计算机科学
COMPUTER SCIENCE

基于服务器无感知计算的网络带宽快速测量方法

杨若萱, 金飞宇, 曲连威, 周子杰, 郑奇斌, 李振华

引用本文

杨若萱, 金飞宇, 曲连威, 周子杰, 郑奇斌, 李振华. 基于服务器无感知计算的网络带宽快速测量方法[J]. 计算机科学, 2026, 53(3): 392-399.

YANG Ruoxuan, JIN Feiyu, QU Lianwei, ZHOU Zijie, ZHENG Qibin, LI Zhenhua. A Serverless-based Approach to Fast Measurement of Network Bandwidth [J]. Computer Science, 2026, 53(3): 392-399.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.250600011

基于服务器无感知计算的网络带宽快速测量方法

杨若萱[1] 金飞宇[1] 曲连威[1] 周子杰[1] 郑奇斌[2] 李振华[1]
1 清华大学软件学院 北京 100084
2 北京大数据先进技术研究院 北京 100195
(rosheenanea@outlook.com)

摘 要 以5G/6G和WiFi 6/7为代表的新一代移动网络在显著提升接入带宽(简称网速)的同时,也加剧了网络性能(如吞吐量、时延和丢包率)的波动,导致测速时间延长、流量开销增加、用户体验下降,并且难以满足基于微内核的泛在计算系统(如MINIX,QNX和seL4)在任务管理与进程间通信时对网络低成本、高实时性的迫切需求。基于云函数和云容器的服务器无感知计算虽然提供了潜在的解决思路,但现有的以Swiftest为代表的快速带宽测量方法深受长尾流量效应的制约,造成了严重的流量浪费。为此,提出一种基于服务器无感知计算的智能化网络带宽快速测量方法。首先,结合底层传输机制与实际测量数据分析了现有方法中长尾效应存在的原因。在此基础上,设计了突发式快启动的数据传输机制,将传统的平滑递增数据包发送策略分解为多轮短时的突发传输模式,通过客户端的动态反馈,实时调控发送速率、缩短传输时间并提高测量精度,规避因控制反馈延迟而引起的长尾流量浪费问题。多个典型网络场景下的实验结果表明,相比Swiftest,所提出的方法将服务端的发包流量减少了85%,测量时间缩短到平均1.6秒,显著降低了服务端资源压力和用户端流量开销,并且具备良好的泛在计算环境工程可部署性。

关键词:网络带宽测量;微内核;泛在计算;服务器无感知计算;突发式快启动
中图分类号 TP393

A Serverless-based Approach to Fast Measurement of Network Bandwidth

YANG Ruoxuan[1],JIN Feiyu[1],QU Lianwei[1],ZHOU Zijie[1],ZHENG Qibin[2] and LI Zhenhua[1]
1 School of Software,Tsinghua University,Beijing 100084,China
2 Advanced Institute of Big Data,Beijing 100195,China

Abstract Next-generation mobile networks,represented by 5G/6G and WiFi 6/7,have substantially increased access bandwidth (i.e.,network speed). At the same time,they have also amplified fluctuations in network performance-such as throughput,latency,and packet loss-thereby prolonging speed tests,increasing traffic overhead,and degrading user experience. More importantly,this volatility makes it difficult to meet the pressing requirement for low-cost and highly real-time networking in microkernel-based ubiquitous computing systems(e.g.,MINIX,QNX,and seL4),where task management and inter-process communication critically depend on timely and lightweight network support. Although serverless computing built on cloud functions and cloud containers offers a potential solution path,existing rapid bandwidth measurement methods(e.g.,Swiftest) remain heavily constrained by the long-tail traffic effect,leading to considerable traffic waste. To address this challenge,this paper propose a serverless-based approach to fast measurement of network bandwidth. We first analyze the causes of the long-tail effect in prior approaches by combining transport-layer mechanism analysis with real-world measurement data. Building on these insights,we design a bursty fast-start transmission mechanism that decomposes the conventional smoothly ramped packet-sending strategy into multiple rounds of short-duration burst transmissions. With dynamic feedback from the client,the sender regulates its transmission rate in real time to shorten measurement duration and improve estimation accuracy,thereby avoiding long-tail traffic waste induced by delayed control feedback. Experiments across multiple representative network scenarios show that,compared with Swiftest,the proposed method reduces server-side transmitted traffic by 85% and shortens the average measurement time to 1.6 seconds. These gains significantly alleviate server resource pressure and reduce client data consumption,while exhibiting strong engineering deployability in ubiquitous computing environments.

到稿日期:2025-06-03 返修日期:2025-09-05
基金项目:国家重点研发计划(2022YFB4500703);国家自然科学基金(62332012,62472245)
This work was supported by the National Key Research and Development Program of China(2022YFB4500703) and National Natural Science Foundation of China(62332012,62472245).
通信作者:曲连威(qlw@mail.tsinghua.edu.cn)

Keywords Network bandwidth measurement, Microkernel, Ubiquitous computing, Serverless computing, Bursty fast start

1 引言

近年来，随着数字通信技术的持续演进，全球网络接入能力经历了前所未有的飞跃式增长。在无线通信领域，WiFi 标准从 802.11n(WiFi 4)升级至 802.11ac(WiFi 5)、再到 802.11ax(WiFi 6)，每一代技术都在传输速率、频谱利用率与连接可靠性方面有显著提升。与此同时，蜂窝网络从 3G 向 4G 和 5G 的演进，也极大拓展了移动终端的带宽边界。尤其是 5G 网络，其商用部署已支持 500 Mbps 以上的典型下行速率和低于 10 毫秒的端到端时延，为移动用户带来了近乎光纤级的体验保障[1]。在有线接入方面，光纤到户与光纤到楼等宽带技术的快速普及，使得终端接入能力从传统的兆比特级跃升至千兆比特级，进一步释放了网络基础设施的能力上限[2-3]。这一系列技术变革不仅在理论性能上实现了突破，更直接催生了大量带宽密集型应用场景的落地，如 4K/8K 超高清视频、云游戏、虚拟现实、泛在计算与分布式计算、远程协同办公与智能网联汽车等。随着网络带宽的持续提升，泛在计算领域获得了新的发展动能，也对系统架构提出了更高要求。以微内核为核心的泛在异构系统在应对任务管理、任务调度及高效进程通信方面不断演进，其关键模块的设计与实现愈加精细高效，以满足复杂网络环境下的实时性与低开销需求。带宽的提升不再只是网络层面的问题，它正逐步成为决定新一代应用体验上限的关键变量。

