



基于优化决策树的时延敏感流智能感知调度

王雪荣¹, 唐政治², 李银川², 齐美玉², 朱建波², 张亮²

(1. 中国电信股份有限公司研究院, 广东 广州 510660;

2. 北京华为数字技术有限公司, 北京 100085)

摘要: 目前流量调度策略无法做到智能按需化, 尤其对于网络突发故障造成的拥塞以及高价值业务的护航场景, 无法按需保障时延敏感的业务体验。通过分析研究不同网络业务流量时延敏感性属性需求, 探索挖掘不同网络业务流量的行为特征与其时延敏感性需求之间的内在关联关系。然后利用 AI 技术对这种内在的关联关系进行学习, 构建其映射关系, 实现了时延敏感流智能感知调度。同时, 考虑 AI 模型的可解释性及可部署性实际问题, 采用强化学习剪枝优化可解释性决策树模型, 提高模型的鲁棒性同时使模型更轻量化, 易于设备部署实现。通过真实网络流量实验, 强化学习优化后的决策树模型在单次推理情形下感知正确率提高 1.75%, 推理速度提升约 30%; 同时, 实验也证明了使用局部微观统计特征多次推理有助于提高模型感知正确率。在所有实验中, 强化学习优化的决策树模型规模缩小了 60.0%~87.2%, 并且 Saras 比 Q-learning 具有更好的优化表现。

关键词: 流量分析; 流量调度; 时延敏感属性; 强化学习; 决策树

中图分类号: TP393

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2023095

Delay-sensitive traffic intellisense scheduling based on optimal decision tree

WANG Xuerong¹, TANG Zhengzhi², LI Yinchuan², QI Meiyu², ZHU Jianbo², ZHANG Liang²

1. Research Institute of China Telecom Co., Ltd., Guangzhou 510660, China

2. Beijing Huawei Digital Technologies Co., Ltd., Beijing 100085, China

Abstract: Currently, network traffic scheduling strategy cannot be intelligent and on-demand, especially in the congestion caused by sudden network failures and escort scenarios of high-value services. They cannot guarantee latency-sensitive service experience on demand. The delay-sensitive attribute requirements of different network traffic were analyzed and studied, and the internal correlation between the behavior characteristics of varying network traffic and its delay sensitivity requirements was explored. Then, AI technology was used to learn this inherent relationship and construct its mapping relationship, realizing a traffic scheduling technical solution based on the intelligent awareness of delay-sensitive traffic. At the same time, considering the practical issues of interpretability and deployability of AI models, reinforcement learning (RL) technology was used to prune and optimize the interpretable

decision tree model, which improved the robustness of the model and made model lighter and easier to implement in equipment deployment. Through experiments by the collected real network traffic, the decision tree model optimized by reinforcement learning could improve the awareness accuracy by 1.75% in a single inference case, and the inference performance was improved by about 30%. The experiment also proved that using micro-statistical features for multiple inferences could help improve the model accuracy; in all experiments, the scale of the decision tree model optimized by RL was reduced by about 60.0%~87.2%, and the Saras had better optimization performance than *Q*-learning.

Key words: traffic analysis, traffic scheduling, delay-sensitive attribute, reinforcement learning, decision tree

0 引言

从 1969 年的高级研究计划局网络 (Advanced Research Project Agency Network, ARPANET) 互联网雏形诞生, 历经 50 多年的发展, 到目前日趋复杂的网络架构及协议, 互联网不仅本身发生了巨大变化, 同时也推动着社会的发展和变化。但是伴随着不断新兴的互联网业务驱动, 传统“尽力而为”的消费型网络服务已难以支撑现状生产型互联网/5G 等业务对差异化服务等级协定 (service level agreement, SLA) 可承诺、低时延及确定性带宽的需求。因此, 在基于现有互联网构架及协议的基础上, 需要更合理地利用网络资源及合理地进行网络流量调度, 才能更好地支持新兴网络业务对网络性能的需求。

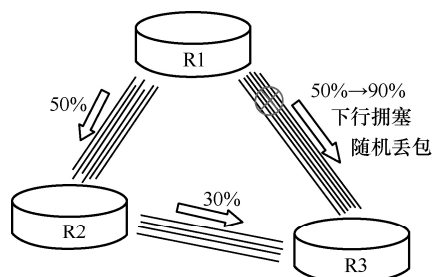
目前, 网络流量调度研究的文献主要分为两个方面。(1) 时间敏感网络 (time sensitive networking, TSN) 流量调度^[1]。时间敏感网络主要服务于工业控制、车载网络等很多行业领域的业务, 实现确定性和低时延的网络数据传输能力。其主要通过流量调度机制, 实现对所有交换机端口的数据调度, 通过确定数据传输顺序和时间, 达到满足各业务流量对时延和带宽等性能的需求, 进而优化网络数据传输的性能。(2) 数据中心网络流量调度^[2]。目前, 数据中心网络广泛采用的路由协议都是基于数据流到路径的静态映射, 比如最短路径优先算法和基于哈希的等价多路径路由等。这些路由协议实现都不考虑链

路的负载和数据流大小等情况, 都会导致链路拥塞和负载不均衡, 使得网络性能下降。因此, 在数据中心网络中, 不论其内部、相互之间还是和用户之间都需要进行高效的流量调度, 进而实现最小化流完成时间、公平性保证及带宽保障等优化目标^[2]。

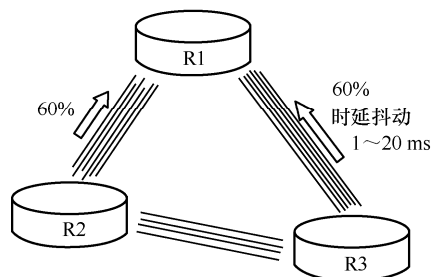
目前, 网络流量的调度策略都是确保某个特定目标最优化, 因而无法做到智能按需化的流量调度。尤其对于网络突发故障造成的拥塞以及高价值业务的护航场景, 现有的流量调度机制无法按需保障时延敏感型业务的体验。例如, 在运营商的骨干网络中存在两个实际问题: 故障发生导致的拥塞如何自动调优; 如何实现高价值业务流量的护航, 抑制网络中占据资源的低价值业务, 降低整体时延。网络业务流量按需化调度场景如图 1 所示。

