

面向超算系统的带内网络遥测架构设计与调度算法研究

曹继军^{1),2)} 徐佳庆^{1),2)} 李晋文^{1),2)} 杨汉芝¹⁾ 孔德永³⁾ 戴 艺^{1),2)}

¹⁾(国防科技大学计算机学院 长沙 410073)

²⁾(国防科技大学并行与分布式计算重点实验室 长沙 410073)

³⁾(湖北经济学院信息工程学院 武汉 430205)

摘 要 网络测量是网络监控管理、配置调优和故障诊断的基础。近年来,带内网络遥测由于实时性和准确性等特点,成为当前网络测量技术研究的热点。针对超算系统(即高性能计算系统)的高速互连网络场景,本文提出了一种轻量级带内网络遥测架构(LW-INT),该架构摒弃对IEEE 1588全局时钟同步协议的依赖,采用分段延迟度量方法,通过在网络芯片端口集成轻量级内建遥测单元(BITU),实现了对互连网络的端到端路由路径、链路状态和逐跳步网络时延的精确测量,且具有硬件逻辑资源占用少,工程实现难度低的显著优势。在此基础上,本文定义了一种最小路径遥测覆盖问题(MPCP-T),证明了该问题为NP完全问题,并提出了两种启发式遥测任务的调度算法(A2A-CB和A2A-EP),分别采用“消减优化”和“增量扩展”策略,以较少的遥测调度次数(即遥测任务量)实现对全网测量对象的完整覆盖。为验证所提架构和算法的有效性,基于FPGA平台完成LW-INT核心模块的硬件实现与资源评估,并在Fat-Tree、Torus、Dragonfly三种超算典型网络拓扑开展多维度实验,同时借助Mininet仿真平台分析遥测机制的网络资源占用特性。实验结果表明,LW-INT架构的BITU模块在关键资源占用方面较IEEE 1588模块降低35%以上,分段延迟度量方法可实现传输延迟的精准测量;在含 P 个主机节点的网络中,A2A-EP算法可将A2A基础算法的遥测任务量由 $O(P^2)$ 降低为 $O(P)$,性能与更高计算复杂度的INTO-CH基准算法相当,A2A-CB算法的性能略逊于A2A-EP算法,但计算复杂度更低,更适合大规模超算网络场景,且两种算法的遥测任务量均显著优于SARG基准算法。综上所述,本文所提出的LW-INT架构及其A2A-CB/EP遥测调度算法,有效解决了超算互连网络数据平面轻量级、高效率测量的关键问题,为下一代超算系统互连网络的状态感知与智能运维提供了重要的技术支撑和实践参考。

关键词 超算系统;互连网络;带内网络遥测;轻量级架构;遥测调度算法

中图法分类号 TP393 **DOI号** 10. 11897/SP. J. 1016. 2026. 01966

Research on In-band Network Telemetry Architecture Design and Scheduling Algorithm for Supercomputing Systems

CAO Ji-Jun^{1),2)} XU Jia-Qing^{1),2)} LI Jin-Wen^{1),2)} YANG Han-Zhi¹⁾ KONG De-Yong³⁾ DAI-Yi^{1),2)}

¹⁾(College of Computer Science and Technology, National University of Defense Technology, Changsha 410073)

²⁾(National Key Laboratory of Parallel and Distributed Computing, National University of Defense Technology, Changsha 410073)

³⁾(School of Information Engineering, Hubei University of Economics, Wuhan 430205)

Abstract Network measurement is the foundation of network monitoring, configuration optimization, and fault diagnosis. In recent years, in-band network telemetry has become a

收稿日期:2025-10-24;在线发布日期:2026-03-24。本课题得到国家重点研发计划项目(No. 2025YFB3003100)及子课题(No. 2025YFB3003102)资助。曹继军(通信作者),博士,副研究员,中国计算机学会(CCF)会员,主要研究领域为高性能互连网络、网络测量与管理、复杂编码电路等。E-mail: caojijun@nudt.edu.cn。徐佳庆,博士,副研究员,主要研究领域为高性能计算和高性能互连网络。李晋文,博士,研究员,主要研究领域为计算机系统结构、高速电路设计等。杨汉芝,学士,工程师,主要研究领域为网络测量与管理技术。孔德永(通信作者),博士,副教授,主要研究领域为计算机网络与通信。E-mail: kdykong@hbue.edu.cn。戴 艺,博士,副研究员,主要研究领域为高性能互连网络、集合通信等。