在网络带宽能力持续提升、应用层负载愈加复杂的背景下，如何对网络实际运行性能进行准确测量与评估，已成为终端用户、服务提供商及网络运营商普遍关注的关键问题。对终端用户而言，带宽测量可用于刻画网络质量是否满足业务需求，从而支持流量管理与应用选择决策，并提升整体使用体验；对运营商而言，测速结果是服务质量评估、网络规划优化与资源调度的重要依据；而对于研究与监管机构来说，精准的测量数据更是理解网络运行现状与制定管理策略的基础[4-5]。在这一背景下，众多带宽测量工具与服务相继涌现。在专业测量领域，NetPerf[6] 和 iPerf[7] 等传统测试工具为网络技术人员提供了高度定制化的性能评估方案，能够全面检测带宽、时延和丢包等关键指标。然而，此类工具通常需要较为复杂的配置与操作流程，交互友好性有限，普通用户的学习与使用成本较高；同时，其测量过程可能产生较大的测试流量开销，从而对正常网络行为造成扰动。面向普通大众用户的网络测速服务，如 Speedtest.net[8]、Fast.com[9] 以及国内友声科技公司推出的“测网速”等，则以网页或移动应用的形式提供了更加简洁直观的测速体验，显著降低了用户使用门槛。这类服务不仅为用户提供实时可视化的网络状态反馈，也持续积累海量测量数据，为网络运营商与研究人员提供了极具价值的参考样本[10-11]。

当前商业领域的主流网速测量服务商通过部署海量且地理分散的测速服务器，以满足大规模用户的测速请求。以国际市场头部服务商为例，Netflix 运营的 FAST 测速系统在欧美地区布局超千个测速节点，而 Ookla 旗下的 Speedtest 构建了超过 16000 个节点的全球测速网络[8]。国内最大的测速服务商友声科技“测网速”应用为支撑 3500 万用户群体，在全国部署了 350 余个测速节点，日均数据传输量超过 60TB。尽管测速服务商为这种高密度服务器集群的建设和运维付出了高额的成本，但采用固定规模的集群存在双重困境。一方面，用户基数持续增长，导致基础设施存在扩容压力；另一方面，业务流量的昼夜周期性波动，造成资源利用率呈现“潮汐效应”，高峰期节点过载与闲时资源空置并存。

云原生计算(Cloud Native Computing, CNC)[10] 的兴起，为低成本弹性部署网络测速服务提供了理论上的可能性。云原生计算基于微服务[11] 架构实现功能解耦，通过容器化[12](如 Docker)封装与编排调度[13](如 Kubernetes)实现资源弹性伸缩，在提升系统服务能力的同时降低部署与运维成本。随着云原生技术的演进，无服务器计算(Serverless Computing, SC)[14] 逐渐成为重要的计算范式之一，其通过函数及服务(Function-as-a-Service, FaaS)模式将业务封装为事件驱动的函数单元。FaaS 平台在事件触发时自动实例化运行环境并执行函数，完成后立即释放资源，其事件驱动模型和细粒度调度相比基于容器的云原生技术具有更彻底的弹性伸缩能力，并且开发者无需管理基础设施。但是，在成本方面，云函数按照应用实际使用的流量收费，而非根据固定大小的带宽线路收费。

传统基于洪泛机制的测速模式存在测量耗时长、流量开销大等问题，而新兴的 Swiftest 方法虽已显著提升测速效率，但仍面临严重的流量浪费问题，特别是由“持续发包＋延迟反馈”引发的长尾流量现象，直接在云端应用现有方法会造成高额的流量开销，难以满足微内核等泛在计算系统在高效任务调度、精细化任务管理及低延迟进程间通信过程中对网络性能的严格要求。因此，本文围绕服务器无感知计算模式下带宽测量的关键挑战，首先分析并量化了 Swiftest 方法的长尾流量问题，并从机制与建模层面深入探讨其成因；随后，本文提出一种基于突发式快启动数据传输机制的测速方法，将传统的平滑递增发包策略重构为多轮短时突发传输，并在每轮传输中引入可主动停止的短时发包与闭环反馈控制机制，从而在保证带宽测量准确性的同时显著减少冗余流量；最后，本文实现了完整的系统原型，并通过实测验证了本文方法在流量消耗、测速准确性与测试耗时上的综合性能表现。

2 相关工作

网络带宽测量是网络性能优化和资源管理的关键技术。传统的洪泛式测量方法虽能提供较准确的带宽评估，但往往伴随高流量开销和网络干扰，而非洪泛式方法则在精度和实时性方面面临挑战。随着网络环境日益复杂，高效、精准、低成本的带宽测量成为研究重点。

2.1 网络带宽测量

网络带宽测量是众多网络系统的关键组件，不仅因为它直接关联到网络服务质量的评估，也因为它对于保障用户体验和网络资源优化具有不可或缺的作用[12-15]。在网络服务

的众多指标中，带宽的测量尤为重要，因为它直接反映了网络的传输能力和效率。近年来，随着网络技术的不断发展和应用的日益广泛，对蜂窝网络、Wi-Fi 网络和物联网等不同网络环境下带宽的统计分析成为了研究的热点[16-21]。

