间隔值草图:融合数据包间隔的多维度大流检测草图算法

林澍,黄家玮,邵婧,李思覃,梁琦,王启乐,赵艺琳
(中南大学计算机学院,湖南 长沙 410083)

摘 要: 在网络通信、网络流量管理等领域,快速、准确地识别大流对流量拥塞控制、恶意流量监测等任务具有重要意义。然而,现实网络环境中的数据流传输速率极高,使得大流检测变得异常复杂和具有挑战性。现有大多数大流检测方法主要依赖单一维度的统计数据,通常仅基于流中数据包的估计值进行流量统计与分析。这种方式的局限性在于其忽略了其他潜在维度的关键信息,如数据包间隔的分布特性,这些信息在准确识别大流时可能起到关键作用。针对上述问题,提出一种名为间隔值草图的大流检测算法。该算法通过引入流的估计值大小和数据包间隔分布特性这 2 个维度的流量特征,优化大流的保护与小流的替换策略。具体地,间隔值草图通过结合数据包间隔特性,有效区分大流与小流,从而在低内存条件下显著提升检测性能。为验证间隔值草图的准确性和有效性,采用 CAIDA 和 MAWI 两个真实网络流量数据集进行分析。实验结果表明,间隔值草图在多种内存设置和流量分布下均表现出显著优势,与现有方法相比,间隔值草图能够在内存资源受限的情况下保持较高的检测精度,F1 值最高可达到现有方法的 2.4 倍。

关键词: 草图算法;大流检测;网络测量;网络通信;准确性;数据包间隔
中图分类号: TP393 **文献标志码:** A **DOI:** 10.19678/j.issn.1000-3428.0252087

IntervalSketch: A Multi-Dimensional Sketch Algorithm for Heavy Flow Detection Integrating Packet Intervals

LIN Shu, HUANG Jiawei, SHAO Jing, LI Sitan, LIANG Qi, WANG Qile, ZHAO Yilin
(School of Computer Science and Engineering, Central South University, Changsha 410083, Hunan, China)

【Abstract】 In the fields of network communication and traffic management, rapid and accurate identification of heavy flows is of great significance for tasks such as traffic congestion control and malicious traffic monitoring. However, extremely high data transmission rates in real-world network environments render heavy flow detection exceptionally complex and challenging. Most existing heavy flow detection methods rely primarily on single-dimensional statistical data; typically, these methods obtain traffic statistics and perform analyses based solely on estimated flow packet counts. The limitation of such an approach lies in its neglect of critical information from other potential dimensions, such as the distribution characteristics of packet inter-arrival times, which may be crucial in accurately identifying heavy flows. To address these issues, this paper proposes a heavy flow detection algorithm called IntervalSketch. By incorporating two-dimensional traffic features—the estimated flow size and packet interarrival time distribution—the algorithm optimizes the protection of heavy flows and the replacement strategy for mouse flows. Specifically, IntervalSketch effectively distinguishes heavy flows from mouse flows by leveraging packet inter-arrival time characteristics, thereby significantly improving the detection performance under low-memory conditions. To validate the accuracy and effectiveness of IntervalSketch, two real-world network traffic datasets, CAIDA and MAWI, are used. The experimental results demonstrate that IntervalSketch exhibits significant advantages across various memory configurations and traffic distributions. Compared with existing methods, it maintains a higher detection accuracy under memory-constrained conditions, achieving an F1 value up to 2.4 times that of state-of-the-art approaches.

【Key words】 sketch algorithm; heavy flow detection; network measurement; network communication; accuracy; packet interval

基金项目: 国家自然科学基金重点项目(62132022);国家自然科学基金青年科学基金项目(62302524)。

作者简介: 林澍,男,硕士研究生,主研方向为计算机网络、网络测量草图算法;黄家玮(通信作者),教授、博士;邵婧、李思覃、梁琦、王启乐、赵艺琳,硕士研究生。

收稿日期: 2025-01-24 **修回日期:** 2025-03-26 **E-mail:** jiaweihuang@csu.edu.cn

计算机工程 www.ecice06.com

0 引言

近年来,网络数据规模和流量大小显著增长[1-2],网络测量对于网络通信、网络安全、流量控制、系统管理等场景非常重要[3-5],网络数据规模和流量大小的变化为转发节点数据平面的网络测量带来了巨大的挑战。部分优化算法虽然通过简化计算流程提升了处理速度,但往往以检测精度为代价,这严重影响了其在部分检测场景中的测量效果,特别是在软件定义网络、5G 网络等新型网络架构中,这种问题变得更加突出。

目前,大流检测在带宽分配、入侵检测、拥塞控制等重要场景中得到大量应用[6-8]。考虑到内存的限制条件,如何快速、准确地检测大流非常关键。现有方法主要分为基于替换的方法和基于草图的方法两类。

基于替换的方法的设计核心是仅记录部分流量,当内存占满时,用新到达的流量替换最小的流量来牺牲部分流量信息,从而减少内存消耗。经典方法包括空间节省(SS)算法[9]、紧凑空间节省(CSS)算法[8]和损耗计算算法[10]。

SS 算法在一个称为“流量摘要”的结构中记录有限数量的流量,如果一个到达的数据包属于现有流量,SS 算法会相应地更新计数器。如果内存没有空间来存储一个之前未记录的新流量,该流量将直接替换现有流量,并将相应的计数器加 1。然而,当内存不足时,小流量可能会错误地替换潜在的大流量。CSS 算法通过一个名为“迷你数据桌”的数据结构,改进 SS 算法,相较于 SS 算法,CSS 算法可以更有效地利用内存。损耗计算算法将数据流分割成许多时间窗口,并在窗口边界减少所有的计数器,同时让新流量替换计数器计数值减少到零的流量。但是,由于计数器的减少是在时间窗口变化时自动触发的,导致结果的误差率较高。随机接纳策略(RAP)[11]是一种排名测量方法,采用选择性替换策略来谨慎地替换小流量,从而提高了测量的准确性。

尽管上述方法具有速度快、吞吐量高的优点,但当内存空间不足并且小流数量较多时,小流可能错误地替换潜在的大流,从而导致大流检测的准确性下降。此外,新流量持续到达,并不断替换那些计数值不是很小的相对最小流量,会导致严重的高估问题。

为了在有限的内存下提高大流检测的准确性,学术界提出了基于草图的方法,该方法的一个固有特征是采用估计的计数值而不是精确的计数值,在内存占用和测量准确性之间取得了良好平衡。

弹性草图[12]通过将结构划分为大流部分和小流部分,以适应不同的流量特征。其中,大流部分采用哈希表来跟踪大流的信息,而小流部分则利用最小计数草图(CM)[13]结构来存储小流。然而,由于哈希冲突的存在,小流部分的测量值往往被严重高估,导致最终报告的结果存在误差。同时,小流部分使用大尺寸计数器来记录小流,不仅增加了内存开销,还造成了一定程度的空间浪费。此外,大流与小流部分之间频繁的数据包替换,也可能引发较高的测量误差。

为改善上述问题,主投票草图(MV-Sketch)[14]采用投票算法对最小计数草图进行优化,其核心结构由 3 个字段组成,即桶中计数值的总和、候选大流的键以及指示计数器。每当数据包到达时,它会被哈希到各行对应的桶中,更新操作包括增加计数值总和字段,并通过比较候选流键来调整指示计数器。最终,该方法结合多个相关桶的测量值计算流量估计结果,显著降低了测量误差。

另一方面,布隆草图[15]针对偏斜的网络流量分布场景进行优化,并通过分层计数器结构与布隆过滤器来缓解高估问题。在这一架构中,小流被存储在低层,而大流被存储在高层,以提高内存利用效率。然而,布隆过滤器固有的假阳性问题严重影响了结果的准确性。波动草图[16]通过引入波动计数器,采用随机递增和递减策略,以缓解计数器的高估现象。部分值草图[17]通过选择流键的不同部分进行估计,提升了吞吐量。冷过滤器[18]采用双层结构和小计数器来记录小流的频率,并在所有哈希计数器溢出时,将数据包标记为大流并传递至后续流处理模块。阶梯过滤器[19]结合最近最少使用算法,在有限内存条件下以较低的时间开销来过滤小流。哈希链草图[20]通过哈希链技术精确定位大流,提高了测量准确性。