(1) 拥塞自动调优

如图 1 (a) 所示, R1 是骨干网络中的一台路由器, R1 通过负载均衡机制向 R2 和 R3 传输数据, 在正常情形下, 链路的利用率都是 50%。若 R1 到 R3 下行主干 (trunk) 部分成员接口发生故障, 主干链路的利用率会由 50% 升高到 90%, 出现网络拥塞现象, 于是下行端口队列开始随机丢包。目前, 现网中设备的流量调度策略还不能进行智能化处理, 一般的策略会随机调度部分流量选择其他路径传输, 以期缓解链路拥塞。由于采取随机调度策略, 因此会存在调度的流量是时延敏感小流的业务, 这样就会造成这些业务时延增大, 同时小流流量的数据量占比小, 可能也缓解不了拥塞。



(a) 故障拥塞场景的流量调度需求



(b) 高价值业务流量调度需求

图1 网络业务流量按需调度场景

(2) 高价值业务护航

如图1(b)所示, R3上行到R1的数据传输分为R3到R1的高价值直连链路和通过R2绕行的链路。对于R3直连骨干R1的低时延平面承载的高价值业务,链路利用率为60%左右,虽然没有丢包,但是链路中存在的时延不敏感的大流会产生微突发,进而导致报文的排队引入了较大的时延抖动,抖动会对一些时延敏感的业务流量产生较大影响。因此,实际需求希望在高价值平面进行实时大流监控分析,将时延不敏感的大流提前绕行到低价值平面,确保高价值平面的时延抖动满足SLA要求。例如,将时延不敏感大流绕行R2,缓解R3到R1链路压力,使R3链路转发时延抖动降低。

因此,针对上述骨干网中存在的流量调度实际问题,需要对流量的时延敏感属性进行精细化感知分析,进而实现流量的按需差异化调度。本文的主要贡献包括以下5个方面。

- 提出了一种网络业务流量时延敏感属性智能感知的方法,通过对业务流量的时延敏感属性进行智能精细化感知,完成业务流量的按需智能化调度策略。

- 分析研究了因特网工程任务组(Internet Engineering Task Force, IETF)组织制定的网络中各种业务流量归属大类服务类别和其所需网络性能要求标准,根据实际场景需求,将所有类别合并为时延敏感属性要求不同的4个等级。
- 利用AI技术强大的数据挖掘和学习能力,探索构建不同业务流量的行为特征与其时延敏感属性之间的内在关联关系。
- 考虑AI模型的可解释性及可部署性实际问题,采用简单的决策树机器学习算法,同时使用强化学习优化决策树模型,提高模型的鲁棒性同时使模型更轻量化,易于设备部署实现。
- 使用采集的真实网络环境流量数据进行实验,结果表明优化后的决策树模型在单次推理情形下感知正确率提高1.75%,同时模型规模缩小约87.2%,推理速度提升约30%。基于感知的业务流量时延敏感属性实现了智能按需流量调度策略。

1 相关工作

网络流量时延敏感属性感知方法主要分为主动方式和被动方式两种。主动方式是指基于已知的业务流量敏感属性,在发送报文时主动设置标识位,中间转发设备解析报文可主动获取得到,如DiffServ方法;被动方式是指需要根据业务流量的行为特征,采用相关感知算法进行被动识别获取,如基于流size的感知方法。流量时延敏感属性感知方法对比见表1。

表1 流量时延敏感属性感知方法对比

方案	局限性
ACL	维护复杂,表资源消耗大,手工配置困难
DSCP	可能出现误配错配,存在不信任问题
DPI	存在隐私安全问题,资源消耗大,加密流识别率低
基于流size	区分粒度粗,时敏大流无保障,小流误识别

DiffServ 方法^[3]主要通过为不同的业务流量报文设置不同的优先级标识,将 SLA 要求高的业务设置为高优先级,SLA 要求低的业务设置为低优先级,高优先级业务走低时延路径并优先调度,从而保证业务流量差异化调度。DiffServ 方法的实现方式有两种,一是通过配置网络设备上的访问控制列表(access control list, ACL)标识不同业务优先级,例如在 ACL 里配置已知业务的五元组信息及优先级信息,进而实现流量的调度。但是这种实现方式随着业务的增多及网络的复杂化,不足也日趋明显。新业务及新用户都需要不断增加或修改 ACL 配置,维护复杂,工作量大,同时对于流量庞大的网络核心设备,配置庞大的 ACL 也不符实际。二是在终端设备上对发送出的不同业务流量报文标识不同差分服务编码点(differentiated services code point, DSCP)值,不同的 DSCP 值标识不同的优先级,中间网络设备可直接读取标识信息进行调度处理。这种方式需要终端应用程序支持,但是让所有运营者对自己的应用程序标记 DSCP 值是不现实的,而且存在误配和故意错配,因此,在开放的网络环境中实现不具有可行性。

基于流 size 感知的方法,学术研究的方法主要包括 pFabric^[4]、PIAS^[5]和 AuTO^[6]等,产业界商用的方法,以思科为例,主要为动态数据包优先级(dynamic packet prioritization, DPP)^[7]和应用感知路由(application-aware routing, AAR)^[8]。文献[4]提出了一种 dequeue 机制,在预先知道流 size 的情况下,根据剩余流量的大小设置优先级,实现最高优先级发送,最低优先级丢弃的流量调度。由于需要提前预知或预设流量大小,在现实场景中难以实现。针对文献[4]的不足之处,文献[5]提出了一种新的调度方案,利用多个优先级队列来实现多级反馈队列调度算法,能够在流大小未知的情况下完成流的调度,降低流的平均完成时间。但是,多级反馈队列的阈值获取依赖于长周期的流量分布,阈值的准确设置较困难。文献[6]指出 PIAS 方法在运行时