商业系统中几乎所有的带宽测量方法都采用了洪泛式的方法，即通过用数据包填满整个通信链路来测试带宽容量。这种方法简单直接，能够快速给出网络的最大传输能力。然而，在学术界，研究者也探索了一些非洪泛式的测量方法[22-24]。这些方法通过设计和发送具有特定模式的专用数据包，并记录这些数据包的传输时间来间接推断出网络带宽。相较于洪泛式方法，非洪泛式方法在理论上对网络造成的干扰更小，更适合于需要持续监测带宽而又不希望影响正常网络使用的场景。然而，非洪泛式方法的准确性和可靠性受到多种因素的影响。其中，时间信息的敏感性是一个主要的挑战。诸如丢包、排队延迟[25]以及确认报文(Acknowledge，ACK)聚合[26]等因素都可能导致测量结果出现偏差。丢包会直接影响数据包传输的连续性和完整性，从而影响带宽测量的准确度。排队延迟在网络繁忙时尤为显著，可能会导致数据包到达时间不确定，进一步影响带宽的计算。ACK 聚合技术尽管可以提高网络传输效率，但在进行带宽测量时可能会因为聚合导致的数据包传输特征改变而影响测量精度。

2.2 模糊拒绝采样

在传统拒绝采样中，目标分布 $T(x)$是已知的，但是在带宽测量中，目标分布 $T(x)$却是未知的。模糊拒绝采样将吞吐量样本点简化为一个模型，其中 N 个样本点的范围从 $V_{\min}$到 $V_{\max}$。其目标是找到一个区间$[V_x, V_y]$，使得该区间内同时具有较高的样本密度和较大的样本数量。为此，定义目标函数 $F(V_x, V_y)$来衡量区间$[V_x, V_y]$的质量[27]：

$$F(V_x, V_y) = Density \times Size = C \cdot \frac{K^2(V_x, V_y)}{V_y - V_x} \tag{1}$$

其中，$K(V_x, V_y)$表示区间$[V_x, V_y]$内样本点的数量；$C = V_{\max} - V_{\min}$是一个用于归一化的常数；$Density$ 表示样本密度，即单位区间内的样本数量；$Size$ 表示区间的大小。另外，为了防止区间过小导致密度过大，设定区间大小的最小阈值为 $(V_{\max} - V_{\min})/N$。在处理吞吐量样本时，选择使得目标函数 $F(V_x, V_y)$最大的区间$[V_x, V_y]$作为关键区间，在实际实现中以间隔 Δt 不断重复计算关键区间。当连续 m 个关键区间的重合度超过阈值 σ 时，就认为吞吐量已经趋于稳定，可以将关键区间内的样本均值作为带宽估计的结果。

在带宽测量过程中，虽然噪声常导致瞬时采样点剧烈波动，但带宽关键区间的存在与位置却基本保持稳定。如图 1 所示，关键区间内始终聚集着大量采样点，密集分布于V_x与 V_y之间。即便随着时间推移，新出现的采样点可能上下跳动，甚至偏离正常范围，但重新计算的关键区间也仅发生微小调整；异常点的出现反而进一步巩固了关键区间的稳定性。得益于关键区间在测速初期即迅速成型，该机制无需长时间持续测量，便能准确锁定有效带宽范围，从而显著减少测速流量开销并大幅缩短测量时长，同时保持较高的测量精度。此外，由于该方法对噪声干扰不敏感，测速服务器无需部署在距离用户较近的位置，因此可以利用低成本、弹性的公有云虚拟机或容器构建微型服务器池，在保障测速性能的同时，显著减少基础设施建设和降低运营成本。

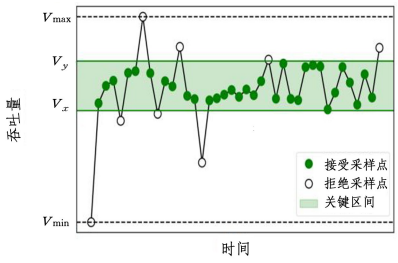

图 1 模糊拒绝采样和关键区间

Fig. 1 Fuzzy reject sampling and critical intervals

2.3 数据驱动的下行带宽测量

在服务端发送数据包进行带宽测量时，选择合适的发送速度尤为重要。不合适的发送速度可能使得客户端带宽达不到饱和，或者浪费太多流量。Swiftest[11]基于对泛在场景下网络带宽概率分布的理解，旨在优化拥塞控制算法的启动过程。该方法利用多模态高斯分布模型来确定初始探测数据率，并结合动态调整策略，快速收敛到实际的网络带宽。多模态高斯分布来源于对实际网络带宽数据的统计分析。在泛在场景下，不同移动接入技术(如 4G，5G，WiFi 等)的带宽分布通常呈现出多个峰值，这些峰值反映了不同网络条件下带宽的典型值。5G 的带宽分布可能在 172 Mbps，289 Mbps，485 Mbps 和 806 Mbps 处出现峰值，这些峰值与互联网服务提供商提供的典型固定宽带计划承诺的带宽相匹配[11]。这种多峰特性是由于网络性能受多种因素影响，包括带宽限制、基础设施状态、服务质量管理策略和信号强度变化等。通过对大规模带宽测量数据的分析，可以发现这些峰值对应的正态分布模式，从而将带宽分布建模为多个独立正态分布的加权组合。

首先，根据特定移动接入技术的带宽概率分布确定最可能的带宽值。带宽概率分布可以描述为多模态高斯分布，即多个独立正态分布的加权组合：

$$P(X) = \sum_{i=1}^{k} w_i N(X \mid \mu_i, \sigma_i) \tag{2}$$

其中，$N(X \mid \mu_i, \sigma_i)$是第 i 个正态分布，w_i是其权重，μ_i是均值，σ_i是标准差。初始探测数据率设置为权重最高的正态分布模式的均值，即$\mu_{\max}$。

在确定了初始探测数据率后，选择与之匹配的合理数量的测试服务器，以固定的时间间隔 Δt 采样一次的频率采集带宽样本 B_t，并检查最新的带宽样本是否低于当前探测数据率。如果带宽样本未低于当前探测数据率，表明网络尚未饱和，则将探测数据率调整到更大的典型带宽值之一，并可能添加更多服务器。如果带宽样本明显低于当前探测数据率，说明已接近或达到网络的饱和点，则保持当前探测数据率不变。