综上所述,基于草图的方法[21-22]因其良好的准确性和较高的处理速度,在大流检测中的应用变得愈发广泛。它们使用哈希函数,将流量数据紧凑地存储在草图结构中,这是一种能够最小化内存使用的数据结构。与基于替换的方法不同,基于草图的方法不仅占用更少的内存,还具有更快的检测速度和更高的检测精度。

当前,基于草图的方法通常依赖于流量大小的估计值来进行替换,但在内存受限的情况下,由于流量大小的数据信息有限,这可能会降低检测精度。此类方法大多忽略了其他关键的网络流量特

征[23-24]，这一缺陷促使本文研究其他特性，并开发出一种利用多维度信息进行统计分析的新型草图。

通过实验探究发现，大流的数据包间隔比小流小得多，这是因为大流具有更高的传输速率，使得大流的数据包出现得十分紧凑，这一现象促使本文提出一种新方法——间隔值草图，它基于草图算法，核心思想是通过在草图的数据结构中监控潜在大流的流键、流量大小以及数据包间隔，来更新候选大流的相关信息。当引入新的流时，间隔值草图能够加强对结构中已存大流的保护，通过利用多维度统计信息，间隔值草图能够在内存资源有限的情况下，在流量大小遵循齐夫分布[25]的条件下，大幅提升检测精度。本文的主要贡献如下：

1)本文设计一种间隔值草图，其综合利用流量估计值大小和数据包间隔 2 个多维度统计信息，在快速处理和低内存占用的前提下，实现高效准确的大流检测。

2)本文的间隔值草图算法可以很好地将网络环境中的大流和小流进行区分，使其更好地适用于不同场景的网络测量任务(大流检测、突变流检测等)。

3)基于 CAIDA 和 MAWI 实验数据进行分析，验证间隔值草图在大多数内存配置下的检测精度。

1 研究背景与动机

1.1 大流检测

大流检测是一个非常重要的网络测量任务，通过持续监控和分析网络流量来识别高速流量。在给定的预设阈值下，如果某个流量在固定时间段内的流量大小超过该阈值，则该流量被识别为大流。大流检测广泛应用于网络管理、网络安全等场景中：在企业网络中，它有助于定位和解决可能导致网络拥塞的设备或应用程序；互联网服务提供商利用大流检测来监控和管理用户的带宽使用情况，从而提供更为优质的网络服务；在网络安全领域，它可以快速识别和响应诸如 DDoS 攻击等异常情况。

1.2 现存问题分析

从特征提取维度来看，现有算法主要存在以下局限性：传统草图算法大多采用单一维度的流量统计特征(如数据包数量或字节大小)进行设计，未能充分挖掘网络流量的多维特征。这种设计范式难以准确刻画复杂的流量行为模式，特别是在需要识别和区分多种流量类型(如大流、小流、突变流等)时，其检测性能往往难以满足实际需求，这直接导致算法在面对突发流量或动态变化的网络环境时，表现出较差的适应性和检测精度。

在内存优化方面，现有算法面临着精度与效率的权衡困境。为应对内存资源受限的场景，部分算法采用数据压缩策略进行存储和更新，虽然这种方案能够降低内存开销，但不可避免地引入了显著的测量误差，特别是在网络流量普遍遵循齐夫分布特性的情况下，少数大流往往占据了主要的网络资源，而现有算法由于哈希冲突、内存竞争以及替换策略的局限性，容易出现“大流错误替换”现象，即重要的大流信息被非关键小流所覆盖，严重影响了检测结果的可靠性。这一问题在网络拥塞或高负载环境下表现得尤为突出，成为制约算法性能提升的关键因素。

综上所述，现有草图算法在多维特征融合、误差控制、多场景测量等方面仍存在显著不足。如何在有限的内存约束下实现多维特征的融合，提升大流检测的准确性和鲁棒性，成为当前网络测量领域亟待解决的关键问题。

1.3 实证分析

本文使用 CAIDA 和 MAWI 的互联网数据[26-27]进行现有的基于草图方法的实证测试与分析，选择默默哈希[28]作为实验的哈希函数。数据集包含 2 900 万个数据包和 0.9 万个数据流，通过 64 bit 的源-目的对定义流量，选取主投票草图[14]和波动草图[16]作为本文的测试对象。通过精确率和召回率指标评估大流检测的准确性，精确率用来衡量报告的大流中真实大流的比例，召回率用来衡量报告的真实大流占所有实际大流的比例。

为了评估主投票草图和波动草图的准确性，本文将内存大小设置为 16～256 KB。如图 1 所示，当内存较大时，精确率和召回率都很好。然而，当内存为 32 KB 及以下时，效果显著下降。在内存有限的情况下，由于哈希冲突的存在，这种单维度的测量方法不足以维持良好的检测准确性。

接下来，探讨其他流量特性以辅助大流检测任务。网络环境中的流量大小遵循齐夫分布[25]，即网络中大多数流量都非常小，少数大流量产生了大部分的网络流量。本文首先定义数据包的平均间隔值。在网络测量周期内，数据包的顺序值表示其在网络流中的出现次序。例如，在一个测量周期内，若捕获了若干条网络流的总共 50 万个数据包，则其顺序值范围为 1～50 万。对于同一条网络流，数据包可以连续到达，也可能间隔性到达。假设某条网络流 f 在测量周期内的数据包顺序值为{10，50，100}，则其数据包平均间隔值为(100－10)/2，即 45。

www.ecice06.com

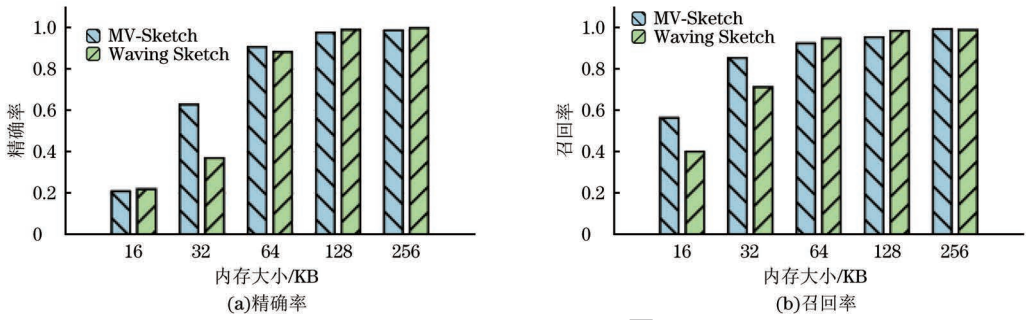

图 1 主投票草图和波动草图的准确性

Fig. 1 Accuracy of MV-Sketch and Waving Sketch

本文使用 CAIDA 数据集进行分析，如图 2 所示，横轴表示网络流的大小(即数据包数量)，纵轴表示网络流的数据包平均间隔值。首先，将网络流按照大小划分为 4 类，之后统计这 4 类网络流对应的数据包平均间隔值。由图 2 可知，网络流越大，其平均数据包间隔越小。因此，使用流中数据包的平均间隔来分析流量大小具有合理性，可以将数据包间隔作为有用的流量特征来辅助大流检测任务。

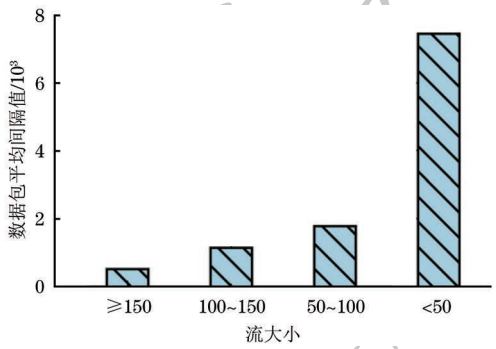

图 2 频率和间隔之间的关系

Fig. 2 Relationship between frequency and interval

综上所述，由于内存有限，现有方法在大流检测中难以保持高精确率和高召回率。流量大小估计的低准确性，是造成这一性能问题的主要原因。通过实验验证发现，数据包间隔是区分大流和小流的有效特征。这一结论启发本文提出一种新的基于草图的方法，利用数据包间隔来提高准确性，同时保持良好的内存效率。