hotspot in current network measurement technology research due to its real-time performance, high accuracy and outstanding characteristics. Aiming at the scenario of high-speed interconnection networks in supercomputing systems (i. e., high performance computing system), this paper proposes a LightWeight In-band Network Telemetry architecture (LW-INT). Abandoning the reliance on IEEE 1588 Precision Time Protocol for global clock synchronization, this architecture adopts a segmented delay measurement method and integrates lightweight Built-In Telemetry Units (BITU) into the ports of network chips, which enables accurate measurements of end-to-end routing paths, link status and hop-by-hop network latency in interconnection networks. It also features the remarkable advantages of low occupation of hardware logic resources and low difficulty in engineering implementation. On this basis, this paper defines the Minimum Path Coverage Problem for Telemetry (MPCP-T), proves that the problem is NP-complete, and proposes two heuristic scheduling algorithms for telemetry tasks (A2A-CB and A2A-EP). Adopting the strategies of “cutback optimization” and “incremental expansion” respectively, the two algorithms achieve complete coverage of all measurement objects in the entire network with fewer telemetry scheduling times (i. e., less telemetry task quantity). To verify the effectiveness of the proposed architecture and algorithms, the hardware implementation and resource evaluation of the core modules of LW-INT are completed based on an FPGA platform, and multi-dimensional experiments are carried out on three typical supercomputing network topologies, namely Fat-Tree, Torus and Dragonfly. Meanwhile, the Mininet simulation platform is used to analyze the network resource occupation characteristics of the telemetry mechanism. Experimental results show that the BITU module of the LW-INT architecture reduces the occupation of key hardware resources by more than 35% compared with the IEEE 1588 module, and the segmented delay measurement method can realize the accurate measurement of transmission latency. In a network with P host nodes, the A2A-EP algorithm reduces the telemetry task quantity of the basic A2A algorithm from $O(P^2)$ to $O(P)$, and achieves performance comparable to that of the INTO-CH benchmark algorithm with higher computational complexity. Although the A2A-CB algorithm is slightly inferior to the A2A-EP algorithm in performance, it has lower computational complexity, making it more suitable for large-scale supercomputing network scenarios. Moreover, the telemetry task quantities of both algorithms are significantly better than that of the SARG benchmark algorithm. In summary, the LW-INT architecture and its A2A-CB/EP telemetry scheduling algorithms proposed in this paper effectively solve the key problems of lightweight and high-efficiency measurement for the data plane of supercomputing interconnection networks, and provide important technical support and practical references for the state perception and intelligent operation and maintenance of interconnection networks in the next-generation supercomputing systems.

Keywords supercomputing system; interconnection network; in-band network telemetry; lightweight architecture; telemetry scheduling algorithm

1 引言

高速互连网络通常是指超算系统(或称为超级计算系统、高性能计算系统、HPC系统)内部用于连接计算节点、管理节点、存储节点的网络。它是计算

系统全局性基础设施,是构成计算系统并实现各个计算部件协同工作的关键部件之一。超算系统的互连网络通常由成千上万的网络接口芯片(Network Interface Chip, NIC)和网络交换芯片(Network Routing Chip, NRC)构成。为了对高速互连网络进行高效可靠管控,需要对网络路由路径、性能指标、

拥塞状态等进行精确测量。网络测量是互连网络监控管理、配置调优和故障诊断的基础。目前,高速互连网络的测量通常采用基于带外管理(Out-Band Management)或带内管理(In-Band Management)的查询式测量方法。具体而言,网络测量的过程依赖于网络的管理平面和控制平面,网络测量的信令通过带外或者带内管理路径传输,网络测量的请求由运行在管理服务器上的管理程序发起,网络测量的响应由被测量对象被动读取相关状态信息产生。这种传统网络测量方式难以追踪数据平面中数据流的实际转发路径及其实时的路径时延性能,不可避免地存在着“观察者效应”和“测不准效应”等问题^[1],这也会导致利用网络测量结果实施网络控制往往难以达到预期效果。此外,随着超算系统规模的不断增大和网络管理与运维智能化发展,传统的网络测量方案已经越来越难以满足当前超大规模HPC系统互连网络管理及运维在实时性和准确性等方面的需求。

带内网络遥测(In-Band Network Telemetry, INT)是一种不需要网络管理平面和控制平面干预的由网络数据平面收集、传输和报告网络状态的网络测量技术。INT技术最早由Barefoot、Arista、Dell、Intel和VMware等公司于2015年共同提出。随后, Kim等人^[2]于同年发表了第一个INT原型实现。P4.org应用工作组对INT规范^[3]的发布与更新,对于推动INT技术的标准化、促进网络技术创新以及支持智能网络发展具有重要作用。由于相比传统网络测量在监控方式、实时性、监控范围、资源占用、准确性等方面具有显著优势,INT技术可以广泛地应用于网络故障诊断^[4]、拥塞控制^[5]、路由决策^[6]、流量工程^[7-8]、运维可视化^[9]、路由验证^[10]等方面。作为一种快速兴起的新型网络测量方案,INT技术日益受到相关领域学术界和工业界的重点关注,并向着遥测架构创新、上层应用支撑、应用场景扩展、任务调度算法优化、安全机制设计等方向不断发展。

