密码学进展

论文报告

日期：2024年5月5日

姓名：盖 乐

学号：21009200991

目录

[全文翻译 4](#_Toc166059449)

[智能网络数据平面的高效设计 4](#_Toc166059450)

[摘要 4](#_Toc166059451)

[1 简介 4](#_Toc166059452)

[2 背景和动机 6](#_Toc166059453)

[3 NetBeacon概述 8](#_Toc166059454)

[4 数据平面认知模型设计 9](#_Toc166059455)

[5 模型部署 10](#_Toc166059456)

[6 执行 13](#_Toc166059457)

[7 评估 14](#_Toc166059458)

[8 讨论 17](#_Toc166059459)

[9 相关工作 19](#_Toc166059460)

[10 结论 19](#_Toc166059461)

[阅读笔记 20](#_Toc166059462)

[Abstract 21](#_Toc166059463)

[I. INTRODUCTION 21](#_Toc166059464)

[Background and Motivation 22](#_Toc166059465)

[2.1 Programmable Data Plane 22](#_Toc166059466)

[2.2 Motivation 23](#_Toc166059467)

[The Overview of NetBeacon 24](#_Toc166059468)

[Data Plane Aware Model Design 24](#_Toc166059469)

[4.1 Feature Engineering 24](#_Toc166059470)

[4.2 Multi-Phase Sequential Models 25](#_Toc166059471)

[Model Deployment 25](#_Toc166059472)

[5.1 Data Plane Model Representation 25](#_Toc166059473)

[Range Marking. 25](#_Toc166059474)

[Range Coding in Feature Tables. 26](#_Toc166059475)

[Handling Forest Models. 26](#_Toc166059476)

[5.2 Stateful Storage Management 26](#_Toc166059477)

[Differentiating Short and Long Flows. 27](#_Toc166059478)

[Handling Storage Index Collisions. 27](#_Toc166059479)

[5.3 Integrated Data Plane Processing Logic 27](#_Toc166059480)

[5.4 Control Plane Logic 28](#_Toc166059481)

[Implementation 28](#_Toc166059482)

[6.1 Computing Flow-level Features 28](#_Toc166059483)

[Aggregate Features. 28](#_Toc166059484)

[Summary Features. 29](#_Toc166059485)

[6.2 Multi-phase Model Inference 29](#_Toc166059486)

[Evaluation 30](#_Toc166059487)

[7.1 Experiment Setup 30](#_Toc166059488)

[Testbed Setup. 30](#_Toc166059489)

[Metric and Features. 30](#_Toc166059490)

[Tasks. 30](#_Toc166059491)

[7.2 End-to-end System Performance 31](#_Toc166059492)

[P2P Application Fingerprinting. 31](#_Toc166059493)

[Covert Channel Detection. 31](#_Toc166059494)

[DDoS Attack Detection. 32](#_Toc166059495)

[Caveats. 32](#_Toc166059496)

[7.3 Model Representation Performance 32](#_Toc166059497)

[Experiment Setup. 32](#_Toc166059498)

[Experimental Results. 33](#_Toc166059499)

[7.4 NetBeacon Deep Dive 33](#_Toc166059500)

[Throughput. 33](#_Toc166059501)

[Determination Thresholds. 33](#_Toc166059502)

[The Size of Stateful Storage 34](#_Toc166059503)

[Imperfection of Hardware Hashing. 34](#_Toc166059504)

[Discussion 34](#_Toc166059505)

[Addressable Market Analysis. 34](#_Toc166059506)

[Scalability Analysis. 34](#_Toc166059507)

[Security Analysis. 35](#_Toc166059508)

[Related Work 35](#_Toc166059509)

[ML-powered Traffic Analysis. 35](#_Toc166059510)

[Applications for programmable switches. 35](#_Toc166059511)

[Conclusion 36](#_Toc166059512)

[Acknowledgments 36](#_Toc166059513)

[References 36](#_Toc166059514)

[大摘要 37](#_Toc166059515)

[中心思想： 37](#_Toc166059516)

[动机与问题： 37](#_Toc166059517)

[相关工作： 39](#_Toc166059518)

[解决思路与手段 39](#_Toc166059519)

[讨论： 40](#_Toc166059520)

[下一步研究点： 41](#_Toc166059521)

[启发与收获： 41](#_Toc166059522)

# 全文翻译

## 智能网络数据平面的高效设计

周广孟，清华大学；刘卓涛，清华大学和中关村实验室；付传普，清华大学；李琦和徐恪，清华大学和中关村实验室

https://www.usenix.org/conference/usenixsecurity23/presentation/zhou-guangmeng

这篇论文收录在《会议论文集》中

第 32 届 USENIX 安全研讨会。

2023年8月9日至11日 • 美国加利福尼亚州阿纳海姆

978-1-939133-37-3

### 摘要

直接在网络数据平面上部署机器学习模型，可以使用数据驱动模型而不是预先定义的协议在线路速度上进行智能流量分析。这种能力称为智能数据平面Intelligent Data Plane (IDP)，这可能会改变现行的网络设计。新兴的交换机用于实现IDP这方面的现有技术分为：

**（i）注重从数据平台提取有用的流量信息，同时将基于学习的流量分析放在平台控制上；**

1. **进一步将学习模型嵌入到数据平面中，而没有使用流级特征来获得较高的学习精度。**

为了提高模型的准确性和模型的部署效率，本文提出了NetBeacon。尤其是NetBeacon提出了一种多级顺序模型架构，通过合并可线速计算的流级特征来提高学习准确性，在流的不同级执行动态数据包分析。此外，NetBeacon设计有效的模型机制表示要解决在网络平面数据上部署基于树的模型时的表边界爆炸问题。最后，NetBeacon通过多个紧密连接的设计来强化其处理流的可扩展性，以管理存储流状态的每个流状态的有状态存储。我们实现了一个原型NetBeacon并广泛评估其在多个流量分析任务中的性能。

### 1 简介

人工智能（AI）在各种网络设计中越来越受欢迎，例如视频比特率自适应[[34](#page17),[56](#page17)]、拥塞控制[[1](#page16),[57](#page17)]、流量优化[[11](#page16)]、路由[[29](#page16),[65](#page17)]、以及网络规划[[66](#page17)]。学习到的模型配置通常在终端主机上[[1](#page16),[34](#page17),[56](#page17),[66](#page17)] 或网络控制平面[[11](#page16),[29](#page16),[57](#page17),[65](#page17)] 执行推理，配备灵活的通用处理器或GPU。

然而，直接在**网络数据平面上**部署流量分析的学习模型是一个相对较少的领域。在网络数据平面上执行模型推理的关键优点是，它能够使用数据驱动的模型而不是预定义的协议，以线速度（即网络转发速度的零降低）进行智能流量分析。这种被称为智能数据平面（IDP）的能力可能会改变广泛的网络设计。例如，复杂的网络攻击通常会绕过经验学习的流量过滤器[[31](#page16)]，流量激发基于学习的恶意检测机制的社区设计，例如[[14](#page16),[37](#page17)]。然而，与这些方法不同的是，IDP实现线速流量分析，而在网络控制平面上部署学习模型的最先进技术只能处理大约 10 Gbps 的流量[[14](#page16)]。另外，将控制平面放置在分析的关键路径上会带来额外的反应时间[[2](#page16)]。除了安全域之外，其他网络设计（例如差异化路由[[29](#page16)]、ECN阈值调整[[57](#page17)]，和中心管理[[19号](#page16)]）也可能受益于IDP。

由于难以在网络数据平面上实现专门的、协议定义的ASIC交换芯片的动态和数据驱动的AI模型，IDP的设计过去是“几乎不可能的”。幸运的是，幸运的是，可编程交换机的进步使得基于协议无关交换架构（PISA）的可定制数据包处理逻辑成为可能。在PISA下，网络数据包通过一个可编程管道进行处理，包括解析器、多个匹配/操作阶段和一个反解析器，实现线速处理。虽然PISA架构目前还不是图灵完备的，不能对数据包字节执行任意计算，但社区已经提出了许多围绕可编程交换机的重要网络安全设计。

**第一类现有技术侧重于从数据平面提取有用的流信息，以支持总体应用。**例如，NetWarden [53] 和 FlowLens [4] 依赖可编程交换机收集流分布信息，基于这些信息在控制平面中执行隐蔽通道检测 [53] 和基于学习的流量分类 [4]。类似地，Poseidon [61] 和 Jaqen [32] 将收集的流信息（例如，流速）与预定义的流量过滤策略（例如，基于阈值的过滤器）进行比较，以减轻容量型DDoS攻击。这类设计与IDP的关键区别在于它们不直接将学习模型嵌入可编程数据平面。

**第二类现有技术更进一步实现了IDP。**例如，pForest [10] 和 SwitchTree [26] 提出在可编程交换机上部署决策树模型，将一个匹配/动作阶段映射到决策树的一层。然而，尽管它们的设计经过了软件交换机BMv2的验证，但在实际硬件上无法部署（详情见 § A）。Planter [63] 扩展了IIsy [55] 并提出了一种包括特征表和代码表的体系结构，以在数据平面上表示集成树。尽管有潜力，但Planter 缺少几个关键设计：例如，尚不清楚Planter 如何同时多次使用一个特征（这在实际中相当常见）。最近的工作Mousika [51] 提出了一种决策树模型，即二叉决策树（BDT），专为可编程交换机数据平面设计。然而，部署BDT所需的表项数量面临组合爆炸的问题（详情见 § A）。至关重要的是，所有这些先前可部署的技术只使用每个数据包的特征进行建模，完全忽略了流级别的特征。正如我们将在 § 2.2 中展示的那样，流级别的特征对于提高各种流量分析任务的学习准确性是必要的。

在本文中，我们提出***NetBeacon，这是一种新颖的IDP设计，推动了模型准确性和模型部署效率的最新进展***。具体而言，NetBeacon具有以下创新设计：

1. **以多阶段顺序模型架构为中心的数据面板操作学习模型设计。**由于流的不同阶段的数据包标注了不同的流级别状态，因此我们的模型在流的不同阶段进行动态分析，减少基于单个推理模型做出过早分类决策所引入的错误。同时，我们的模型使用前面设计的流级和每个数据包特征，可在数据平面上进行线速计算，以确保可部署性。
2. **NetBeacon 提出了一种有效的模型表示机制，以解决将决策树或森林模型表示为数据平面匹配表时出现的表项爆炸问题。**与最先进的相比，Mousika [[51](#page17)],NetBeacon在一些情况下显著减少了表项消耗 （高达75%）。
3. **我们进一步加强了NetBeacon 的可扩展性，以处理并发流量，通过区分短流和长流的处理逻辑，并在观察到存储索引冲突时允许安全存储多路复用。**这可能使NetBeacon 能够处理的并发流量比用于维护每个流状态的寄存器总数还要多。

我们使用Tofino交换机作为可编程数据平面实现了NetBeacon的原型，并通过三个用例对其进行了广泛评估。实验结果表明，NetBeacon在流量分析/分类准确性和硬件表消耗方面均优于最新技术。我们还定量研究了交换机硬件（例如，哈希的不完善性和未来硬件升级）可能如何影响NetBeacon。

### 2 背景和动机

#### 2.1 可编程数据平面

传统交换机配备了定制的ASIC芯片，用于数据包处理以实现高速转发。因此，在ASIC芯片上实现新的网络协议需要设备制造商进行设计、制造和严格测试，这是一个昂贵的过程。为了实现灵活的协议开发，大约两十年前，社区提出了软件定义网络（SDN），通过OpenFlow协议允许软件控制器在交换机上安装定制的流表项[36]。SDN取得了巨大成功，在多年的时间里得到了广泛的部署，特别是在数据中心[13, 20]。最近，围绕协议无关交换架构（PISA）的新兴可编程交换技术将网络可编程性提升到了另一个水平。交换管道本身具有足够的灵活性，可以直接通过特定领域的编程语言（如P4）进行编程，而不是完全依赖于软件控制器。

在PISA交换管道中（图1），网络数据包首先进入解析器进行数据包头解析，然后进入多个匹配/动作阶段进行数据包操作，最后到达解组器进行数据包序列化。解析器、匹配/动作和解组器都可以编程实现所需的协议。匹配/动作支持精确匹配、三元匹配和最长前缀匹配（LPM）。每个匹配对应一个动作，其中可以执行特定的计算和存储修改。相互依赖的动作需要放置在不同的阶段。数据包头和元数据实例使用无状态存储存储，随着新数据包的到达而重新初始化。PISA还提供了有状态和持久存储，例如计数器、计量器和寄存器。最后，PISA提供了各种机制（例如重新提交、重新循环、数据包生成）来进一步扩展编程能力。

尽管PISA具有灵活性，但它具有以下计算和存储限制。首先，它**支持**布尔运算、移位、加法和减法，但**不支持**乘法和除法。浮点运算、循环操作和复杂的条件操作也不受支持。主要的计算逻辑是使用匹配/动作阶段实现的，但这些阶段并非无限制的（例如，Tofino 1只有12个阶段）。同样，存储资源是有限的，例如，在Tofino 1上，每个流水线的静态随机存取存储器（SRAM）为120MB，而三态内容寻址存储器（TCAM）为6.2MB。也许最令人“惊讶”的存储限制是，当数据包穿过交换管道时，每个寄存器只能被访问一次。因此，诸如读取然后更新寄存器等操作在原生环境中不受支持。

#### 2.2 动机

PISA的灵活性激发了对智能数据平面（IDP）潜力的重要研究。我们将先前的工作分为三个子类，如表1所总结的。首先，NetWarden [53] 和 FlowLens [4] 代表了使用可编程交换机收集有用的流信息，例如数据包间延迟分布，然后在控制平面上进行流量分析的设计。由于数据平面和控制平面之间的交互延迟，流量不是以线速进行分析。

Poseidon [61] 和 Jaqen [32] 具有与NetWarden [53] 和FlowLens [4] 相似的设计，唯一的区别是收集的流信息可以直接在数据平面上以线速对容量型DDoS攻击进行分类（请注意，它们的某些操作仍涉及控制平面中的服务器）。然而，它们的分类逻辑是基于阈值驱动的流量过滤器，而不是机器学习模型。因此，它们的方法不能推广到无法准确表示为手工制作的过滤器的流量分析逻辑的用例。

Planter [63] 和Mousika [51] 通过使用匹配/动作表来表示决策树分支，为在网络数据平面嵌入决策树模型迈出了初步步伐。然而，它们的设计有两个主要限制：在数据平面上表示学习模型的效率低，正如 § 7.3 中所示，并且仅考虑无状态的每个数据包特征，而忽略了有状态的流级别特征。