2 主要方法

2.1 关键设计

为了解决在小内存条件下大流被小流错误替换的问题，本文设计了一种间隔值草图算法。该算法利用二维流量特征，每个桶获得哈希到桶中数据包的流键、流估计值大小和数据包间隔。间隔值草图中的替换策略同时考虑流量的估计大小和数据包间隔，优先保留大流量，从而提高检测的准确性。总结来说，本文方法旨在实现以下目标：

1)高检测准确性。间隔值草图支持非常准确的大流量检测任务。

2)小内存占用。间隔值草图使用较小的内存占用来保持紧凑的数据结构。

3)高处理速度。间隔值草图具有很高的数据包处理速度。

2.2 数据结构

图 3 展示了间隔值草图的数据结构，它由一个具有 d 行和 w 列的二维桶数组组成，用 $B(i, j)$ 表示第 i 行、第 j 列的桶($1\leqslant i\leqslant d, 1\leqslant j\leqslant w$)。$B(i, j)$由 3 个字段组成：键字段 $B(i, j).k$ 存储候选大流的 ID；计数字段 $B(i, j).v$ 存储候选大流的总包数；顺序值字段 $B(i, j).s$ 是被哈希到该桶中的已存数据包或当前更新后的顺序值。此外，每个数据包在进入草图结构之前已经按顺序进行了编号。

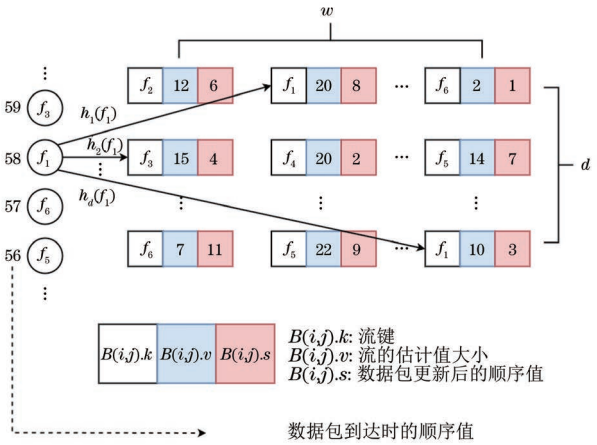

图 3 间隔值草图的数据结构

Fig. 3 Data structure of IntervalSketch

2.3 具体操作

插入与更新操作的伪代码如算法 1 所示。

算法 1 插入与更新操作

1. Initialize:
2. for j=1 to w do

```
3.    for i=1 to d do
4.      B(i, j).k=null;
5.      B(i, j).v=0;
6.      B(i, j).s=0;
7.    end for
8.  end for
9. For arrival packet P of flow f and sequence Ps:
10.  begin
11.    for i=1 to d do
12.      if B(i, hi(f)).k==f || B(i, hi(f)).k==
null then
13.        B(i, hi(f)).k=f;
14.        B(i, hi(f)).v=B(i, hi(f)).v+1;
15.        B(i, hi(f)).s=Ps;
16.      else
17.        if rand(0, 1)<1/(B(i, hi(f)).v×B(i,
hi(f)).s) then
18.          B(i, hi(f)).k=f;
19.          B(i, hi(f)).s=Ps;
20.        end if
21.      end if
22.    end for
```

1)更新。当一个数据包到达时,已经知道它的当前顺序值 Ps 和流键(ID)。然后,将流键为 f 的数据包通过哈希函数 hi(f)映射到桶中,下面是 3 种情况:

(1)检查桶中的情况。如果 f 不在任何桶中,并且至少有一个桶是空的,则将 $B(i, \mathrm{hi}(f)).k$ 设置为 f,并将 $B(i, \mathrm{hi}(f)).v$ 设置为 1。这是一种最简单的情况。

(2)将数据包哈希到每一行的一个桶中,先将流键与桶中 $B(i, \mathrm{hi}(f)).k$ 进行比较,若 $B(i, \mathrm{hi}(f)).k=f$,则将 $B(i, \mathrm{hi}(f)).v$ 增加 1,并将 $B(i, \mathrm{hi}(f)).s$ 更新为数据包的当前最新顺序值。

(3)如果桶既不为空也无法匹配到来的数据包,则需要进行更复杂的处理,即在桶中进行数据包的替换操作。替换操作的核心思想是不仅利用流量大小的估计值信息,还利用数据包的间隔大小信息。数据包的间隔值越大,它属于小流的可能性就越大。

间隔值草图使用概率替换策略来决定桶中现有的流是保留还是被到来的流 f 替换。其中,替换概率的计算公式如式(1)所示。一旦选择了一个候选的大流进行替换,桶中的 3 个字段将随之更新,以反映新的流数据信息。

$$P=\frac{1}{B(i,\mathrm{hi}(f)).v\times B(i,\mathrm{hi}(f)).s} \tag{1}$$

替换概率的计算公式体现了多维度的思想:$B(i, \mathrm{hi}(f)).v$ 越大,表明桶中已存流的数据包数量估计值越大,越有可能是一条大流;$B(i, \mathrm{hi}(f)).s$ 越大,表明桶中已存流的顺序值越靠后,和当前哈希过来的数据包顺序值的差值越小,数据包的间隔值越小,属于大流的可能性越大。因此,综合计算得出的替换概率越小,对已存储在草图中的流起到了很好的保护作用。

2)查询。在查询流 f 的大小时,间隔值草图确定流 f 被哈希到的特定桶。流 f 的估计值大小是通过哈希到这些桶中 $B(i, \mathrm{hi}(f)).v$ 的最小值来确定的。因此,如果流的数据存在桶中,间隔值草图将根据这个最小值提供流大小的估计值。

2.4 示例分析

为了更好地理解上述方法,本文提供几个示例:

1)示例 1。如图 4 所示,当一个流键为 f_1、顺序值为 15 的数据包到达时,其通过哈希函数被映射到 2 个桶中。此时可以看到,这 2 个桶都存储了属于流 f_1 的数据包。然后,增加存储该流的估计值大小字段 $B(i, j).v$ 的值,并将 $B(i, j).s$ 更新为当前数据包的顺序号,即 15。

2)示例 2。如图 5 所示,这是一个更复杂的情况,涉及本文算法中的替换操作。当数据包到达时,发现哈希映射到的 2 个桶都已满,并存储着来自其他流的数据包,需要尝试找到一个桶并替换其中现有的流。第一行哈希映射的桶中 $B(i, j).v$ 和 $B(i, j).s$ 的值分别为 3 和 4,第二行哈希映射的桶中 $B(i, j).v$ 和 $B(i, j).s$ 的值分别为 3 和 9。因此,替换概率分别为 1/12 和 1/27,显然第一行哈希桶的替换概率更大(更有可能是小流)。因此,选择第一行的桶进行替换,并更新该桶中的 3 个字段值。

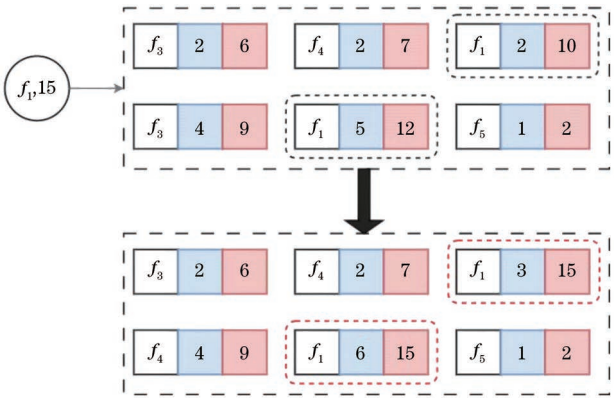

图 4 示例 1

Fig. 4 Example 1

2.5 大流和突变流检测

在一个预定义的测量周期 t 内,大流的判定基于

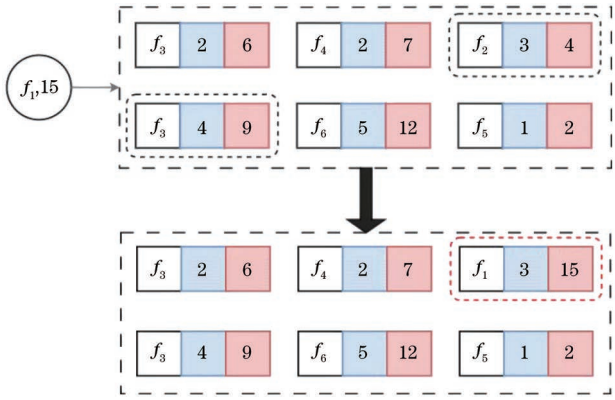

图 5 示例 2

Fig. 5 Example 2

阈值 θV,其中,θ 为默认大流比例阈值(取 0.000 8),V 表示该周期内的总数据包数。因此,大流阈值会随着流量规模的变化而动态调整。在每个测量周期结束后,间隔值草图会遍历并统计草图结构中的所有流信息,对每个流 f 统计其估计值,若该值超过阈值 θV,则该流被判定为大流。

