计算机科学
COMPUTER SCIENCE

EvoTrace:基于非线性数据包遥测和批处理的轻量级带内网络遥测方法

王攀祥, 崔允贺, 申国伟, 郭春, 陈意, 钱清

引用本文

王攀祥, 崔允贺, 申国伟, 郭春, 陈意, 钱清. EvoTrace:基于非线性数据包遥测和批处理的轻量级带内网络遥测方法[J]. 计算机科学, 2025, 52(5): 291-298.

WANG Panxiang, CUI Yunhe, SHEN Guowei, GUO Chun, CHEN Yi, QIAN Qing. EvoTrace:A Lightweight In-band Network Telemetry Method Based on Nonlinear Embedding and Batch Processing [J]. Computer Science, 2025, 52(5): 291-298.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240100164

EvoTrace:基于非线性数据包遥测和批处理的轻量级带内网络遥测方法

王攀祥[1,2,3] 崔允贺[1,2,3] 申国伟[1,2,3] 郭 春[1,2,3] 陈 意[1,2,3] 钱 清[4]
1 贵州大学计算机科学与技术学院文本计算与认知智能教育部工程研究中心 贵阳 550025
2 贵州大学计算机科学与技术学院公共大数据国家重点实验室 贵阳 550025
3 贵州省软件工程与信息安全特色重点实验室 贵阳 550025
4 贵州财经大学信息学院 贵阳 550025
(panxiangw21@163.com)

摘 要 带内网络遥测(In-band Network Telemetry,INT)使数据包能够携带网络状态信息,具有较高的测量准确性和精度。然而,这种提升是以增加数据平面开销为代价的。遥测信息的嵌入会导致数据平面的网络开销过大。同时,现有的遥测方法通常对大流的数据包进行大量的测量,忽略了对中小流数据包的测量。为了克服上述问题,提出了一个轻量级的 INT 方法——EvoTrace。EvoTrace 设计了一种非线性数据包遥测方法,根据网络流的属性监测不同流的数据包。此外,EvoTrace 还采用元数据批处理的方式对遥测元数据进行聚合,以减少网络带宽占用和遥测数据包的数量。在 OpenvSwitch(OVS)上实现了 EvoTrace 并进行了测试,实验结果表明,与现有方法相比,EvoTrace 在提高网络流监测覆盖率的同时,节省了 40%以上的 INT 带宽占用。

关键词:网络遥测;可编程数据平面;带内网络遥测;网络测量;软件定义网络
中图分类号 TP393

EvoTrace:A Lightweight In-band Network Telemetry Method Based on Nonlinear Embedding and Batch Processing

WANG Panxiang[1,2,3],CUI Yunhe[1,2,3],SHEN Guowei[1,2,3],GUO Chun[1,2,3],CHEN Yi[1,2,3] and QIAN Qing[4]
1 Engineering Research Center of Text Computing & Cognitive Intelligence,Ministry of Education,College of Computer Science & Technology,Guizhou University,Guiyang 550025,China
2 State Key Laboratory of Public Big Data,College of Computer Science & Technology,Guizhou University,Guiyang 550025,China
3 Provincial Key Laboratory of Software Engineering and Information Security,Guizhou University,Guiyang 550025,China
4 School of Information,Guizhou University of Finance and Economics,Guiyang 550025,China

Abstract In-band network telemetry(INT) enables packets to carry network state information,achieving high monitoring accuracy and precision. However,this advancement comes at the cost of increased data plane overhead. The embedding of telemetry information results in excessive network overhead within the data plane. Meanwhile,existing telemetry methods usually measure large number of packets from large flows,fails in measuring packets from small and medium flows. To address these issues,this paper proposes a lightweight INT method—EvoTrace. EvoTrace introduces a nonlinear packet telemetry method that monitors packets from different flows according to the attributes of network flows. Additionally,EvoTrace also employs a metadata batching method to aggregate the telemetry metadata,for reducing network bandwidth occupancy and the number of telemetry packets. EvoTrace is implemented on OpenvSwitch(OVS) and tested,experimental results demonstrate that,compared with the existing methods,EvoTrace achieves larger network flow monitoring coverage while saves more than 40% of INT bandwidth occupancy.

Keywords Network telemetry,Programmable data plane, In-band network telemetry, Network measurement, Software-defined network

到稿日期:2024-01-22 返修日期:2024-04-02
基金项目:国家自然科学基金(62102111);贵州省科技计划项目(黔科合基础-ZK[2022]重点 011);贵州省高等学校大数据安全与网络安全创新团队(黔教技[2023]052)
This work was supported by the National Natural Science Foundation of China(62102111),Guizhou Provincial Science and Technology Plan(Qian Ke He Jichu-ZK[2022] Key 011) and Big Data Security and Network Security Innovation Team of Guizhou Provincial High Education Institution([2023]052).
通信作者:崔允贺(yhcui@gzu.edu.cn)

1 引言

细粒度和全网可见性是数据中心网络等高密度和超大规模网络中的关键问题[1-2]。新型业务和现代网络基础设施对网络测量技术提出了新要求[3]。然而,传统的网络测量技术已不能满足如今网络测量准确性、实时性和通用性的需求。例如,SNMP[4]采用轮询机制收集整个网络状态信息,无法实时获取网络数据流量信息。NetFlow[5]和 sFlow[6]通过对网络流量进行采样来分析相关的网络活动,不能满足细粒度和实时性的测量要求。

近年来,可编程数据平面(Programmable Data Plane, PDP)[7]技术的出现催生了网络遥测技术[8],这是一种自动采集、处理和分析网络状态信息的新型网络测量技术。不同于传统网络测量技术,带内网络遥测(In-band Network Telemetry, INT)[9]将数据包转发与网络测量结合,网络节点将每个数据包、流和交换机的状态信息嵌入遥测数据包,对网络进行实时测量,此过程无需网络控制平面的干预。因此,带内网络遥测具有实时性强、测量粒度细和无需控制平面干预等优点。