当前,INT标准主要针对的是可编程有线网络,然而由于其先进性,研究者已经将INT数据平面测量的思想扩展到IP-over-Optical网络^[9]、无线传感器网络6TiSCH^[11]、802.11无线网络^[12]、数据包光网络^[13]、混合网络^[14]等场景,而且随着网络可编程性的提高而不断创新发展。高速互连网络和以太网在网络架构和性能指标等方面存在着明显差异。将上述带内网络遥测方法直接应用于高速互连网络将存在不少缺陷:首先,对数据平面流量进行随路测量时,没有考虑不同虚通道(Virtual Channel, VC)之间的

实时延迟性能差异,因此不能精确测量高速互连网络转发路径不同虚通道的网络传输性能。其次,通过带内遥测测量网络逐跳步时延时,需要对遥测报文打上时间戳(Timestamp)。这要求网络能够支持IEEE 1588精密时钟同步协议(IEEE 1588 Precision Time Protocol, IEEE 1588)。然而,高速互连网络通常不支持该协议,为其增加该协议需要较大的实现代价,且目前尚不具备足够强的应用场景需求。2016年,国产“天河”专用互连网络^[15]提出并实现了带内网络管理机制和协议。然而,由于不具备数据流随路测量能力,因此仍属于传统的网络测量技术。可见,针对超算系统的高速互连网络场景,有必要研究设计一种低资源消耗的带内遥测方法,以实现随路由路径和网络时延等关键随路信息的测量。

本文面向超算系统的带内网络遥测技术开展研究,创新点及主要贡献如下:(1)针对超算系统高速互连网络的遥测需求,提出一种轻量级的带内网络遥测架构(LightWeight In-Band Network Telemetry, LW-INT),通过分段延迟度量方法,在无需IEEE 1588全局时钟同步的前提下,实现了端到端路由路径、链路状态及逐跳步时延的精准测量,具有硬件逻辑资源占用少且易于实现等优点;(2)定义了最小路径遥测覆盖(MPCP-T)问题,证明了该问题为NP完全问题,并设计了两种启发式调度算法(A2A-CB、A2A-EP),通过“消减优化”和“增量扩展”的差异化策略,以更少遥测任务量实现全网络策略对象覆盖,其中A2A-EP算法将A2A算法的任务量从 $O(P^2)$ 降低至 $O(P)$,A2A-CB算法则在性能与计算复杂度间实现平衡;(3)基于FPGA验证平台,评估了LW-INT架构核心单元BITU的资源消耗,并验证了分段延迟度量中传输延迟的可行性。同时,通过在Fat-Tree、Torus和Dragonfly三种典型超算拓扑上的实验,验证了所提出的两种调度算法在不同规模网络中的有效性。此外,借助Mininet网络实验平台,对本文提出的带内遥测机制进行了全面评估,分析了其在网络资源占用(包括带宽消耗和CPU占用)以及分段延迟度量等关键性能和功能指标上的表现。

2 相关工作

与本文内容相关的研究工作主要涉及带内遥测架构、调度算法、应用场景,以及超算系统网络测量与管理,简述如下。

2.1 带内遥测架构

带内网络遥测按原理可分为 AM-PM^[16-17] 和 INT^[3] 两类。AM-PM 借助数据包头部的标记字段测量丢包率、时延等性能,精度较高,但需网络设备支持时间同步;INT 则通过专用遥测报文携带指令与数据,可采集更丰富的网络状态。在时延测量方面, Kim 等人^[2] 提出基于可编程数据平面的 INT 技术,通过遥测报文携带路径标记和交换机排队延迟信息,实现 HTTP 请求延迟测量及延迟剧增原因分析; IntMon^[18] 则依托 ONOS 控制器和 BMv2 交换机实现 UDP 封装的 INT 方案,通过为 INT 报文标记时间戳完成网络时延指标测量。在路径验证方面, Sel-INT^[19] 通过概率采样逐跳步收集设备 ID,检测流表配置错误导致的转发路径不一致问题; CherryPick^[10] 和 PathDump^[20] 针对思科 In-situ OAM 平台设计,均由交换机插入路径 ID,其差异在于前者由控制器验证路径,后者由用户终端捕捉数据包并统计转发路径。