识别突变流的过程是找到在 2 个相邻周期之间数据包数量变化超过某个阈值的流。

3 理论分析

3.1 空间与时间复杂度

假设间隔值草图的参数配置为 $d=\mathrm{lb}\ \delta^{-1}$,$w=2\varepsilon^{-1}$,其中,$\varepsilon(0<\varepsilon<1)$是近似参数,$\delta(0<\delta<1)$是错误概率。

定理 1 间隔值草图的空间复杂度为 $O(\varepsilon^{-1}\ \mathrm{lb}\ \delta^{-1}\mathrm{lb}\ n)$,每包更新时间为 $O(\mathrm{lb}\ \delta^{-1})$,检测所有大流的时间为 $O(\varepsilon^{-1}\mathrm{lb}^2\delta^{-1})$。

证明 间隔值草图中的每一个桶存储着一个流键字段、一个流估计值大小字段和一个顺序值字段,因此,每一个桶的内存使用为 $O(\mathrm{lb}\ n+2^{-1}\ \mathrm{lb}\ n+2^{-1}\ \mathrm{lb}\ n)=O(2\ \mathrm{lb}\ n)$。因为 2 是常数,所以每个桶的空间复杂度为 $O(\mathrm{lb}\ n)$,因此,间隔值草图的内存使用为 $O(dw\times\mathrm{lb}\ n)=O(\varepsilon^{-1}\ \mathrm{lb}\ \delta^{-1}\ \mathrm{lb}\ n)$。

每个数据包更新时需要进入到 d 个桶中,因此,需要进行 $d=\mathrm{lb}\ \delta^{-1}$ 次哈希操作,时间为 $O(\mathrm{lb}\ \delta^{-1})$。

查询所有的大流需要遍历 $d\times w$ 个桶,遍历完成后,对于 $B(i,\ j).v$ 超过阈值的那些桶,检测 d 个桶来获取估计值,$d\times dw=d^2w$,因此,检测所有大流的时间为 $O(d^2w)=O(\varepsilon^{-1}\ \mathrm{lb}^2\delta^{-1})$。

3.2 检测的误差边界值

本节通过理论分析来证明本文方法的大流检测不存在较高的估计误差。

定理 2 对于给定的流 f,用 $V_t(f)$和 $\hat{V}_t(f)$分别表示在特定时间 t 下流的真实值和估计值,其中,$\hat{V}_t(f)\leqslant V_t(f)$。

证明 在最开始 $t=0$ 时,$\hat{V}_t(f)$和 $V_t(f)$都为 0,定理成立。在时间 t 时,有 2 种情况:

1)如果到来的数据包匹配流 f,那么 $\hat{V}_t(f)=\hat{V}_{t-1}(f)+1$,且 $V_t(f)=V_{t-1}(f)+1$,定理成立。

2)如果到来的数据包不匹配流 f,需要利用之前所述的替换概率进行替换操作。为了降低高估所带来的误差,无论是否成功进行了替换,$B(i,\ j).v$ 的值均不变。因此,$\hat{V}_t(f)=\hat{V}_{t-1}(f)$,且 $V_t(f)=V_{t-1}(f)$,可得 $\hat{V}_t(f)\leqslant V_t(f)$,定理成立。

定理 3 给定一个正数 ε 和一个大流 f,在一个周期过后,如果 $S(f)$存在桶中,则有:

$$\Pr\{|(S(f)-\hat{S}(f)|\geqslant\lceil\varepsilon N\rceil\}\leqslant\frac{1}{\varepsilon^2 dw^2} \tag{2}$$

证明 该定理的核心是通过切比雪夫不等式推导出误差的概率界限。首先,对证明过程中出现的参数进行定义:

1)$S(f)$表示流 f 的真实流量,即流 f 的总数据包数。

2)$\hat{S}(f)$表示由算法估计出的流 f 的流量值大小。

3)误差 $E=S(f)-\hat{S}(f)$表示真实值与估计值之间的差异。

4)N 表示整个网络的总流量,即所有流的总数据包数。

首先进行误差的方差分析,每行 i 的干扰误差来源于其他小流 f',每个 f'落入流 f 对应桶的概率为$\frac{1}{w}$,可得单行误差的值为:

$$E_i=\sum_{f'\neq f}X_{f'} \tag{3}$$

$X_{f'}$表示小流 f'落入流 f 的桶的干扰量,那么对 $X_{f'}$进行期望计算可得:

$$\mathbb{E}[X_{f'}]=\frac{1}{w}\cdot S(f') \tag{4}$$

则 $X_{f'}$的方差为:

$$\mathrm{Var}[X_{f'}]=\frac{1}{w}\left(1-\frac{1}{w}\right)S(f')^2 \tag{5}$$

单行误差 E_i 的方差为:

$$\mathrm{Var}[E_i]=\sum_{f'\neq f}\mathrm{Var}[X_{f'}]=\sum_{f'\neq f}\frac{1}{w}\left(1-\frac{1}{w}\right)S(f')^2 \tag{6}$$

总误差 E 是 d 行误差的平均值:

计算机工程 www.ecice06.com

$$E=\frac{1}{d}\sum_{i=1}^{d}E_i \tag{7}$$

根据方差的性质，误差 E 的方差为：

$$\mathrm{Var}[E]=\frac{1}{d}\mathrm{Var}[E_i] \tag{8}$$

代入单行误差 E_i 的方差，可得：

$$\mathrm{Var}[E]=\frac{1}{d}\cdot\sum_{f'\neq f}\frac{1}{w}\left(1-\frac{1}{w}\right)S(f')^2 \tag{9}$$

当 w 较大时，可以忽略$\frac{1}{w}$的高阶项，进一步将式(9)化简为：

$$\mathrm{Var}[E]\approx\frac{1}{dw}\sum_{f'\neq f}S(f')^2 \tag{10}$$

接下来，进入切比雪夫不等式的代入部分。根据切比雪夫不等式可得：

$$\Pr(E\geqslant\varepsilon N)\leqslant\frac{\mathrm{Var}[E]}{\varepsilon^2N^2} \tag{11}$$

代入误差 E 的方差表达式后可得：

$$\Pr(E\geqslant\varepsilon N)\leqslant\frac{\frac{1}{dw}\sum_{f'\neq f}S(f')^2}{\varepsilon^2N^2} \tag{12}$$

记 $S_{\mathrm{SumSq}}=\sum_{f'\neq f}S(f')^2$ 为小流流量的平方和，进一步将式(12)化简：

$$\Pr(E\geqslant\varepsilon N)\leqslant\frac{S_{\mathrm{SumSq}}}{\varepsilon^2dwN^2} \tag{13}$$

如前文所述，网络环境中流量大小遵循 Zipf 分布，在流量满足 Zipf 分布($\alpha=1$)时，可认为 $S_{\mathrm{SumSq}}\approx N^2/w$，代入后可得：

$$\Pr(E\geqslant\varepsilon N)\leqslant\frac{1}{\varepsilon^2dw^2} \tag{14}$$

综上，可得误差的概率界限为：

$$\Pr\{|(S(f)-\hat{S}(f)|\geqslant\lceil\varepsilon N\rceil\}\leqslant\frac{1}{\varepsilon^2dw^2} \tag{15}$$

3.3 对比与总结

本文对间隔值草图和现有算法进行了比较与分析，包括持续到达草图[29]、主投票草图[14]和计数堆草图[30]。这些方法的复杂度和误差界限如表 1 所示。

由于草图的基本数据结构几乎相同，因此间隔值草图、持续到达草图和主投票草图 3 种算法的空间复杂度相同。但是对于计数堆草图而言，其还有一个额外的堆数据结构，因此空间复杂度为 $O(\varepsilon^{-1}\ \mathrm{lb}\ \delta^{-1}+H\ \mathrm{lb}\ n)$。