尽管 INT 可以提供详细的遥测信息,但其可能给网络带来较大的开销,严重影响网络应用程序的性能。具体地,根据数据包的修改级别和遥测信息的传输方式,INT spec v2.1[10]提出了 3 种 INT 应用模式:INT-XD(In-band Network Telemetry eXport Data),INT-MX(In-band Network Telemetry eMbed instruct(X)ions)和 INT-MD(In-band Network Telemetry eMbed Data)。在 INT-MD 模式下,由于转发路径上的每个节点都将实时状态嵌入到每个经过的数据包中,因此数据包的大小和带宽开销随路径长度呈线性增加。此外,由于受到网络的最大传输单元(Maximum Transmission Unit, MTU)的限制,因此可能无法收集到路径后面部分节点的遥测信息。在 INT-XD 和 INT-MX 模式中,每个遥测数据都需要产生一个数据包。因此,这两种模式也会在网络中产生大量的小型遥测数据包,增加网络带宽开销和处理冗余。为了解决上述问题,一些 INT 方案使用采样方法以减少网络开销[11-17]。其中,sINT[11]根据网络状态调整遥测元数据的插入比例,在降低网络开销的同时实现对数据平面事件的检测。与原始 INT 相比,其网络开销减少了 37%。Suh 等[12]提出了 FS-INT——一种灵活的基于采样的 INT 方案。FS-INT 中,INT 源节点每隔 R 个数据包插入一个 INT 头。INT-label[16]定期将设备内部状态标记到业务数据包,使遥测能够以最小的带宽开销覆盖整个网络。DINT[17]根据网络流吞吐量的变化调整插入遥测信息的时间间隔,能够以较小的遥测开销收集细粒度的网络状态信息。因此,基于采样的 INT 方案成为最流行的解决方案之一。然而,现有的基于采样的网络遥测方法和 INT 模式存在以下问题。

1)缺乏监测中小流的能力。大多数现有 INT 方案在选择网络遥测嵌入位置时没有考虑流的大小,因此,这些方案在占网络流总数的极少部分的大流中嵌入了过量的遥测信息。尽管这些方案有效地保留了包级遥测信息,但无法较完整地保留小流或中流的流级遥测信息。对于网络异常检测等需要较完整的流级遥测信息的场景,这些方案将会产生较大的估计误差。

2)缺乏聚合小型遥测数据包的能力。在 INT-MD 模式下,一旦遥测信息采集完成,其将在数据平面产生遥测报告并传输给监控程序。通常,以太网数据包的最小长度为 64 字节。当元数据较小时,需要在数据包尾部填充无效数据。此外,INT 元数据的数量与路径的长度成正比,会在网络中产生大量小型数据包,从而导致链路后续交换机处理冗余和增加监控引擎工作负载。在文献[18]中,NeetSeer 使用流事件批处理最小化开销。然而,由于网络事件之间可能存在较长的间隔,NeetSeer 可能需要较长的时间来获取一些网络事件,因此其不适合一些事件优先级较高的应用。

为了解决上述问题,本文提出了一种基于非线性数据包遥测和元数据批处理的轻量级带内网络遥测方法——EvoTrace。本研究的目标是在最小化数据平面开销的同时最大程度地保留 INT 实时性强、测量粒度细、网络视图详细等优点。为了实现这一目标,EvoTrace 设计了一种非线性数据包遥测方法,其中数据包的监测概率随着流大小的增加而降低。因此,EvoTrace 能够提高对中小流数据包的测量能力,同时避免在大流中测量过多的数据包。为了解决聚合小型遥测数据包的问题,EvoTrace 在 Linux 内核中引入了元数据批处理方法。EvoTrace 将多个遥测信息封装在一个数据包中,然后将其传输到监控服务器。该方法不仅减少了带宽和处理开销,还可以缓解监控服务器的过载问题。同时,为了保证遥测信息的及时传递,EvoTrace 引入了超时机制。一旦环形缓冲区满或超过时间阈值,缓冲区中的所有遥测信息将被封装在数据包中并发送到监控服务器。

本文的主要贡献如下:

1)提出了一种基于非线性数据包遥测和元数据批处理的轻量级带内网络遥测方法——EvoTrace。EvoTrace 设计了一种非线性数据包遥测方法,通过以与流量大小相适应的概率对每个遥测信息进行采样,获取所有流的信息。这不仅保证了应用需求,降低了遥测系统的带宽和处理开销,而且提高了对中小流数据包的测量效率。此外,EvoTrace 还提出了一种元数据批处理方法,在保证遥测信息不变的情况下减少了带宽开销和链路中的遥测报告数量,同时减轻了网络遥测服务器的压力。

2)在广泛使用的可编程虚拟交换机 Open vSwitch (OVS)[19]上实现了 EvoTrace。据我们所知,现有的方案[20-21]仅在 OVS 上实现了 INT-MD 模式,EvoTrace 是首个在 OVS 上尝试实现 INT-MX 模式的方案。

3)对 EvoTrace 进行了性能评估。实验结果显示,EvoTrace 在提高网络流遥测覆盖率的同时,将遥测数据包数量减少了 20 倍以上。在遥测开销为 0.5%时,带宽开销能降低 40%以上。

2 INT 概述

根据数据包修改程度,INT spec v2.1 提出了 3 种应用模式:INT-XD,INT-MX 和 INT-MD。表 1 列出了 3 种 INT 应用模式的综合比较。在 INT-XD 模式下,INT 节点根据其流表中配置的 INT 指令,将元数据直接从数据平面发送到监控

系统，无须修改数据包。在 INT-MX 模式下，INT 源节点将 INT 指令嵌入数据包头部，然后源节点、中间节点和宿节点按照数据包中嵌入的 INT 指令直接将元数据发送到监控系统。宿节点在将数据包转发给接收方之前剥离 INT 指令，将数据包恢复原样。数据包修改仅限于头部。在 INT-MD 模式下，INT 指令和元数据都嵌入数据包中。这是经典的逐跳 INT，其中源节点嵌入 INT 指令，中间节点和宿节点嵌入元数据。宿节点从数据包中剥离 INT 指令和元数据堆栈，并选择性地将元数据发送到监控服务器。在这种模式下数据包修改最多，但它最小化了监视系统从多个 INT 节点整理报告的开销。

表 1 各种 INT 应用模式的比较

Table 1 Comparison of various INT application modes

应用模式	MTU 限制	遥测报告数量	实时性	灵活性	数据包修改
INT-XD	无	高	高	中	无
INT-MX	无	高	高	高	指令
INT-MD	是	低	低	高	指令和元数据