2.2 带内遥测调度

单个遥测任务仅能测量部分遥测对象,如何调度多个遥测任务实现对指定范围或全网内遥测对象的覆盖,是带内遥测调度的核心问题。 INTO^[21] 首次建模该问题,提出集中与平衡两类优化问题,在证明其为 NP-Complete 问题后,设计了启发式算法求解。 INTOPP^[22] 引入机器学习,将遥测任务形式化为整数线性规划模型,最大化任务与网络流的时空相关性匹配,实验表明其异常检出率较 INTO 算法大幅提升。 IntOpt^[23-24] 是面向虚拟网络功能服务链的监控框架,以满足监控需求情况下最小化监控流为目标,采用模拟退火随机贪心启发式 (Simulated Annealing Random Greedy Heuristic, SARG) 算法降低开销。针对概率型 INT 架构^[25], Wei Chen 等人^[26] 提出了优化调度算法,通过流量数据包分配实现多并行 PINT 任务的高效调度。 Yuan Peng-Y 等人^[27] 提出了 SDN 中基于图分割的自适应带内网络遥测探测路径配置 (Adaptive Configuration for In-Band Network Telemetry based on Graph Segmentation, ACGS) 方法,利用拓扑划分限制探测路径长度,并在划分的子拓扑中求解欧拉回路得到探测路径。

2.3 带内遥测场景

带内网络遥测最初面向可编程网络提出并制定 INT 标准,后随其优势凸显而拓展至多元应用场景。例如, POINT^[13] 针对数据包光网络设计意图驱动遥测架构,通过跨层次映射查询意图及响应,解决光网

络层次结构与设备不可编程性带来的 INT 部署挑战。 ML-INT^[9] 面向 IP-over-Optical 网络,基于 P4 的遥测架构分模块采集光信道与 IP 层遥测参数,并通过流采样和数据包采样来降低遥测开销。此外, Karaagac 等人^[11] 首次将 INT 扩展到工业无线传感器网络 6TiSCH,采用机会捎带机制以满足自组织和分布式的遥测需求; Haxhibeqiri 等人^[12] 针对 IEEE 802.11 无线网络,设计统一的 INT 节点架构并与无线 SDN 框架深度融合,支撑网络重配置与验证。针对实际部署中部分设备不支持遥测的异质网络, Zaballa 等人^[14] 提出适用于 NGS 和 MPLS 混合网络的 INT 方案,保障了其过渡部署的可行性。

2.4 超算网络测量与管理

超算与数据中心场景下的网络测量与管理,是当前高性能网络领域的研究热点。例如, LightGuardian^[28] 实现了低开销、高鲁棒性的流级信息采集与异常定位; INAM^[29] 专为 MPI (Message Passing Interface) 应用设计,提供全面的跨栈通信性能分析; DTA^[30] 支持基于 RDMA (Remote Direct Memory Access) 的高速遥测数据聚合,显著提升了数据收集效率; STREAM^[31] 是可扩展的 HPC 遥测平台,能够整合多系统数据,为运维与科研提供实时、可靠的系统信息; OmniWindow^[32] 突破了可编程交换机的资源限制,支持高效、灵活的网络遥测框架; RD-Probe^[33] 通过黑盒监控技术,能够构建复杂网络环境下的实际网络拓扑; ByteTracker^[34] 是首个无代理探测系统,实现了对大规模数据中心网络的准确的路径跟踪。这些研究工作为超算及数据中心系统的网络测量、运维管理与性能优化提供了重要的理论基础与技术支撑。 UFM^[35] 遥测功能支持对网络带宽、拥塞情况、错误信息及延迟数据的追踪,提供多属性监测、基于端口的带宽数据等网络状态信息,并将其流式传输至指定的数据库,以供深度分析。

综上所述,与上述相关工作相比,本文研究工作的创新之处体现于:在带内遥测场景方面,针对超算系统高速互连网络场景展开研究,拓展了带内遥测的应用场景;在带内遥测架构方面,根据超算互连网络低直径、无精确全局时钟同步等特点,提出轻量级的带内网络遥测架构;在带内遥测调度算法方面,以解决最小路径遥测覆盖问题为目标,设计启发式遥测任务调度算法。与 IntOpt^[24] 的 SARG 算法相比,本文方法针对超算互连网络的大规模全链路监控需求,简化了问题模型,采用启发式算法,以实现更少

的遥测任务为目标。而 SARG 算法则聚焦于通用虚拟网络的监控流优化,其目标是通过最小化封装和转发开销来实现效率提升。该算法考虑了遥测频率一致性和监控流容量阈值等通用网络约束,因而更适合用于通用虚拟网络的精细化开销优化;相比文献[27]提出的 ACGS 方法,本文方法完全基于数据流路径进行遥测,无需配置探测路径,从而避免了将基于欧拉回路的遥测应用于超算互连网络时可能引发的路由死锁问题。此外,ACGS 方法试图解决的探测路径过长导致探测报文长度超过网络 MTU 限制的问题,在低时延低网络跳步数的超算互连网络中并不存在。