在连续多次测量中，如果最新的 m 个带宽样本显示出较小的波动(最大值和最小值之间的差异比小于 k)，则视为带宽测量基本收敛。此时终止测试，并通过带宽样本估算最终的带宽测量结果：

$$\text{最终带宽} = \frac{1}{m} \sum_{t=1}^{m} B_t \tag{3}$$

以上技术表明，通过智能化的算法设计，即使在受限的网

络环境下也能实现高效且准确的带宽测量。尽管这些方法在它们各自特定的应用场景中表现出色，但它们仍然存在局限性。例如，依赖于 TCP 慢启动机制的方法可能会因为网络状况的突然变化导致测量结果提前收敛，影响测量的准确性。而一些方法（如 Swiftest）由于需要和客户端交互而存在长尾流量的问题。同时，在网络环境发生变化，如添加新的服务器时，很多测量方法需要重新进行测速以适应新的网络状况，这增加了网络管理的复杂性。

3 问题描述与动机分析

3.1 Swiftest 的长尾流量问题分析

Swiftest[11] 作为较前沿的带宽测量技术，通过一套数据驱动的机制实现亚秒级的带宽快速测量，其核心思想是通过不断调整服务端发送速率，并结合客户端的反馈判断带宽是否饱和，以此估计用户侧的真实接入带宽。相比于传统洪泛式带宽测量方法，Swiftest 能在更短的时间内完成带宽估计，缩短了用户端的测试等待时间和降低了网络资源消耗，在大规模移动终端测速场景中表现出良好的实用性与推广价值。

以 5G 网络环境为例，在一次典型的 5G 下行带宽测试中，在探测过程中首先根据先验的 5G 带宽概率分布确定其中的模态带宽为[172，289，485，806] Mbps，对应不同程度的网络质量和用户实际体验带宽的可能取值。Swiftest 以其中概率最高的 172 Mbps 为初始探测速率，服务端最初将以 172 Mbps的速率向客户端发送测试数据包。客户端在接收到数据后，会实时进行带宽采样和分析，判断接收带宽是否达到了发送速率，即是否饱和。如果客户端检测到其接收带宽已接近 172 Mbps，此时客户端会立即向服务端发送“未饱和”反馈，请求服务端继续升高发送速率以进一步测试是否还有剩余带宽可用。接收到客户端的未饱和反馈后，服务端不会采用线性提升的方式增加发送速率，而是根据多模态高斯分布中排序的模态向下一个高概率模态跃迁。例如，此时将发送速率调整为 289 Mbps，再次进入测试阶段。服务端继续以新的速率发送数据包，而客户端继续监测带宽接收情况。如果再次未饱和，客户端继续发出反馈，直到某一轮客户端带宽无法跟上服务端发送速率，即接收带宽远低于发送速率时，说明此速率已超过客户端真实可用带宽，测量过程终止。此时，Swiftest 会回溯最近的一轮测试结果，认为该轮所处的客户端接收速率即为客户端的可用下行带宽近似估计值。Swiftest 方法的核心在于，通过交替的“提速→收敛→反馈”过程逼近客户端的带宽瓶颈点，在提升测速准确性的同时降低测量流量，是一种兼顾效率与轻量的现代测速设计，可以较为准确地评估用户实际的网络承载能力。

然而，在实际应用中，Swiftest 方法仍面临一个较为突出的性能瓶颈——长尾流量问题。长尾流量是指在网络带宽测量过程中，由于客户端带宽已计算出接收速率，而服务端仍在持续发送测试数据包，从而产生的一段额外且无效的数据流量。这部分流量并不会对最终的测量结果产生任何贡献，却会额外占用网络资源，造成流量浪费。

为系统评估 Swiftest 方法在不同网络环境下的流量消耗特性，特别是其长尾流量的实际占比情况，本文设计并执行了一组覆盖广泛接入方式和网络路径的实验。通过对典型网络类型下的测试过程进行采样记录与数据分析，采集了包括测试流量、长尾流量和测试耗时在内的核心性能指标，旨在揭示 Swiftest 在实际应用中可能存在的流量浪费问题。

本次测试选取 3 台手机作为客户端设备，其操作系统版本分别为 Android 11/12/13。服务端部署在两台云主机上，分别位于阿里云华北与华南区域，均配置为固定的 1 000 Mbps上下行带宽，以模拟不同地理路径下的真实网络环境。网络接入环境方面，客户端分别通过 3 类常见接入方式接入互联网，即 Wi-Fi 网络、4G 蜂窝网络以及 5G 蜂窝网络，其中蜂窝网络覆盖中国移动和中国联通两家运营商。测试过程中尽可能保持客户端处于空闲状态，避免后台任务对网络性能的干扰。

每次测速的流程如下：客户端与服务端建立连接通道，随后客户端发起测速请求，服务端根据客户端反馈动态调整发包速率。在客户端带宽收敛后，通知服务端停止发包，并记录收集到的带宽样本与传输数据。测试完成后，客户端输出总接收流量、带宽收敛时间与各类关键日志。在统计指标定义上，本文将客户端在收敛前接收到的用于带宽计算的数据量称为“测试消耗流量”，收敛后仍继续接收到的无效流量定义为“长尾流量”。两者之和为一次测速过程中服务端的总发包量。测试耗时则定义为客户端从发起测速请求到完成带宽收敛之间的时间区间。每台设备在每种网络环境下均进行不少于 10 次的测速实验，并取平均值进行对比分析。为了避免网络波动带来的离群影响，实验中剔除了明显异常的样本点，并统一在工作日白天的非高峰时段进行测量。

为直观展示 Swiftest 测速方法在不同网络环境下的流量消耗情况，特别是长尾流量的占比，本文在 3 类典型网络接入方式（4G，5G 和 WiFi）下分别进行了 30 次的测速实验，最后对 3 种类型网络的测试消耗流量和长尾流量进行了均值计算（All），并绘制如图 2 所示的对比柱状图。