容易与当前/真实大小分布不匹配,而在不匹配的情况下,性能下降可能高达 38.46%。因此,文献[6]提出了使用深度强化学习(deep reinforcement learning, DRL)方法根据环境自动配置算法参数,减少人工配置的时间成本以及人为错误导致的性能降低。但是,强化学习本身也存在训练收敛慢的问题,因此对流量分布变化的反应会比较迟缓,灵活性比较差。思科 Nexus 9000 系列交换机采用的 DPP 方法核心思路是跟踪入口端流,将其区分为大象/老鼠流,分别应用单独的排队策略。将每个流中的前 N (设置的 DPP 最大数据包计数) 个数据包划分为老鼠流,将第 $N+1$ 个数据包及以后的数据包划分为大象流。将区分的老鼠流置于高优先级(丢包/时延敏感),而大象流置于加权队列(带宽)。DPP 方法中固定的阈值 N 设置会存在误识别的情况,例如,会误识别新兴的虚拟现实(virtual reality, VR)流量,同时阈值的设置需要大量数据分析得出。相对而言,基于应用识别的 AAR 方法对流量的调度更精细化。AAR 方法通过对链路进行监视和测量获取链路的性能(丢包率、时延和抖动),然后将基于深度包检测(deep packet inspection, DPI)识别出的应用程序流量按照其所需网络性能调度到相应的传输隧道,得到精准的流量调度。但是 DPI 需要在设备上分析流报文信息,资源消耗大,同时存在隐私安全问题和加密流量分析正确率低等问题。

鉴于目前流量时延敏感属性感知调度方法存在的不足,本文提出了一种网络流量时延敏感属性智能精细化感知的方法,基于精细化感知的结果实现了流量的按需精细化调度需求。利用 AI 技术挖掘学习不同网络业务流量的行为特征与其时延敏感性需求之间的内在关联关系,并构建其映射关系,实现了基于时延敏感属性智能感知的流量调度技术方案。

2 基于时延敏感属性智能感知流量调度

基于时延敏感属性智能感知流量调度的流程如图 2 所示。对于网络中不同的业务流量,首先

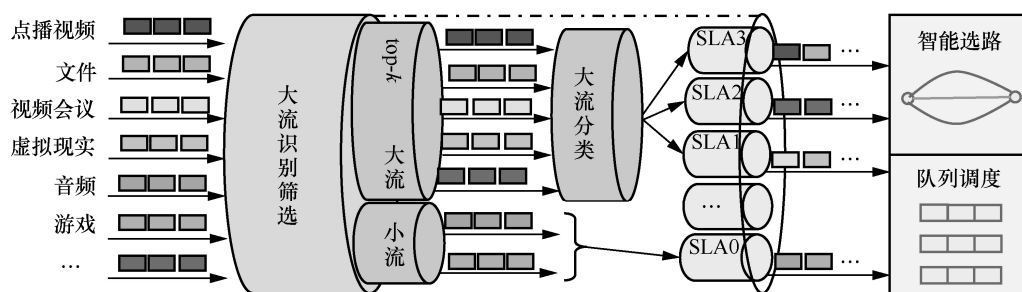


图2 基于时延敏感属性智能感知流量调度的流程

通过大流识别机制可自动识别区分流量中的大流和小流。对大小流区分的研究比较多,如传统基于 $\text{top-}k$ 思路^[9-10]以及基于机器学习的方法^[11-13]。鉴于实际部署网络设备计算资源的限制,本文主要研究基于 $\text{top-}k$ 检测区分出的大流的时延敏感属性智能感知调度方法,对大流的识别筛选采用现存的 $\text{top-}k$ 思路检测算法。主要对检测识别出来的大流进行行为统计特征测量,然后使用机器学习算法(考虑可解释性和部署性难度,本文采用决策树算法)感知大流的 SLA 需求差异性,将大流细分映射到不同的 SLA 等级。根据获取到的业务流量的 SLA 等级,可进行调度的策略包括智能路由选择和设备内的出端口队列调度等,进而实现业务流量的差异化调度处理。

2.1 网络流量时延敏感属性标准

全网流量涉及的应用和业务很多,衡量和确定这些应用或业务具体的时延敏感属性等级,需要一个可信和规范化标准。本文对网络流量的时延敏感属性等级划分参考国际 IETF 组织制定的草案 RFC4594。在草案 RFC4594 中^[14],根据应用程序/服务的不同流量特征和所需要的网络性能定义了 12 个服务类别。网络流量服务类别划分及对应的网络性能要求见表 2。表 2 给出了 12 种服务类别,包括网络控制类、信令类和多媒体会议类等。流量特征列给出了每种类别对应的流量行为特征,网络性能列给出了每种类别对应的网络性能要求。根据这些行为特征和网络性能要求可以将相似的当前和未来应用程序/服务流量映射

到这 12 个服务类别中。应用举例列给出了每个服务类别大致归属的应用/服务,如网络控制类应用主要包含网络路由相关协议。

草案 RFC4594 还对应给出了这 12 种服务类别的 DSCP 值,用于终端应用对发出的报文进行参考标记,但在实际场景中却并没有应用起来。考虑到实际网络设备流量调度并不会用到这么多的队列,根据表 2 网络性能和 DSCP 列提供的信息,本文对具有相似或相近网络性能需求的服务类别进行合并,将它们标注为相同的时延敏感属性等级。例如,将 IP 语音类和实时交互类等标注为 SLA (这里的 SLA 主要指时延敏感度)等级 0。因此可以将这 12 个服务类别映射到 SLA 的 4 个等级上,即 0~3 等级,网络业务流量时延敏感属性等级见表 3。其中,从 0 到 3,优先级依次降低。在实现中,通过大小流检测过程识别出的小流对应到 SLA 0 等级,大流则通过决策树模型精细化分类对应到 SLA 1~3 等级。按照感知的业务流量时延敏感属性 SLA 等级即可对业务流量进行按需精细化调度。

2.2 网络流量时延敏感属性智能感知方法

2.2.1 网络流量处理及特征提取

考虑可解释性和设备实际部署难度问题,本文采用决策树算法作为感知模型。因为决策树算法属于有监督机器学习,需要对采集的无标签流量进行处理,包括流的应用/服务标注、流的时延敏感属性等级标注和流的特征提取,本文对流量的处理都是针对以五元组(源 IP 地址、目的 IP