3 轻量级带内网络遥测架构

3.1 系统总体架构

为了适应多种应用场景需求,INT 规范 2.1^[3]定义了三种工作模式,分别为 INT-XD(eXport Data)、INT-MX(eMbed instructions)和 INT-MD(eMbed Data)。其中,INT-XD 的每个 INT 节点收到数据报文后根据流量监控列表(WatchList)收集元数据并直接发送到遥测服务器;INT-MX 通过在数据报文中嵌入 INT 指令,使每个 INT 节点根据 INT 指令收集 INT 元数据,并将其发送到监控系统;INT-MD 将 INT 指令和 INT 元数据均嵌入到数据报文中, Sink 节点在接收到遥测报文时,将全部遥测信息转发到遥测服务器。相比而言,INT-MD 的遥测效率最高且易于实现,因此本文采用类似 INT-MD 模式,设计了一种轻量级遥测系统架构,并将其命名为 LW-INT 架构。

LW-INT 总体架构如图 1 所示。在该架构中,每个 NRC 和 NIC 芯片的端口模块中集成了内建遥测单元(Built-In Telemetry Unit, BITU),其功能是:发送和接收各类带内网络遥测报文,并在报文中携带网络的局部互连信息和本地计时信息,从而实现从源节点和目的节点之间路由路径和逐跳步时延性能的测量。运行在遥测服务器(INT-SVR)上的遥测调度器根据上层应用的遥测需求生成单个或多个遥测任务,遥测任务通过与遥测主机接口的交互完成带内遥测过程。中小规模的超算系统通常只需部署单个遥测服务器即可满足所有上层应用的遥测需求。对于超大规模超算系统(主机节点数目达到 10^4 量级)的互连网络,可以部署多个遥测服务器共同处理不同的并发遥测任务。

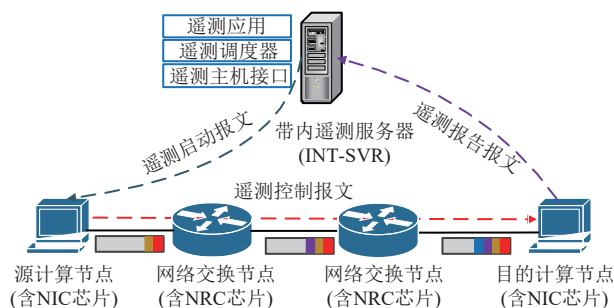


图1 LW-INT的总体架构

单个遥测任务仅仅完成对指定源节点到指定目的节点之间单向路径的状态测量,即遥测调度器通知源节点产生遥测报文并启动单次带内遥测过程。遥测报文在从源节点到目的节点的传输过程中,随路收集网络路径信息和逐跳步的时延信息。目的节点获得遥测报文后报告给遥测调度器。具体而言,遥测调度器通过网卡上的 NIC 芯片下发遥测描述符, NIC 将遥测描述符转换为遥测启动报文(INT_STA),该报文通过网络数据平面到达源节点 NIC 芯片并进入其内建遥测单元。该单元根据遥测启动报文(INT_STA)内容产生遥测控制报文(INT_CTL)。该报文通过网络数据平面逐跳步路由达到目的节点 NIC 芯片,并进入其内建遥测单元。该单元产生遥测报告报文(INT_RPT),该报文通过网络数据平面返回到遥测服务器,并上传给遥测调度器。遥测控制报文(INT_CTL)在从源节点逐跳步传输到目的节点的过程中,每次进入网络端口和从网络端口发送时,网络端口内的内建遥测单元将路径和时延信息添加在报文中。遥测调度器通过解析和处理遥测报告报文(INT_RPT)中的遥测信息,获得从源节点到目的节点的端端路径和逐跳步时延信息,供给上层遥测应用使用。

3.2 遥测主机接口

在 LW-INT 架构中,运行在遥测服务器的遥测调度器通过遥测主机接口启动带内网络遥测过程并接收遥测信息。遥测主机接口如图 2 所示,其处理过程包括:(1)遥测描述符提交(如图中序号 1~5 所示):遥测调度器创建的虚端口(Virtual Port, VP)在主存中分配专用的描述符发送队列(SDQ),用于该 VP 向 NIC 芯片提交用于构造遥测启动报文的描述符,描述符以固定长度为单位进行读写操作。VP 向 SDQ 中写入 1 个描述符后,写指针就加“1”。NIC 芯片为 VP 设置了一个计数器 SDQ_NUM,用于记录该 SDQ 中未被预取的描述符的个数。遥测调度器将描述符写入 SDQ 后,再向 NIC 芯片中对应的