我们通过实验证明了流级别特征（由其5元组标识的流）对于提高准确性的必要性。通常情况下，流级别特征比每个数据包的特征更重要。为此，我们考虑了三种情况：P2P应用指纹识别、隐蔽通道检测和DDoS攻击检测。对于每个任务，我们使用每个数据包的特征（例如，第n个数据包的TTL、数据包大小）和流级别特征（例如，前n个数据包的数据包大小的均值和方差）训练一个决策树模型。然后，我们根据特征的信息增益绘制了图2上半部分的特征重要性。流级别特征明显显示出更高的重要性分数。

其次，我们展示了流感知分类器比不考虑流的分类器输出更准确的结果。具体来说，我们训练了两个决策树模型：一个不考虑流的模型，只使用每个数据包的特征，与Mousika [51]中的情况相同；另一个是考虑流的模型，同时使用流级别和每个数据包的特征，按其重要性排序。这两个模型使用相同数量的特征。我们在图2的下半部分报告了分类准确率结果。结果表明，整合流级别特征显著提高了准确性：P2P应用指纹识别提高了11%，隐蔽通道检测提高了43%，DDoS攻击检测提高了21%（有关这些任务的详细描述，请参见第7.1节）。

**设计目标。**与先前的工作（表1）相比，我们设计NetBeacon旨在通过在网络数据平面上同时实现线速和高度准确（基于学习的）流量分析来推动最新技术。其他的工作[45, 50]使用智能网卡而不是可编程交换机作为数据平面。这两种类型的硬件具有截然不同的特性：可编程交换机具有更高的吞吐量，但计算能力有限。在本文中，我们专注于基于PISA的IDP设计。

假设、威胁模型和限制。我们假设NetBeacon可以访问与任务相关的训练数据集。机器学习本身存在安全漏洞，例如数据和模型投毒攻击，但这超出了范围。我们假设托管NetBeacon的可编程交换机是安全的。由于流级别特征消耗可编程交换机上的有状态存储，因此NetBeacon可以同时维护每个流状态的并发流数量受到硬件存储的限制，尽管NetBeacon具有专门的设计来改善可扩展性。同时，可以通过并行部署多个NetBeacon实例来扩展流量分析能力。NetBeacon不支持在数据平面上难以计算的流级别特征（例如，数据包大小的百分位数）。我们使用5元组（即，源/目标IP、源/目标端口和协议）来标识一个流，但NetBeacon本身不受任何流定义的限制。

### 3 NetBeacon概述

如图3所示，从架构上讲，NetBeacon围绕两个主要组件进行设计：数据平面感知的模型设计和高效的模型部署。数据平面感知的模型设计是一种协同设计方法，用于生成适合硬件的学习模型。为此，我们的特征工程依赖于可在交换管道上以线速提取或计算的特征。此外，考虑到流级别特征（例如，数据包大小的均值）随着流的进行而变化，NetBeacon提出了一种多阶段顺序模型架构，可以在流进行时做出多个推理决策，直到系统足够自信以做出最终决定。

模型部署的关键设计是模型表示模块。它将学习到的模型转换为数据平面上的多个特征表和一个模型表，其中特征表将特征值编码为称为范围标记的数据结构，这些范围标记进一步映射到模型表中存储的推理结果。NetBeacon设计了高效的编码机制，大大减少了表示模型时的表项消耗。

此外，NetBeacon设计了一个有状态存储管理模块，在数据平面上实现了对每个流的有效状态管理。一方面，该模块使NetBeacon能够使用纯粹的每个数据包特征（即，对于短流不维护每个流状态），处理短流，其中短流使用学习模型进行分类。另一方面，NetBeacon利用硬件哈希来实现存储多路复用。特别是，当新流的5元组哈希到已占用的寄存器时（即，存储冲突），如果存储的流已经被分类确定或超时，则新流可以占用此寄存器；否则，NetBeacon回退到使用无状态的每个数据包特征进行新流处理。如果数据包属于存储的流，则更新寄存器并计算特征进行模型推理，即查询特征表和模型表。

一旦数据包的推理结果确定，用户可以根据结果设计定制的后处理，例如进行二进制的拒绝或允许决策，或者相应地分配不同的细粒度服务优先级。

### 4 数据平面认知模型设计

#### 4.1 特征工程

在NetBeacon中，决策树学习模型可以使用每个数据包的特征和流级别的特征。每个数据包的特征可以从单个数据包中获取，通常基于数据包头部的字段，例如数据包大小和生存时间（TTL）值。流级别的特征是通过组合同一流中其他数据包的属性来获得的。

我们将流级别的特征分为聚合特征和摘要特征两类。**聚合特征**表示为 F = aggr(a, c, d)，其中 a 表示 F 中考虑的属性，c 是对属性施加的条件，d 表示一旦数据包满足 c，则更新 F 值的预定规则。例如，特征 F = aggr(数据包大小，[96, 112)，+1) 记录了数据包大小在 [96, 112) 范围内的流中的数据包数量。与聚合特征不同，**摘要特征**的计算不能通过预先确定的更新规则轻松表示。代表性的摘要特征包括最大值/最小值、均值和方差，甚至涉及到硬件上原生不支持的乘法或除法。我们在第6.1节中指明了摘要特征的计算方式。

我们在表2中总结了NetBeacon中考虑的特征。对于特定任务，我们根据其信息增益计算的重要性选择顶级特征。

#### 4.2 多阶段顺序模型

与现有技术类似[[26](#page16),[51](#page17),[63](#page17)],NetBeacon坚持基于决策树的学习模型，因为与涉及大量非线性计算的神经网络相比，它们的结构更类似于数据平面上的匹配/动作操作。采用我们最先进的决策树森林模型，即随机森林（ RF）和XG-Boost（XGB）。考虑到硬件上表项的数量是有限的，我们可以通过限制森林中的树数、最大树深度、最大叶子数等来控制模型大小。

与对所有数据包进行同等对待的按数据包分析不同，流级分析中的数据包位于流的不同阶段，因此表示不同的流级状态。因此，流级特征随着流的进行而变化。在图4中，我们使用两个数据集定义可视化不同流阶段的两个流级特征（即数据包间延迟IPD的随手和数据包大小的轴向）的动态性。显然，这两个功能随着时间的推移都经历了显着的变化。因此，使用基于流级特征的单一模型进行推理，精度较低，如图所示[5](#page6)。

结果观察促使我们设计一个多阶段模型架构，以便在不同阶段应用不同的模型的流程。在每个阶段，NetBeacon使用在该阶段计算的特征进行训练和推理，即第n个数据包的流级特征是基于前n个数据包计算的。我们的模型做出推理决策的数据包称为推理点。推理点的准确排列取决于任务。特别是，每个推理点本质上代表了我们的模型处理推理点之前的n个数据包之后流的分析结果。因此，可以根据任务统一或具体地放置推理点。例如，如果模型使用磁场作为特征，则由于硬件限制，推理点应放置在 2 的磁场位置（参见§[6.1](#page10)）。数据包不选择作为推理点的推理点使用其前一个和最近的推理点的推理结果。因此，两个推理点之间的间隔不宜夸张。

NetBeacon为每个推断阶段设置了决策阈值。当特定推断点处的分类概率高于相应的决策阈值时，这表明多阶段顺序模型有信心在不使用后续推断点的情况下预先决定流的类别。这个设计有助于数据平面上的有状态存储管理（如5.2节所述）。除了流感知的学习模型外，NetBeacon还使用了一个流不可知的模型来对无法使用流级别特征进行分析的数据包进行分类，如5.2节所述。

### 5 模型部署

#### 5.1 数据平面模型表示

我们的多阶段顺序模型架构和流程无法得知树模型中的各个模型以相同的方式表示。在本节中，除非另有说明，否则我们将介绍如何在数据平面上表示单个模型（即单个决策树）模型或具有多个决策树的森林模型）。在§[6.2](#page11)，我们展示了如何合并多相顺序模型的模型表示。

我们从特征单个决策​​树的数据平面表示开始。在决策树中，叶子节点代表分类结果，从根节点到叶子节点的节点代表该叶子节点的节点规则，通常是多个范围的串联。例如叶子节点1的路径[6](#page7) 是 f 1ε[0, 25), f 2ε[0, 46) 和 f 3ε[0, 10)。对于未出现在路径上的要素，其范围是最大允许范围。例如，到节点5的路径是f 1ε[65, 103), f 2ε[0, 256) 和 f 3ε[10, 256)。因此，如果设备交换机支持范围匹配，则可以轻松实现决策树模型，如图中的模型表（1）[6](#page7)，其中键是多个特征范围的集群，值是叶节点。

不幸的是，的是传递开关并不普遍支持范围匹配，并且必须将其编码为三元匹配（三元边界中的位是 0、1 或\*）。基于经典的关联方法[[46](#page17)], [65, 103) 需要 8 个三元边界，范围 [0, 256) 需要 1 个三元边界，范围 [10, 256) 需要 6 个三元边界。因此，为了同时满足所有三个范围，需要 48 (8 × 1 × 6)叶子。

当探索模型表中的叶子组合时，NetBeacon删除冲突的组合。（我）特征 价值冲突： 阐述叶子1-1[9](#page9) 有f1< 25 在其路径上并且叶子 2-2 有 f1其路径上≥35，因此排除上述两个叶子的组合。（二）特征语义冲突：如果一个节点最大（数据包大小）< 10 在一条叶路径和一个节点上随（数据包大小）> 20 在另一个叶子路径上，两个以上叶子的组合被修复。通过保留所有兼容的叶子组合，上述机制恢复了原始森林模型的准确性。

#### 5.2 有状态存储管理

为了利用流级特征，NetBeacon依赖有状态存储来维护每个流的状态。现有技术[[4](#page16),[53](#page17)] 涉及控制平面在接收新流时不连通的存储索引。实现线速流量分析，NetBeacon相反，依赖数据平台上随时可用的硬件散列来分配存储索引。特别是，假设有 N 个有状态注册可用于存储流状态，NetBeacon将流的存储索引计算为H(5-元组)%N，其中H是一个哈希函数。

然而，基于哈希的存储索引分配存在冲突的问题，即两个不同的流（具有不同的五元组）可能接收到相同的存储索引。因此，有必要将真实的流ID（例如，5 元）组）与存储索引一起存储，FormNetBeacon发现碰撞。

一旦发生存储冲突，如果新流覆盖存储，则原始流和新流的每个流状态都会变脏。所以，NetBeacon提出了两种设计来解决这个问题：一种设计可以减少总体碰撞机会，另一种设计可以实现安全存储覆盖、区分短流和长流、如下介绍的[2.2](#page4), 在之上合并流级功能的原因 每个数据包的特征是为了提高分类准确性。由于流量分析的主要目标是提高所有流的总体数据包分类准确性，因此为具有不明数据包的流维护每个流状态的边际返回比为我们使用 P2P 应用程序指纹识别任务定义来证明了这一观察结果。特别是，我们考虑五种情况：仅针对超过 8 个（16、1024、 2048）个数据包（即预备8、预备16等）的流合并流级特征，并考虑流级特征为所有的流量。我们在前面介绍了类似的所有流量的总体数据包分类准确性[10](#page9)显然，之前明显情况的总体精度非常接近。

鉴于这一观察，NetBeacon建议仅针对长流维护每个流的状态。这大大减少了竞争有限状态存储的流的数量。实际上，很难提前知道流长。因此，NetBeacon引入了一个长短流二元分类模型，仅利用每个数据包的特征来确定数据包是否属于长流。具体来说，我们根据集中流长度的第80个百分位来划分长流和短流。取长（短）流中的前N个报文作为然后长（短）流样本。使用每个数据包的特征进行训练和分类。长短流二元分类模型是特定于任务的。例如，在P2P应用程序指纹识别任务中，N=7，准确率82%。

处理存储索引冲突。当存储索引发生冲突时，NetBeacon如果现有流的允许推理类已确定或流已完成（即，其最后一个数据包到达时间超过预定义的超时），则新流使用占用的寄存器。NetBeacon退回到对新流使用每个数据包无状态功能。

如果流已通过多阶段顺序模型中定义的最后一个推理点，则确定流的推理类别/结果（§[4.2](#page6)）或者模型对流程在中间推理点的推理结果有足够的信心。如果是这样，NetBeacon将流程的推理结果保存在 流类表，由流的 5 元组作为关键字，以匹配来自该流的后续数据包。同时，它更新流的存储以标记其推理结果已确定，表明占用的寄存器已准备好存储在索引冲突时被未来的流覆盖。在从数据界面接收到指示流的类别已确定的消息后，控制界面将新的条目动态插入到流类别表中。同时，需要定期删除流类别表中的一些条目（基于先进性先）出或LRU原则）以防止表溢出。

#### 5.3 集成数据平面处理逻辑

我们现在介绍集成的数据平面处理逻辑。当一个数据包时磷到达，其流轴线H1通过对数据包的5元组进行散列计算（Alg.1中的第1行）。[1](#page10)）。然后将哈希值的最后一个索引大小设置为存储指数SID（Alg 中的第 2 行[1](#page10)）。如果磷是一个普通数据包（即不是重提交数据包，如下所述），它与流类别表T匹配系统（Alg 中的第 4 行[1](#page10)）。如果匹配，则直接将数据包的推理结果/类别指定为匹配的类别。

否则，NetBeacon检查是否为数据包分配了每个流状态存储磷在状态存储中。因此，它使用以下方法检索存储的真实流ID（例如，5元组）SID把其与磷的流ID。如果它们符合，NetBeacon识别存储在流中的新数据包SID，然后相应地更新流的状态。同时，如果磷正好是一个推论点，NetBeacon计算流级特征FF，基于此NetBeacon与每个数据包一起执行模型推理 Fp摘自磷。如果分类概率大于预定义阈值dt，则确定流的类别。之后，数据首先界面更新流的存储注册当然 指示其类别已确定，然后向控制界面发出信号以将流插入流类别中

桌子。如果磷不是推理点，它使用存储的推理结果（即最近的推理点得到的结果）作为自己的分类结果。

相反，如果磷没有现有存储的流，NetBeacon仅使用每个数据包的特征来获得其分类结果磷。如果磷被整理为长流数据包，NetBeacon检查存储是否开启SID为空或准备被覆盖。如果是这样，NetBeacon启动每流状态磷使用索引存储的流程SID因此，当存储的流被确定（或超时）时，存储寄存器被延迟释放，同时新的流被散列到寄存器。