表2 网络流量服务类别划分及对应的网络性能要求

服务类别	流量特征	网络性能			DSCP	应用举例
		丢包率	时延	抖动		
网络控制类	报文大小可变, 主要为非弹性短消息, 但也存在突发	低	低	是	CS6	网络路由协议
IP 语音类	固定大小的小报文, 发包率固定, 非弹性和低速率	非常低	非常低	非常低	EF	IP 语音承载
信令类	报文大小可变, 一些是突发的且存活时间短的流量	低	低	是	CS5	IP 语音信令
多媒体会议	报文大小可变, 传输间隔固定, 速率自适应, 丢包敏感	低—中	非常低	低	AF4	H.323/V2 视频会议 (自适应)
实时交互	实时传输协议/用户数据报协议流, 非弹性, 大部分是变速率	低	非常低	是	CS4	视频会议和交互性游戏
多媒体数据流	报文大小可变, 弹性且变速率	低—中	中	是	AF3	点播流媒体视频和音频
广播视频	固定或可变速率, 非弹性且无突发	非常低	中	低	CS3	广播电视和直播
低时延数据	可变速率、弹性的且存活时间短的弹性流量	低	低—中	是	AF2	电子商务及金融交易
操作维护管理	报文大小可变, 弹性或者非弹性的流量	低	中	是	CS2	操作维护管理保障
高吞吐数据	可变速率、突发的且存活时间长的弹性流量	低	中—高	是	AF1	存储和转发应用
标准	都涉及	无具体要求			DF (CS0)	未分化应用
低优先级数据	非实时性和弹性的流量	高	高	是	CS1	任何不需要带宽保障的流量

表3 网络业务流量时延敏感属性等级

服务类别	网络性能			DSCP	SLA 等级
	丢包率	时延	抖动		
网络控制类	低	低	是	CS6	1
IP 语音类	非常低	非常低	非常低	EF	0
信令类	低	低	是	CS5	1
多媒体会议	低—中	非常低	低	AF4	0
实时交互	低	非常低	是	CS4	0
多媒体数据流	低—中	中	是	AF3	2
广播视频	非常低	中	低	CS3	2
低时延数据	低	低—中	是	AF2	1
操作维护管理	低	中	是	CS2	2
高吞吐数据	低	中—高	是	AF1	3
标准	无具体要求			DF (CS0)	3
低优先级数据	高	高	是	CS1	3

地址、源端口、目的端口和传输层协议) 为标识的流。按照文献[15]中对网络报文的数学表达式定义, 将每个报文定义表示为 $p^i = (k^i, l^i, t^i)$, $i = 1, 2, \dots, n$, 其中, 第一个元素 k^i 表示第 i 报文的五元组信息, 第二个元素 l^i 表示第 i 报文的大小 (byte),

第三个元素 t^i 表示第 i 报文到达的时间。于是, 以五元组信息唯一标识的流 $f \in F$ 可定义为 $f: \{p^1 = (k^1, l^1, t^1), p^2 = (k^2, l^2, t^2), \dots, p^m = (k^m, l^m, t^m)\}$, $t^1 < t^2 < \dots < t^m$, 其中, 所有报文的五元组信息一样, 即 $k^1 = k^2 = \dots = k^m$ 。



(1) 应用/服务标注：借助 DPI 等识别工具或者 IP 与应用对应关系表等将采集的流量按照五元组信息标注好其归属的应用/服务，则最后所有标注好的流可表示为 $F = \{f^1 : \text{app}^1, f^2 : \text{app}^2, \dots, f^N : \text{app}^N\}$ 。

(2) 流时延敏感属性等级标注：按照第 2.1 节划分好的流时延敏感属性等级标准，将 (1) 中标注好的应用/服务流归属到表 3 中对应的服务类别，进而映射标注出该流对应的敏感属性 SLA 等级。则用数学表达式可表示为 $F = \{f^1 : (\text{app}^1, \text{SLA}^1), f^2 : (\text{app}^2, \text{SLA}^2), \dots, f^N : (\text{app}^N, \text{SLA}^N)\}$ 。

(3) 流特征提取：对标注完标签的所有网络流 F 提取感知模型需要的行为统计特征。文献[16]提出了网络流量可用行为统计特征共 240 多个，但大部分特征针对 TCP 流。为了提取的行为统计特征具有更好的可解释性和鲁棒性，本文综合了业务专家经验知识和数据驱动特征工程算法。在数据驱动特征筛选算法上，选择基于 filter 思想的算法，按照发散性或相关性对各个特征进行评分，对设定阈值或者所需特征的个数进行筛选。最后，结合设备可承受特征提取的计算资源，提取了 11 维特征，具体特征计算方式如下。

- 平均流速 (FlowRate)：监测周期内单位时间传输的数据字节数，其数学表达式为：

$$\text{FlowRate} = \sum_{i=1}^M (l^1 + l^2 + \dots + l^M) / (t^M - t^1)。$$
- 最大突发度 (MaxBurst)：监测周期内报文到达时间间隔小于某个阈值的所有报文字节总和最大值，其数学表达式为：

$$\text{MaxBurst} = \max(\text{burst}^1, \text{burst}^2, \dots, \text{burst}^q, \dots)，$$
其中， $\text{burst}^i = \sum (l^{h-1} + l^h + \dots)$, if $t^h - t^{h-1} < \text{threshold}$ 。
- 报文大小均值 (AvgPktSize)：监测周期内所有报文长度的均值，其数学表达式为：

$$\text{AvgPktSize} = \sum_{i=1}^M (l^1 + l^2 + \dots + l^M) / M。$$
- 最大报文长度 (MaxPktSize)：监测周期内

所有报文长度最大值，其数学表达式为：

$$\text{MaxPktSize} = \max(l^1, l^2, \dots, l^M)。$$

- 最小报文长度 (MinPktSize)：监测周期内所有报文长度最小值，其数学表达式为：

$$\text{MinPktSize} = \min(l^1, l^2, \dots, l^M)。$$
- 最大报文间隔 (MaxPktInterval)：监测周期内报文到达时间间隔最大值，其数学表达式为：

$$\text{MaxPktInterval} = \max(t^2 - t^1, t^3 - t^2, \dots, t^M - t^{M-1})。$$
- 最小报文间隔 (MinPktInterval)：监测周期内报文到达时间间隔最小值，其数学表达式为：

$$\text{MinPktInterval} = \min(t^2 - t^1, t^3 - t^2, \dots, t^M - t^{M-1})。$$
- 报文间隔均值 (AvgPktInterval)：监测周期内报文到达时间间隔的均值，其数学表达式为：

$$\text{AvgPktInterval} = (t^M - t^1) / M。$$
- 报文大小一阶差分均值 (AvgPktDiff)：监测周期内报文大小序列一阶差分后序列绝对值的均值，其数学表达式为：

$$\text{AvgPktDiff} = \sum (|l^2 - l^1| + |l^3 - l^2| + \dots + |l^M - l^{M-1}|) / (M - 1)。$$
- 报文大小均方差 (StdPktSize)：监测周期内报文大小序列的均方差，其数学表达式为：

$$\text{StdPktSize} = \sqrt{\sum (l^i - \text{AvgPktSize})^2 / M}。$$
- 最大报文持续时间 (MaxPktDurTime)：监测周期内报文到达时间间隔小于某个阈值的所有报文持续时间的最大值，其数学表达式为：

$$\text{MaxPktDurTime} = \max(\text{pktdurtime}^1, \text{pktdurtime}^2, \dots, \text{pktdurtime}^q, \dots)，$$
其中， $\text{pktdurtime}^i = \sum (\dots + |t^h - t^{h-1}| + \dots)$, if $|t^h - t^{h-1}| < \text{threshold}$ 。