从图 2 中可以看出，Swiftest 方法在所有网络类型下均产生了较为显著的长尾流量。以 5G 网络为例，单次测试的平均测试消耗流量为 32.89 MB，而平均长尾流量则达到了 36.15 MB，二者几乎持平，浪费率超过 100%。在 4G 网络中，测试流量仅为 8.29 MB，但长尾流量却高达 24.55 MB，是前者的近 3 倍。即便在带宽相对稳定的 Wi-Fi 网络下，测试消耗和长尾流量分别为 13.95 MB 与 18.58 MB，后者同样显著高于有效测试数据。根据 3 种类型网络平均值结果，长尾流量依旧略高于测试流量本身。

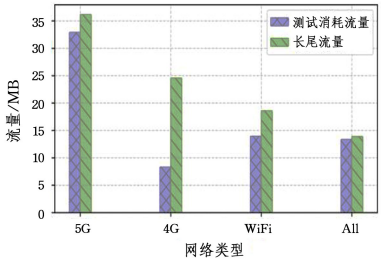

图 2 不同网络类型下 Swiftest 测试消耗流量与长尾流量

Fig. 2 Comparing Swiftest test consumption traffic with long tail traffic under different network types

为进一步分析流量浪费是否与测试耗时有关，对比了 3 种网络类型的平均测试耗时，同时统计了 3 种类型的平均值(All)，计算结果如表 1 所列。可以看到，在所有网络类型中，Swiftest 的测试耗时整体保持在亚秒级(即低于 2 s)，其中 WiFi 网络相对较为稳定，平均耗时最低，仅为 1.1 s 左右。尽管其耗时较短，但长尾流量比例仍高，说明该现象并非源于测试时间过长，而更可能与 Swiftest 本身的测试机制存在结构性问题。

表 1 不同网络类型下 Swiftest 测试平均耗时

Table 1 Average time spent on Swiftest tests under different network types

网络类型	平均耗时/s
5G	1.58
4G	1.42
WiFi	1.11
All	1.31

3.2 长尾流量产生的原因

如图 3 所示，Swiftest 的测速过程大致如下：客户端首先向服务端发起测速请求，服务端开始以某一初始速率向客户端发送数据包。

客户端根据收到的数据包计算带宽，当判断当前带宽尚未饱和时，将请求服务端提升发包速率；若检测到带宽已达饱和，则发送终止请求，要求服务端停止发包。然而，在服务端接收到“提升速率”或者“终止请求”前的这段时间内，其仍会按照当前速率持续发包(见图 3 中黄色背景部分)。这部分数据在客户端收敛判断后才被接收，已不再对带宽计算起作用，因此构成了所谓的“长尾流量”。这一现象的本质，是发送带宽冗余和客户端与服务端之间的控制延迟共同作用的结果。

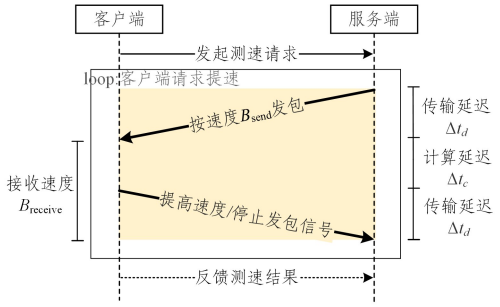

图 3 Swiftest 测速流程时序图(电子版为彩图)

Fig. 3 Timing diagram of Swiftest speed measurement flow

本文引入如下符号对测速过程进行建模：B_{send}，服务端发包带宽；$B_{receive}$，客户端接收带宽；Δt_d，服务端到客户端的单程网络延迟；Δt_c，客户端带宽计算收敛所需时间。

测速开始于时刻 t_0，服务端以带宽 B_{send} 开始发包，数据包经过 Δt_d 延迟后到达客户端。客户端耗时 Δt_c 完成带宽收敛计算，并立即发出“停止发包”信号，该信号再次经过 Δt_d 到达服务端。因此，服务端从 t_0 到 $t_0+2\Delta t_d+\Delta t_c$ 的时间段都在持续发包。

此时，服务端总共发送的流量为：

$$\text{总发包流量}=B_{send}\cdot(2\Delta t_d+\Delta t_c) \tag{4}$$

客户端用于测速计算的流量为：

$$\text{有效测试流量}=B_{receive}\cdot\Delta t_c \tag{5}$$

因此，长尾流量为：

$$\text{长尾流量}=B_{send}\cdot(2\Delta t_d+\Delta t_c)-B_{receive}\cdot\Delta t_c \tag{6}$$

整理得：

$$\text{长尾流量}=2B_{send}\cdot\Delta t_d+(B_{send}-B_{receive})\cdot\Delta t_c \tag{7}$$

其中，第一项 $2B_{send}\cdot\Delta t_d$ 表示数据往返传播所带来的基础延迟所对应的冗余发包；第二项 $(B_{send}-B_{receive})\cdot\Delta t_c$ 表示在客户端仍处于收敛阶段期间，由于服务端发包速率高于接收能力而引发的冗余流量积累。

Swiftest 的设计目标是通过其“数据驱动的速率调节机制”，使得服务端带宽 B_{send} 能够快速逼近客户端真实带宽 $B_{receive}$，以期最小化发送带宽冗余，即尽可能压缩第二项中的带宽差因子 $(B_{send}-B_{receive})$。该策略对于稳定网络环境下的带宽精确探测较为有效。然而，在实际应用中，由于网络动态波动、客户端响应延迟和速率调整步长等多种因素的共同作用，Swiftest 很难做到实时精准地收敛于 $B_{receive}$，从而导致第二项中的带宽冗余积累仍然显著。

基于这一观察，本文选择了不同于 Swiftest 的优化方向：不试图进一步逼近 $B_{receive}$，而是主动控制 Δt_c，通过主动停止、小窗口的猝发发送机制，尽早终止不必要的发送行为，从而减少冗余带宽持续存在的时间。该思路的核心逻辑是：即便相较于 $B_{receive}$ 仍存在一定误差，但只要能够及时发现并停止不再贡献无效的数据流量，就能显著抑制冗余数据的累积。这一思想直接启发了本文在下一章中提出的一种突发式快启动数据传输机制，其设计目的正是通过主动停止与频繁反馈，在时间维度上打断冗余数据的累积过程，从而显著降低长尾流量。本质上，本文从“带宽域最小化带宽差”转向了“时间域最小化持续冗余窗口”，实现了从速率调优向时序调控的优化范式转变。