在整个流量分析过程中，NetBeacon使用重新提交的数据包（我）更新现有流程的新推理结果或（二）启动新流的存储。原来目的是重新提交修改是修改触发前面阶段登记的（不是修改数据包本身），我们可以通过将数据包镜像到环回端口来触发，而不是重新提交或者再循环，这样原始数据包就不会被延迟。另外，只有触发推理结果更新的推理点数据包才会被镜像，仅占数据包的一小部分，如§中测量的那样[7.4](#page14)。

#### 5.4 控制平面逻辑

在NetBeacon，控制平面负责（我）一开始就将特征表和模型表安装到平面数据上，（二）收到请求后更新流类表（消化在确定流类别时，从数据平面获取（在 Tofino 交换机中）。请注意，更新流类别表的延迟不会影响流量分析，因为与流类别表不匹配的数据包将改为常规模型推理管道因此，控制平面去掉了数据包分类的关键路径NetBeacon，确保线速流量分析。

### 6 执行

我们实现了一个原型[1](#page10)的NetBeacon在Barefoot Tofino 1交换机上，包括12个MAU级、每条管道120MB SRAM和6.2MB TCAM。该交换机的开发工作包括约1300行P416数据平面的代码和控制平面的300行Python代码。我们在这里介绍两个关键实现，即流级特征的计算和多阶段模型推理。更多实施延迟到§[C](#page19)。

#### 6.1 计算流级特征

如§中所述[4.1](#page5)，流级特征分为聚合特征和聚合特征。聚合特征。一个聚合特征形式化为F =聚合(a、c、d)。[11](#page11) （第1行到第5行），NetBeacon为每个聚合特征创建一个表，其中 属性a是键，条件c是表项，特征值在匹配时更新。摘要功能包括顶部、简短、倾斜和意思是具有某些属性， 例如， 数据包大小和数据包间延迟（IPD）。最高和最低限度,NetBeacon将相应注册中存储的值与新值进行比较，然后相应地更新注册。意思是涉及到除法，数据平面不支持除法。解决此限制的一种方法相当于阈值乘以相应推理点的位置，这可以在获得多相顺序模型后离线完成。另一种方法是依赖转移然而，操作要求推理点的位置是 2 的幂的计算缘Var(X) = E(X2)−E(X)2涉及两者分配和支架。数据平面支持估计支架计算仅考虑值的前4位。例如，0b101000001 的平方和 0b101011111 (351) 的平方相同。分配眉毛只能通过以下方式解决转移操作。这本质上要求推理点的位置是 2 的幂，如果缘用于学习模型中的特征。

NetBeacon使用两个预设来计算备份：一个是total\_bytes，用于记录流大小（以字节为单位）（第7行），另一个是size\_square\_sum，用于记录数据包大小的平方和（第8行）。当数据包数量为2的幂时（例如，第9行中的8），首先根据total\_bytes计算平均数据包大小（第10行），然后对size\_square\_sum求溶剂（第11行）。然后，计算平均数据包大小（第10行）。数据包大小的平方（第12行）。最后，两者相减返回火箭（第13行）。由于计算支撑，火箭计算需要4个阶段，其中total\_size和size\_square\_sum在第一阶段计算，size\_mean和size\_square\_sum\_mean在第二阶段计算，size\_mean\_square在第三阶段计算，size\_var在第四阶段计算。

#### 6.2 多层次模型推理

我们的多阶段顺序模型架构在不同的推理点应用不同的模型。仔细观察，我们分别表示每个阶段每个模型，它自己的特征表和模型表。或者，我们可以将它们的表示。如图[12](#page11)，特征表和模型表都有一个额外的键，名为total\_packets，用于区分不同阶段的模型。考虑到这些模型可能使用不同的特征，如果某个特征对应特定模型使用，则提示特征的范围标记设置为\*代表任何范围标记。一般来说，模型推理分为两个阶段：一个阶段用于匹配特征表，一个阶段用于匹配（聚合）模型表。

### 7 评估

（一）改进性能改进：NetBeacon与基线方法相比，流量分析精度有了显着提高，并且我们的结果与无限状态存储的理想情况相当；

（二）数据平面上的高效模型表示：NetBeacon实现更高的分类精度，同时相比中央消耗更少的数据平面表项；

(三)我们研究各种移动部件NetBeacon。

#### 7.1 实验设置

为了进行最终实验，我们将交换机连接到测试Linux服务器。一台服务器重播聚氯乙烯文件通过tc预播放，另一台服务器捕获从交换机接收的数据包指标和特征。我们使用数据包级宏观精度（定义为记起对于不同的类别）作为紧迫性。当数据集平衡时，宏观精度等于准确度。除非有说明，我们在报告评估结果时可兼容使用宏观精度和精度。每个数据包的完整功能集与[[51](#page17)]。规定的流级特征是IPD和数据包大小的聚合特征和聚合特征。对于不同的任务，我们选择最重要的特征。模型训练利用Pythonsklearn数据库。

我们评估NetBeacon使用以下任务：

* P2P应用程序指纹识别。此任务对P2P应用程序流量进行分类。我们使用来自三个P2P应用程序的流量PeerRush 数据集中的（eMule、uTorrent 和 Vuze）[[41](#page17)]。

因此，这是一个三类分类任务。

该任务识别由抗审查工具 Facet 编码的通道通道流量 [[28](#page16)] 来自良性 Skype 流量。我们使用 FacetTraffic 数据集 [[3](#page16)]，选择12.5%类型作为通道流量。

* DDoS 攻击检测。该任务从良性流量中识别 DDoS 流量。我们收集了八种高级 DDoS 流量，包括放大攻击（即通过实际测量披露的 RIPv1、CLDAP、NTP、DNS 和 Memcached）[22号](#page16)]）和脉冲DDoS攻击（根据[中的三个原始设置][33](#page17)]）。为了避免现实世界的损害，我们在拥有 650 个虚拟机的虚拟笔记本云中执行攻击。我们使用 MAWI 数据集 [[49](#page17)] 在主干网络（2020年1月15日）中作为良性背景流量收集。

#### 7.2 端到端系统性能

在本节中，我们展示了NetBeacon在各种网络负载下的性能。与[53]类似，我们使用每秒到达的新流数量来表示网络负载。IndexSize为16，意味着系统可以同时维护最多65536个流状态。我们在表4中总结了主要设置，更多细节延迟至§ D。我们将NetBeacon与以下五种方法进行比较。为了公平比较，我们在NetBeacon和这些方法中使用相同数量的特征。

• Mousika [51]：Mousika的再现版本。我们选择BDT和后蒸馏BDT中准确率较高的模型。

• NetBeacon(Pkt)：NetBeacon的简化版本，仅使用每个数据包的特征。

• NetBeacon(Flow&Pkt) w/o Optimization：NetBeacon的简化版本，使用流级别和每个数据包的特征，但没有适当管理有状态存储（即，没有短流/长流区分和存储索引冲突管理）。

• NetBeacon(Flow&Pkt) Full Version：NetBeacon的完整版本。

• Ideal(Flow&Pkt)：理想情况（模拟）。我们使用纯Python软件代码模拟基于匹配/动作的数据包处理，并为每个流分配存储空间。

P2P应用指纹识别。Peer-Rush数据集中的每个pcap文件都代表一个小时的流量。为了创建不同的网络负载（即每秒新流的媒体数量），我们控制同时回放的流数量。中等（高）网络负载大约是正常网络负载的2（3）倍。所使用的七个最重要的特征是数据包大小的最大值、最小值、平均值和方差，IPD的最小值，以及数据包大小在[48, 64)和[80, 96)范围内的数据包数量。推断点位于第{2、4、8、32、256、512、2048}个数据包。

（一）专业版的NetBeacon只需使用每个数据包的特征即可获得 Mousika。我们在§中提供了进一步的解释[7.3](#page13)。通过使用流级功能，优化后期的NetBeacon明显的灯光Mousika卡。（二）完整版NetBeacon具有与理想解决方案相当的性能，展示了状态存储管理的有效性。较小的精度损失是因为一小部分流量（例如，高网络负载情况下的 0.85%）由于硬件限制而回落到每个数据包纹。

(三)在这个任务中，NetBeacon使用12级，以及17.29%的SRAM和31.25%的T​​CAM。我们的实验是使用第一代Tofino芯片进行的。在最新的Tofino芯片中，级、TCAM和SRAM资源的数量几乎翻了一番，因此消耗的资源NetBeacon是可以接受的。

通道通道检测。通道通道检测任务中只有2000个流，其中1000个是良性的，其余1000个是恶意的。由于数据集是Skype流量，因此所有流量都很长。因此，我们添加来自Univ2数据集的后台流量[[6](#page16)] 因此整个流量包括更均衡的短流和长流分配。我们调整聚氯乙烯文件重播率来控制网络负载。使用的功能NetBeacon分别是数据包大小为 [96, 112)、[112, 128)、[128, 144)、[144, 160)、[160, 176] 和 [176, 192] 的数据包数量。推理点位于第{512、1024、1500、2048、3000、4096、6000、8192、10000}个数据包。

#### 7.3 模型表示性能

在本节中，我们展示了数据平面模型表示NetBeacon与其他相比效率更高IDP作品。我们使用 Mousika 和 Mousika (RF Distilled) [[51](#page17)Mousika 代表直接训练的 BDT 模型。 Mousika(RF Distilled) 表示对训练的RandomForest 模型进行得到的 BDT 模型。实验设置。除了上述 P2P 应用指纹识别、伊斯坦布尔通道检测和 DDoS 攻击检测这三个任务之外，我们在这部分还添加了流量类型分类任务。我们使用 ISCXVPN 数据集 [[12](#page16)]此任务的流量类型，包括六种类型：聊天、P2P、电子邮件、VoIP、流媒体和文件传输。选择随机森林来进行流量类型分类任务。为了确保公平比较，NetBeacon和 Mousika 使用相同的数据训练。我们首先 从原始数据包中提取特征作为数据训练NetBeacon由于Mousika中的模型是二元决策树（BDT），因此我们将特征转换为二进制作为Mousika的训练数据。

所有特征表和模型表中的入境数。Mousika中的模型使用一张巨大的桌子来表示。为了NetBeacon，我们调整树的数量和最大树深度和不同的学习精度和不同的表入境数量。Mousika没有类似的机制来平衡精度和表入境数量，因此调整训练数据的大小来为Mousika生成不同的模型。

实验结果。我们在图中展示了结果[14](#page13)。 （我） 当消耗数量相同的表入口时， NetBeacon与Mousika相比，在所有四项任务上都取得了更高的准确度。有两个可能的原因可能有助于提高性能。第一个是BDT本身使用的特征是二进制特征，可能不具有表达式第二个是BDT本身的设计可能比使用最先进的基于树的模型效率低。NetBeacon，即RF和XGB。（二）为了表示两个性能相似的模型，NetBeacon在数据平面上使用了更少的表边界（例如，海峡通道检测减少了75%）。这证明了我们在中引入的范围标记和范围编码机制的有效性。事实上，我们认识到 BDT 本身的设计倾向 大量生成表项；我们在§中提供了详细的讨论[A](#page18)。

#### 7.4 NetBeacon深入研究

我们讨论影响因素NetBeacon因素。重型交换机上的 PISA 管道可确保任何编译的程序都以线速运行。在 100Gb 转发端口上，P2P 应用程序指纹识别、通道检测和 DDoS 攻击检测的测量吞吐量分别为 99.12 Gbps、99.13 Gbps 和 99.18 Gbps。由于一个100Gb环回端口被管道中的16个转发端口共享，因此每个转发端口的镜像报文比例平均应小于1/16，以保证环回端口不会成为瓶颈在我们的任务中，数据包镜像率分别为3.8%、0.09%和1.28%。

推理点数。多阶段顺序模型中采用的推理点/阶段的数量是影响性能的重要因素。如中所介绍的[4.2](#page6), 早期的流量级别信息可能还为时过早阶段，这可能会影响分类准确性。从上面我们可以看出[15a](#page14)，当我们添加额外的推理点时，准确性就会增加。

确定阈值。如§中所示[4.2](#page6)，我们为每个推理点设置一个判定阈值。[15b](#page14)，我们探索了一个确定阈值。较低的确定阈值允许较早分类和重新租用仓库注册，但可能导致错误分类。我们可以看到，当可用注册的数量缩小（即缩小的IndexSize）时，较小的下限阈值允许较早释放登记，这个补偿了阶段早期做出推理决策的准确性损失。因此，较低的判定阈值比较高的判定阈值获得更好的准确度结果。正好，当存储注册的数量基线时，采用最强的判定阈值更为有利。因此，应考虑可用的硬件资源来确定判定阈值。

有状态存储的大小（即IndexSize）。有状态存储维护每个必要的流状态和计算流级别特征是的。因此，区分不同流引起的存储索引关系问题依次。[15c](#page14), 我们展示了状态存储和分类的大小 准确度呈正相关。同时，我们的状态存储管理在有限的状态存储的基础上提高了数据包分类的准确性。

硬件分区的不完善。我们获得散列到每个仓库登记的数量流，并分区分布图[15天](#page14)理想情况下，所有登记应该具有相同的预定次数，即图中的红色虚线[15天](#page14)。实际分配确实实现了理想分配，这可能会引入额外的存储变量冲突。

### 8 讨论

可指定市场分析。基于现有技术的流量分析的学习[[4](#page16),[14](#page16),[37](#page17)] 将特征工程和/或ML模型推理放在控制（平面例如辅助服务器）上。因此，他们可以使用相当复杂的特征和模型，但产生的分析吞吐量比线速度小。另一种艺术类别[[2](#page16),[32](#page17),[61](#page17)] 主要考虑通过编码防御规则来解析 DDoS在输入交换机上，实现线速流量过滤。然而，他们的方法无法推广到实现其他基于学习的流量分析。相比之下，NetBeacon合伙人IDP将通用学习模型直接嵌入到网络数据平面中，以实现线速智能流量分析。与现有相比IDP相关设计[[10](#page16),[26](#page16),[51](#page17),[55](#page17),[63](#page17)],NetBeacon在分析精度限制和模型表示效率上都超越了它们。然而，由于 PISA 管道的限制，特征和模型架构都受到了影响NetBeacon，这可能会影响复杂任务的分析精度，例如表中量化[5](#page13)除了转移交换机之外，现有技术还考虑了SmartNIC [[45](#page17),[50](#page17)] 和 DPDK 驱动的主机网络 [[35](#page17)]作为智能数据平面，可实现更灵活的计算和存储，但代价是吞吐量低（例如，NetFPGA-PLUS具有200 Gbps，而Tofino 1货运交换机总共具有6.4 Tbps）。

