大规模部署或许会逐步开展.不过 IBN 的部署会引起它与传统网络的并存问题.因为 IBN 中存在与传统网络不兼容的协议和网络设备,会带来一些性能问题,例如网络的可靠性、路由问题等;文献<sup>[9]</sup>借鉴 SDN 中采用增量部署的方式,可采用增量部署或者部分部署的方式来逐渐部署 IBN.此外,还要考虑 IBN 的可拓展性,因为在 IBN 的逐步部署过程中,IBN 本身也会不断的调整,从而更好的实现用户的意图,因此 IBN 的可拓展性也需要进一步的研究.

### (4)IBN 跨域通信问题的研究

目前大多数对于 IBN 的研究局限于在一个单独的网络域内,但是却鲜有人考虑跨域通信的问题<sup>[64]</sup>.IBN 的实现很大程度上依赖于对于全局网络信息的感知,意图转译工作需要参考全局网络信息,策略的验证工作更是基于全局的网络信息进行的.然而,在实际中不同网络域的域内信息由于利益关系、隐私保护等因素无法对外公布,因此在一个域内可能无法获取另一个域内的网络信息,这会导致意图的转译工作和策略的验证工作无法正常运行.因此,IBN 中跨域通信、多域通信的问题是 IBN 落地无法避免的,这需要今后进一步的研究.

### (5)IBN 全局优化问题的研究

IBN 的实现是一个完全自动化的过程,因此在整个系统的运行期间需要考虑优化问题,从而提高系统的运行效率.目前,虽然 IBN 实现过程中的各个步骤都有一定的方法,但是目前还没有工作给出一个统一的实现架构,今后的工作需要对协调各个模块的接口,协调优化每一个实现模块.此外,系统的实时性以及及时反馈也是系统

优化的一个部分,为使 IBN 灵活运转,那么就需要不断实时地收集网络状态信息,并及时验证网络策略,并及时给出反馈.因此,IBN 的全局优化工作是一个值得讨论的问题.

#### (6)IBN 安全问题的研究

网络安全问题随着网络的发展一直存在,因此 IBN 中的安全问题也是一个值得思考的内容.IBN 有别于传统的网络,它可以根据用户的意图自动化地部署网络策略,然而这种自动化的部署带来了新的安全问题.首先,用户与网络之间的信任关系是一个值得研究的问题,用户是否可以完全信任网络不会受到恶意应用的干扰从而实现自己的意图;其次,隐私问题也值得注意,IBN 的自动化意味着需要向外界公布一些自己的内部信息,因此隐私保护问题值得研究;最后,IBN 虽然为网络的管理带来了自动化,但是这同样也带来了网络的脆弱性,恶意应用可以攻击 IBN 的转译模块,使得 IBN 无法正常运转.因此,对于 IBN 的安全的研究将确保 IBN 可以平稳安全的发展.

## 5 总结

IBN 是当前网络领域热门技术之一,之所以得到广泛的关注,在于它可以根据用户意图自动实现网络业务编排,这是其他类型网络无法比拟的.首先,网络业务自动化部署将网络管理员从繁重的网络管理工作中解放出来,加速业务和应用的部署和更新;其次,网络策略的自动化验证确保网络转发行为与用户需求一致,无需用户多次参与业务部署;最后,IBN 根据当前网络情况,自动优化网络配置,提升网络性能.鉴于 IBN 的发展潜力,学术界和产业界已经深入研究其关键技术,并将 IBN 应用到网络测量、网络业务编排等领域.然而,IBN 要想真正地落地还需要克服意图语义挖掘、策略验证和跨域通信等挑战.因此,要发挥 IBN 的优势,不仅要不断改进现有技术,还要开发新技术,完善 IBN 的优化闭环.只有这样,IBN 才会更好地服务网络智能管理,最终实现网络自治.