2.2.2 基于强化学习剪枝优化决策树感知模型

决策树模型训练构建过程通过基尼系数或者信息熵等计算进行子节点的切分，按照一定的固定策略方向进行训练收敛^[17]。这种方式训练出来的模

型可能会依赖训练样本数据集的分布,造成在实际环境应用过程中存在泛化性问题;另外,模型的构建以结果的好坏为导向,可能训练好的模型在分类的时间和空间上都不具有最优性,模型较大对设备的部署不友好。基于此原因,考虑使用强化学习进行预剪枝训练优化决策树模型。

(1) 强化学习: 强化学习的基本思想是在某一时刻的 $s (s \in S)$ 状态下, 智能体 (agent) 采取动作 $a (a \in A)$ 获得收益期望, 环境 (environment) 会根据 agent 的动作反馈相应的回报 (reward) $r (r \in R)$, 如图 3 所示。本文使用的强化学习为异策略 (off-policy) 的 Q-learning 和同策略 (on-policy) 的 Saras, 两者本质的区别在于动作策略和需要评估的策略是否是同一个策略。Saras 在每次迭代过程中使用相应策略 (如 ε -贪婪策略) 选择动作 a' 更新 $Q(s, a): Q(s, a) = Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$, $s \leftarrow s', a \leftarrow a'$, 其中 γ 为折扣因子, 取值范围为 $[0, 1]$, 并且动作 a' 作为下一轮迭代采取的动作。而 Q-learning 在每轮迭代中使用 $\max_a Q(s, a)$ 更新 $Q(s, a): Q(s, a) = Q(s, a) + \alpha[r + \gamma \max_a Q(s', a') - Q(s, a)]$, $s \leftarrow s'$, 但是在下一轮迭代中重新采用相应策略 (如 ε -贪婪策略) 选择新的动作 a 。

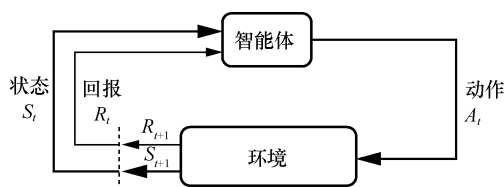


图3 强化学习基本思想流程

(2) 强化预剪枝建树: 决策树模型在训练过程中有个重要的超参数, 即树的深度。这个参数如果设置得太高, 就会过拟合, 设置得太低又会欠拟合, 两者都无法很好地学习出数据的模式。因此, 本文采用强化预剪枝方法构建决策树, 通过限制叶子节点的数目来预剪枝, 同时修改决策树构建过程, 使用堆的方式构建决策树。该方式中, 节点在入堆前, 除了节点的样本, 还有通过

算法计算的当前节点的特征。出堆后, 计算的是当前节点的左右子节点的特征, 以及相对于父节点获得的得分提升, 并将子节点入堆。树在叶子节点达到设置的阈值时结束构造。堆中的排序方式为: 节点相对于父节点获得的得分提升, 即先构造提升最大的节点, 不纯度高的节点会优先被处理, 最大化降低节点不纯度, 构造一棵泛化能力较好的树。

在上述通过堆的方式构建决策树过程中, 叶子节点数目难以确定, 因此, 可以与强化学习结合, 通过强化学习搜索最佳的叶子节点数目。通过搜索策略, 搜索最大叶子节点数到最小叶子节点数区间的叶子数目, 期望寻找一条最优、 r 最大的轨迹。这条轨迹代表最优叶子节点集合, 实现整体性能最优的分类决策树模型。

通过强化学习预剪枝训练决策树模型的过程如图 4 所示。

- 智能体模块里环境状态: 当前指定的叶子节点数目, 用于指导构建决策树。叶子节点的数目决定了预构建决策树的复杂程度, 也直接影响剪枝时需要修剪的幅度。
- 动作空间行动: 增加或者减少叶子节点数目, 调整的幅度根据迭代步数进行缩小。这里的动作设定为增加或者减小叶子节点的数目, 而不是直接把节点数目作为动作, 这样做的好处是一方面可以减小动作空间, 加快收敛速度; 另一方面, 可以将单步的调整转变为多步寻优, 引入轨迹的概念, 避免陷入局部最优。
- 环境返回的奖励: 每一次迭代奖励为根据新的状态值叶子节点数生成的决策树在测试集上的正确率相对于基础决策树的评分的提升累计。这里采用累积提升, 这样可以尽可能地避免强化学习陷入局部最优。因为探索时, 某个动作可能会在测试数据集上表现较好, 但是却不具备很强的泛化

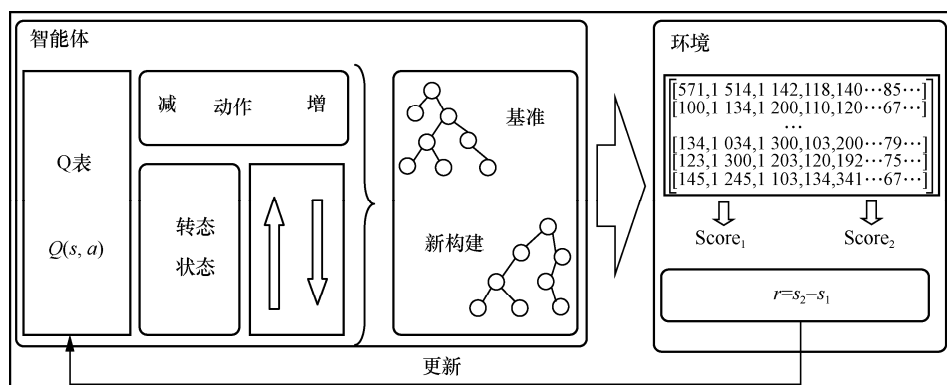


图4 通过强化学习预剪枝训练决策树模型的过程

性。但是，沿着一条回报都很高的轨迹剪枝得到的决策树，其性能会比较稳定，在实测数据上也会具备较强的泛化性。

- 学习策略：在进行动作选择时，并不总是选择期望最大的动作，根据贪婪度的设定，训练开始时让强化学习自己探索，随着训练的进行，在探索比例逐渐衰减到 90% 的时刻选择期望最大的动作，剩下的 10% 让强化学习探索，而且根据迭代步数，不断地衰减动作的幅度，这种流程可以获得更快的收敛速度，也可以获得更高的精度。