可扩展性分析。NetBeacon具有以下流量分析层次结构：流类别表存储已确定分类类别的流（算法1中的第4-5行）。[1](#page10)）。当与流类别表匹配时，数据包将不会被 通过流类表后部署的ML模型重新分析。否则，数据包进入典型的ML推理。如果NetBeacon在寄存器 (SRAM) 中维护数据包的每个流状态（算法 1 中的第 11-19 行）[1](#page10)），ML推理将使用流级特征；否则，推理主要基于每个数据包的特征（Alg.1中的第21行）。[1](#page10)）。因此，对可处理的并发流的数量没有硬性限制NetBeacon，因为它可以在必要时回退到使用每个数据包的功能。理论上每流的最大数量表现NetBeacon可以同时维持由 注册（SRAM）的大小和正在使用的流级功能。在可扩展性等重要的DDoS任务中，Tofino 1交换机上的一条管道（总共4条管道）最多可支持注册（SRAM）中的140,000个流索引。 Tofino 3 中的 SRAM 大小增加了 80%，管道总数增加了一倍。从架构上，我们可以部署多个NetBeacon每个实例，并在这些实例之间进行负载平衡。NetBeacon实例以线速独立处理流量。

安全分析。我们讨论了自适应（或白盒）攻击NetBeacon第一种类型是资源消耗攻击，其中攻击者生成大量的固件流以溢出交换机上的可用SRAM。这本质上是一种拒绝能力（DoC）攻击的形式[[38](#page17)]。幸运的是，DoC 攻击不会完全崩溃NetBeacon因为它可以退回到使用每个数据包的功能。针对DoC攻击的可能安全防御是通过专用设置协议（例如，Portcullis [[40](#page17)]）。 实际上，DoC攻击可以通过聚合流量或者配置多个来缓解NetBeacon硬件实例。

第二类攻击旨在利用尾部系统参数NetBeacon。两个典型的例子是：（我）攻击者先发送良性数据包，然后再发送恶意数据包的隐形攻击NetBeacon确定流量等级；（二）低速率长流攻击，攻击者强制流超时。一般来说，这些类型的攻击具有进入壁垒的作用，因为对手需要探测参数设置。为了减弱隐形，NetBeacon可以对流类别表中的流量强制执行最大流长度限制。为了减弱低速率长流攻击，NetBeacon可以确定每个流的特定于流的超时（例如，观察到的最大IPD的倍数）。

### 9 相关工作

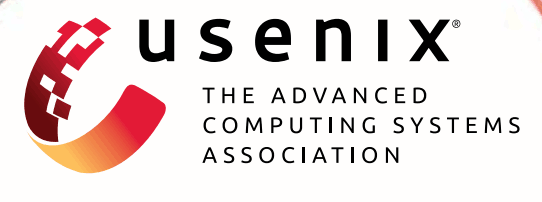
机器学习驱动的流量分析。基于学习的流量已经被研究了很长时间[[7](#page16),[38](#page17)]。近年来，人工智能的发展和网络场景的丰富鼓励了这一领域的研究。有些作品集中于功能设计。[[5](#page16),[14](#page16),[15](#page16),[42](#page17)] 对特征分别进行矩阵变换、压缩和频率变换。一些工作重点在加密流量下的分类。[16](#page16),[27号](#page16),[47](#page17),[59](#page17)]根据加密流量场景构建不同的模型。一些工作通过采用人工智能的最新技术来解决流量分类的常见问题，例如深度学习[[42](#page17)]，自动编码器[[37](#page17)]，数据强化[[21](#page16)]、自动化机器学习[[17号](#page16)]等。然而，这些艺术并不是主要关注点IDP。

指挥开关的应用。指挥开关由于其灵活性而被广泛的领域所评估，包括开关功能优化[[18](#page16),[39](#page17),[60](#page17)]、网络测量与监控[[48](#page17),[58](#page17),[62](#page17)]，网络安全[[23](#page16),[52](#page17),[54](#page17),[64](#page17)]、全方位系统[[25](#page16),[44](#page17)]， ETC。

### 10 结论

智能数据平面（IDP）可能会通过使用数据驱动模型而不是预先定义的协议实现线速智能流量分析来实现网络设计的新范式。在本文中，我们提出NetBeacon,最先进的IDP在学习模型精度和模型表示效率方面均相当于现有技术的设计。其核心是，NetBeacon被赋予权力（我）包含多阶段顺序模型架构的数据操作模型设计，（二）有效的模型机制表示，以及(三)有状态存储管理设计。我们实现了一个原型NetBeacon并广泛评估其在多个流量分析任务中的性能。

# 阅读笔记



**An Efficient Design of Intelligent Network Data Plane**

(**32nd USENIX Security Symposium**

**August 9–11, 2023 • Anaheim, CA, USA 978-1-939133-37-3**)

阅读人员： 盖 乐

日 期： 2024/05/08

## Abstract

本文提出了一种名为NetBeacon的智能网络数据平面设计，旨在提高模型的准确性和部署效率。NetBeacon通过多阶段顺序模型架构，在数据流的不同阶段动态分析数据包，同时结合了可在线路速率下计算的流级特征，以提升学习准确率。此外，NetBeacon设计了高效的模型表示机制，解决了在网络数据平面上部署基于树的模型时表项爆炸的问题。NetBeacon还通过多种紧密耦合的设计来管理用于存储每个流状态的状态存储，增强了其处理并发流的可扩展性。研究者们实现了NetBeacon的原型，并在多种流量分析任务上广泛评估了其性能。实验结果表明，NetBeacon在流量分析/分类准确性和硬件表消耗方面均优于现有技术。此外，研究者们还定量研究了交换机硬件（例如哈希的不完善和未来硬件升级）可能对NetBeacon产生的影响。*（介绍了NetBeacon，这是一种高效的智能网络数据平面设计，它通过在数据流的不同阶段动态分析数据包，并结合流级特征来提升学习模型的准确性。NetBeacon的设计包括一个多阶段顺序模型架构，它能够在数据流传输过程中的不同阶段进行动态数据包分析，利用可在线路速率下计算的流级特征来提高学习准确率。此外，NetBeacon还设计了高效的模型表示机制，解决了在网络数据平面上部署基于树的模型时遇到的表项爆炸问题。为了处理并发流，NetBeacon采用了多种紧密耦合的设计来管理状态存储，增强了其可扩展性。研究者们实现了NetBeacon的原型，并在多种流量分析任务上进行了广泛的性能评估。结果显示，NetBeacon在流量分析/分类准确性和硬件表项消耗方面均优于现有技术，并且研究者们还研究了交换机硬件特性对NetBeacon性能的潜在影响。）*

## INTRODUCTION

*在介绍部分，论文讨论了人工智能（AI）在多种网络设计中的日益普及，包括视频比特率自适应、拥塞控制、流量优化、路由和网络规划等。这些学习模型通常部署在终端主机或网络控制平面上进行推理，这些平台配备了灵活的通用处理器或GPU。然而，直接在网络数据平面上部署学习模型进行流量分析是一个较少探索的领域。在数据平面上执行模型推理的主要优势在于它能够使用数据驱动模型而非预定义协议，在线路速率下实现智能流量分析，这种能力被称为智能数据平面（IDP），有潜力转变广泛的网络设计。 论文指出，随着可编程交换机（例如P4交换机）的出现，为实现IDP提供了关键的硬件支持。现有的相关工作主要分为两类：一类专注于从数据平面提取有用的流信息，同时将基于学习的流量分析放在控制平面上；另一类则进一步将学习模型嵌入到数据平面中，但未能使用对实现高学习准确率至关重要的流级特征。 作者提出了NetBeacon，以在模型准确性和模型部署效率方面推进现有技术。NetBeacon提出了一种多阶段顺序模型架构，通过在数据流的不同阶段进行动态数据包分析，同时纳入了可在线路速率下计算的流级特征以提高学习准确率。此外，NetBeacon设计了高效的模型表示机制，以解决在网络数据平面上部署基于树的模型时表项数量爆炸的问题。最后，NetBeacon通过多种紧密耦合的设计来管理用于存储每个流状态的状态存储，增强了其处理并发流的可扩展性。作者实现了NetBeacon的原型，并在多个流量分析任务上广泛评估了其性能。*

介绍部分主要总结了以下几点内容：

1. ****AI在网络设计中的应用****：人工智能在多种网络功能设计中变得越来越流行，包括视频比特率适应、拥塞控制、流量优化、路由和网络规划等。
2. ****智能数据平面（IDP）的概念****：提出了在网络数据平面上直接部署机器学习模型的概念，以实现基于数据驱动模型的智能流量分析，而不是依赖预定义的协议。
3. ****现有技术的局限性****：指出了现有技术在实现IDP方面的两大类别及其局限性，包括仅在控制平面上进行流量分析和在数据平面上嵌入学习模型但忽视流级特征的问题。
4. ****NetBeacon的提出****：介绍了NetBeacon这一新的IDP设计，它通过多阶段顺序模型架构和流级特征来提高学习准确率，并设计了高效的模型表示机制和状态存储管理，以解决表项爆炸问题并增强并发流处理能力。
5. ****原型实现与评估****：作者实现了NetBeacon的原型，并在多个流量分析任务上进行了性能评估，证明了NetBeacon在准确性和硬件表项消耗方面相较于现有技术的优越性。
6. ****硬件影响的考量****：论文还考虑了交换机硬件特性，如哈希函数的不完善和未来硬件升级，对NetBeacon性能的潜在影响。 总的来说，介绍部分为读者提供了论文研究背景、问题陈述、提出的解决方案以及预期贡献的概览。

## Background and Motivation

### 2.1 Programmable Data Plane

该段介绍了传统交换机与可编程交换机的不同之处，以及可编程交换机如何实现更高层次的网络可编程性。涉及了以下几个关键点：

1. ****传统交换机的局限性****：传统交换机使用专用的ASIC芯片进行数据包处理，以实现高速转发。但这也意味着，要在这些芯片上实现新的网络协议，需要设备制造商进行设计、制造和严格的测试，这是一个昂贵且耗时的过程。
2. ****软件定义网络（SDN）****：为了解决上述问题，社区提出了SDN，它允许软件控制器通过OpenFlow协议在交换机上安装定制的流条目，从而实现协议的敏捷开发。SDN在数据中心等场景中得到了广泛应用。
3. ****可编程交换机技术****：最近，以PISA（Protocol-Independent Switch Architecture）为中心的可编程交换机技术进一步提升了网络的可编程性。与完全依赖软件控制器的SDN不同，可编程交换机的切换管道本身足够灵活，允许通过特定领域的编程语言（如P4）直接编程。
4. ****PISA切换管道****：PISA管道包括解析器、多个匹配/动作阶段和反解析器。这些组件都可以编程，以实现所需的协议。匹配/动作阶段支持精确匹配、三元匹配和最长前缀匹配（LPM），每个匹配对应一个动作，可以执行特定的计算和存储修改。
5. ****PISA的计算和存储限制****：尽管PISA非常灵活，但它仅支持布尔、位移、加法和减法运算，不支持乘法和除法。浮点运算、循环操作和复杂的条件操作也不被支持。此外，匹配/动作阶段的数量是有限的，例如Tofino 1有12个阶段。存储资源也是有限的，例如Tofino 1的SRAM为120MB，TCAM为6.2MB。
6. ****状态存储****：PISA还提供了状态存储，如计数器、计量器和寄存器，以及一些机制（如重新提交、重新循环、数据包生成）来进一步扩展编程能力。但每个寄存器在数据包穿过交换管道时只能被访问一次，因此不支持如读取后更新寄存器的操作。 通过这一节，作者阐述了可编程交换机如何通过PISA架构克服了传统网络设备在灵活性和可扩展性上的不足，同时也指出了在设计智能数据平面时需要考虑的硬件限制。

### 2.2 Motivation

在该段中主要讨论了智能数据平面（IDP）的潜力和必要性，以及NetBeacon的设计动机。作者指出，尽管可编程交换机技术如PISA架构为网络数据平面的灵活编程提供了可能，但现有工作在实现IDP时存在一些限制。这些限制包括：依赖控制平面进行流量分析导致的线速处理瓶颈，以及在数据平面上直接嵌入学习模型时对流级特征的忽视。流级特征，如数据包大小的统计信息，对于提高学习模型的准确性至关重要，但以往的工作往往只考虑了单个数据包的特征。 作者通过实验展示了流级特征在P2P应用指纹识别、隐蔽通道检测和DDoS攻击检测等任务中的重要性，并强调了将流级特征纳入模型训练和推理的必要性。此外，NetBeacon的设计目标是同时实现线速处理和高准确率的流量分析，这要求它不仅要在数据平面上高效地表示学习模型，还要能够有效地管理有限的硬件存储资源，以支持对并发流的处理。 为了解决上述挑战，NetBeacon提出了一系列创新设计，包括多阶段顺序模型架构、高效的模型表示机制以及对状态存储的管理。这些设计使得NetBeacon能够在保持高准确率的同时，优化数据平面上的模型部署和流量分析性能。

#### Design Goals.

在论文的"Design Goals"部分，作者阐述了NetBeacon的设计目标，这些目标指导了NetBeacon架构和功能的开发：

1. ****线速和高准确率的流量分析****：NetBeacon旨在实现在网络数据平面上进行线速（line-speed）和高准确率的流量分析。这意味着NetBeacon需要能够在不影响网络转发速度的前提下，准确地执行流量分类和分析任务。
2. ****利用可编程交换机的硬件能力****：NetBeacon设计考虑了PISA（Protocol Independent Switch Architecture）架构的特性，利用可编程交换机的硬件能力来实现智能数据平面（IDP）。
3. ****流级特征的计算和使用****：NetBeacon强调了流级特征的重要性，这些特征是从整个数据流中计算得出的，对于提高学习模型的准确性至关重要。因此，NetBeacon的设计包括了对这些特征的有效计算和使用。
4. ****高效的模型表示****：为了解决在数据平面上部署决策树或森林模型时可能遇到的表项爆炸问题，NetBeacon提出了一种高效的模型表示机制，显著减少了所需的表项数量。
5. ****可扩展性****：NetBeacon设计了一种状态存储管理模块，以提高处理并发流的可扩展性。这包括对短流和长流的不同处理逻辑，以及在存储索引冲突时的安全存储多路复用。
6. ****硬件友好的模型设计****：NetBeacon的设计考虑了硬件的计算和存储限制，确保所提出的模型能够在实际的硬件上高效运行，同时保持高准确率。
7. ****安全性****：虽然机器学习模型本身可能存在安全漏洞，如数据和模型投毒攻击，但NetBeacon假设其运行的可编程交换机是安全的，并且通过设计来提高系统的安全性。
8. ****性能优化****：NetBeacon的设计还考虑了性能优化，包括减少状态存储的使用和提高数据平面的处理效率。 总体而言，NetBeacon的设计目标是创建一个既能利用数据驱动模型进行智能流量分析，又能在硬件上高效运行的系统。这些目标共同推动了NetBeacon在模型准确性、部署效率和可扩展性方面的创新设计。