SDQ_NUM 写入描述符的个数,从而触发遥测过程。NIC 芯片每次读取 1 个描述符,就将对应的 SDQ_NUM 减“1”。NIC 芯片为 VP 设置了一个读指针 SDQ_RP,用于记录硬件下一次准备从该 SDQ 读取描述符的物理地址。(2)遥测报告报文接收(如图中序号 7~8 所示):遥测调度器为所用 VP 在主存中分配独享的报文接收队列(MPQ),用于接收 NIC 芯片发送的遥测报告报文。NIC 芯片为该 VP 设置了一个计数器 MPQ_NUM,用于记录该 VP 的 MPQ 中未处理的 MP 消息个数。NIC 芯片向 MPQ 中写入 1 个遥测报告报文,则 MPQ_NUM 加“1”;软件从 MPQ 中读取 1 个遥测报告报文,则向 MPQ_NUM 执行 1 次 PIO 写操作,使得 MPQ_NUM 减“1”。为了快速处理报文接收队列中的遥测报告报文,可以将报文接收队列非空定义为一种中断类型,并通过 SMI 中断机制通知 CPU 处理。图中序号 6 表示报文在互连网络中的传输过程。

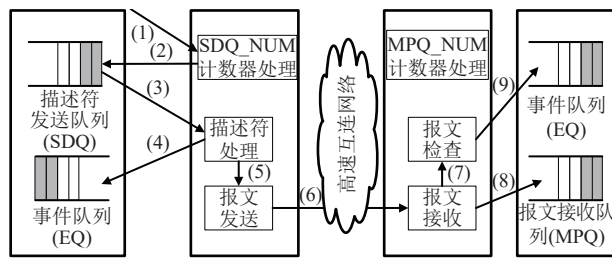


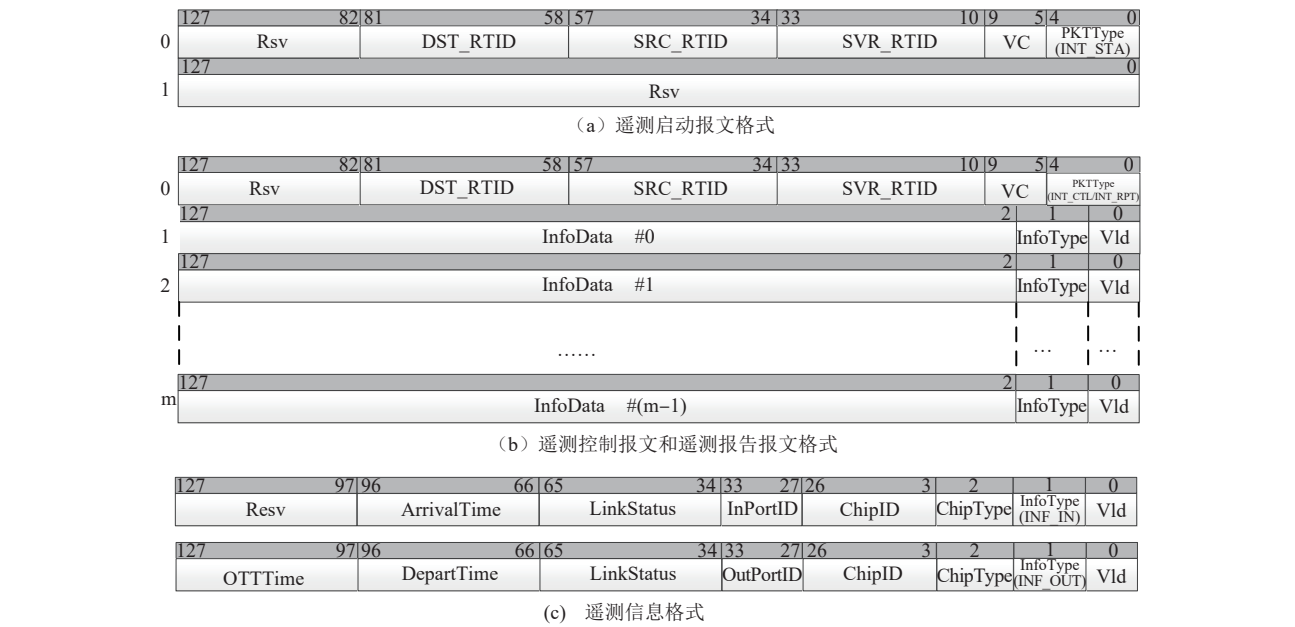
图2 带内网络遥测主机接口

3.3 遥测报文格式

如 3.1 节所述,遥测过程需要定义三种遥测报文,分别是遥测启动报文(INT_STA)、遥测控制报文(INT_CTL)和遥测报告报文(INT_RPT)。不同于传统带内网络管理报文在管理平面内路由,这三种报文都在数据平面内传输。其中,INT_STA 报文和 INT_RPT 报文分别用于启动遥测过程和接收遥测结果,但它们不对各自传输路径上的状态进行测量,INT_CTL 报文负责对源节点和目的节点之间的单向数据传输路径的状态进行测量。此外,为了支持分段延迟度量(详见 3.5 节)过程中精确测量往返时间(Round-Trip Time, RTT),需要定义另外两种本地时延测量报文,分别是 RTT 测量请求报文和 RTT 测量响应报文。