数据来源：INT spec v2.1[10]整理。

与经典的 INT-MD 不同，INT-XD 和 INT-MX 的遥测元数据并不是依次嵌入转发的数据包中，而是在需要进行遥测的数据包经过转发设备时直接将当前的元数据信息组成遥测报告数据包发送到监控服务器。INT-MD 是逐跳承载数据，INT-XD 和 INT-MX 是逐跳上报元数据，因此 INT-XD 和 INT-MX 的遥测数据包的数量远远大于 INT-MD。相比 INT-MD，INT-XD 和 INT-MX 具有以下优点。

1）不受数据包 MTU 限制。INT-MD 嵌入数据包中的遥测数据的数量受限于其原始大小与 MTU 之间的差异，达到 MTU 后，无法收集路径后面节点的信息。

2）遥测数据不会随业务数据包一起丢失，这对网络管理员再现业务数据包丢失前的转发路径状态至关重要。

3）适用于组播。INT-MD 在组播场景下不可避免地会带来大量重复的 INT 信息收集，增大了网络带宽。

虽然 INT-XD 模式和 INT-MX 模式具有上述诸多优点，然而随着网络规模的增长，遥测报告数据包的数量与路径长度呈比例增加，从而在链路中产生大量小的遥测报告数据包，这不但会增加网络带宽和设备开销，而且存在扩展性问题。此外，INT-XD 根据流监控列表收集遥测信息，而 INT-MX 与 INT-MD 由数据包头携带遥测指令位图。因此，INT-XD 难以灵活调整测量对象，测量灵活性更低。综上所述，这 3 种操作模式在 MUT 限制、实时性、灵活性等方面需要权衡。

3 基于非线性数据包遥测和元数据批处理的轻量级带内遥测框架

EvoTrace 的架构如图 1 所示，其主要工作流程分为以下 3 个步骤。

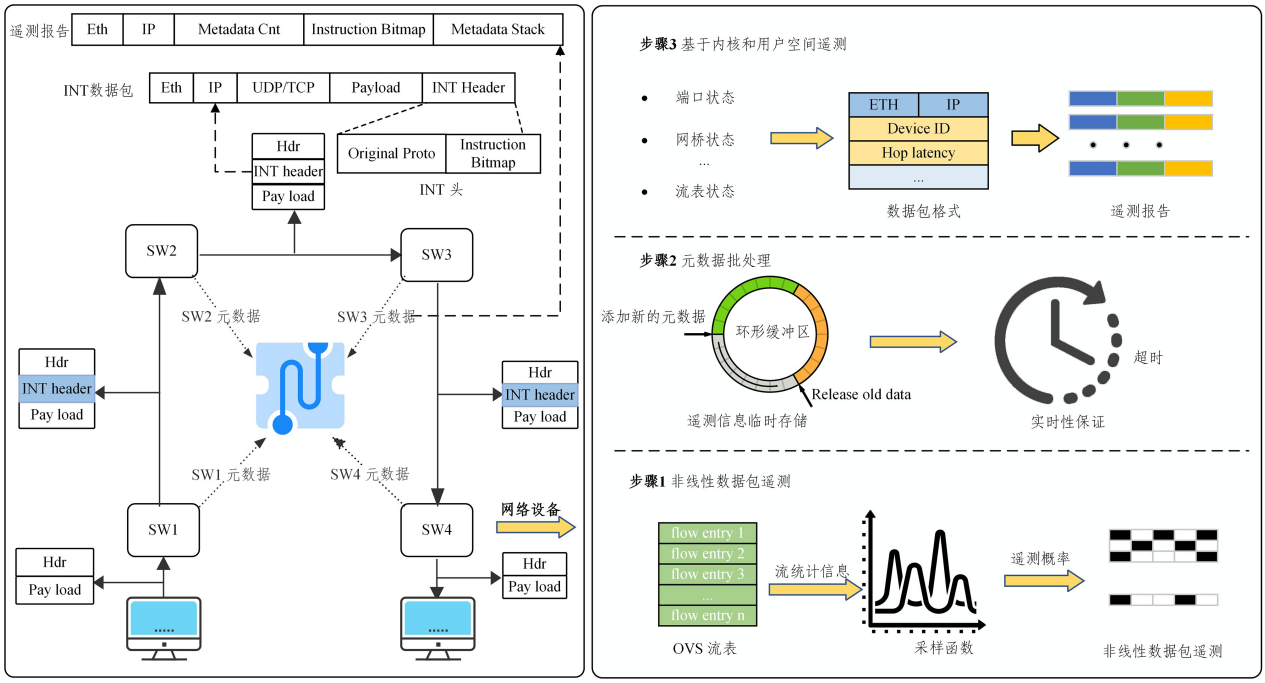


图 1 EvoTrace 架构

Fig. 1 EvoTrace architecture

步骤 1 非线性数据包遥测。EvoTrace 首先获取 SDN 交换机中的网络流信息；然后设计了一个簇采样函数，使数据包遥测概率为所属流当前大小的递减函数，并通过采样函数和当前流大小计算数据包监测概率；最后根据数据包监测概率确定是否收集遥测信息，若是，则收集遥测信息并嵌入数据包，否则直接转发数据包。

步骤 2 元数据批处理。当需要遥测的数据包经过交换机时，EvoTrace 设计了环形缓冲区，使用该缓冲区临时存储其遥测信息，直到可以将其复制到遥测报告中。同时，为了保证遥测信息的实时传递，EvoTrace 设计了超时机制。一旦缓冲区满或遥测信息在数据平面的存储时间间隔超过阈值，则将缓冲区中的所有元数据封装成一个数据包发送到监控服务器。

步骤 3 基于内核和用户空间的网络遥测。OVS 主要由一个内核模块和一个用户空间进程组成。在实现 INT 特性时，需对这两个组件进行扩展。INT 源节点将 INT 收集指令

作为 INT 头的一部分嵌入数据包中。随后，沿着数据包转发路径的所有中间节点将解析 INT 头并收集所请求的数据平面状态。在最后一跳时，INT 宿节点会删除 INT 头，然后将数据包发送到目的地。通过该方式，监控系统对终端主机变得透明。此外，EvoTrace 引入了一种新的遥测数据包格式。