## References:

- [1] Li Y, Yin X, Wang Z, et al. A Survey on Network Verification and Testing with Formal Methods: Approaches and Challenges[J]. IEEE Communications Surveys & Tutorials, 2018, 21(1): 940-969.
- [2] Veriflow. Network Complexity, Change, and Human Factors are Failing the Business. <https://www.veriflow.net/survey/>.
- [3] M.Muuss, "The story of the ping program,"1983. [Online]. Available: [http://mirrors.pdp-11.ru/\\_vax/www.bandwidthco.com/wHITEPAPERS/netforensics/icmp/The%20Story%20of%20the%20PING%20Program.pdf](http://mirrors.pdp-11.ru/_vax/www.bandwidthco.com/wHITEPAPERS/netforensics/icmp/The%20Story%20of%20the%20PING%20Program.pdf).
- [4] "traceroute." [Online]. Available: <ftp://ftp.ee.lbl.gov/traceroute.tar.gz>.
- [5] McKeown N. Software-Defined networking. In: Proc. of the INFOCOM Key Note. 2009. <http://infocom2009.ieee-infocom.org/technicalProgram.htm>.
- [6] Feamster N, Rexford J, Zegura E. The Road to SDN: An Intellectual History of Programmable Networks. ACM, 2014.
- [7] Han Y, Li J, Hoang D, et al. An intent-based network virtualization platform for SDN[C]// International Conference on Network & Service Management. IEEE, 2017.
- [8] C. Janz, "Intent nbi-definition and principles," Open Networking Foundation, Version, vol. 2, 2015.
- [9] Elkhatib Y, Coulson G, Tyson G. Charting an intent driven network.2017 13th International Conference on Network and Service Management (CNSM). IEEE Computer Society, 2017.
- [10] Zhang H, Wang Y, Qi X, et al. Demo abstract: An intent solver for enabling intent-based SDN[C]//2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2017: 968-969.
- [11] Kiran M, Pouyoul E, Mercian A, et al. Enabling intent to configure scientific networks for high performance demands[J]. Future Generation Computer Systems, 2017: S0167739X1730626X.
- [12] Cohen R, Barabash K, Rochwerger B, et al. An intent-based approach for network virtualization[C]// Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on. IEEE, 2013.
- [13] "Intent: Don't Tell Me What to Do! (Tell Me What You Want)." [Online]. Available: <https://www.sdxcentral.com/articles/contributed/networkintentsummitperspectivedavidlenrow/2015/02/>.
- [14] 向基于意图的网络迈进. 2018. [https://www.cisco.com/c/m/zh\\_cn/express/case\\_center/en/anren008369.html](https://www.cisco.com/c/m/zh_cn/express/case_center/en/anren008369.html).