#### Assumptions, Threat Model, and Limitations.

该段中作者对NetBeacon的设计理念进行了限制性说明和假设界定。NetBeacon的设计基于几个关键假设：首先，它假定能够获取与特定任务相关的训练数据集，这些数据集对于训练机器学习模型至关重要。其次，它假设NetBeacon运行的可编程交换机是安全的，没有考虑机器学习模型本身可能存在的安全漏洞，如数据和模型投毒攻击。此外，NetBeacon的设计考虑了硬件存储的限制，因为它需要使用状态存储来维护每个流的状态信息，而存储资源是有限的。尽管如此，NetBeacon通过设计来提高可扩展性，例如通过区分长短流和处理存储索引冲突，以支持更多的并发流。 威胁模型方面，NetBeacon考虑了可能针对其系统参数的攻击，例如，攻击者可能通过探测系统参数设置来发动隐蔽攻击或低速率长流攻击。为了缓解这些威胁，NetBeacon采取了一些措施，如设置最大流长度限制和确定特定流的超时时间。 至于局限性，NetBeacon由于PISA流水线的计算和存储限制，不能支持所有可能的机器学习模型和特征。这意味着在处理需要复杂模型和特征的流量分析任务时，可能无法实现最高的分析准确性。此外，NetBeacon的设计也不支持在数据平面上难以计算的流级特征。尽管如此，NetBeacon通过其创新的设计，如多阶段顺序模型架构和高效的模型表示机制，仍然在多个流量分析任务上实现了显著的性能提升。

## The Overview of NetBeacon

在该部分论文提供了NetBeacon架构的全面概览，展示了其主要组件和设计哲学。NetBeacon的设计围绕两个核心组件：数据平面感知模型设计和高效模型部署。 首先，数据平面感知模型设计是NetBeacon的一个共创方法，它侧重于生成适合硬件特性的机器学习模型。NetBeacon通过精心设计的特征工程，确保所选特征能够在网络上以线速提取或计算。此外，NetBeacon采用多阶段顺序模型架构，根据不同阶段的数据包动态分析流的状态，从而减少基于单一推理模型过早分类引入的错误。模型使用可在数据平面上以线速计算的流级和每包特征，确保了部署的可行性。 其次，模型部署的关键设计是模型表示模块，它将学习到的模型转换为数据平面上的特征表和模型表。特征表通过一种称为范围标记的数据结构编码特征值，这些范围标记进一步映射到模型表中存储的推理结果。NetBeacon设计了高效的编码机制，显著减少了表示模型时的表项消耗。 此外，NetBeacon设计了状态存储管理模块，以实现数据平面上高效的每流状态管理。该模块允许NetBeacon仅对长流使用流级特征，而短流则使用纯每包特征进行分类。NetBeacon利用硬件哈希来实现存储多路复用，当新流的5元组哈希到已占用的寄存器时，如果存储的流已被分类确定或超时，则新流可以使用该寄存器。如果数据包属于存储的流，则更新寄存器并计算特征进行模型推理。 NetBeacon的架构还包括了一个集成的数据平面处理逻辑，它定义了当数据包到达时的处理流程，包括流哈希的计算、存储索引的派生、流类别的确定、状态的更新以及基于推理结果的自定义后处理。 最后，控制平面逻辑负责在系统开始时在数据平面上安装特征表和模型表，并在数据平面确定流类别时更新流类别表。控制平面由于不参与数据包分类的关键路径，因此不会影响数据包的线速处理。 整体而言，NetBeacon的设计目标是通过结合先进的机器学习模型和高效的数据平面表示方法，实现在网络数据平面上进行高效准确的流量分析。

## Data Plane Aware Model Design

## 4.1 Feature Engineering

论文详细介绍了NetBeacon如何进行特征工程以支持其决策树学习模型。

1. ****特征类型****：NetBeacon的决策树学习模型可以使用两种类型的特征：每包特征（per-packet features）和流级特征（flow-level features）。每包特征通常基于数据包头部的字段，如数据包大小和生存时间（TTL）值。流级特征则是通过结合同一流中其他数据包的属性获得的。
2. ****流级特征分类****：流级特征进一步被细分为聚合特征（aggregate features）和摘要特征（summary features）。聚合特征通过特定的属性、条件和更新规则来表达，而摘要特征则包括最大/最小值、平均值和方差等，这些特征的计算可能涉及硬件不支持的操作，如乘法或除法。
3. ****特征计算****：论文特别指出了如何在数据平面上计算摘要特征，例如，通过位移操作来近似除法，以及使用近似的平方计算来处理方差。由于数据平面的计算限制，这些计算需要精心设计以适应硬件能力。
4. ****特征选择****：对于特定的任务，NetBeacon会根据特征的重要性来选择顶部特征，这些重要性是基于信息增益计算得出的。这确保了模型能够在保持准确性的同时，减少所需的计算和存储资源。
5. ****动态流级特征****：NetBeacon认识到流级特征会随着数据流的进行而变化，因此提出了一种多阶段顺序模型架构，可以在数据流的不同阶段应用不同的模型，以提高分类的准确性。

## 4.2 Multi-Phase Sequential Models

在4.2节“Multi-Phase Sequential Models”中，论文详细阐述了NetBeacon如何设计多阶段顺序模型架构以提高流量分析的准确性。NetBeacon采用决策树基础的学习模型，特别是随机森林（Random Forest, RF）和XGBoost（eXtreme Gradient Boosting, XGB），因为这些模型的构建与数据平面上的匹配/动作操作更为相似。与处理所有数据包相同的每包分析不同，NetBeacon认识到流级分析中的数据包处于流的不同阶段，携带不同的流级状态，因此流级特征是动态变化的。为了解决使用单一模型进行推断时准确率低的问题，NetBeacon提出了一种多阶段顺序模型架构，该架构可以在流进行的不同阶段应用不同的模型。

在每个阶段，NetBeacon根据该阶段计算的特征进行训练和推断，即基于前n个数据包计算第n个数据包的流级特征。模型做出推断决策的数据包称为推断点，推断点的确切安排取决于具体任务。NetBeacon为每个推断阶段设置了确定阈值，当特定推断点的分类概率超过相应阈值时，表明多阶段顺序模型足够有信心在不使用后续推断点的情况下预先决定流的类别。

此外，NetBeacon还采用了一个流感知模型来对无法使用流级特征进行分析的数据包进行分类，如在5.2节中详细描述的。NetBeacon的设计允许在保持高准确率的同时，减少状态存储的使用，这对于有限硬件资源的可编程交换机来说至关重要。通过这种多阶段顺序模型架构，NetBeacon能够在数据流的不同阶段进行更细致的流量分析，从而提高了整个系统的准确性和效率。

### Model Deployment

## 5.1 Data Plane Model Representation

在该段中，论文详细描述了如何在数据平面上表示单个决策树模型，并且介绍了NetBeacon如何通过创新的方法来高效地将学习模型转换为数据平面上的结构：

1. ****决策树的表示****：在数据平面上，决策树的每个叶子节点代表一个分类结果，而从根节点到叶子节点的路径则代表该叶子节点的匹配规则，这通常是多个特征范围的串联。
2. ****范围匹配到三元匹配****：由于数据平面上的可编程交换机通常不支持范围匹配，NetBeacon必须将范围匹配转换为三元匹配（使用0、1或\*表示）。这种转换显著增加了所需的表项数量，导致了组合爆炸问题。
3. ****范围标记机制****：为了解决表项数量爆炸的问题，NetBeacon引入了一种名为“范围标记”的新机制。该机制确保每个叶子节点在模型表中只消耗单个三元条目，无论与该叶子节点相关的特征表条目有多少。
4. ****连续范围编码（CRC）算法****：NetBeacon提出了一种新颖的连续范围编码（CRC）算法，用于在特征表中表示特征值范围。CRC算法利用了特征值范围的连续性和覆盖所有可能值的特性，通过优化前缀的数量来减少所需的表项。
5. ****森林模型的处理****：对于森林模型，NetBeacon可以单独表示每棵树，并将每棵树的推断结果聚合以得到最终结果。对于像随机森林这样的模型，最终结果是由各个树给出的推断类别的多数决。然而，对于梯度提升树模型（如GBDT、XGBoost），由于最终推断概率是通过非线性函数聚合来自各个树的结果，NetBeacon则采用合并模型表示的方法。
6. ****模型表示的合并****：NetBeacon通过合并不同树的特征表和模型表来处理森林模型，从而在保持原始森林模型准确性的同时，减少了所需的数据平面表项数量。 总体而言，5.1节强调了NetBeacon在数据平面上高效表示学习模型的能力，通过范围标记和连续范围编码算法，NetBeacon显著提高了模型部署的效率，同时保持了高准确率的流量分析。

### Range Marking.

NetBeacon通过引入一种新的机制，名为范围标记（Range Marking），解决了条目组合爆炸问题。在决策树模型中，每个特征通常有多个阈值，例如，在图6中，特征f1有三个阈值：25, 65和103，将整个可能的值范围(即[0,256))分成四个连续且不重叠的基础范围，即[0,25), [25,65), [65,103)和[103,256)。范围标记的目标是使用三元组位字符串（即与0和1都匹配的0, 1和\*）表示这些范围，使得（i）每个基础范围映射到唯一的位字符串，（ii）通过组合多个连续的基础范围获得的任意新范围也映射到唯一的位字符串。

### Range Coding in Feature Tables.

该段内容主要讨论了CRC（连续范围编码）算法在处理数据平面中特征值范围的编码问题。CRC算法的目标是找到最优的父范围[pstart,pend)，使得表示该父范围所需的前缀数量最小化。为了实现这一点，CRC将表示[pstart,pend)的前缀分为接受前缀和拒绝前缀两部分，并优先考虑拒绝前缀。

CRC算法通过解决优化问题来找到最优的父范围。这个优化问题的复杂度是O(lgR^2)，其中R是范围的大小。CRC还提出了一种求解器，其基本思想是将范围列表中的第一个前缀和第二个前缀逐渐合并为一个前缀。此外，CRC不需要对最后一个基础范围[rn,R)进行编码，因为其范围标记机制会默认给该范围分配0值，因此无需在特征表中表示它。

最后，作者通过图8展示了与Prefix [46]相比，CRC算法可以大大减少条目使用量的优势。

### Handling Forest Models.

在处理森林模型的部分，NetBeacon采用了一种特别的方法来优化和简化决策树森林在数据平面上的表示。NetBeacon在数据平面上对森林模型的处理涉及将多个决策树合并为一个统一的模型表示，这样做的目的是为了避免单独表示每棵树时可能出现的效率低下问题。在随机森林等模型中，最终的推断结果是通过聚合每棵树的推断类别来得到的，通常是通过多数投票的方式来确定最终的分类结果。对于这类模型，NetBeacon通过增加一个额外的匹配表来实现这一点，这个表在每棵树的模型表之后，用于聚合每棵树的推断结果。

然而，对于梯度提升树模型（如GBDT、XGBoost），由于它们的最终推断概率是通过非线性函数（例如Sigmoid函数）从各个树的结果聚合得到的，这在数据平面上直接执行是非常困难的。因此，NetBeacon不会单独表示每棵树，而是将它们的特征表和模型表合并，创建一个新的特征表，其中考虑了所有树中出现的特征值范围。合并特征表的过程类似于创建一个新的特征表，同时使用范围标记算法来表示这些范围。

对于模型表，每个条目代表来自不同树的叶子的一个组合。因此，模型表条目的键是考虑了所有到达这些叶子的节点（代表特征范围）的路径上的范围标记，而对应的值是这些叶子的聚合结果，例如通过Sigmoid函数聚合的输出，这可以离线计算得到。

NetBeacon在合并模型表示时，会排除那些存在冲突的叶子组合，包括特征值冲突和语义冲突。通过保留所有兼容的叶子组合，这种机制保留了原始森林模型的准确性。通过这种方法，NetBeacon能够在保持模型准确性的同时，显著减少在数据平面上表示模型所需的表项数量，从而提高了模型部署的效率。

## 5.2 Stateful Storage Management

论文详细讨论了NetBeacon如何管理状态存储以支持流级特征的计算和利用：

1. ****状态存储的需求****：为了利用流级特征，NetBeacon依赖于状态存储来维护每个流的状态。这与仅使用每包特征的方法相比，可以提高流量分析的准确性。
2. ****硬件哈希分配****：NetBeacon采用硬件哈希功能来分配存储索引，而不是依赖控制平面进行分配。这种方法可以确保线速流量分析，并且通过哈希函数和模运算来计算流的存储索引。
3. ****存储索引冲突****：由于哈希分配可能导致不同的流获得相同的存储索引（即存储索引冲突），NetBeacon设计了两种机制来缓解这一问题：减少冲突发生的整体机会，以及在冲突发生时允许安全的存储覆盖。
4. ****长短流区分****：NetBeacon提出了区分长短流的设计，只为长流维护状态信息。这是基于观察到的流量特性，即短流对于维护状态信息的边际收益较低。NetBeacon使用一个基于每包特征的二分类模型来预测流的长度，以此来决定是否为一个流分配状态存储。
5. ****存储冲突处理****：当存储索引发生冲突时，NetBeacon允许新流使用被占用的寄存器，如果原始流的类别已被确定或流已超时。否则，NetBeacon会回退到使用无状态的每包特征来处理新流。
6. ****流类别表****：NetBeacon引入了流类别表，用于存储和匹配流的5元组和推断结果，以便后续数据包的处理。控制平面负责动态更新流类别表，以反映流的分类结果。
7. ****存储效率****：通过上述设计，NetBeacon能够高效地管理有限的状态存储资源，并在保持高准确率的同时，处理更多的并发流。 总体而言，5.2节强调了NetBeacon在状态存储管理方面的创新，通过减少存储冲突和优化存储资源的使用，NetBeacon能够更有效地支持复杂的流量分析任务，同时保持数据平面的线速处理能力。

### Differentiating Short and Long Flows.

在5.2节中提到的“Differentiating Short and Long Flows”部分，NetBeacon通过区分长短流来优化状态存储管理，以支持更有效的流量分析。NetBeacon认识到，并非所有网络流都需要维护状态信息来提高分类准确性。短流（Short Flows）由于数据包数量较少，即使利用流级特征，对分类准确性的提升也相对有限。因此，NetBeacon提出了一种机制，仅对长流（Long Flows）维护状态信息，以此来减少对有限状态存储资源的需求。