带内网络遥测报文格式如图 3 所示,各个位域的含义说明如下:(1)PKTType:报文类型,有效的类型包括 INT_STA、INT_CTL、INT_RPT、RTT_REQ、RTT_ACK。(2)VC:虚通道编号,互连

网络中不同类型的报文通常通过不同的虚通道传输,其位宽取决于芯片内设计的虚通道数目,通常可设置为 3 位或 4 位。(3)SVR_RTID:遥测服务器 NIC 芯片的路由标识,是 INT_RPT 报文从目的节点路由到遥测服务器 NIC 芯片的依据,通常设置为 20 位。(4)SRC_RTID:源节点 NIC 芯片的路由标识,是 INT_STA 报文从遥测服务器路由到源节点 NIC 芯片的依据,与 SVR_RTID 位宽相同。(5)DST_RTID:目的节点 NIC 芯片的路由标识,是 INT_CTL 报文从源节点路由到目的节点 NIC 芯片的依据,与 SVR_RTID 位宽相同。(6)Vld:遥测信息有效位。所有遥测报文采用固定长度,INT_CTL 和 INT_RPT 报文可携带多组遥测信息($\langle \text{InfoType}, \text{InfoData} \rangle$),每组遥测信息用该信息域表示其是否有效。当源节点 NIC 芯片产生 INT_CTL 报文时,报文中的 m 个有效位均初始化为 0。INT_CTL 报文在从源节点向目的节点逐跳步路由时,报文中每增加一条带内遥测信息,则该信息对应的有效位更新设置为 1。(7)InfoType:遥测信息类型,有效的类型包括:输入端遥测信息(INF_IN)、输出端遥测信息(INF_OUT)。(8)InfoData:遥测数据。遥测信息类型不同,则带内遥测数据格式不同。其中输入端遥测信息格式为 $\langle \text{ChipType}, \text{ChipID}, \text{InPortID}, \text{LinkStatus}, \text{ArrivalTime} \rangle$,输出端遥测信息格式为 $\langle \text{ChipType}, \text{ChipID}, \text{OutPortID}, \text{LinkStatus}, \text{DepartTime}, \text{OTTTime} \rangle$ 。(9)ChipType:芯片类型,用来区分 NIC 芯片和 NRC 芯片。(10)ChipID:芯片管理编号,用来标记互连网络中不同的网络芯片。考虑到所能支持的互连网络系统规模,可为芯片编号设置 20 位芯片编号。(11)InPortID, OutPortID:分别是输入端口编号和输出端口编号,用来记录遥测控制报文通过 NRC 芯片所经过的输入和输出端口。假设 NRC 芯片共 N 个端口,则设置 $\log_2 N$ 取上整数位端口编号信息,各个端口分别编号为 0、1、 $\dots$ 、 $N-1$ 。(12)ArrivalTime, DepartTime, OTTTime:分别是到达时间、离开时间和单向传输时延,其中 ArrivalTime 用来记录遥测控制报文到达 NRC 芯片带内遥测模块的本地计时器值,DepartTime 用来记录遥测控制报文离开 NRC 芯片带内遥测模块的本地计时器值,OTTTime 用来记录 NRC 芯片内建遥测单元与邻接芯片内建遥测单元之间的报文单向传输时延,该值是报文往返时间测量值(RTT)的一半。(13)LinkStatus:网络端口链路状态信息,通常



是通道锁定位图(BlockLockBitmap)、报文重传计数(RetryCount)和链路速率(LinkRate)等。

在上述报文格式定义中,假设了互连网络的数据平面采用了分布式查表路由,即在报文传输的每一跳步,NRC根据报文中携带的路由标识查表确定该报文的输出口。

3.4 内建遥测单元

互连网络芯片的逻辑层次包括物理层(PHY)、链路层(LINK)、事务层(TRANSACTION)、核心层(CORE)等,其中物理层、链路层、事务层构成端口逻辑,NIC芯片通常包含较少数量(≤ 4)的端口逻辑,NRC芯片包含较多数量(≥ 64)的端口逻辑。BITU嵌入在每个端口逻辑的事务层与链路层之间。在数据发送方向,BITU模块接收事务层发送逻辑的数据处理后发送给链路层发送逻辑;在数据接收方向,BITU模块接收链路层接收逻辑的数据处理后发送给事务层接收逻辑。