(3) 整体探索路径策略规划：在使用强化学习搜索的过程中，奖励的设定是极为重要的，其次是动作的幅度，不合适的设定将会导致训练收敛慢或者陷入局部循环无法收敛。

在训练开始时，预先建立一棵最大的树，这棵树将会作为基准决策树，作为模型正确率提升的基线。在探索过程中，不同状态（叶子节点数）对应的决策树会有不同的结果，将所有结果和基准决策树进行比较并计算提升值，可以根据提升值的变化趋势找到一个很好的探索方向。若奖励为负，则所探索的方向是错误的，将减少在此状态下指定方向上的探索期望；若奖励为正，则提高在当前状态下指定方向的探索期望，鼓励在当前方向上的探索欲。将前 N 步累计的回报作为当前状态下的奖励。

还有动作的设计，动作不只有增减两个操作，还有动作幅度大小的设定。大的动作幅度更容易在探索过程中跳过局部循环，探索更多的状态；根据迭代步数的增加，减少动作幅度，小的动作幅度更加容易在局部探索最优状态。强化学习探索动作的轨迹和对应决策树模型的正确率的关系如图 5 所示，可以看出探索动作的轨迹可以很好地跳出局部最优解，其中 action 指探索动作的轨迹，Acc 指正确率。在多个 epoch 下，每个 epoch 探索一条路径，并更新 Q 表，逐步填充 Q 表下状态和动作的期望值。

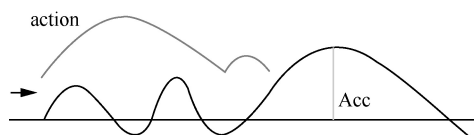


图5 强化学习探索动作的轨迹和对应决策树模型正确率的关系

3 实验对比与分析

3.1 实验数据集说明

实验中用以测试验证强化学习剪枝优化决策树模型性能的网络流量数据集来源包括两方面：一是通过搭建采集环境，收集了 400 多个应用程序产生的流量，总数据量为 187 GB，采集应用的平台包括 Windows、Android 和 iOS 操作系统；二是从某运营商的广域网环境中采集，数据经过去隐私处理后使用，总数据量为 78 GB。采集的

数据都是无标签的原始网络流报文数据,按照第2.2.1节进行标注和特征提取等相关处理工作。

3.2 实验与分析

由于在实际网络设备上部署,捕获的现网流量经过时延敏感属性感知后还需要执行策略动作,因此,不可能使用完整的一条流进行特征提取处理。同时由于转发类网络设备内存限制以及处理的时延性问题,也不可能先缓存大量数据再进行特征提取和模型推理。所以,在实现时,设置设备每隔200 ms计算提取一次特征,一条流的提取特征持续时长至多为20 s,相当于一组流至多可提取100个点的特征。

3.2.1 模型单次推理情形

模型单次推理是指以五元组信息标识的一条流只做一次特征提取和推理,得到这条流时延敏感属性等级。通过Q-learning预剪枝训练得到的决策树(以下简称优化决策树)模型和原始决策树模型的正确率对比如图6所示。从图6可以看出,与原始决策树模型相比,优化决策树模型在相同的测试集数据上,正确率提升了1.75%。同时,训练集和测试集的正确率之间差距明显缩小,模型泛化性得到提高。原始决策树模型训练集正确率达99.98%,和测试集正确率差了15.94%,模型存在一定程度的过拟合,这是因为训练集和测试集正确率相差越小模型泛化性就越好。优化决策树模型在训练集和测试集之间正确率差距为4.02%,与原始决策树模型相比,差距缩小了11.92%,模型的过拟合程度有所改善。

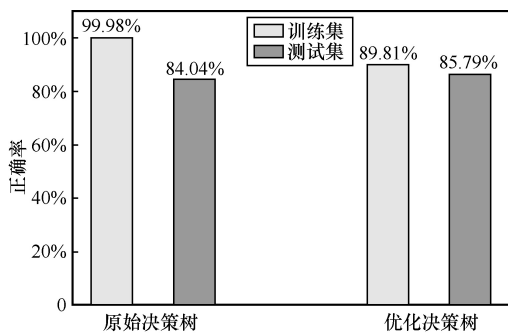


图6 优化决策树模型与原始决策树模型的正确率对比

优化决策树模型和原始决策树模型的规模和推理耗时对比见表5。原始决策树是在默认参数情况下,任其自由分支生长构成的最大决策树。因此,模型在训练过程中为了保证在训练集上得到高正确率,叶子节点数高达69 694个。而优化决策树模型,在测试集正确率提升1.75%情况下,叶子节点的个数也只有8 904个,比原始决策树模型缩小了约87.2%。优化后的模型更轻量化,更易于在网络设备上部署。

表5 优化决策树模型和原始决策树模型的规模和推理耗时对比

模型	叶子节点数/个	推理耗时/s
原始决策树模型	69 694	0.333 1
优化决策树模型	8 904	0.231 3

强化学习和暴力穷举搜索最优决策树模型效率对比如图7所示。可以看出,使用暴力穷举法搜索叶子节点参数时,需要遍历所有的叶子节点数,对比所有搜索结果选择出最优的决策树模型。对于叶子节点数为69 694个的基准模型,需要搜索69 594次(设置最小叶子节点数至少为100个)。而使用强化学习搜索叶子节点数参数时,可以增减叶子节点数和动态调整增减动作的幅度,不需要逐个遍历,可大幅度提高搜索的效率。如图7所示,强化学习只需要搜索7次即可找到最优解,搜索效率提升了99.99%。