为了实现这一区分，NetBeacon采用了基于每包特征的二分类模型来预测流的长度。这个模型使用训练集中流长度的80百分位数来划分长短流，并以此作为分类依据。通过这种方式，NetBeacon能够在数据流的早期阶段就判断出一个流是否值得为其分配状态存储，从而为长流保留存储资源。

此外，NetBeacon还通过硬件哈希函数来分配存储索引，这允许系统在数据平面上以线速处理流量，同时避免了控制平面的介入。在发生存储索引冲突时，NetBeacon能够安全地处理这种情况，允许新流使用被占用的存储寄存器，前提是原有流的类别已经被确定或者流已经超时。

通过这种长短流的区分，NetBeacon能够在保持高准确率的同时，更高效地利用状态存储资源，提升了系统处理并发流量的能力。这种方法不仅优化了资源使用，还提高了网络数据平面上智能流量分析的可扩展性。

### Handling Storage Index Collisions.

这些内容主要描述了NetBeacon在处理存储索引冲突时的策略。当两个流的存储索引发生冲突时，NetBeacon允许新流使用已占用的寄存器，前提是现有流的推理类别已经确定，或者该流已经结束（即，其最后一个数据包到达时间超过了预定义的超时时间）。如果满足这些条件，NetBeacon将把流的推理结果保存在一个以流的5元组为键的流类表中，以便与该流的后续数据包匹配。同时，它会更新流的状态，标记其推理结果已确定，表明已占用的寄存器可以由未来的流覆盖。控制平面在收到数据平面指示一个流的类别已确定的消息后，会动态地将新条目插入到流类表中。同时，为了预防表溢出，需要定期从流类表中移除一些条目（基于FIFO或LRU原则）。

## 5.3 Integrated Data Plane Processing Logic

本段内容主要讨论了数据平面处理逻辑、模型表的表示以及决策树模型在交换机中的实现问题。

NetBeacon的数据平面处理逻辑是在数据包到达时触发的一系列操作。首先，系统计算数据包的5元组哈希值，并使用该哈希值的低位来派生存储索引。如果数据包是新流的一部分，NetBeacon会检查是否已经有为该流分配的状态存储，如果没有，则会初始化一个新的状态存储。

对于每个到达的数据包，NetBeacon会首先检查是否存在于流类表（Flow Class Table）中。如果数据包的流类别已知，则直接应用相应的分类结果。如果未知，则NetBeacon会根据数据包的特征，包括流级特征和每包特征，进行模型推理。在多阶段顺序模型架构中，NetBeacon在不同的推断点使用不同的模型进行分类，以适应流状态的动态变化。

当数据包到达推断点时，NetBeacon会计算流级特征，并结合每包特征进行分类。如果分类概率超过预设的确定阈值，NetBeacon将更新流的状态存储以标记类别已确定，并将流的类别记录到流类表中，以便后续数据包的处理。

在整个流量分析过程中，NetBeacon利用重提交（Resubmit）机制来更新现有流的推断结果或为新流初始化状态存储。重提交操作用于触发先前阶段的寄存器修改，而不改变数据包本身，从而避免了延迟。

此外，NetBeacon的数据平面处理逻辑还包括控制平面的交互，控制平面负责在系统启动时安装特征表和模型表，并在数据平面确定流类别时更新流类表。控制平面由于不参与数据包分类的关键路径，因此不会影响数据包的线速处理。

总体而言，5.3节强调了NetBeacon如何在数据平面上实现一个综合的处理逻辑，该逻辑不仅能够处理每包特征，还能够利用流级特征进行高效的流量分析，同时与控制平面协同工作，以实现动态的流量分类和状态管理。

## 5.4 Control Plane Logic

在NetBeacon中，控制平面负责在数据平面上安装特征表和模型表（在非常初期），并在数据平面确定流类别时更新流类表。需要注意的是，更新流类表的延迟不会影响流量分析，因为未被流类表匹配的数据包会通过正常的模型推断流程。因此，在NetBeacon中，控制平面不处于数据包分类的关键路径上，确保了线速流量分析。

## Implementation

在Barefoot Tofino 1可编程交换机上实现了NetBeacon的原型，该交换机包括12个MAU（Memory Access Unit）阶段，每条流水线有120MB的SRAM和6.2MB的TCAM。对于数据平面的开发工作包括了大约1300行P416代码，而对于控制平面则包括了300行Python代码。在此介绍了两项关键实现：流级特征的计算和多阶段模型推断，更多的实现细节被推迟到第C节。

## 6.1 Computing Flow-level Features

### Aggregate Features.

在NetBeacon中，“Aggregate Features”指的是流级特征的一类，它们是通过聚合同一流中多个数据包的特定属性来计算的。这些特征可以捕捉到流量的统计特性，如数据包大小的分布、间隔时间等，对于理解和分类网络流量非常有用。 NetBeacon中的聚合特征包括对数据流中的数据包进行计数、求和、平均、最小值和最大值等操作。这些操作通常涉及到流的状态信息，需要在数据流的多个数据包之间进行累积和比较。例如，一个聚合特征可能计算一定数量数据包的平均大小，或者记录在某个时间窗口内数据包的出现次数。 在P4编程语言中，NetBeacon通过创建特定的表来实现聚合特征的计算。每个表对应一个聚合特征，其中表的键是数据包的属性，表项包含了基于该属性的聚合操作的条件和更新逻辑。当数据包的属性值与表项匹配时，相应的聚合特征值会根据预定义的更新规则进行更新。 由于聚合特征的计算需要跨多个数据包进行，NetBeacon设计了一种机制来确保在数据流的生命周期内这些特征能够被正确地累积和更新。这通常涉及到在数据平面上维护状态信息，例如使用寄存器来存储中间的聚合结果。 聚合特征对于提高NetBeacon的流量分析准确性至关重要，因为它们提供了对流量行为更深入的理解。通过结合这些流级特征和每包特征，NetBeacon能够更准确地识别和分类网络流量，从而提高整个系统的效能。

### Summary Features.

在NetBeacon中，总结特征（summary features）它们提供了对整个数据流的统计摘要。这些特征通常包括流的最小值、最大值、平均值和方差等统计量，它们能够描述数据流的关键特性：

1. ****计算方法****：摘要特征的计算通常涉及到对流中所有数据包的某个属性（如大小或间隔时间）进行累积和统计分析。例如，计算整个流的数据包大小总和和平均值，或者流的最后一个数据包与第一个数据包之间的最大间隔时间。
2. ****硬件限制****：由于数据平面的硬件限制，如不支持乘法和除法操作，直接在数据平面上计算这些统计量可能具有挑战性。NetBeacon通过使用移位操作和近似计算方法来解决这些问题。
3. ****多阶段计算****：对于一些摘要特征，如方差，可能需要在数据平面上进行多阶段计算。例如，NetBeacon可能会在不同的匹配/动作阶段分别计算流的总大小、平方总和、平均大小和方差。
4. ****优化存储****：为了减少状态存储的使用，NetBeacon可能会采用特定的策略来优化这些摘要特征的计算。例如，只有在流的特定阶段才计算方差，或者使用流的当前状态来预测和更新摘要特征。
5. ****重要性****：摘要特征对于提高流量分析的准确性至关重要，因为它们能够揭示流量的内在模式和异常行为，这对于许多网络应用和安全监控任务非常有用。
6. ****模型训练****：在模型训练阶段，摘要特征被用来训练机器学习模型，以便在数据平面上进行高效的流量分类和异常检测。
7. ****实时更新****：NetBeacon设计了机制来实时更新这些摘要特征，以反映流的当前状态，这对于动态和实时的流量分析场景非常重要。 总体而言，摘要特征在NetBeacon的流量分析框架中扮演着重要角色，它们提供了对网络流量深层次理解的基础，使得基于数据驱动的模型能够在数据平面上实现高精度的流量分类和异常检测。

## 6.2 Multi-phase Model Inference

该文本片段主要讨论了NetBeacon采用多阶段顺序模型架构，不同的模型被应用于不同的阶段。每个阶段都对应于流中的一个特定点，称为推断点，NetBeacon在这些点上使用当前累积的流级特征和每包特征进行分类决策。 在数据平面上，这种多阶段推理是通过一系列匹配和动作表来实现的。每个阶段都有其对应的特征表和模型表，它们包含了该阶段所需匹配的特征范围和相应的动作。例如，第一个阶段可能使用流的初始几个数据包来计算特征，而随后的阶段则使用更多的数据包来细化这些特征的值。 为了区分不同阶段的模型，NetBeacon在模型表中引入了一个额外的键，通常是total\_packets，它表示到目前为止流中已经观察到的数据包数量。这个键确保了在正确的阶段应用正确的模型。 在P4伪代码中，NetBeacon定义了多个动作来标记特征值和设置分类结果。每个特征表都对应于一个特定的特征，并在匹配时应用相应的动作。模型表则使用这些特征的标记值来确定最终的分类结果。 此外，NetBeacon还考虑了如何在数据平面上有效地表示森林模型，例如随机森林或梯度提升树。对于这些模型，NetBeacon合并了各个决策树的特征表和模型表，以减少所需的表项数量，并优化了数据平面上的匹配过程。 通过这种多阶段模型推理方法，NetBeacon能够适应流状态的动态变化，从而在不同的流阶段提供更准确的分类结果。这种方法不仅提高了流量分析的准确性，而且还优化了数据平面上的资源使用，使得NetBeacon能够在保持高性能的同时，处理复杂的流量分析任务。

## Evaluation

NetBeacon在流量分析/分类精度和硬件表消耗方面都优于现有的最先进技术。我们对交换机硬件（例如哈希的不完善以及未来的硬件升级）如何影响NetBeacon进行了定量研究。实验结果显示，在相同的表项数量下，NetBeacon在所有四项任务上的准确率都高于Mousika。NetBeacon在数据平面上使用的表项数量比Mousika少得多，这表明我们引入的范围标记和范围编码机制是有效的。我们还展示了NetBeacon在多个流量分析任务中的实施原型，并对其性能进行了广泛的评估。

## 7.1 Experiment Setup

### Testbed Setup.

在端到端实验中，我们将可编程交换机与两个Linux服务器相连。其中一个服务器通过tcpreplay重放pcap文件，另一个服务器捕获来自可编程交换机的包。

### Metric and Features.

我们使用包级别的宏观准确率（定义为不同类别的召回率平均值）作为评价标准。当数据集是平衡的，宏观准确率等于准确度。除非另有说明，我们在报告评估结果时可互换使用宏观准确率和准确度。完整的每个数据包特征集与[51]中的相同。完整的流级特征集包括IPD和包大小的聚合特征和摘要特征。对于不同的任务，我们从中选择最重要的特征。模型训练利用Python的sklearn库。

### Tasks.

本文通过三个不同的任务来评估NetBeacon的性能：P2P应用指纹识别、隐蔽通道检测和DDoS攻击检测。在P2P应用指纹识别任务中，使用PeerRush数据集中来自三个P2P应用（eMule、uTorrent和Vuze）的流量，将P2P应用流量分类为三类。隐蔽通道检测任务旨在从良性Skype流量中识别Facet工具编码的隐蔽通道流量，使用FacetTraffic数据集，并选择12.5%的类型作为隐蔽通道流量。DDoS攻击检测任务则是从良性流量中识别出DDoS流量，收集了八种先进的DDoS流量，包括放大攻击（如RIPrv1、CLDAP、NTP、DNS和Memcached）和脉冲DDoS攻击，并在具有650个虚拟机的虚拟私有云中执行攻击，使用MAWI数据集中的流量作为良性背景流量。这些任务覆盖了不同的网络流量分析场景，用于全面评估NetBeacon的有效性和准确性。

## 7.2 End-to-end System Performance

在这部分，作者评估了NetBeacon在不同网络负载下的端到端系统性能。网络负载通过每秒到达的新流媒体数量来表示。实验设置了不同的网络负载情况，包括正常、中等和高负载，通过调整同时重放的流量数量来控制。在P2P应用指纹识别任务中，使用PeerRush数据集中的流量，控制重放流量的数量来模拟不同的网络负载情况。隐蔽通道检测任务中，由于数据集规模较小，通过添加背景流量来平衡短流和长流的分布，并调整流量的重放速率来控制网络负载。DDoS攻击检测任务通过合并良性流量和DDoS流量来创建网络流量，并同样通过调整重放流量的数量来控制网络负载。 实验结果表明，NetBeacon在所有任务中均展现出了高准确率和较低的数据平面表项消耗。与Mousika相比，即使在仅使用数据包级特征的简化版本中，NetBeacon也表现出了更高的准确率。当引入流级特征后，NetBeacon的性能进一步提升，与理想情况下的软件模拟方法相当。此外，NetBeacon在资源消耗方面也是可接受的，例如在DDoS攻击检测任务中，NetBeacon使用了9个处理阶段，消耗了11.11%的SRAM和1.85%的TCAM。这些结果证明了NetBeacon在保持线路速度的同时，能够有效地执行复杂的机器学习模型进行流量分析。

### P2P Application Fingerprinting.

在PeerRush数据集中，每个pcap文件包含一小时的流量。为了创建不同的网络负载（即每秒新流的数量），我们控制同时重放的流的数量。中等（高）网络负载大约是正常网络负载的2（3）倍。使用的七个最重要的特征包括数据包大小的最大值、最小值、平均值和方差、IPD的最小值，以及数据包大小在[48,64)和[80,96)之间的数据包数量。推理点位于{第2个，第4个，第8个，第32个，第256个，第512个，第2048个}数据包处。我们在图13的左子图中绘制了结果。（i）即使只使用每个数据包特征的NetBeacon精简版本也超过了Mousika。我们在§ 7.3中提供了进一步的解释。当使用流级特征时，优化后的NetBeacon显著优于Mousika。（ii）完整版本的NetBeacon与理想解决方案的性能相当，展示了有状态存储管理的有效性。由于硬件限制，一小部分流量（例如，在高网络负载情况下的0.85%）回落到使用每个数据包特征，导致轻微的精度损失。（iii）在这个任务中，NetBeacon使用了12个阶段，以及17.29%的SRAM和31.25%的TCAM。我们的实验是在第一代Tofino芯片上进行的。在最新的Tofino芯片中，阶段、TCAM和SRAM资源几乎翻了一番，因此NetBeacon消耗的资源是可以接受的。