3.1 非线性数据包遥测

如图 2 所示，基于系统采样的数据包遥测方法以固定的数据包间隔监视数据包，基于随机采样数据包遥测方法以一定的概率监测处理数据包。这两种方法都以固定的概率遥测数据包，没有考虑网络中不同大小网络流的数据包比例。由于网络流的流量大小服从幂律分布，因此，基于系统采样和随机采样的方法从占网络流总数的极少部分的大流中嵌入了过量的遥测信息，而占网络流总数的绝大多数的中、小流则被嵌入了很少甚至 0 个遥测信息。基于系统采样和随机采样的方法的流越小，流信息丢失的概率越大。同时，上述方法的采样间隔越大或概率越小，流信息丢失的概率越大。最后，在使用上述方法时，流传输的路径越短，流信息丢失的概率越大。为解决以上问题，本文设计了一种非线性数据包遥测方法。

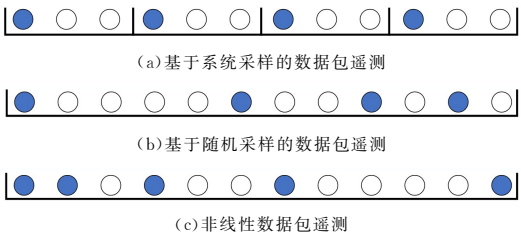


图 2 不同遥测方法

Fig. 2 Different telemetry methods

数据包并非同时到达交换机，因此无法为大流和小流分别选择不同的固定嵌入概率，以达到降低大流的遥测信息嵌入数量的目的。然而，SDN 交换机的流表实时记录了每条数据流的统计信息。非线性数据包遥测的思想是，对于数据流 F（设其流量为 s）的第 $i\in[1,s]$ 个数据包，其遥测信息嵌入概率 $P(i)$ 为 i 的单调递减函数。对于大流，随着其流量的增加，嵌入概率逐渐减小，从而避免了被过多嵌入；对于小流，该方法提升了小流的遥测概率。

考虑到幂函数 $y=x^{-\alpha}(\alpha>0)$ 和指数函数 $y=\beta^{x}(0<\beta<1)$ 均是自变量的减函数。由于一条流中数据包数大于等于 1，即 $x\geqslant 1$。当 $x\geqslant 1$ 时，指数函数 $y=\beta^{x}<1(0<\beta<1)$，故不能保证一条流一定嵌入了遥测信息。而对于幂函数 $y=x^{-\alpha}(\alpha\geqslant 0)$，当 $x\geqslant 1$ 时，$0<y\leqslant 1$。又由于幂函数，当 $x=1$ 时，$y=1$。这个特征可以用来监视数据包，以表示必须监测第一个数据包，确保每条流的信息都能被收集到。因此，数据包遥测概率计算函数定义为：

$$P(i)=\begin{cases}\dfrac{1}{(1-k)^{\alpha}}, & \text{if } i>k\\ 0, & \text{if } i\leqslant k\end{cases} \tag{1}$$

其中，$i\geqslant 0$，$\alpha\geqslant 0$，i 表示网络流的第 i 个数据包，$P(i)$ 表示分组遥测概率，α 为预设值。对于 TCP 数据包，前 3 个数据包用于建立通信，不承载用户数据。因此，前 3 个数据包不需要遥测。如果用户数据包采用 TCP 协议，则 $i\geqslant 3$，$k=3$。对于其他数据包类型，$k=0$。在非线性数据包遥测中，第 1 个，第 2 个，…，第 s 个数据包被监测的概率分别为 $P(1)$，$P(2)$，…，$P(s)$，$P(1)\geqslant P(2)\geqslant,\cdots,\geqslant P(s)$。

在 SDN 交换机中，具有相同特征的网络数据包集合被描述为流，这与本文的定义是一致的。为了处理流，OVS 交换机定义了流表项。每个流表项对应一个特定的流，不同流的数据包应该执行不同的操作。流表项结构如图 3 所示。流表项的计数器实时统计了一条流与流表项匹配成功的数据包数和字节数等信息。当数据包到达 OVS 时，其将与流表项进行匹配，如果流表项与到达的数据包匹配，则将增加与相应流表项相关联的数据包数量。因此，可以从 OVS 流表项计数器中获取当前流的大小。当数据包流表匹配失败时，EvoTrace 将当前流的大小 i 设置为 1；相反，当数据包流表匹配成功时，将 i 设置为流表项计数器加 1。

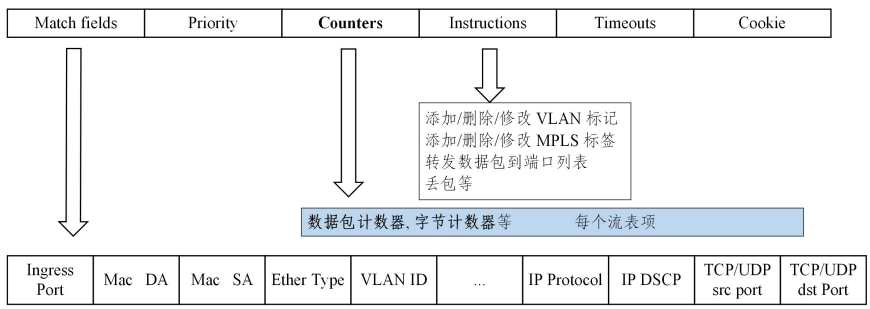


图 3 流表项结构

Fig. 3 Flow entry structure

EvoTrace 可以通过参数 α 控制遥测系统的精度与开销。针对不同的上层应用，EvoTrace 为 α 设置不同的值，以便遥测精度和遥测开销之间取得合理的平衡。当 $\alpha=0$ 时，数据包遥测概率恒等于 1，相当于原始 INT。α 越大，遥测概率下降得越快，嵌入的遥测信息越少，开销越小，遥测精确性越低；α 越小，遥测概率下降得越慢，嵌入的遥测信息越多，开销越大，遥测精确性越高。当 k 固定时，随着流增大，遥测概率逐渐减小，保证了大流的准确性不会下降得过快。网络流中监测数据包的数量反映了该算法遥测信息的约减程度。对于由 n 数据包组成的流，h 跳后，监测的数据包的数量如下：

$$M=\sum_{i=0}^{n}\frac{h}{i^{\alpha}} \tag{2}$$

综上所述，EvoTrace 能够确保始终测量每个流的第一个业务数据包，从而能够收集所有流的遥测信息，即使是数据包较少的小流。同时，对于大流，随着其流量增加，其遥测信息嵌入概率逐渐减小，从而避免了嵌入过多的遥测信息。