对于时间复杂度来说，前 3 种算法在进行插入和查询过程中都需要遍历 $d\times w$ 个桶，遍历完成后，对于超过阈值的那些桶，检测 d 个桶来获取估计值，$d\times dw=d^2w$，因此，检测所有大流的时间为 $O(d^2w)=O(\varepsilon^{-1}\ \mathrm{lb}^2\delta^{-1})$。而计数堆草图在更新与查询的过程中需要访问堆结构，其时间复杂度为 $O(H)$，随着 H 的增大而增加。

表 1 算法复杂度与误差界限比较

Table 1 Comparison of algorithm complexity and error bound

算法	空间复杂度	时间复杂度	误差的概率界限
间隔值草图	$O(\varepsilon^{-1}\mathrm{lb}\ \delta^{-1}\ \mathrm{lb}\ n)$	$O(\varepsilon^{-1}\ \mathrm{lb}^2\delta^{-1})$	$\frac{1}{\varepsilon^2dw^2}$
持续到达草图	$O(\varepsilon^{-1}\ \mathrm{lb}\ \delta^{-1}\ \mathrm{lb}\ n)$	$O(\varepsilon^{-1}\ \mathrm{lb}^2\delta^{-1})$	$\frac{1}{\varepsilon w}$
主投票草图	$O(\varepsilon^{-1}\ \mathrm{lb}\ \delta^{-1}\ \mathrm{lb}\ n)$	$O(\varepsilon^{-1}\ \mathrm{lb}^2\delta^{-1})$	$\frac{1}{2\varepsilon}$
计数堆草图	$O(\varepsilon^{-1}\ \mathrm{lb}\ \delta^{-1}+H\ \mathrm{lb}\ n)$	$O(H)$	$\frac{1}{\varepsilon}+H$

在误差的概率界限方面，间隔值草图有着最小的误差界限，这是因为其采用了多维度测量分析的思想，能够结合数据包间隔特征，更好地对流的特性进行区分，在误差控制方面具有更好的表现。

4 实验结果分析

4.1 实验设置

实验的服务器配备了六核 Intel Xeon W-2235 3.80 GHz 处理器和 64 GB 内存。每个 CPU 核心拥有 64 KB L1 缓存、1 MB L2 缓存以及 8.25 MB 共享 L3 缓存。服务器运行 Ubuntu 18.04.5 操作系统。实验中使用以下 2 种数据集：

1) CAIDA 数据集[26]。CAIDA 是收集于 Equinix-New York 数据监控器的 CAIDA 2019 年数据集，流量跟踪持续约 1 h。本文将跟踪数据分为 5 个部分以获取平均结果。每个周期设置为 1 min，包含约 3 100 万个数据包。本文测试了 10 个周期，以获取 1 min 的平均结果。

2)MAWI 数据集[27]。该数据集由 MAWI 工作组收集，旨在研究日本广域网的网络性能。本文选择了一个监控时长为 40 min 的 2020 年跟踪数据，包含约 1 027 万个数据包。

在对比方法及相关参数方面，使用源地址/目的地址对作为流键，流键字段设置为 64 位。本文选择了默默哈希[28]作为实验的哈希函数。参数 $d=4$，设置为 4 是因为在当前大多数草图算法实验中，草图的桶数据结构都设置为 4 行，其可以在测量准确率和空间利用效率之间取得良好平衡。w 的值由

计算机工程 www.ecice.com

实验中内存分配的大小计算得出。大流的判定基于阈值 θV，其中，θ 为默认大流比例阈值（默认取 0.000 8），V 表示该周期内的总数据包数。选取默默哈希的原因是其计算速度快，适合高吞吐量的应用场景，同时它的延迟较低，在网络测量等对延迟敏感的应用中，低延迟特性是非常关键的。默默哈希生成的哈希值具有均匀性和随机性，能有效减少哈希冲突，提升算法的准确性。本文将间隔值草图（简称为 Interval）与 5 种基线方法进行对比，包括主投票草图（简称为 MV）[14]、波动草图（简称为 Waving）[16]、计数堆草图（简称为 Ch）[30]、弹性草图（简称为 Elastic）[12] 以及持续到达草图（简称为 UA）[29]。前 4 种基线方法均为单维度的草图算法，而持续到达草图是一种多维度草图方法，利用网络流中数据包连续到达的特性，来辅助进行网络流量的统计与分析。

在评估指标方面，本文使用精确率（Precision）、召回率（Recall）、F1 值（F1）和相对误差（ARE）这 4 个指标来评估算法性能，在实验的最后还测量了吞吐量（Packet Rate）以衡量处理速度的快慢。

1）精确率表示正确报告的大流数占报告的所有流数量的比例。

2）召回率表示正确报告的大流数占真实大流数的比例。

3）F1 值表示精确率和召回率的调和平均值。

4）相对误差表示绝对误差与被测量真实值之比。

5）吞吐量表示相对于测量周期的整个持续时间，已添加到草图中的数据包所占的比例。本文使用每秒百万个数据包（Mp/s）作为数据包速率的量化单位。

4.2 准确性结果

1）基于不同内存大小的大流检测准确性。图 6 和图 7 展示了在 CAIDA 和 MAWI 数据集上大流检测的准确性。当内存低于 64 KB 时，流量测量对内存的需求较高，其他方法的准确性急剧下降，导致性能较差。MV、Waving 和 Elastic 方法的单一维度替换策略导致大流被错误驱逐，进一步影响了准确性。相比之下，基于多维度的方法 Interval 在 2 个数据集上都始终取得了更高的准确性，尤其是在内存有限的情况下。

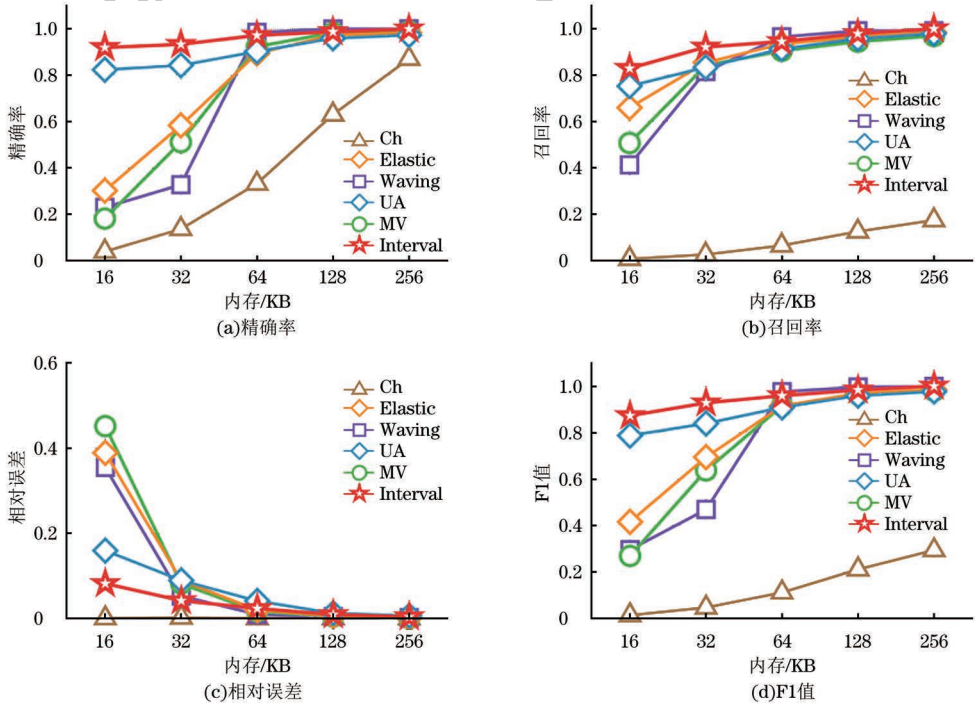

图 6 大流检测结果（CAIDA 数据集）

Fig. 6 Heavy flow detection results (CAIDA dataset)