### Covert Channel Detection.

在隐蔽通道检测任务中，NetBeacon旨在从Skype良性流量中识别出利用Facet工具编码的隐蔽通道流量。为了构建更平衡的短流和长流分布，研究者们将Univ2数据集中的背景流量与Skype流量合并。通过调整流量的重放速率，控制了网络负载。NetBeacon使用的是数据集中Skype流量的子集，其中长流（超过4000个数据包）被20%的Skype流量替换，作为测试集。其余的Skype流量则作为训练集。在NetBeacon的实现中，使用了单个决策树模型，其最大深度为10，并且在九个不同的推断阶段中，每个阶段都使用了不同大小的模型。具体来说，模型大小在各个阶段分别为{2\*5, 3\*6, 3\*6, 3\*6, 3\*7, 2\*7, 3\*6, 3\*5, 3\*4}。实验结果显示，即使在没有优化的版本中，NetBeacon的性能也与理想方法相当，这表明了NetBeacon在处理隐蔽通道检测任务时的有效性。此外，NetBeacon在此任务中消耗了12个处理阶段，以及相对较小的SRAM和TCAM资源，分别为13.44%和34.03%。这些结果进一步证明了NetBeacon能够在保持高吞吐量的同时，有效地执行复杂的机器学习模型进行流量分析。

### DDoS Attack Detection.

在DDoS攻击检测任务中，通过合并良性流量和DDoS流量来创建网络流量，并通过调整同时重放的流的数量来控制网络负载。此任务中使用的特征是最小IPD（Inter-Packet Delay）和数据包大小。推理点位于第{2个、5个、8个、16个}数据包处。实验结果显示在图13的最右边子图中展示。（i）在各种网络负载下，NetBeacon的流量无关版本比Mousika的准确率高6%到9%。（ii）引入流级特征后，检测准确率提高了20%，并且非常接近理想方法。我们注意到，在此任务中，有状态存储管理的量化好处并不显著。这是因为可以在存储冲突发生之前的一个更早的阶段获得准确的流类别。(iii) NetBeacon在此任务中占用了9个阶段，以及11.11%的SRAM和1.85%的TCAM资源。

### Caveats.

在论文的"Caveats"部分，作者讨论了NetBeacon在实际应用中可能面临的一些限制和挑战。由于硬件资源的限制，NetBeacon在特征选择、模型大小（例如树的深度）以及推断点的位置方面存在一定的局限性，这可能会影响复杂任务的分析准确性。此外，作者还指出，尽管NetBeacon在保持线路速度的同时实现了智能流量分析，但其设计中使用的机器学习模型和特征可能受到PISA流水线的约束，从而限制了模型的复杂性。 作者还提到了在实际部署中可能遇到的一些攻击，例如资源耗尽攻击，其中攻击者可能会生成大量并发的虚假流量以耗尽交换机上的SRAM资源。尽管这类拒绝能力（DoC）攻击不会完全破坏NetBeacon的功能，因为系统可以回退到使用每包特征，但防御这类攻击通常需要通过专门的设置协议来实现。此外，还有针对NetBeacon系统参数的攻击，例如隐蔽攻击和低速率长流攻击。对于这些攻击，NetBeacon可以采取一些措施进行防御，例如实施最大流长度限制和确定特定流的超时时间。

## 7.3 Model Representation Performance

在这部分，作者探讨了NetBeacon在数据平面上模型表示的性能，并与其他智能数据平面（IDP）工作进行了比较。作者选择了Mousika及其通过随机森林模型蒸馏得到的BDT模型作为基线进行比较。在实验设置中，除了之前提到的三个任务（P2P应用指纹识别、隐蔽通道检测和DDoS攻击检测），还额外添加了一个流量类型分类任务，使用了ISCXVPN数据集，包括聊天、P2P、电子邮件、VoIP、流媒体和文件传输六种流量类型，并选择了随机森林模型进行流量类型分类。 为了确保公平比较，NetBeacon和Mousika使用了相同的训练数据。NetBeacon从原始数据包中提取特征作为训练数据，而Mousika由于其模型是二进制决策树（BDT），因此将特征转换为二进制形式。评估指标包括分类准确率和表示模型所需的数据平面表项数量。NetBeacon通过调整树的数量和最大树深度来获得不同的学习准确性和不同数量的表项。而Mousika由于缺乏类似机制来平衡准确性和表项数量，因此通过调整训练数据的大小来产生不同的模型。

实验结果显示，当消耗相同数量的表项时，NetBeacon在所有四个任务中都比Mousika实现了更高的准确率。这可能是由于BDT使用的二进制特征可能表达性不足，以及BDT本身的设计可能不如NetBeacon使用的基于树的模型（如RF和XGB）高效。此外，为了在数据平面上表示性能相似的模型，NetBeacon使用的表项数量大大减少（例如，在隐蔽通道检测中减少了75%）。这证明了NetBeacon在§5.1中介绍的范围标记和范围编码机制的有效性。实际上，作者发现BDT的设计本身就倾向于生成大量的表项；提供了详细的讨论。

### Experiment Setup.

在论文的“Experiment Setup”部分，作者详细描述了用于评估NetBeacon性能的实验设置。实验平台将可编程交换机连接到两台Linux服务器，其中一台服务器使用tcpreplay工具重放pcap文件，另一台服务器捕获从可编程交换机接收到的数据包。评估使用的指标是包级别的宏准确率（macro-accuracy），该指标定义为不同类别的召回率（Recall）的平均值，在数据集平衡时，宏准确率等同于准确率。实验中使用了完整的每包特征集以及流级特征的聚合特征和摘要特征，根据不同任务选择最重要的特征。模型训练使用的是Python的sklearn库。 在实验中，作者使用了三个不同的任务来测试NetBeacon的性能：

1. P2P应用指纹识别任务，该任务对三种P2P应用（eMule、uTorrent和Vuze）的流量进行分类，使用PeerRush数据集中的流量。
2. 隐蔽通道检测任务，该任务旨在从Skype流量中识别出使用Facet工具编码的隐蔽通道流量，使用FacetTraffic数据集中的流量。
3. DDoS攻击检测任务，该任务从良性流量中识别出DDoS流量，收集了包括放大攻击和脉冲DDoS攻击在内的八种高级DDoS流量，并使用MAWI数据集中的流量作为良性背景流量。 实验设置了不同的网络负载情况，通过调整同时重放的流量数量来控制。对于P2P应用指纹识别任务，每个任务使用了不同的模型大小和推断点位置。对于隐蔽通道检测任务，由于数据集规模较小，研究者们将Univ2数据集中的背景流量与Skype流量合并，以获得更平衡的短流和长流分布。DDoS攻击检测任务则通过合并MAWI数据集和恶意流量数据集来创建网络流量，并调整重放流量的数量来控制网络负载。 此外，作者还报告了在每个任务中NetBeacon的性能，包括误报率（FPR）和漏报率（FNR），并讨论了由于硬件限制，设计的机器学习模型在特征、模型大小和推断点位置选择上的局限性。作者强调，尽管存在这些限制，NetBeacon在流级宏准确率方面表现良好，对于入侵检测/防御系统中的恶意流量检测尤其有用。

### Experimental Results.

该部分中展示了在图14中的实验结果。当消耗相同的表项数量时，NetBeacon在所有四项任务上实现了比Mousika更高的准确率。可能有两个原因贡献于这种性能提升。第一个原因是BDT本身使用的特征是二进制特征，可能不具备表达能力。第二个原因是BDT的设计可能不如NetBeacon使用的最先进的基于树的模型（即RF和XGB）高效。为了表示两个表现相似的模型，NetBeacon在数据平面上使用了更少的表项（例如，在隐蔽信道检测中减少了75%）。这证明了作者在第5.1节中引入的范围标记和范围编码机制的有效性。

## 7.4 NetBeacon Deep Dive

在这一部分探究了影响NetBeacon性能的因素。吞吐量方面，可编程交换机上的PISA流水线确保任何编译后的程序都能以线速运行。在100Gb的转发端口上，P2P应用指纹识别、隐蔽信道检测和DDoS攻击检测的测量吞吐量分别为99.12 Gbps、99.13 Gbps和99.18 Gbps。由于一个100Gb回环端口被管道中的16个转发端口共享，每个转发端口的镜像数据包比率应平均小于1/16，以确保回环端口不会成为瓶颈。在我们进行的任务中，每个任务的数据包镜像比率分别是3.8%、0.09%和1.28%。

### Throughput.

多阶段顺序模型中采用的推断点/阶段数量是影响性能的一个重要因素。根据第4.2节的内容，流级别信息在早期阶段可能是不成熟的，这可能会影响分类准确性。如图15a所示，随着添加额外的推断点，准确率会提高。

### Determination Thresholds.

在论文的“Determination Thresholds”部分，作者探讨了在NetBeacon的多阶段顺序模型架构中，为每个推断点设置确定阈值对于系统性能的影响。确定阈值用于判断在特定推断点上，流量的分类结果是否足够自信，从而可以提前确定流量的类别，而不必等待后续更多数据包的到来。这样做可以提前释放状态存储资源，提高系统的效率。 实验中，作者考虑了不同的确定阈值设置，并观察了它们对系统准确率的影响。较低的确定阈值允许系统在更早的阶段对流量进行分类，这可能会导致一些误分类，但同时也能够更早地释放存储寄存器。在存储寄存器数量较少的情况下，较低的确定阈值有助于补偿由于在早期阶段做出推断决定而导致的准确率损失。因此，在资源受限的情况下，较低的确定阈值可能会带来更好的准确率结果。相反，当存储寄存器数量较多时，使用较高的确定阈值更为有利。 作者通过实验发现，确定阈值的选择应该基于可用的硬件资源和具体的应用场景来决定。例如，在DDoS攻击检测任务中，由于需要处理大量并发流量，可能需要设置较高的确定阈值以减少误报。而在P2P应用指纹识别任务中，可能可以承受较低的确定阈值，以便更快地对流量进行分类。 总体而言，确定阈值是NetBeacon中一个重要的参数，它影响着系统的存储效率和分类准确率，需要根据具体的硬件能力和网络流量特性进行细致调整。

### The Size of Stateful Storage

为了维护每个流的状态并计算流级别特征，状态存储是必要的。因此，解决由不同流引起的存储索引冲突问题至关重要。在图15c中，论文展示了状态存储的大小与分类准确率之间的正相关关系。同时，状态存储管理在有限的状态下显著提高了数据包的分类准确率。

### Imperfection of Hardware Hashing.

在论文这一部分，作者研究了硬件哈希函数的不完美性对NetBeacon性能的影响。在NetBeacon中，为了高效地管理状态存储，使用硬件哈希函数来为每个流分配存储索引。理想情况下，所有存储寄存器应该均匀地被流所使用，即每个寄存器被流哈希到的次数大致相同。然而，实际的硬件哈希函数可能无法完全实现这一理想分布，导致一些寄存器可能会有过多的哈希冲突，而其他寄存器则可能使用不足。 作者通过实验测量了流量被哈希到每个存储寄存器的实际分布情况，并发现实际分布与理想分布存在偏差。这种偏差可能会导致额外的存储索引冲突，影响NetBeacon的状态存储管理效率。具体来说，如果某些寄存器过于频繁地被哈希到，那么这些寄存器上的流状态可能会被错误地覆盖，从而引发状态丢失或错误更新。 为了量化这种影响，作者统计了每个寄存器上流量的命中次数，并绘制了命中次数的分布图。从图中可以看出，实际的分布并不是完全均匀的，一些寄存器的命中次数远高于或远低于平均值。这种不均匀性可能会对NetBeacon的性能产生负面影响，尤其是在高负载或高并发场景下，存储冲突的概率会增加。 尽管存在这种硬件哈希的不完美性，NetBeacon仍然能够通过其设计中的状态存储管理机制来有效处理存储冲突，保持高性能的流量分析。此外，作者也指出，可以通过采用更优的哈希算法或增加存储资源的方式来减少这种不完美性对系统性能的影响。 总体而言，硬件哈希的不完美性是NetBeacon在实际部署时需要考虑的一个问题，但通过合理的设计和优化，可以在一定程度上缓解由此带来的性能损失。

## Discussion

## Addressable Market Analysis.

在此部分中，作者讨论了NetBeacon的潜在市场和应用前景，主要内容概述如下：

1. ****现有技术的局限性****：作者指出，现有的基于学习流量分析的研究工作通常将特征工程和机器学习模型推理部署在控制平面上，如辅助服务器。这些方法虽然可以使用复杂的特征和模型，但提供的分析吞吐量远低于线路速度。
2. ****特定应用的解决方案****：另一类研究工作主要关注使用可编程交换机进行DDoS缓解，通过在交换机上编码防御规则来实现线路速度流量过滤。然而，这些方法并不适用于实现其他基于学习的流量分析。
3. ****NetBeacon的优势****：与现有技术相比，NetBeacon倡导智能数据平面（IDP），直接将通用学习模型嵌入到网络数据平面中，以实现线路速度的智能流量分析。NetBeacon在分析准确性和模型表示效率方面都超越了现有的IDP设计。
4. ****硬件限制****：由于PISA流水线的约束，NetBeacon中的特征和模型架构可能受到限制，这可能会影响复杂任务的分析准确性。
5. ****可扩展性分析****：NetBeacon没有硬性限制可以处理的并发流的数量，因为它可以在必要时回退到使用每包特征。此外，可以通过部署多个NetBeacon实例并进行负载均衡来提高系统的可扩展性。
6. ****安全性分析****：作者讨论了针对NetBeacon的几种自适应（或白盒）攻击，包括资源耗尽攻击和利用NetBeacon系统参数的攻击。对于这些攻击，NetBeacon可以采取一些防御措施，如设置最大流长度限制和确定特定流的超时时间。
7. ****市场应用****：NetBeacon的技术可以应用于广泛的网络设计，包括但不限于网络安全、流量优化、数据中心网络管理等领域，具有很大的市场应用潜力。 总体而言，"Addressable Market Analysis"部分强调了NetBeacon在智能数据平面领域的创新性和潜在的市场应用价值，同时也指出了在实际部署时需要考虑的一些挑战和限制。

## Scalability Analysis.

NetBeacon设计了一种状态存储管理模块，用于在数据平面上高效地管理每个流的状态。一方面，该模块使NetBeacon能够使用纯数据包特征处理短期流（即，对短期流不维护任何流状态），短期流使用学习模型进行分类。另一方面，NetBeacon利用硬件哈希技术实现存储复用。特别是当一个新流的5元组被哈希到已占用的寄存器时（即发生存储冲突），如果存储流已被确定类别或超时，新流就可以使用这个寄存器；否则，NetBeacon会回退并使用无状态的数据包特性来处理新流。