3.2 元数据批处理

针对小型遥测数据包缺乏聚合能力的问题，EvoTrace 设计了元数据批处理方法，对不同数据包的遥测信息进行聚合。通过将多个遥测元数据封装到一个遥测包中，EvoTrace 实现了这一目标。为了避免聚合等待时间过长，设计了一种实时遥测保证机制。

EvoTrace 在 OVS 交换机内核模块开辟了一段固定长度的内存，将收集的遥测元数据按处理顺序放入。在内存用尽后，元数据从这段内存的起始位置开始存放，覆盖旧的数据，本文将这段内存称为环形缓冲区。环形缓冲区临时存储遥测信息，直到其可以被复制到遥测报告数据包中进而传输到远程监控服务器。为了实现上述功能，环形缓冲区使用特殊的环形队列数据结构实现，每个队列条目的长度为元数据的大小。每个收集的元数据都被压入队列中进行临时缓存。由于每个数据包收集的遥测信息是由 INT 头指令位图确定的，因此每个缓冲队列项的大小并不相同。每个传入的元数据都放在环形缓冲区中用于临时缓存。考虑到以太网数据包长度的限制，环形缓冲区大小被设置为：

$$N=\frac{MTU\ Length-Packet\ Header\ Length}{INT\ Metadata\ Length} \tag{3}$$

环形缓冲区的有效地址索引为$(0,N-1)$，其占用的内存不超过一个包的大小，所以 OVS 交换机的性能几乎不会下降。

由于网络拥塞、延迟、重负载或其他因素，数据包到达处理节点的时间并不规律。在元数据批处理方法中，随着数据包遥测概率的降低，监测数据包之间的间隔增大，遥测信息在缓冲区中存储的时间更长，从而导致遥测系统的实时性下降。为了解决这个问题，EvoTrace 引入了超时机制。当环形缓冲区满或遥测间隔超过时间窗口阈值时，环形缓冲区内的所有元数据将被封装成一个数据包并发送到监控服务器。

本文用 i 表示数据包处理顺序。当遥测信息放置在缓冲区的首部位置时，加载节点当前时间，记为放入时间 T_s。当遥测信息放置在缓冲区的非首部位置$(i\%N\neq 0)$时，获取节点当前时间，记为当前时间 T_c。T_c 仅用于处理当前遍历的数据包，对添加到环缓冲区的每个元数据进行判断，不需要被记录。

对于每个 T_c，计算 T_c 和 T_s 之间的差值。当差值超过预定义阈值 $\delta(T_c-T_s>\delta)$，将遥测信息封装到一个新数据包中并传输到监控服务器。与原始的 INT-MX 相比，通过批处理减少的数据包数量 R 可以表示为 $0\leqslant R\leqslant S-(S/N)$，其中 S 表示原始遥测报告数据包数。当两个相邻包的当前时间 T 的差值很大时，可能导致当前时间与放入时间之间的差值超过阈值，此时遥测报告数据包中可能携带了少量遥测信息。

3.3 基于内核和用户空间的遥测策略

OVS 的数据包转发功能由两个部分组成，分别是内核空间的 datapath 模块和用户空间的 ovs-vswitchd 模块。OVS datapath 模块作为 Linux 内核中的一个模块运行，直接与底层网络设备交互，实现数据平面功能，负责高性能的数据包处理和转发，处理大部分的数据流量。ovs-vswitchd 是一个用户空间守护进程，包含所有 OpenFlow 规则，并在控制平面执行功能。这两个组件协同实现 OVS 的各种功能和特性。EvoTrace 主要在内核模块实现。

由于非线性数据包遥测方法对较早到达的数据包会产生较高的遥测概率，因此，位于其前面的数据包的大小可能会迅速达到 MTU。为解决此问题，EvoTrace 的实现采用了 INT-MX 模式。当需要遥测的数据包经过转发设备时，直接将当前的元数据信息作为遥测报告包发送给监控服务器，而不是随着数据包转发将元数据逐个嵌入数据包中。下面将分别介绍 INT 源节点、INT 中间节点和 INT 宿点的实现。

1）INT 源节点。在 INT 源节点处，对收到的数据包插入一个包含 INT 指令的 INT 头。INT 头是 INT 模块根据遥测任务修改原始报文添加的。对于 OVS，收发数据包都是在内核空间进行处理，数据结构称为 sk_buff，该结构已经分配了存储数据包的内存。在 OVS 处理期间，数据包操作被限制在 sk_buff 内，应该确保添加一个可变长度的 INT 头不破坏 sk_buff 结构。为了减少数据复制操作，INT 头添加在原始数据包负载后面，作为数据包负载的一部分，无须移动数据。INT 头包含两个字段：原始协议和指令位图。原始协议表示原始 IP 协议的备份值，是为了在宿节点恢复原始数据包。数据包在最后一跳转发给用户前，需将 INT 头删除，将数据包恢复原样。数据包在源节点、中间节点和宿节点的转发行为是不同的。当数据包到达源节点时，IP 首部的协议设置为 INT_T，作为 INT 数据包的协议标识符。将带 INT 头的数据包称为 INT 数据包，将携带元数据的数据包称为遥测报告。这使得网络设备能够识别需要遥测的数据包，并对其进行专门的处理和调度。

2）INT 中间节点。中间节点需执行非线性数据包遥测和元数据批处理生成遥测报告。当 IP 数据包的协议字段为 INT_T 时，中间节点解析 INT 头，然后收集指令位图指定的上层应用所需的数据平面状态。通过指令位图调整测量对象，可实现灵活遥测。表 2 列出了 EvoTrace 支持的元数据值的示例。跳时延为出端口时间戳减入端口时间戳。中间节点需要在数据平面按需构建新的数据包，并将遥测信息封装到这些数据包中进而传输到监控服务器。如图 4 所示，遥测报文包含以太网头部、IP 头部、元数据个数、指令位图和元数据堆栈。IP 头的协议字段设置为 INT_R，以区别其他普通数据包和 INT 数据包。

表 2 元数据说明

Table 2 Metadata description

遥测数据	说明
u32 saddr	源 IP 地址
u32 daddr	目的 IP 地址
u16 ingress_port	入端口号
u16 egress_port	出端口号
u16 latency	跳时延
u64 n_hit	流表匹配成功的数据包个数
u64 n_missed	流表匹配失败的数据包个数
u64 n_lost	丢失的数据包个数