2）基于不同阈值的大流检测准确性。图 8 展示了在 CAIDA 数据集上大流阈值对算法性能的影响。

实验中分配了 64 KB 的内存，并不断调整大流阈值，从 0.000 1 逐步增加至 0.002 4。结果表明，随着阈值的提高，报告的大流数量有所减少，所有算法的性能均有所改善。然而，在小阈值条件下，其他算法由于只基于单维度进行统计分析，性能明显下降。UA 采用了数据包的连续到达特性进行辅助分析，但是因为在网络环境中数据包的持续到达值未

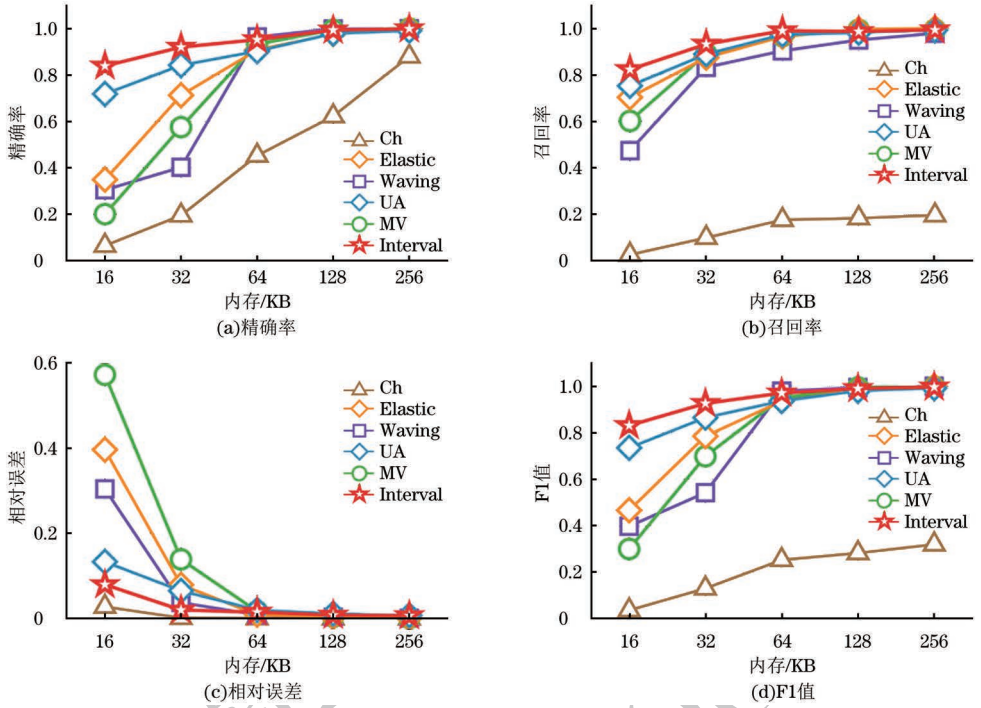

图 7 大流检测结果(MAWI 数据集)

Fig. 7 Heavy flow detection results (MAWI dataset)

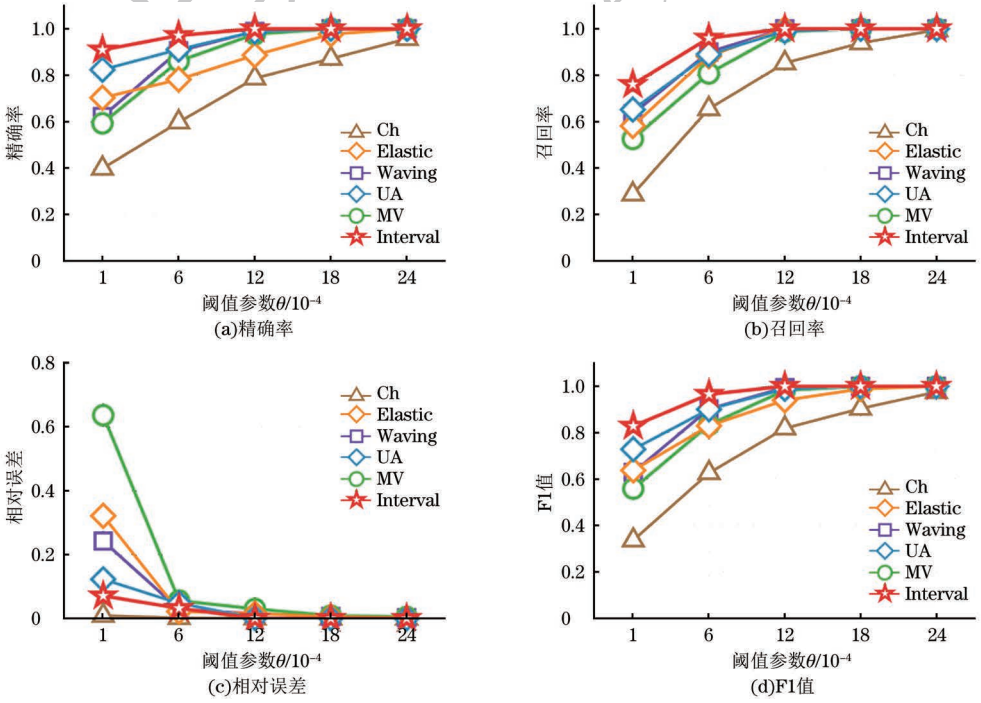

图 8 大流阈值的影响

Fig. 8 Impact of heavy flow threshold

必很大,导致其测量准确性不如本文 Interval 方法。Interval 即使在较小的阈值下仍能保持良好的性能,精确率超过 90%,召回率高于 75%,综合指标 F1 值也优于其他算法。这种优势主要得益于 Interval 利用数据包平均间隔值作为辅助特征进行分析,不仅增强了对大流的识别能力,还有效减少了对小流误判的情况。相比传统方法,多维度策略能更精准地捕捉流量的特征,提高了对大流的区分能力,即便在低阈值情况下仍能维持较高的检测精度。

3)基于字节计数的大流检测准确性。本文评估了在大流检测中使用字节计数的实验效果,其中计数字段记录了流的总字节数。图 9 展示了这种测量方法的准确性结果。在基于字节计数的大流检测

中，本文方法 Interval 在内存较小时精确率略低于 Elastic，这是因为 Interval 的替换概率由网络流大小和间隔值计算得出，而字节统计的网络流大小会略大于数据包统计方法，导致计算概率误差略高，但 Interval 在 F1 值上依旧领先于所有方法，包括另一种多维度方法 UA。

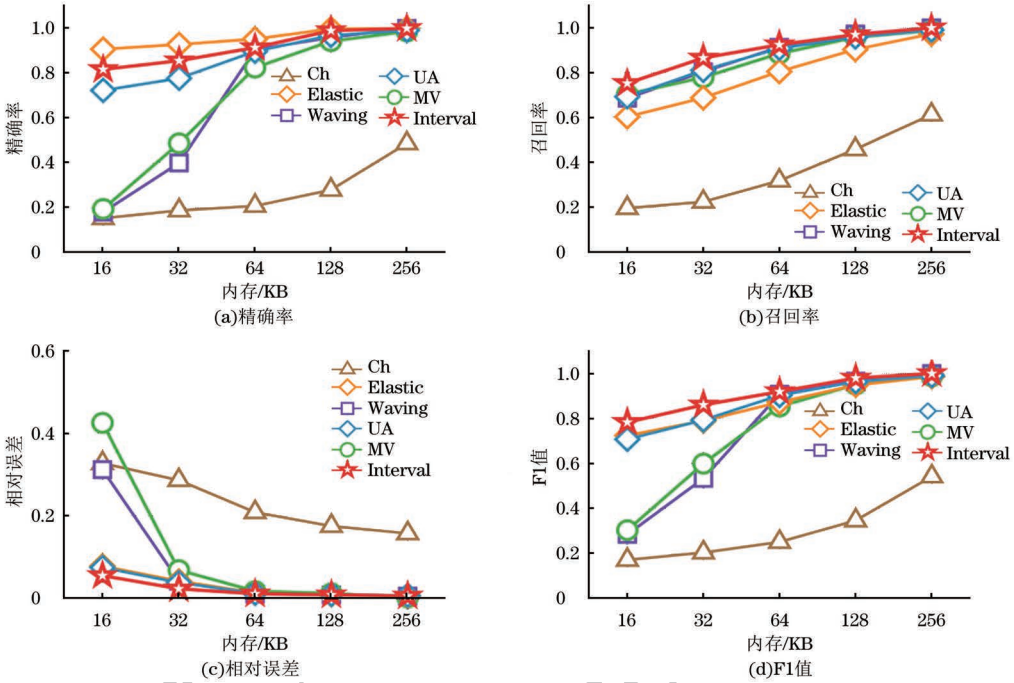

图 9 基于字节计数的大流检测结果

Fig. 9 Heavy flow detection results based on byte counting

4)突变流检测准确性。图 10 和图 11 展示了在 CAIDA 和 MAWI 数据集上突变流检测的准确性。本文方法在小内存条件下表现优秀，超越了其他经典方法。Ch 在大内存和小内存场景中都表现不佳，这是因为 Ch 使用了辅助堆结构，导致内存效率大幅降低。Interval提供了更优的流量大小估计策