4 基于服务器无感知计算的网络带宽快速测量方法

4.1 算法设计与机制说明

为解决 Swiftest 方法中存在的持续发包导致冗余数据积累的问题，利用服务器无感知计算低成本的特性，设计了一种突发式快启动数据传输机制的快速带宽探测方法。该方法通过将服务端的发包行为划分为主动停止的多轮短时间突发发包，并在每轮发包后根据客户端反馈动态调整测试策略，从而在保证带宽测量精度的前提下，有效抑制了长尾流量的产生。

与传统“持续发包+控制信令终止”的策略不同，突发式快启动传输机制具体步骤如下。

突发式发送：服务端在接收到测速请求后，选择一个初始探测速率(一般遵循多模态高斯分布)，并在一个短时间窗口内进行一次自动停止的突发发包。

客户端反馈判断：客户端接收数据并使用模糊拒绝采样算法[27]对带宽样本进行实时聚类，判断带宽是否进入“关键区间”并趋于稳定。该算法通过在样本分布中寻找密度最高的稳定区间，并监测该区间在时间轴上的重合度，实现对带宽收敛状态的鲁棒判断。若样本显示已稳定，则立即反馈“终止”信号；否则，反馈“继续探测”信号。

多轮反馈迭代：服务端根据反馈结果，决定是否进行下一轮突发发包，并在每一轮中动态调整发送速率(通常按模态带宽区间逐级上升)。通常整个过程在 3～5 轮内即可收敛。

测试终止与结果输出：一旦客户端反馈带宽饱和，服务端立即停止发包，客户端统计有效样本并输出最终带宽估计值。

该算法采用迭代式探测结构，在每轮发包之后立即根据客户端反馈，决定是否进行下一轮，从而在控制冗余发包的同时快速收敛至准确带宽估计值。在该算法中，每轮突发发送的时间窗口 T_{burst} 一般设置为 100～200 ms，客户端通过关键区间采样判断带宽是否饱和。该方法的关键思想在于，将传统测速中“持续发包、直到饱和”这一过程改造为“短时发送、自动停止、反馈决策”的模式，从而实现带宽估计过程中对无效数据发包的最小化。

长尾流量的理论估计可表示为：

$$L_{\text{total}}=\sum_{i=1}^{n}\left(B_{\text{send}}^{(i)}\cdot T^{(i)}-B_{\text{recv}}^{(i)}\cdot T_{\text{eff}}^{(i)}\right) \tag{8}$$

其中，$B_{\text{send}}^{(i)}$ 表示第 i 轮的服务端发送速率，$T^{(i)}$ 为该轮发包时长，$T_{\text{eff}}^{(i)}$ 为客户端在该轮中判定为“有效测量”的部分时间窗口。通过限制每轮 $T^{(i)}$ 的长度并提前终止测试，该方法在大多数情况下可将 $T_{\text{eff}}^{(i)}$ 快速压缩至较小区间。相比 Swiftest 方法，虽然突发式快启动数据传输机制方法可能在反馈轮数上略多，但其每轮发包时长极短、控制点更密集，从而在整体测试耗时可控的前提下，显著降低了延迟反馈导致的冗余流量。

另外，该客户端判断机制得益于模糊拒绝采样算法对网络噪声的鲁棒性，使得每轮发包无需等待长时间的样本积累即可进行判断，从而进一步增强了突发式快启动数据传输机制的时效性与流量控制能力。与原始模糊拒绝采样方法依赖长时间持续采样不同，本文方法通过每轮小窗口突发采样即可触发关键区间判断，从而将带宽稳定性的判断与冗余控制更紧密地结合于时间窗口内，实现更高效的实时判断。

4.2 系统实现与测试

基于服务器无感知计算的快速带宽探测方法的系统整体采用客户端-服务端架构设计，客户端运行于移动设备上，负责测速任务的发起、样本采集与反馈控制；服务端部署在云平台上，提供按需的测速容器资源，并承担实际的数据发送与调速控制任务。

客户端测速功能集成于原生 Android 应用中，兼容 Android 8 至 Android 13 系统版本。应用启动后会首先与服务端建立控制通道和数据通道。在测速过程中，客户端实时接收服务端发送的突发包，并对接收到的吞吐样本运行模糊拒绝采样算法，以动态判断是否已达带宽饱和状态。一旦收敛完成，客户端将带宽估计结果和终止信号发送回服务端，并输出测速结果。

服务端以轻量级云容器的形式部署在公有云平台中，每个测速任务由调度中心动态分配至一个测速容器实例。宽带探测服务模块（Bandwidth Probing Service，BPS）接收到测速请求后，将根据客户端反馈动态调整突发发包速率，并在收到终止信令后立刻停止发包。服务端的容器实例具有无状态特性，具备快速启动与释放能力。

测速系统的核心通信流程主要分为以下 4 个阶段。

初始化与连接建立：客户端启动后，首先通过 HTTPS 控制通道向服务端发送测速请求，并建立一对用于控制与数据传输的 TCP/UDP 通信通道。服务端响应请求并分配测速所需端口资源。

突发包发送与样本收集：服务端按照预定的速率列表依次发送短时间窗口的突发数据包，客户端实时接收数据并记录每轮的吞吐量样本。

样本分析与反馈判断：客户端使用模糊拒绝采样算法对当前轮次内的带宽样本进行关键区间判定，若发现样本稳定聚集于某一高密度带宽段并达到设定重合度阈值，则向服务端返回“终止”信号；否则，发送“继续”信号并进入下一轮。

终止与结果输出：服务端接收到终止信号后立即停止发包，客户端汇总所有轮次中的关键区间，计算最终估计带宽值。

在具体实现上，客户端以原生 Android Java 编写，底层使用 DatagramSocket 进行 UDP 发包与监听，并通过 WebSocket 建立控制通道，实现与服务端之间的低延迟指令交互。模糊拒绝采样与关键区间判断模块由独立线程异步运行，确保在数据采样过程中不阻塞主线程。服务端采用基于服务器无感知计算的轻量级容器部署方式，每个测速容器都具备独立的速率控制逻辑和会话状态管理能力。