3）INT 宿节点。为恢复原始业务数据包，在将 INT 数据包转发到目的地之前，须将 INT 头删除，以确保 INT 监控对终端主机是透明的。在数据包转发过程中，如果下一跳 IP 等于目的 IP，则当前节点为最后一跳，即宿节点。与路由器不同，OVS 交换机不知道下一跳 IP 地址，无法根据下一跳 IP 与目的 IP 是否匹配来判断当前节点是否为宿节点。OVS 的优势之一是可以通过 OpenFlow 协议进行控制，为每个节点配置不同的功能。在 SDN 体系结构中，控制器拥有网络的

全局视图，利用全局视图，网络管理员可以指定宿节点。EvoTrace 为宿节点添加了一个专门的 OpenFlow 规则，该规则能够自动删除交换机内任何流中的 INT 头。由于网络中宿节点的比例极低，因此该规则引入的延迟开销几乎可以忽略。而在其余节点，不添加宿节点处理规则，其仍按原有流表规则进行匹配转发，不会引入额外的数据转发延迟。由于目前 SDN 控制器不支持自定义规则，因此通过 CLI 命令手动配置。

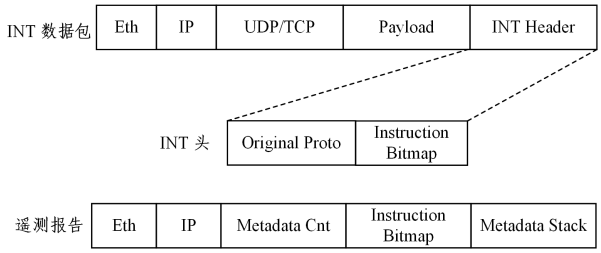


图 4 遥测报文结构

Fig. 4 Telemetry packet structure

4 实验与分析

4.1 实验设置

本文使用 Mininet[22] 创建了一个 $K=4$ 的胖树数据中心网络拓扑，由 4 台核心交换机、8 台汇聚交换机、8 台接入交换机和 16 台服务器组成。运行环境为 i5-10400 CPU、8 GB 内存和操作系统为 Ubuntu 20.04 的虚拟机。数据流量通过 nping 网络工具生成，其中流的产生时间间隔固定，而流的大小符合 Pareto 分布。流量通信模式采用随机模式，即任意一台主机随机向网络中其他主机发送数据包。

从流遥测覆盖率、遥测报告占用率和带宽占用率 3 个方面评估了 EvoTrace 的整体性能。其中，原始 INT 表示为 O-INT。EvoTrace 主要由两种方法组成，即非线性数据包遥测和元数据批处理。本文分别评估了这两种方法的开销，记为 NL-INT 和 BP-INT。其中，数据包占用率等于遥测报告数除以所有包数，带宽占用率等于遥测报告占用的带宽除以所有流量占用的带宽。INT 指令位图收集如表 2 所列的 8 种类型的 INT 元数据。

4.2 结果与分析

4.2.1 EvoTrace 性能评估

在本节中，将 EvoTrace 与 FS-INT[12]，PINT[14] 和 DINT[17] 进行了比较。FS-INT 采用基于系统采样的数据包遥测，PINT 采用基于随机采样的数据包遥测。首先，对比了 EvoTrace，FS-INT，PINT 和 DINT 的流遥测覆盖率；接着，对上述方法的数据包占用率和带宽占用率进行了比较。

图 5 显示了 EvoTrace，FS-INT，PINT 和 DINT 的流遥测覆盖率。图中，FS-INT 的横轴表示采样间隔，分别为 1，2，3，4，分别代表隔 20，40，60，80 个数据包；PINT 横轴上的 1，2，3，4 分别代表 1/20，1/40，1/60，1/80 的不同采样比；EvoTrace 横轴上的 1，2，3，4 分别代表 α 为 1/4，1/2，1，2；DINT 横轴上的 1，2，3，4 分别代表最大采样间隔为 10 ms，20 ms，30 ms，40 ms。从图中可以看出，EvoTrace 优于 FS-INT，PINT 和 DINT。例如，EvoTrace 的所有流遥测覆盖率均为 100%。相比之下，FS-INT 的流遥测平均覆盖率为 95.0%，最大覆盖率为 98.5%，最小覆盖率为 92.75%。PINT 的平均、最大、最小流遥测覆盖率分别为 89.06%，94.87%，81.75%。而 DINT 的平均、最大、最小流遥测覆盖率分别为 83.1%，92.88%，72.38%，明显小于 EvoTrace 的流遥测覆盖率。EvoTrace 与 FS-INT，PINT 和 DINT 相比，流遥测平均覆盖率分别提高了 5%，10.94% 和 16.9%。实验结果表明，EvoTrace 能够收集所有流的遥测信息，而 FS-INT，PINT 和 DINT 会丢失部分流的遥测信息。这是因为 EvoTrace 能够精确地测量中、小流的数据包，而 FS-INT，PINT 和 DINT 则不能。因此，与 FS-INT，PINT，DINT 相比，EvoTrace 可以获得更大的流遥测覆盖率。

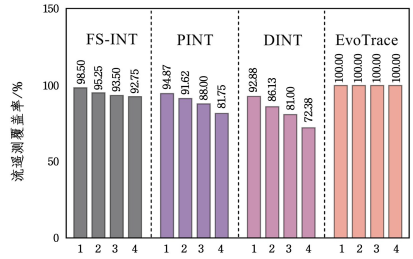
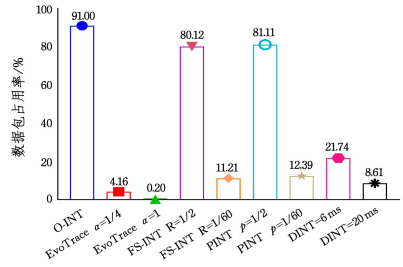