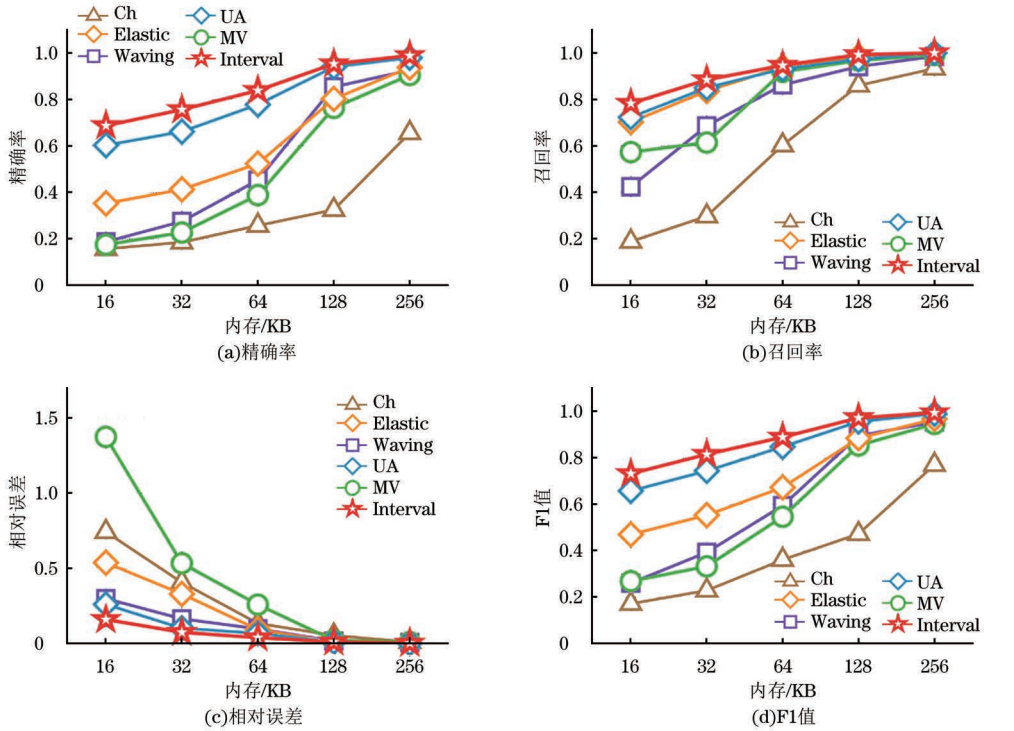

图 10 突变流检测结果(CAIDA 数据集)

Fig. 10 Bursty flow detection results (CAIDA dataset)

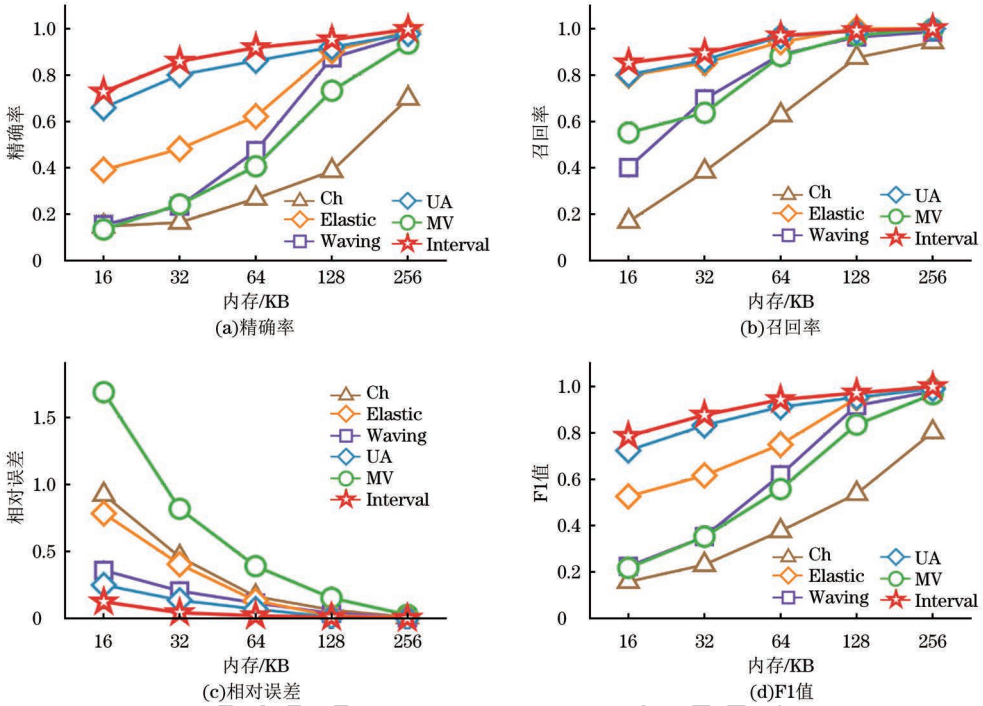

图 11 突变流检测结果(MAWI 数据集)

Fig. 11 Bursty flow detection results (MAWI dataset)

略,提高了突变流识别的准确率。

在第 3 节的理论分析部分,通过理论推导可知,与其他方法相比,Interval 具有最小的误差边界值,而本节的实验结果印证了前文的推导结果。本文方法的准确性在多种场景和多种统计测量方式下均优于对比方法,其拥有良好的准确性。

4.3 速度结果

除了上述算法准确性实验之外,本文还测试了 Interval 方法的处理速度,分别在 CAIDA 和 MAWI 数据集上进行数据包处理速度的测试。如图 12 和图 13 所示,本文方法的吞吐量达到了一个非常高的水平。以 CAIDA 数据集下的实验为例,与 Waving、Ch 相比,本文方法吞吐量性能分别提高了 293.43%、1 300.39%。UA 与本文方法一样,均采用了多维度的思想,但由于 UA 是基于连续值进行分析的,需要统计网络流中数据包的最大连续值和当前连续值,算法的计算开销较大,因此吞吐量不如 Interval。

在第 3 节的理论分析部分,对比了现有方法的空间与时间复杂度,现有方法与 Interval 均采用了标准的草图数据结构,因此它们拥有类似的空间与时间复杂度,但不同方法的具体计算过程不同,因此,最后算法的执行速度存在差异。Interval 不仅使用流大小的估计值作为判断条件,还利用数据包的间隔值来辅助大流检测任务,这对处理速度产生了轻微影响,然而其准确性的提升是巨大的。本文方法在确保优秀处理速度的同时,显著提高了准确性。

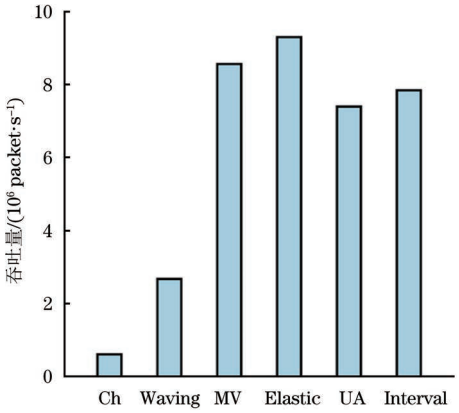

图 12 速度对比结果(CAIDA 数据集)

Fig. 12 Speed comparison results (CAIDA dataset)