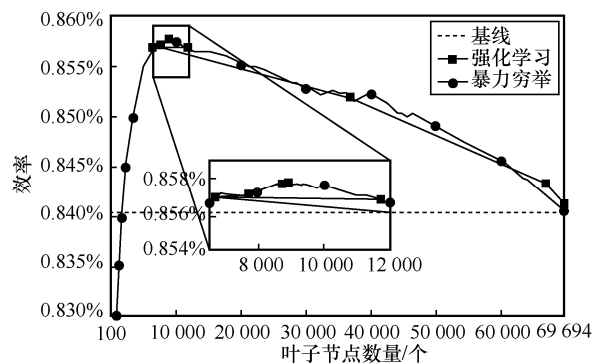


图7 强化学习和暴力穷举搜索最优决策树模型效率对比

3.2.2 不同推理方式情形

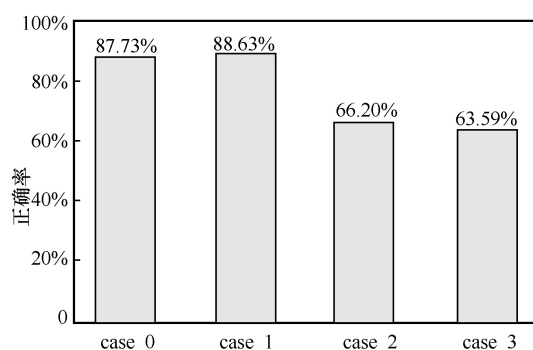
实际部署时考虑到设备的性能和资源问题,



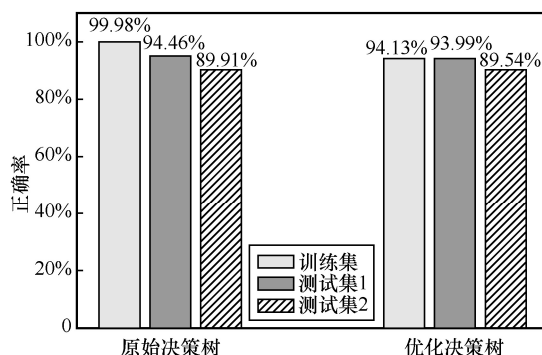
每隔 200 ms 提取一次特征。但是如果只对一条流的五元组信息进行特征计算和时延敏感属性感知,由于只利用了一条流,200 ms 只包含很少信息,会造成推理结果达不到实际需求指标。因此,通过离线实验的方式探索不同推理方式对模型感知结果的影响。推理方式对模型正确率的影响如图 8 所示。在图 8 (a) 中,对比分析了 4 种推理方式的情况。其中,case_0 表示 2 s 为一次预测区间,即每隔 200 ms 提取一次特征,缓存 10 个点特征后,就达到 2 s,输入模型推理,则相当于有 10 个样本,推理结果通过投票表决得到最终结果;case_1 表示 4 s 为一次预测区间,则表示缓存 20 个点的特征,相当于 20 个样本,模型通过推理投票表决最终结果;case_2 将 case_1 中缓存的 10 个点特征计算均值,合并为一个样本输入模型推理;case_3 将 case_1 中缓存的 20 个点特征计算均值,合并为一个样本输入模型推理。从图 8 (a) 可以看出,使用更多的局部特征进行多次推理投票可以获取更好的正确率,例如,case_1 相对于 case_0 多使用了 10 个点局部特征,正确率提升了 0.9%,达到了 88.63%。相反,将更多的局部特征均值化处理,会造成大量局部信息丢失,使得识别推理的正确率严重下降。例如,case_2 相对于 case_0 做统计特征均值处理后,正确率下降了 21.53%;case_3 相对于 case_1 做统计特征均值处理后,正确率下降了 25.04%。因此,在推理方式上,相对于使用长时间流的均值化宏观行为统计特征,使用多个局部微观统计特征进行多次推理更有助于提升识别推理的正确率。

在实际设备部署时,考虑设备性能问题,采取对一条以五元组标识的流每隔 200 ms 提取一次特征,缓存 100 个点特征后进行推理投票得到结果。在进行多次推理时,优化决策树模型相对于原始决策树模型在测试集 1 上并没有表现出很大提升,只相差 0.47%,但是模型规模缩小了约 67.4% (原始决策树叶子节点数为 68 475,优化决

策树叶子节点数为 22 295)。图 8 (b) 中,测试集 1 表示提取缓存 100 个点特征进行 100 次推理投票的情形,测试集 2 表示提取缓存的特征点数不确定,存在小于 100 个点特征的情形。实际离线采集的报文数据流由于人为截断等因素,存在一些流不满足持续时长 20 s。这些流若按照每隔 200 ms 提取一次局部统计特征,则会不满足 100 个点特征,也就无法满足 100 次推理投票的推理方式。测试集 2 和测试集 1 的差值表示在提取缓存特征点数不确定的情形下,推理识别的正确率有所下降,优化决策树的正确率下降约 4.45%。



(a) 不同推理方式下模型正确率对比



(b) 不同推理次数下正确率对比

图 8 推理方式对模型正确率的影响

考虑网络业务流量上下流行为的差异性,对流量的上行和下行流量行为分别训练模型,分析其对推理结果的影响。上下行流模型性能对比见表 6。对于下行流,优化前后的模型在测试数据集上的表现相当,优化决策树模型的测试集正确率比原始决策树提升了 0.49%;对于上行流,优化决策树模型比原始决策树模型略微降低 0.11%。

表6 上下行流对应模型性能对比

模型	原始决策树			优化决策树		
	训练集正确率	测试集正确率	叶子节点数/个	训练集正确率	测试集正确率	叶子节点数/个
下行流模型	99.99%	86.39%	56 169	93.64%	86.88%	22 259
上行流模型	99.94%	98.05%	5 683	98.97%	97.94%	2 493
综合	90.57%			90.85%		

可以看出,在测试集上,上行流模型明显正确率更高,达到98%左右,下行流模型只有86%左右,这主要是因为很多业务的下行流行为存在相似性,区分度不高。整体综合上下行流来看,优化决策树模型在正确率方面高于原始决策树模型0.28%,同时模型规模缩小了60.0%左右。