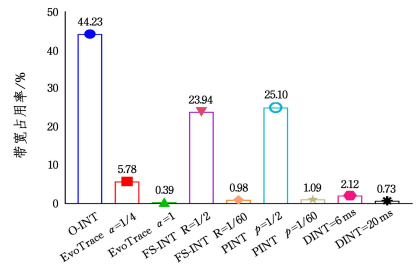
图 5 流遥测覆盖率比较

Fig. 5 Comparison of flow telemetry coverage ratios

图 6(a)和图 6(b)分别显示了 EvoTrace，FS-INT，PINT 和 DINT 的数据包占用率和带宽占用率。



(a)数据包占用率比较



(b)带宽占用率比较

图 6 数据包占用率和带宽占用率的比较

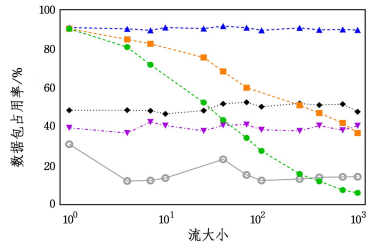
Fig. 6 Comparison of packet occupancy and bandwidth occupancy ratios

实验结果表明，当 $\alpha=1$ 时，EvoTrace 的遥测数据包减少 90%以上，遥测带宽减少 40%以上。可见，EvoTrace 显著减少了遥测数据包的数量，同时也减少了大量的带宽。EvoTrace 的平均带宽占用率为 3.09%，最大带宽占用率为 5.78%，最小带宽占用率为 0.39%。FS-INT 和 PINT 的平均、最大和最小带宽占用率分别为 12.46%，13.1%，23.94% 和 25.1%，0.98% 和 1.09%。而 DINT 的平均、最大和最小带宽占用率分别为 1.43%，2.12%，0.73%。实验结果验证

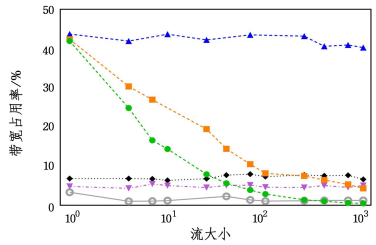
了 EvoTrace 在降低网络开销方面的有效性。

4.2.2 非线性数据包遥测(NL-INT)的性能评估

图 7(a)和图 7(b)分别显示了不同大小流的数据包占用率和带宽占用率。在 EvoTrace $\alpha=1/2$ 和 $\alpha=1$ 时，数据包占用率和带宽占用率都随着流量的增加而逐渐降低，而 O-INT，FS-INT 和 PINT 几乎不变，DINT 由于与网络状态相关，变化稍大。这表明，较大流中包含的遥测信息比例相比原始 INT 越来越低。这种趋势可以归因于这样一个事实：α 的值越大，随着流大小的增加，概率下降得越快，导致监测的数据包之间的间隔更大。根据网络流量的特点，占网络流总数的极少部分的大流在整体流量中占相当大的比例。因此，减少大流的开销可以显著降低系统的整体开销。而对于 O-INT，FS-INT，PINT 和 DINT，数据包占用率和带宽占用率随流大小的增加上下波动，变化很小。这表明，所提出的非线性数据包遥测能够有效降低网络开销。



(a)流大小对数据包占用率的影响



(b)流大小对带宽占用率的影响

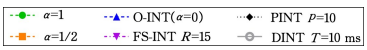
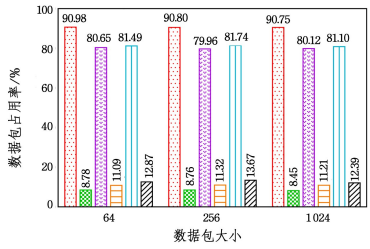


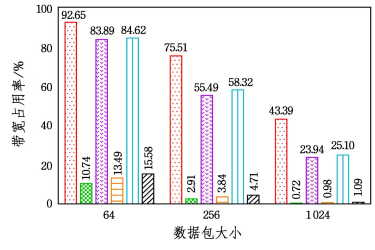
图 7 流大小对数据包占用率和带宽占用率的影响

Fig. 7 Effect of flow size on packet occupancy and bandwidth occupancy

图 8(a)和图 8(b)分别显示了 NL-INT、系统采样[12]和随机采样[14]的数据包占用率和带宽占用率的比较。本文分别使用 64 字节、126 字节和 1 024 字节的数据包进行了 3 组实验。系统采样每隔 R 个数据包插入元数据，而随机采样每个数据包嵌入元数据的概率为 p。如图 8(b)所示，随着数据包长度的增加，3 种方法的带宽占用率都呈下降趋势。这表明流中数据包越大，遥测开销所占的比例越小。然而，数据包越大，可嵌入的遥测信息越少，可能导致数据包碎片或遥测信息收集不完整。如图 8(a)所示，无论数据包的大小如何变化，数据包占用率都能保持一致。这说明 EvoTrace 不受 MTU 限制，不会造成业务数据包分片或遥测信息收集不完整。当 $\alpha=1/4$ 时，NL-INT 的带宽开销略低于 $R=2$ 的系统采样和 $p=1/2$ 的随机采样。当 $\alpha=1$ 时，与 $R=60$ 的系统采样和 $p=1/60$的随机采样相比，NL-INT 表现出略低的带宽开销。当 $\alpha=1$ 时，NL-INT 平均减少约 82.18%的元数据，表明 NL-INT 和系统采样与随机采样，能一样有效降低开销。



(a)不同数据包大小下数据包占用率比较



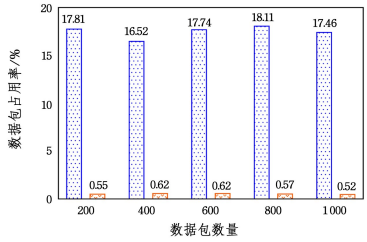
(b)不同数据包大小下带宽占有率比较

图 8 NL-INT 数据包占用率和带宽占用率的比较

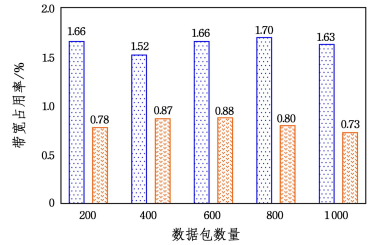
Fig. 8 Comparison of NL-INT packet occupancy and bandwidth occupancy