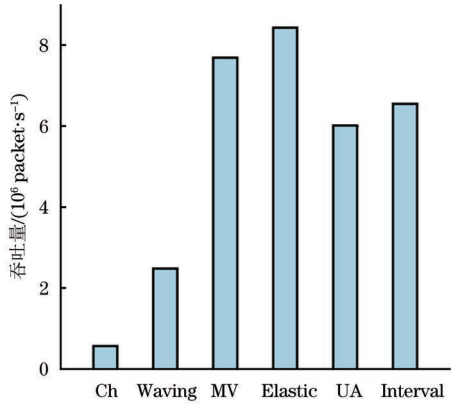

图 13 速度对比结果(MAWI 数据集)

Fig. 13 Speed comparison results (MAWI dataset)

5 结束语

本文提出一种基于多维数据统计的间隔值草图方法，旨在快速、准确且高效地识别大规模高速网络中的大流。该方法不仅结合了流量估计值大小，还融入了数据包的间隔特征，从而显著提升了流量替换操作的准确性。实验结果表明，即使在内存资源受限的情况下，间隔值草图在检测准确性和吞吐量方面依然优于传统方法。

参考文献

[1] 周伟林，杨莞，徐明伟. 网络功能虚拟化技术研究综述[J]. 计算机研究与发展，2018，55(4)：675-688.
ZHOU W L，YANG Y，XU M W. Network function virtualization technology research[J]. Journal of Computer Research and Development，2018，55(4)：675-688.（in Chinese）

[2] LU J，CHEN H C，ZHANG Z. LUSketch：a fast and precise sketch for top-k finding in data streams[C]//Proceedings of the International Conference on Computer Communications and Networks（ICCCN）. Washington D. C.，USA：IEEE Press，2022：1-10.

[3] 戴冕，程光，周余阳. 软件定义网络的测量方法研究[J]. 软件学报，2019，30(6)：1853-1874.
DAI M，CHENG G，ZHOU Y Y. Survey on measurement methods in software-defined networking[J]. Journal of Software，2019，30(6)：1853-1874.（in Chinese）

[4] CURTIS A R，MOGUL J C，TOURRILHES J，et al. DevoFlow：scaling flow management for high-performance networks[C]//Proceedings of the ACM SIGCOMM 2011 Conference. New York，USA：ACM Press，2011：254-265.

[5] KABBANI A，ALIZADEH M，YASUDA M，et al. AF-QCN：approximate fairness with quantized congestion notification for multi-tenanted data centers[C]//Proceedings of the 18th IEEE Symposium on High Performance Interconnects. Washington D. C.，USA：IEEE Press，2010：58-65.

[6] 王光明，李冬青，蒋从锋. 不平衡数据集下的数据中心网络流量异常检测[J]. 计算机工程，2025，51(8)：227-237.
WANG G M，LI D Q，JIANG C F. Network traffic anomaly detection for data centers in imbalanced datasets[J]. Computer Engineering，2025，51(8)：227-237.（in Chinese）

[7] DEMAINE E D，LÓPEZ-ORTIZ A，MUNRO J I. Frequency estimation of Internet packet streams with limited space[EB/OL]. [2024-12-05]. https://link.springer.com/chapter/10.1007/3-540-45749-6_33.

[8] 林耘森箫，毕军，周禹，等. 基于 P4 的可编程数据平面研究及其应用[J]. 计算机学报，2019，42(11)：2539-2560.
LIN Y S X，BI J，ZHOU Y，et al. Research and applications of programmable data plane based on P4[J]. Chinese Journal of Computers，2019，42(11)：2539-2560.（in Chinese）

[9] METWALLY A，AGRAWAL D，EL ABBADI A. Efficient computation of frequent and top-k elements in data streams[EB/OL]. [2024-12-05]. https://link.springer.com/chapter/10.1007/978-3-540-30570-5_27.

[10] MANKU G S，MOTWANI R. Approximate frequency counts over data streams[EB/OL]. [2024-12-05]. https://www.vldb.org/conf/2002/S10P03.pdf.

[11] BEN BASAT R，EINZIGER G，FRIEDMAN R，et al. Randomized admission policy for efficient top-k and frequency estimation[C]//Proceedings of the IEEE Conference on Computer Communications. Washington D. C.，USA：IEEE Press，2017：1-9.

[12] YANG T，JIANG J，LIU P，et al. Elastic sketch：adaptive and fast network-wide measurements[C]//Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. New York，USA：ACM Press，2018：561-575.

[13] CORMODE G，MUTHUKRISHNAN S. An improved data stream summary：the count-min sketch and its applications[J]. Journal of Algorithms，2005，55(1)：58-75.

[14] TANG L，HUANG Q，LEE P P C. MV-Sketch：a fast and compact invertible sketch for heavy flow detection in network data streams[C]//Proceedings of the IEEE Conference on Computer Communications. Washington D. C.，USA：IEEE Press，2019：2026-2034.

[15] ZHOU Y，JIN H，LIU P，et al. Accurate per-flow measurement with bloom sketch[C]//Proceedings of the IEEE Conference on Computer Communications Workshops（INFOCOM WKSHPS）. Washington D. C.，USA：IEEE Press，2018：1-2.

[16] LI J Z，LI Z K，XU Y F，et al. WavingSketch：an unbiased and generic sketch for finding top-k items in data streams[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York，USA：ACM Press，2020：1574-1584.

[17] MIAO R J，ZHANG Y D，ZHENG Z H，et al. CocoSketch：high-performance sketch-based measurement over arbitrary partial key query[C]//Proceedings of the IEEE/ACM Conference on Networking. Washington D. C.，USA：IEEE Press，2023：2653-2668.

[18] ZHOU Y，YANG T，JIANG J，et al. Cold filter：a meta-framework for faster and more accurate stream processing[C]//Proceedings of the 2018 International Conference on Management of Data. New York，USA：ACM Press，2018：741-756.

[19] LI Y P，WANG F Y，YU X，et al. LadderFilter：filtering infrequent items with small memory and time overhead[J]. Proceedings of the ACM on Management of Data，2023，1(1)：1-21.

[20] HUANG J W，ZHANG W L，LI Y J，et al. ChainSketch：an efficient and accurate sketch for heavy flow detection[J]. IEEE/ACM Transactions on Networking，2023，31(2)：738-753.

[21] SIVARAMAN A，SUBRAMANIAN S，ALIZADEH M，et al. Programmable packet scheduling at line rate[C]//Proceedings of the 2016 ACM SIGCOMM Conference. New York，USA：ACM Press，2016：44-57.

[22] LI S T，HUANG J W，ZHANG W L，et al. PA-sketch：a fast and accurate sketch for differentiated flow estimation[C]//Proceedings of the 31st IEEE International Conference on Network Protocols（ICNP）. Washington D. C.，USA：IEEE Press，2023：1-11.

[23] LAKHINA A，CROVELLA M，DIOT C. Characterization of network-wide anomalies in traffic flows[C]//Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement. New York，USA：ACM Press，2004：201-206.

计算机工程 www.ecice06.com

[24] HU Z P, REN H R, SHI P. Synchronization of complex dynamical networks subject to noisy sampling interval and packet loss[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(8): 3216-3226.
[25] POWERS D M W. Applications and explanations of Zipf's law[EB/OL]. [2024-12-05]. https://aclanthology.org/W98-1218.pdf.
[26] equinix-nyc. dirA.20190117-125910.UTC.anon.pcap[EB/OL]. [2024-12-05]. http://www.caida.org/data/overview/.
[27] MAWI_201901011400.pcap [EB/OL]. [2024-12-05]. http://mawi.wide.ad.jp/mawi/.
[28] GLADIS K, CHEN Y. The untapped potential of ascon hash functions: benchmarking, hardware profiling, and application insights for secure IoT and blockchain systems[J]. Sensors, 2025, 25(19): 5936.
[29] YE J, LI L, ZHANG W L, et al. UA-sketch: an accurate approach to detect heavy flow based on uninterrupted arrival[C]// Proceedings of the 51st International Conference on Parallel Processing. New York, USA: ACM Press, 2022: 1-11.
[30] CHARIKAR M, CHEN K, FARACH-COLTON M. Finding frequent items in data streams[EB/OL]. [2024-12-05]. https://www.vldb.org/pvldb/vol1/1454225.pdf.

文字编辑 吴云芳
栏目编辑 宋 圆

计算机工程 www.ecice06.com