NetBeacon假设托管它的可编程交换机是安全的。由于流级别特征消耗了可编程交换机上的有状态存储空间，因此NetBeacon可以同时维护的并发流数量受到硬件存储的限制，尽管NetBeacon有专门的设计来提高可扩展性。尽管如此，通过并行部署多个NetBeacon实例，可以扩展流量分析能力。NetBeacon不支持在数据平面上难以计算的流级别特征（例如，数据包大小的百分位数）。我们使用了5元组来标识流。

## Security Analysis.

作者对NetBeacon可能面临的安全威胁和挑战进行了分析，并提出了相应的防御措施。

## Related Work

## ML-powered Traffic Analysis.

本部分，作者深入探讨了与NetBeacon相关的研究工作，并对智能数据平面（IDP）领域的现有技术进行了全面的分析。文中提到了多个研究方向，包括基于机器学习的流量分析、加密流量的分类技术、利用深度学习等先进AI技术进行流量识别的研究，以及自动化机器学习在流量分析中的应用。此外，还讨论了可编程交换机在网络优化、网络测量、网络安全和分布式系统等领域的应用情况。 作者还对比了NetBeacon与当前IDP领域的其他设计，如pForest、SwitchTree、IIsy、Planter和Mousika等。这些研究工作尝试在数据平面上实现决策树模型或其他机器学习模型，但受限于硬件资源和模型表示的效率。NetBeacon在这些方面的设计更为先进，它通过多阶段顺序模型架构、高效的模型表示机制和状态存储管理设计，实现了在数据平面上对机器学习模型的高效部署和高准确率的流量分析。 总体而言，"Related Work"部分不仅总结了当前领域的研究进展，而且突出了NetBeacon在智能数据平面设计中的创新点和优势，为读者提供了该领域研究背景和技术发展的全面视角。

## Conclusion

作者总结了NetBeacon的核心贡献和研究成果。作者指出，NetBeacon作为一种先进的智能数据平面（IDP）设计，通过直接在网络数据平面上嵌入通用学习模型，实现了在线路速度下的智能流量分析。NetBeacon在模型准确性和模型表示效率方面均优于现有技术。 作者强调了NetBeacon的三个关键创新点：首先，NetBeacon采用了多阶段顺序模型架构，能够动态地在数据流的不同阶段进行分析，提高了学习准确率；其次，NetBeacon设计了高效的模型表示机制，显著减少了在数据平面上表示决策树或森林模型时所需的表项数量；最后，NetBeacon通过多种紧密耦合的设计，强化了对并发流的处理能力，通过管理状态存储，提高了系统的可扩展性。 此外，作者还提到了NetBeacon的原型实现，并在多种流量分析任务上广泛评估了其性能。实验结果表明，NetBeacon在保持高吞吐量的同时，能够有效地执行复杂的机器学习模型进行流量分析，展现出了其在智能数据平面领域的领先性能。

# 大摘要

## 中心思想：

智能数据平面（IDP）是网络领域的一项创新技术，它允许直接在网络数据层面部署机器学习算法，以进行智能的流量分析。与那些依赖于固定协议的传统方法不同，IDP通过数据驱动的模型实现了实时的流量处理，这可能对网络架构的设计产生重大的影响。目前的研究主要分为两个方向：一种方向是将基于学习的流量分析技术应用于控制层面，同时从数据层面提取有价值的信息；另一种方向则是将学习算法直接整合到数据层面，但这种方法并没有充分利用流量级别的特征来提升分析的精确度。 为了解决这些问题并提升模型的精确度和部署的效率，我们提出了一种名为NetBeacon的新型系统。NetBeacon采用了一个分阶段的顺序模型架构，它能够在处理流程的不同阶段对数据包进行动态分析。该架构通过整合可以在实时速度下计算的流量级别特征，显著提高了模型的准确性。此外，NetBeacon还设计了一种高效的模型表示方法，有效解决了在网络数据层面部署基于树的模型时可能遇到的表项数量急剧增加的问题。为了增强系统处理大量并发流量的能力，特别是对于需要管理每个流量状态的有状态存储，NetBeacon还进行了多项设计上的优化。

## 动机与问题：

这篇论文着眼于探索智能数据平面（IDP）的潜力，特别是在利用可编程交换机的灵活性方面。传统的SDN方法中，数据平面与控制平面之间的互动存在延迟，导致流量分析无法达到线速。虽然一些现有工作尝试在控制平面上进行流量分析，但受到性能限制。因此，基于PISA的新型可编程交换技术应运而生，其交换管道具备更高级别的网络可编程性。这项技术激发了对线速智能流量分析的研究，而NetBeacon则是其中的一个重要成果。

以往的研究主要集中在利用可编程交换机收集有用的流信息，然后在控制平面上进行流量分析，或者直接在数据平面上对容量型DDoS攻击进行分类。然而，这些方法存在一些局限性，比如分析速度不够快，或者分类逻辑基于阈值驱动的流量过滤器而不是机器学习模型。

NetBeacon的研究动机在于解决这些问题和挑战，旨在通过在网络数据平面上实现线速和高度准确的流量分析来推动最新技术。相较于以往的工作，NetBeacon采用了一种全新的设计方法，利用PISA架构下的可编程交换机，同时充分利用了流级别特征以提高准确性。NetBeacon的设计目标是克服传统方法中的性能瓶颈，提高流量分析的效率和准确性。

1. 主要贡献：

1.**NetBeacon框架创新：** 本文创新性地提出了NetBeacon框架，这一框架的设计理念是为了克服传统流量分析方法在效率和精确度上的局限。NetBeacon融合了机器学习技术、硬件加速技术以及流量级别的特征提取技术，以实现在高速网络环境中的即时流量分析。

2.**分阶段顺序模型架构：** NetBeacon采用了一种创新的分阶段顺序模型架构，该架构允许在处理流程的不同阶段应用差异化的模型，从而动态调整以适应流量的实时变化，显著提升了分类的准确性。

3.**数据平面感知模型设计：** 论文中还设计了一种特别针对数据平面优化的模型，该模型能够生成适合硬件加速学习的学习模型。通过精心设计的特征选择和模型部署策略，NetBeacon能够在数据平面上高效地执行模型推理。

4.**有状态存储管理模块：** NetBeacon包含一个精心设计的有状态存储管理模块，它负责维护网络中每个流的状态信息。该模块不仅能够有效地管理有限的存储资源，还解决了存储索引冲突的问题。

5.**硬件加速技术的集成应用：** 论文展示了在Barefoot Tofino交换机上实现的NetBeacon原型，并证明了其在高速网络环境中的实时性能。通过集成硬件加速技术，NetBeacon能够以线速执行流量分析，同时保持高吞吐量和低延迟的特性。

1. 相关工作与背景描述：

背景知识：

网络流量分析是指通过监控和分析网络数据流来获取有关网络活动的信息。这项技术在网络管理、安全监控、流量优化等领域起着关键作用。传统的网络流量分析方法通常基于规则匹配、统计分析等技术，但随着网络流量的不断增长和变化，这些方法已经不能满足实时性、准确性和效率的要求。因此，研究人员提出了许多新的流量分析技术，以应对这些挑战。

## 相关工作：

**1.规则匹配方法：**传统的流量分析方式之一是基于规则匹配的方法，它使用预定义的规则集合来识别网络流量中的特定模式或行为，例如访问控制列表（ACL）或防火墙规则。然而，在处理复杂流量模式和大规模流量时，规则匹配方法的效率较低。

**2.机器学习方法：**近年来，随着机器学习技术的发展，越来越多的研究开始将机器学习应用于网络流量分析。这些方法利用机器学习算法从大量的网络数据中学习模式和行为，并用于识别异常流量、入侵检测等任务。常见的机器学习方法包括决策树、随机森林和支持向量机等。

**3.硬件加速技术：**为了提高流量分析的性能和效率，一些研究开始尝试利用硬件加速技术，例如FPGA（现场可编程门阵列）和ASIC（应用特定集成电路）。这些技术能够提供更快的数据处理速度和更高的吞吐量，适用于高速网络环境和大规模数据流。

**4.流级特征提取技术：**流级特征提取是流量分析中的关键步骤，它可以从网络流量中提取各种关于流的特征，如数据包大小、传输速率和流持续时间等。传统方法通常基于数据包级别的特征提取，但随着流量的增长和变化，流级特征提取变得越来越重要。

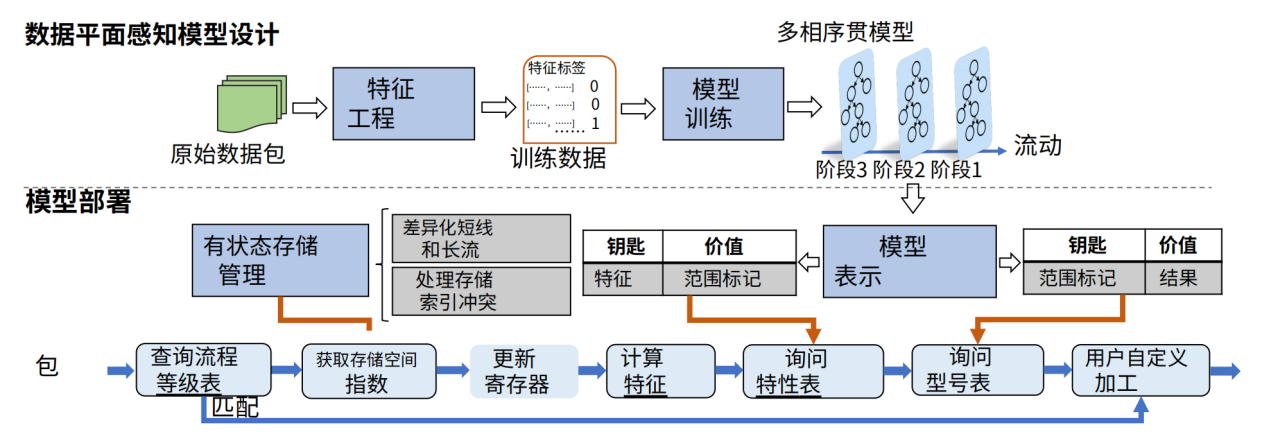
## 解决思路与手段

这篇论文致力于解决网络中智能流量分析的问题，其主要手段和思路如下：

1. **设计智能数据平面（IDP）解决方案**：论文提出了一种名为NetBeacon的IDP解决方案，旨在实现线速智能流量分析。这一解决方案的设计是为了提高网络的智能化水平，使其能够在数据平面上实时分析流量并做出智能决策。

2. **引入多阶段顺序模型架构**：为了实现智能流量分析，NetBeacon采用了多阶段顺序模型架构，将数据操作和分析任务分解为多个阶段，以便逐步处理和决策流量。

3. **优化模型机制表示和状态存储管理**：NetBeacon设计了有效的模型机制表示和状态存储管理，以确保在网络数据平面上能够高效地表示和处理流量分析模型，并实现对流状态的有效管理和维护。



## 讨论：

创新点：

**1.流级特征计算方法**：NetBeacon提出了一种聚合和摘要特征计算的方法，能够从数据包和流级别获取关键信息。这种方法有效地利用了硬件支持的特征提取功能，提高了流量分析的效率和准确性。

**2.多阶段模型推理架构**：NetBeacon采用了一种多阶段顺序模型架构，根据流的不同阶段应用不同的模型，并根据每个阶段计算的特征进行推理。这种分阶段应用不仅能够减少计算负载，还可以根据不同的需求优化模型的使用，提高了分类的准确性。

**3.硬件优化和存储管理**：NetBeacon充分利用了硬件支持的功能范围，设计了算法和模型，使其能够在性能和资源利用率之间取得平衡，实现高效的流量分析。此外，NetBeacon提出了一种解决存储冲突的方法，以确保数据的一致性和准确性。

不足之处：

**1. 硬件依赖性**：论文在设计中依赖于特定的硬件设备（例如Tofino交换机），这可能限制了该解决方案在其他平台上的实际应用。

**2. 安全性考量不足**：虽然论文提及了一些可能的攻击方式，并提出了一些防御方法，但对于安全性的讨论相对较少，对于可能的安全风险和威胁的深入分析和探讨不足。

## 下一步研究点：

**1. 跨平台适用性研究**：进一步研究如何设计和实现具有更广泛适用性的智能数据平面解决方案，使其能够在不同硬件平台上实现智能流量分析。

**2. 安全性增强**：进一步研究如何提高智能数据平面解决方案的安全性，包括对可能的攻击方式和威胁进行更深入的分析，并提出更有效的防御方法。

**3. 性能优化**：进一步优化智能数据平面解决方案的性能，包括提高流量分析的速度和效率，降低硬件资源的消耗，以满足不断增长的网络流量和分析需求。

## 启发与收获：

在NetBeacon的实现中，我们见证了一种旨在提高数据平面上流量分析效率和准确性的创新方法。这一方法集成了多个关键技术，包括流级特征计算和多阶段模型推理。通过这些技术的融合，NetBeacon在Barefoot Tofino 1交换机上构建了一个原型，为高速流量分析开辟了新的可能性。

NetBeacon采用了聚合和摘要的特征计算方法，从数据包和流级别提取关键信息。这种方法使得NetBeacon能够有效地利用硬件支持的特征提取功能，实现数据平面上的高效流量分析。通过追踪流级特征的变化，NetBeacon能够更准确地分类流量，并在处理流的不同阶段应用适当的模型。

其次，NetBeacon的多阶段模型推理架构为不同阶段的流量分析提供了灵活性和精确度。通过在不同的推理点应用不同的模型，并根据每个阶段计算的特征进行推理，NetBeacon能够更好地适应流量的变化，并提高分类准确性。这种分阶段模型应用不仅能够减轻计算负载，还可以根据需求优化模型的使用。

NetBeacon的实现还充分利用了硬件支持的功能，并设计了算法和模型，在性能和资源利用率之间取得了平衡，实现了高效的流量分析。此外，NetBeacon还提出了解决存储冲突的方法，以确保数据的一致性和准确性。

这些启示为流量分析领域带来了深远的思考。NetBeacon的实现为我们提供了一个创新的示范，展示了如何利用现有的硬件和算法来应对流量分析中的挑战。通过结合特征工程、模型推理和硬件优化，我们能够设计出更高效、更准确的流量分析系统，为网络安全、流量管理等领域提供更好的支持。

通过借鉴NetBeacon的思想和方法，我们可以进一步推动流量分析领域的发展，为构建更安全、更高效的网络提供更好的支持。NetBeacon的实现为我们提供了重要的参考，帮助我们更好地理解流量分析的挑战和机遇。