通过上述系统构建与流程实现，突发式快启动数据传输机制在实际网络条件下展现出良好的低开销与高响应能力。下一章将对比 Swiftest 方法，系统评估该机制在准确性、流量消耗与测速耗时等维度的性能表现。

5 实验验证与性能评估

为评估基于服务器无感知计算的快速带宽测量方法在实际环境中的性能表现，在典型的 4G，5G 及 WiFi 网络条件下，对比测试了本文方法与 Swiftest 方法在流量消耗、测量准确性与测试耗时 3 项核心指标上的表现。每种网络环境下均进行不少于 30 组的独立测速实验，并取平均值进行对比分析。

表 2 列出了两种方法在不同网络环境下的总服务端发包流量对比，以及统计了 3 种类型网络的平均值。可以看出，Swiftest 方法在所有网络类型中均表现出明显的流量开销，平均每次测速的发包量在 50 MB 左右，5G 条件下的平均每次测试流量更是高达 93.1MB。而突发式快启动数据传输机制方法显著减少了服务端数据发送量，平均发包流量控制在 7.6MB 左右，节省比例超过 85%。

进一步地，图 4 将发包流量拆解为“有效测试消耗”和“长尾流量”两部分，同时对 3 种类型的网络进行了均值计算（All）。Swiftest 方法中，长尾流量占比普遍高达 50%以上，尤其在 4G 网络下，长尾流量甚至超过测试本身所需。而反馈式突发包方法中，长尾部分基本被压缩至可忽略的水平，验证了其在时域控制上的优势。

表 2 两种机制在不同网络类型下的服务端消耗流量

Table 2 Server-side traffic consumption under different network types for two mechanisms

（MB）

机制	5G	4G	WiFi	All
Swiftest	93.1	14.6	48.3	52.9
突发式快启动数据传输机制	12.1	2.0	8.0	7.6

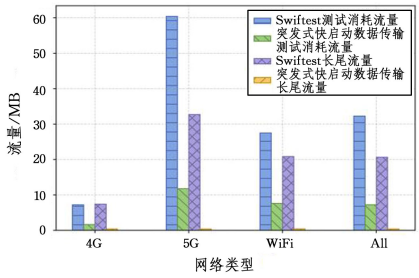

图 4 突发式快启动传输机制与 Swiftest 在不同网络类型下的测试消耗流量与长尾流量

Fig. 4 Testing traffic consumption and long-tail traffic for the bursty fast-start transmission mechanism and Swiftest in different network types

图 5 展示了两种方法测得带宽值与真实带宽之间的相对误差比较。本文以标准洪泛式方法测量出的终端带宽作为真实带宽，将两种方法输出的带宽估计结果与测试终端实际带宽进行对比，计算其相对误差率。在 90% 以上的实验样本中，突发式快启动数据传输机制方法的估计值与真实带宽之间的误差低于 10%，高于 Swiftest 方法。

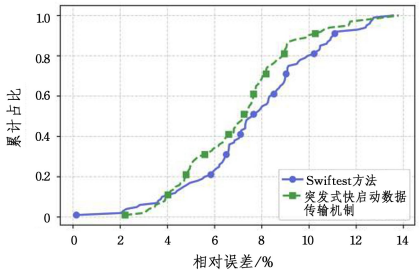

图 5 突发式快启动数据传输机制与 Swiftest 真实带宽的相对误差累计分布

Fig. 5 Cumulative distribution of relative error between the bursty fast-start transmission mechanism and Swiftest actual bandwidth

表 3 列出了两种测速方法在不同网络环境下的平均测试耗时。尽管突发式快启动数据传输机制方法在测试耗时上略高于 Swiftest，但最多没有超过 0.3 s。

表 3 突发式快启动数据传输机制与 Swiftest 在不同网络类型下的测速耗时

Table 3 Bursty fast-start transmission mechanism and Swiftest speedtesting elapsed time for different network types

(s)

机制	5G	4G	WiFi	All
Swiftest	1.58	1.42	1.11	1.31
突发式快启动数据传输机制	1.87	1.63	1.30	1.60

这一差异可以通过理论模型进行解释：设每轮突发发送的时长为 t_{send}，服务端初始发包速率为客户端可接收带宽 k 倍，即 $B_{send}=k\cdot B_{receive}$，其中 $k>1$。客户端与服务端之间的单向网络延迟为 t_d，则每轮测速需经历一次完整的控制反馈往返以及数据传输时间。若整个过程在 N 轮内完成收敛，则理论测试总耗时可估算为：

$$T_{total}=N\cdot(2t_d+k\cdot t_{send}) \tag{9}$$

以实验中常见配置为例，设 $t_{send}=100$ ms，$N=3$，$t_d=30\sim50$ ms，$k=1.2\sim2$，代入计算，有：

1）若 $k=1.2$，$t_d=30$ ms，则 $T_{total}=3\times(2\times30+1.2\times100)=3\times(60+120)=540$ ms；

2）若 $k=2$，$t_d=50$ ms，则 $T_{total}=3\times(100+200)=900$ ms。

因此，即使在理想情况下，该方法的最小测试耗时也在 500～900 ms 内，主要来源于其轮询反馈机制而非效率瓶颈。这一耗时代价是要实现更高效的流量控制与长尾流量压缩所必须付出的合理代价。值得强调的是，尽管该方法在理论上耗时略高，但其总测试时长依然保持在 1～2 s 内，远低于传统洪泛式测速工具的 10～30 s 等级，充分满足移动端用户对测速“秒级响应”的使用预期。

综上，在多类网络环境中，反馈式自控突发包测速方法在流量控制方面展现出显著优势，同时仍保持较高的测量准确性与可接受的测试时延。尤其是在移动端设备与计费流量敏感的使用场景下，该方法具备更强的实用性与推广价值。