- [15] Comer D, Rastegarnia A. OSDF: An Intent-based Software Defined Network Programming Framework[J]. 2018.
- [16] Bi J, Cheng Y, Xie C, et al. Policy-Based Management Framework for the Simplified Use of Policy Abstractions (SUPA)[J]. Policy, 2018.
- [17] Behringer M, Pritikin M, Bjarnason S, et al. Autonomic networking: Definitions and design goals (RFC 7575) [J]. Internet Research Task Force, 2015, 10.
- [18] Jiang S, Carpenter B, Behringer M. General gap analysis for autonomic networking (RFC 7576) [J]. Internet Engineering Task Force, 2015, 10.
- [19] Clemm A, Nobre J, Granville L, et al. Autonomic Networking Use Case for Distributed Detection of Service Level Agreement (SLA) Violations[J]. 2018.
- [20] Sourì A, Norouzi M. A State-of-the-Art Survey on Formal Verification of the Internet of Things Applications[J]. Journal of Service Science Research, 2019, 11(1): 47-67.
- [21] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM CCR, 2008,38(2):69–74. [doi: 10.1145/1355734.1355746]
- [22] Bosshart P, Daly D, Gibb G, et al. P4: Programming protocol-independent packet processors[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 87-95.
- [23] Openaylight sdn controller. [Online]. Available: <https://www.opendaylight.org/>.
- [24] NEMO. Network model. <http://www.nemo-project.net/>.
- [25] Van Deursen A, Klint P. Domain-specific language design requires feature descriptions[J]. Journal of Computing and Information Technology, 2002, 10(1): 1-17.
- [26] <https://www.cisco.com/c/en/us/products/cloud-systems-management/dna-center/index.html>
- [27] Pham M, Hoang D B. SDN applications-the intent-based northbound interface realisation for extended applications[C]//2016 IEEE NetSoft Conference and Workshops (NetSoft). IEEE, 2016: 372-377.
- [28] Onos intent framework. [Online]. Available: <https://wiki.onosproject.org/display/ONOS/Intent+Framework/>.
- [29] Sanvito D, Moro D, Gulli M, et al. ONOS Intent Monitor and Reroute service: enabling plug&play routing logic[C]//2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft). IEEE, 2018: 272-276.
- [30] Foster N, Harrison R, Freedman M J, et al. Frenetic: A network programming language[J]. ACM Sigplan Notices, 2011, 46(9): 279-291.
- [31] Anderson C J, Foster N, Guha A, et al. NetKAT: Semantic foundations for networks[J]. Acm sigplan notices, 2014, 49(1): 113-126.
- [32] Yu Y, Wang ZL, Bi J, Shi XG, Yin, X. Survey on the languages in the northbound interface of software defined networking. Ruan Jian Xue Bao/Journal of Software, 2016,27(4):993–1008 (in Chinese). <http://www.jos.org.cn/1000-9825/5028.htm>.
- [33] Tian B, Zhang X, Zhai E, et al. Safely and automatically updating in-network ACL configurations with intent language[C]//Proceedings of the ACM Special Interest Group on Data Communication. ACM, 2019: 214-226.
- [34] Prakash C, Lee J, Turner Y, et al. Pga: Using graphs to express and automatically reconcile network policies[C]//ACM SIGCOMM Computer Communication Review. ACM, 2015, 45(4): 29-42.
- [35] Gruber T R. A translation approach to portable ontology specifications[J]. Knowledge acquisition, 1993, 5(2): 199-220.
- [36] Liao B, Gao J. An automatic policy refinement mechanism for policy-driven grid service systems[C]//International Conference on Grid and Cooperative Computing. Springer, Berlin, Heidelberg, 2005: 166-171.
- [37] Riftadi M, Kuipers F. P4/O: Intent-Based Networking with P4[C]//2019 IEEE Conference on Network Softwarization (NetSoft). IEEE, 2019: 438-443.
- [38] Beigi M S, Calo S, Verma D. Policy transformation techniques in policy-based systems management[C]//Proceedings. Fifth IEEE International Workshop on Policies for Distributed Systems and Networks, 2004. POLICY 2004. IEEE, 2004: 13-22.
- [39] Wang P, Huang L, Xu H, et al. Rule Anomalies Detecting and Resolving for Software Defined Networks[C]. IEEE Global Communications Conference. IEEE, 2016:1-6.
- [40] Al-Shaer E, Al-Haj S. FlowChecker: configuration analysis and verification of federated openflow infrastructures[C]. ACM Workshop on Assurable and Usable Security Configuration. ACM, 2010:37-44.