4.2.3 元数据批处理(BP-INT)的性能评估

图 9(a)和图 9(b)分别显示了 BP-INT 和原始 INT 的数据包占用率和带宽占用率的比较。



(a)BP-INT 数据包占用率比较



(b)BP-INT 带宽占用率比较

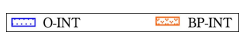


图 9 BP-INT 数据包占用率和带宽占用率比较

Fig. 9 Packet occupancy and bandwidth occupancy ratios comparison of BP-INT

从图 9(a)可以明显看出，元数据批处理通过将多个遥测信息合并到单个数据包中，显著减少了遥测报告的数量。然而，如图 9(b)所示，尽管元数据批处理也减少了带宽，但相对于数据包数量的减少，其减少幅度相对较小。原因是元数据批处理仅将多个元数据聚合在一个遥测报告包中，并没有减少元数据的数量，只减少了遥测报告的头部带宽开销。因此，

元数据批处理在未减少遥测信息的情况下有效减少了链路中的遥测报告数量，同时在一定程度上降低了带宽开销。

结束语 本文提出、实现并测试了一种基于非线性数据包遥测和元数据批处理的轻量级带内遥测框架 EvoTrace。该框架在保证流遥测覆盖的同时，显著降低了网络开销。首先，通过使用与网络流的大小相适应的概率监测每个数据包，保证了对中、小流信息的收集，同时避免了过多测量大流数据包。此外，针对 INT-MX 模式无法对小型遥测数据包进行聚合的问题，EvoTrace 对遥测信息进行聚合，进一步降低了带宽和处理开销，同时可以缓解监测引擎过载问题。EvoTrace 设计了以流量大小为输入参数的非线性函数以控制采样概率，但当出现大量流量时，概率值的计算可能会给 SDN 设备的 CPU 带来负担。在未来的研究中，将结合网络流量模式的变化调整遥测概率，进一步降低遥测开销。

参 考 文 献

[1] YU M. Network telemetry: towards a top-down approach. [J]. Computer Communication Review, 2019, 49(1): 11-17.

[2] ZHAO Y, YANG K, LIU Z, et al. LightGuardian: A full-visibility, lightweight, in-band telemetry system using sketchlets[C]// 18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21). 2021: 991-1010.

[3] LYU H R, LI Q, SHEN G B, et al. Survey on In-band network telemetry[J]. Journal of Software, 2023, 34(8): 3870-3890.

[4] CASE J D, FEDOR M, SCHOFFSTALL M L, et al. RFC1157: Simple network management protocol (SNMP) [EB/OL]. https://www.rfc-editor.org/rfc/rfc1157.

[5] CLAISE B. RFC3954: Cisco Systems NetFlow Services Export Version 9 [EB/OL]. https://dl.acm.org/doi/pdf/10.17487/RFC3954.

[6] PHAAL P, PANCHEN S, MCKEE N. RFC3176: InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks[EB/OL]. https://www.rfc-editor.org/rfc/rfc3176.

[7] BIFULCO R, RÉTVÁRI G. A survey on the programmable data plane: Abstractions, architectures, and open problems[C]// 2018 IEEE 19th International Conference on High Performance Switching and Routing (HPSR). IEEE, 2018: 1-7.

[8] XIAO Z B, CUI Y H, CHEN Y, et al. EAGLE: A Network Telemetry Mechanism Based on Telemetry Data Graph in Kernel and User Mode[J]. Computer Science, 2024, 51(2): 311-321.

[9] KIM C, SIVARAMAN A, KATTA N, et al. In-band network telemetry via programmable dataplanes[C]// ACM SIGCOMM. 2015: 1-2.

[10] P4.org Applications Working Group. In-Band Network Telemetry (INT) Dataplane Specification[EB/OL]. https://p4.org/p4-spec/docs/INT_v2_1.pdf.

[11] KIM Y, SUH D, PACK S. Selective in-band network telemetry for overhead reduction[C]// 2018 IEEE 7th International Conference on Cloud Networking (CloudNet). IEEE, 2018: 1-3.

[12] SUH D, JANG S, HAN S, et al. Flexible sampling-based in-band network telemetry in programmable data plane[J]. ICT Express, 2020, 6(1): 62-65.

[13] TANG S, LI D, NIU B, et al. Sel-INT: A runtime-programmable selective in-band network telemetry system[J]. IEEE Transactions on Network and Service Management, 2019, 17(2): 708-721.

[14] BEN BASAT R, RAMANATHAN S, LI Y, et al. PINT: Probabilistic in-band network telemetry[C]// Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication. 2020: 662-680.

[15] CHOWDHURY S R, BOUTABA R, FRANÇOIS J. Lint: Accuracy-Adaptive and Lightweight In-Band Network Telemetry [C]// 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, 2021: 349-357.

[16] SONG E, PAN T, JIA C, et al. INT-label: Lightweight in-band network-wide telemetry via interval-based distributed labelling [C]// IEEE INFOCOM 2021 - IEEE Conference on Computer Communications. IEEE, 2021: 1-10.

[17] BRUM H B, DOS SANTOS C R P, FERRETO T C. Providing Fine-grained Network Metrics for Monitoring Applications using In-band Telemetry[C]// 2023 IEEE 9th International Conference on Network Softwarization (NetSoft). IEEE, 2023: 116-124.

[18] ZHOU Y, SUN C, LIU H H, et al. Flow event telemetry on programmable data plane[C]// Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication. 2020: 76-89.

[19] PFAFF B, PETTIT J, KOPONEN T, et al. The design and implementation of open {vSwitch}[C]// 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15). 2015: 117-130.

[20] GULENKO A, WALLSCHLÄGER M, KAO O. A practical implementation of in-band network telemetry in open vswitch [C]// 2018 IEEE 7th International Conference on Cloud Networking (CloudNet). IEEE, 2018: 1-4.

[21] ZHENG Y, PAN T, LIN X, et al. Enabling In-band Network Telemetry in Software-based Virtual Switches[C]// 2021 IEEE Global Communications Conference (GLOBECOM). IEEE, 2021: 1-6.

[22] LANTZ B, HELLER B, MCKEOWN N. A network in a laptop: rapid prototyping for software-defined networks[C]// Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks. 2010: 1-6.

WANG Panxiang, born in 1996, postgraduate, is a member of CCF (No. T3193G). His main research interests include software defined networking, network and information security and network telemetry.

CUI Yunhe, born in 1987, Ph. D, associate professor, is a member of CCF (No. F3600M). His main research interests include edge computing, network security, software defined networks, data center networks and network telemetry.

(责任编辑：何杨)