3.2.3 不同强化学习优化效果对比

不同强化学习优化前后决策树模型正确率对比如图9所示。从图9可以看出,在测试集上,不论使用哪种强化学习,相对于原始决策树模型,优化决策树模型的正确率都会提升,提升约1%。同时,Saras具有更好的表现,相比Q-learning算法在测试集上正确率提升了0.28%,模型也更小,相比原始决策树模型缩小了约63.0%,而Q-learning比原始决策树模型缩小了约41.1%。优化前后决策树模型规模对比见表7。

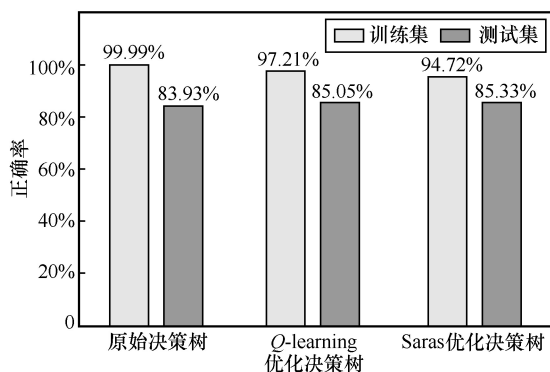


图9 不同强化学习优化前后决策树模型正确率对比

表7 优化前后决策树模型规模对比

模型	叶子节点数/个
原始决策树	69 525
Q-learning 优化决策树	40 930
Saras 优化决策树	25 750

4 结束语

随着网络业务的不断增加,网络承载的流量爆炸式增长,网络的架构也日趋复杂化。因此,网络流量的调度技术在网络领域的研究中尤其重要。目前,网络流量的调度技术都是确保特定目标最优化,以合理最大化资源利用。但是基本无法做到智能按需的流量调度,尤其对于网络突发故障造成的拥塞以及高价值业务的护航场景。针对此问题,本文提出了网络流量时延敏感属性智能感知方法,使用AI技术将网络流量的行为特征映射得到其时延敏感性,基于感知的时延敏感优先级对流量进行按需调度。考虑实际设备的部署问题,采用了可解性强的决策树模型,同时采用强化学习优化决策树模型,使模型更具鲁棒性和更加轻量化。通过采集的真实网络流量对提出的方法进行实验验证,实验表明基于强化学习优化的决策树模型规模缩小了60.0%~87.2%,单次推理模型正确率提升约1.75%,推理速度提升约30%,同时,模型的泛化性也有所改善。

参考文献:

- [1] 张彤,冯佳琦,马延滢,等.时间敏感网络流量调度综述[J].计算机研究与发展,2022,59(4):747-764.
ZHANG T, FENG J Q, MA Y Y, et al. Survey on traffic scheduling in time-sensitive networking[J]. Journal of Computer Research and Development, 2022, 59(4): 747-764.
- [2] 李文信,齐恒,徐仁海,等.数据中心网络流量调度的研究进展与趋势[J].计算机学报,2020,43(4):600-617.
LI W X, QI H, XU R H, et al. Data center network flow scheduling progress and trends[J]. Chinese Journal of Computers, 2020, 43(4): 600-617.
- [3] AURELI D, CIANFRANI A, DIAMANTI A, et al. Going



- beyond DiffServ in IP traffic classification[C]//Proceedings of NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium. New York: ACM Press, 2020: 1-6.
- [4] ALIZADEH M, YANG S, SHARIF M, et al. pFabric: minimal near-optimal datacenter transport[J]. ACM SIGCOMM Computer Communication Review, 2013, 43(4): 435-446.
- [5] BAI W, CHEN L, CHEN K, et al. PIAS: practical information-agnostic flow scheduling for commodity data centers[C]//Proceedings of IEEE/ACM Transactions on Networking. Piscataway: IEEE Press, 2017: 1954-1967.
- [6] CHEN L, LINGYS J, CHEN K, et al. AuTO: scaling deep reinforcement learning for datacenter-scale automatic traffic optimization[C]//Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. New York: ACM Press, 2018: 191-205.
- [7] Intelligent buffer management on Cisco Nexus 9000 series switches white paper[EB]. 2017.
- [8] Application-aware routing[EB]. 2022.
- [9] WU M K, HUANG H, SUN Y, et al. Active Keeper: an accurate and efficient algorithm for finding top- k elephant flows[J]. IEEE Communications Letters, 2021, 25(8): 2545-2549.
- [10] XIE K, TIAN J Z, WANG X, et al. Efficiently inferring top- k elephant flows based on discrete tensor completion[C]//Proceedings of IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2019: 2170-2178.
- [11] LOU K H, YANG Y J, WANG C C. An elephant flow detection method based on machine learning[C]//Proceedings of 2019 International Conference on Smart Computing and Communication. Cham: Springer, 2019: 212-220.
- [12] AL-SAAD M, KHAN A, KELEFOURAS V, et al. Unsupervised machine learning-based elephant and mice flow identification[C]//Proceedings of the 2021 Computing Conference. Cham: Springer, 2021: 357-370.
- [13] ZUO P, SHU Y A. An elephant flows scheduling method based on feedforward neural network[C]//Proceedings of 2021 World Conference on Computing and Communication Technologies. Piscataway: IEEE Press, 2021: 16-20.
- [14] Configuration guidelines for DiffServ service classes[EB]. 2006.
- [15] WANG W, ZHU M, ZENG X W, et al. Malware traffic classification using convolutional neural network for representation learning[C]//Proceedings of 2017 International Conference on Information Networking. Piscataway: IEEE Press, 2017: 712-717.
- [16] MOORE A W, ZUEV D. Discriminators for use in flow-based classification[R]. 2005.
- [17] SUTHAHARAN S. Decision tree learning[M]// Machine learning models and algorithms for big data classification. Boston: Springer, 2016: 237-269.

[作者简介]



王雪荣（1997-），女，中国电信股份有限公司研究院工程师，主要研究方向为 IP 网络技术与下一代互联网。



唐政治（1991-），男，博士，北京华为数字技术有限公司高级工程师，主要研究方向为网络流量分析、网络人工智能技术。



李银川（1994-），男，博士，北京华为数字技术有限公司高级工程师，主要研究方向为人工智能技术。

齐美玉（1983-），女，博士，北京华为数字技术有限公司技术专家，主要研究方向为网络人工智能技术。

朱建波（1979-），男，北京华为数字技术有限公司技术专家，主要研究方向为骨干路由器架构设计。

张亮（1982-），男，博士，北京华为数字技术有限公司技术专家，主要研究方向为智能故障分析、网络流量分析和网络优化。