- [41] Khurshid A, Zou X, Zhou W, et al. Veriflow: Verifying network-wide invariants in real time[C]//Presented as part of the 10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13). 2013: 15-27.
- [42] Gleirscher M, Marmosoler D. Formal Methods: Oversold? Underused? A Survey[J]. arXiv preprint arXiv:1812.08815, 2018.
- [43] Legay A, Delahaye B, Bensalem S. Statistical model checking: An overview[C]//International conference on runtime verification. Springer, Berlin, Heidelberg, 2010: 122-135.
- [44] Kovács L, Voronkov A. First-order theorem proving and Vampire[C]//International Conference on Computer Aided Verification. Springer, Berlin, Heidelberg, 2013: 1-35.
- [45] Cadar C, Sen K. Symbolic execution for software testing: three decades later[J]. Commun. ACM, 2013, 56(2): 82-90.
- [46] Armand M, Faure G, Grégoire B, et al. A modular integration of SAT/SMT solvers to Coq through proof witnesses[C]//International Conference on Certified Programs and Proofs. Springer, Berlin, Heidelberg, 2011: 135-150.
- [47] S. S. Huang, T. J. Green, and B. T. Loo, "Datalog and emerging applications: an interactive tutorial," in Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011, 2011, pp. 1213–1216.
- [48] D. Jackson, Software Abstractions - Logic, Language, and Analysis. MIT Press, 2006. [Online]. Available: <http://mitpress.mit.edu/catalog/item/default.asp?tttype=2&tid=10928>.
- [49] Son S, Shin S, Yegneswaran V, et al. Model checking invariant security properties in OpenFlow[C]//2013 IEEE international conference on communications (ICC). IEEE, 2013: 1974-1979.
- [50] Lewis B, Fawcett L, Broadbent M, et al. Using P4 to Enable Scalable Intents in Software Defined Networks[C]//2018 IEEE 26th International Conference on Network Protocols (ICNP). IEEE, 2018: 442-443.
- [51] Gupta A, Harrison R, Canini M, et al. Sonata: Query-driven streaming network telemetry[C]//Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. ACM, 2018: 357-371.
- [52] Laffranchini P, Rodrigues L, Canini M, et al. Measurements As First-class Artifacts[C]// IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. IEEE, 2019
- [53] Beckett, R., Mahajan, R., Milstein, T. D., Padhye, J., and Walker, D. Don't mind the gap: Bridging network-wide objectives and device-level configurations. In ACM SIGCOMM (SIGCOMM) (2016).
- [54] Beckett, R., Mahajan, R., Milstein, T. D., Padhye, J., and Walker, D. Network configuration synthesis with abstract topologies. In 38th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI) (2017)
- [55] El-Hassany, A., Tsankov, P., Vanbever, L., and Vechev, M. T. Network-wide configuration synthesis. In 29th International Conference on Computer Aided Verification (CAV) (2017).
- [56] El-Hassany, A., Tsankov, P., Vanbever, L., and Vechev, M. T. NetComplete: Practical network-wide configuration synthesis with autocompletion. In 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI) (2018).
- [57] Estan C, Keys K, Moore D, et al. Building a better NetFlow[J]. ACM SIGCOMM Computer Communication Review, 2004, 34(4): 245-256.
- [58] Phaal P, Panchen S, McKee N. InMon corporation's sFlow: A method for monitoring traffic in switched and routed networks[J]. 2001.
- [59] Pan T, Song E, Bian Z, et al. INT-path: Towards optimal path planning for in-band network-wide telemetry[C]//IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, 2019: 487-495.
- [60] Zhang P, Zhang C, Hu C. Fast Data Plane Testing for Software-Defined Networks With RuleChecker[J]. IEEE/ACM Transactions on Networking, 2018, 27(1): 173-186.
- [61] Stoenescu, R., Popovici, M., Negreanu, L., and Raiciu, C. Symnet: Scalable symbolic execution for modern networks. In ACM SIGCOMM (SIGCOMM) (Aug. 2016).
- [62] Beckett, R., Gupta, A., Mahajan, R., and Walker, D. A general approach to network configuration verification. In ACM SIGCOMM (SIGCOMM) (2017).
- [63] Hu ZG, Tian CQ, Du L, Guan XQ, Cao F. Current research and future perspective on IP network performance measurement. Ruan Jian Xue Bao/Journal of Software, 2017,28(1):105–134 (in Chinese). <http://www.jos.org.cn/1000-9825/5127.htm>

- [64] Arezoumand S, Dzeparoska K, Bannazadeh H, et al. MD-IDN: Multi-domain intent-driven networking in software-defined infrastructures[C]//2017 13th International Conference on Network and Service Management (CNSM). IEEE, 2017: 1-7.

**附中文参考文献:**

- [32] 于洋,王之梁,毕军,施新刚,尹霞.软件定义网络中北向接口语言综述.软件学报,2016,27(4):993-1008.  
<http://www.jos.org.cn/1000-9825/5028.htm>
- [63] 胡治国,田春岐,杜亮,关晓蕾,曹峰.IP网络性能测量研究现状和进展.软件学报,2017,28(1):105-134.  
<http://www.jos.org.cn/1000-9825/5127.htm>