结束语 本文面向服务器无感知计算架构，研究新一代移动网络环境下的快速带宽测量问题。针对 5G/6G 与 WiFi 6/7 在提升接入带宽的同时，加剧了测速耗时增加与流量开销上升的问题，本文通过传输机制分析与实测数据验证，揭示了 Swiftest 等现有方法的长尾流量成因：持续发包叠加反馈延迟，使得客户端收敛后服务端仍发送大量冗余数据，造成显著流量浪费并加重服务端与终端负担。

为解决上述问题，本文提出了基于突发式快启动数据传输机制的带宽快速探测方法，通过将发包过程拆分为多轮自停短时突发，并引入实时反馈机制动态调整测试速率，实现了对冗余发包的有效控制。实验结果表明，该方法在大幅度降低服务端流量的同时，依然保持了较高的带宽估计准确性与用户可接受的测试耗时，验证了其在实际移动网络场景下的实用性。

参 考 文 献

[1] YOU X H, PAN Z W, GAO X Q. 5G Mobile Communication Trends and Key Technologies [J]. Scientia Sinica, 2014, 44(5): 551-563.

[2] Federal Communications Commission. Measuring broadbandamerica fixed broadband report: A report on consumer fixed broadband performance in the US[R]. Federal Communications Commission, 2014.

[3] BURGER E W, KRISHNASWAMY P, SCHULZRINNE H. Measuring broadbandamerica: A retrospective on origins, achievements and challenges [J]. ACM SIGCOMM Computer Communication Review, 2023, 53(2): 11-21.

[4] BISCHOF Z S, OTTO J S, SÁNCHEZ M A, et al. Crowdsourcing ISP characterization to the network edge[C]// Proceedings of W-MUST. 2011: 61-66.

[5] LI Z, LI X, YANG X, et al. Fast uplink bandwidth testing for internet users [J]. IEEE/ACM Transactions on Networking, 2023, 31(4): 1886-1901.

[6] HEWLETTPACKARD. HewlettPackard / netperf [EB/OL].

[2025-03-06]. https://github.com/HewlettPackard/netperf.
[7] GUEANTV. iPerf-The TCP,UDP and SCTP network bandwidth measurement tool[EB/OL]. [2025-03-06]. https://iperf.fr/.
[8] OOKLA. Speedtest by Ookla-The Global Broadband Speed Test [EB/OL]. [2025-03-06]. https://www.speedtest.net/.
[9] NETFLIX. Internet Speed Test |Fast.com[EB/OL]. [2025-03-06]. https://fast.com./.
[10] QAMAR F,HINDIA M N,DIMYATI K,et al. Interference management issues for the future 5g network:a review[J]. Telecommunication Systems,2019,71:627-643.
[11] YANG X,LIN H,LI Z,et al. Mobile access bandwidth in practice:Measurement,analysis,and implications[C]// Proceedings of SIGCOMM. 2022:114-128.
[12] LI Y,DENG H,PENG C,et al. iCellular:Device-customized cellular network access on commodity smartphones[C]// Proceedings of NSDI. 2016:643-656.
[13] SUNDARESAN S,DENG X,FENG Y,et al. Challenges in inferring internet congestion using through put measurements [C]// Proceedings of IMC. 2017:43-56.
[14] DENG H,PENG C,FIDA A,et al. Mobility support in cellular networks:A measurement study on its configurations and implications[C]// Proceedings of IMC. 2018:147-160.
[15] LI F,NIAKI A A,CHOFFINES D,et al. A large-scale analysis of deployed traffic differentiation practices[C]// Proceedings of SIGCOMM. 2019:130-144.
[16] HUANG J,XU Q,TIWANA B,et al. Anatomizing application performance differences on smart-phones[C]// Proceedings of MobiSys. 2010:165-178.
[17] HUANG J,QIAN F,GERBER A,et al. A close examination of performance and power characteristics of 4G LTE networks [C]// Proceedings of MobiSys. 2012:225-238.
[18] HUANG J,QIAN F,GUO Y,et al. An in-depth study of LTE:Effect of network protocol and application behavior on performance[J]. ACM SIGCOMM Computer Communication Review,2013,43(4):363-374.
[19] XU D,ZHOU A,ZHANG X,et al. Understanding operational 5G:A first measurement study on its coverage performance and energy consumption[C]// Proceedings of SIGCOMM. 2020:479-494.
[20] NARAYANAN A,RAMADAN E,CARPENTER J,et al. A first look at commercial 5G performance on smartphones[C]// Proceedings of WWW. 2020:894-905.
[21] NARAYANANA,ZHANG X,ZHU R,et al. A variegated look at 5G in the wild:Performance,power,and QoE implications [C]// Proceedings of SIGCOMM. 2021:610-625.
[22] HU N,STEENKISTE P. Evaluation and characterization of available bandwidth probing techniques[J]. IEEE Journal on Selected Areas in Communications,2003,21(6):879-894.
[23] DISCHINGER M,HAEBERLEN A,GUMMADI K,et al. Characterizing residential broadband networks [C]// Proceedings of IMC. 2007:43-56.
[24] YANG T,JIN Y,CHEN Y,et al. RT-WABest:A novel end-to-end bandwidth estimation tool in IEEE 802.11 wireless network [J]. International Journal of Distributed Sensor Networks,2017,13(2):1550147717694889.
[25] HU N,LI L,MAO Z M,et al. Locating internet bottlenecks:Algorithms,measurements,and implications[J]. ACM SIGCOMM Computer Communication Review,2004,34(4):41-54.
[26] MELANDER B,BJORKMAN M,GUNNINGBERG P. Regression-based available bandwidth measurements[C]// Proceedings of SPECTS. 2002:14-19.
[27] YANG X,WANG X,LI Z,et al. Fast and light bandwidth testing for internet users[C]// Proceedings of NSDI. 2021:1011-1026.

YANG Ruoxuan, born in 2000, postgraduate. Her main research interests include network measurement and cloud computing.

QU Lianwei, born in 1995, Ph. D, research associate, is a member of CCF (No. V0464G). His main research interests include privacy protection, cloud computing and cellular network.

（责任编辑：柯颖）