BITU模块实现的功能主要包括遥测报文的生成、捕获、处理和发送,相邻端口链路层之间RTT时间的测量,NRC交换核心层处理时间的测量等。BITU模块实现的内部结构如图4所示。

在数据接收方向,BITU模块从接收数据流中识别出各种遥测报文,并根据报文类型进行相应的处理:(1)如果是INT_STA报文,而且SRC_RTID等于ChipID,则表示本芯片就是遥测源节点,为此需要捕获该报文并触发INT_CTL报文生成。(2)如果是INT_STA报文,但是SRC_RTID不等于

ChipID,则表示本芯片只是从遥测服务器传输INT_STA报文到遥测源节点的中转节点,则将其传输给事务层接收模块进行后续处理。(3)如果是INT_CTL报文,而且DST_RTID等于ChipID,则表示本芯片就是遥测目的节点,为此需要捕获该报文并触发INT_RPT报文生成。(4)如果是INT_CTL报文,但是DST_RTID不等于ChipID,则表示本芯片只是从遥测源节点传输INT_CTL报文到遥测目的节点的中转节点,则更新本报文,即在报文中增加遥测信息<INF_IN, ChipID, PortID, LinkStatus, ArrivalTime>。其中ArrivalTime来自计时器,PortID和ChipID来自配置寄存器。(6)如果是RTT_REQ报文,则需要捕获该报文并触发RTT_ACK报文生成。(7)如果是RTT_ACK报文,则捕获该报文并触发OTT统计器记录RTT_ACK报文半值返回时间。(8)对于其他类型数据报文,BITU模块直接将其传输给事务层接收模块进行后续处理。

在数据发送方向,BITU模块不但需要完成上述INT_CTL报文生成、INT_RPT报文生成、INT_CTL报文更新、RTT_ACK报文生成过程新报文的调度发送,而且需要处理两类报文:(1)周期性触发RTT_REQ报文生成并发送,即从计时器获取本地时钟周期计数信息,根据该信息周期性触发产生RTT_REQ报文,发送给报文调度器,同时触发OTT统计器记录RTT_REQ报文发送时间。(2)更新从事务层接收到的INT_CTL报文并发送,

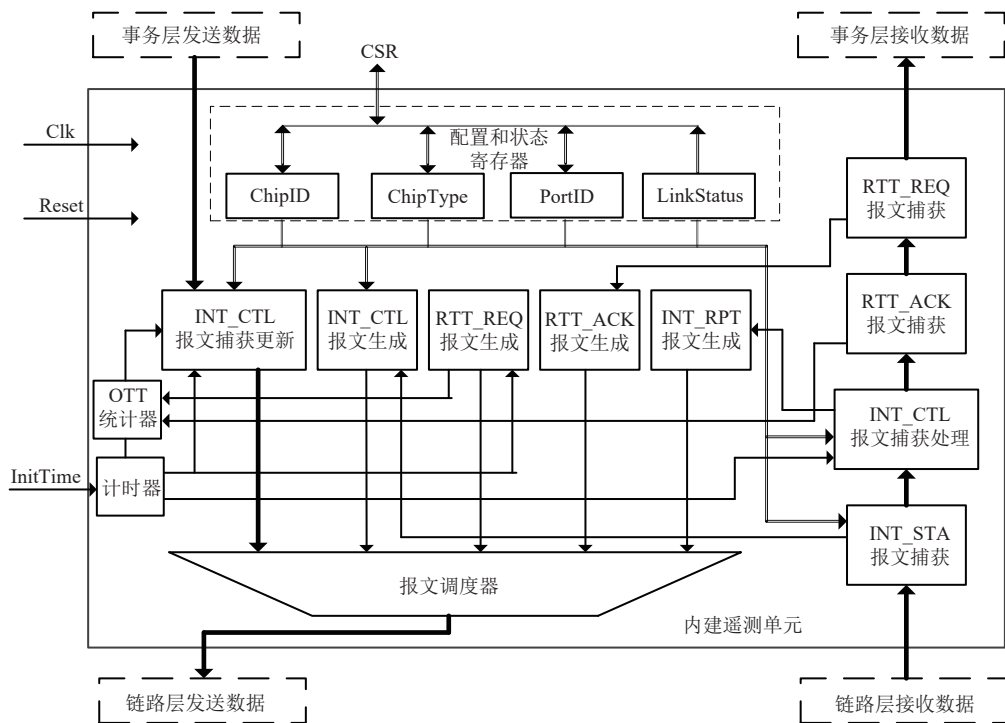


图4 内建遥测单元(BITU)内部结构

即从发送数据流中识别出INT_CTL报文,则在报文中增加遥测信息,分别是<INF_OUT, ChipID, PortID, LinkStatus, DepartTime, OTTTime>。其中DepartTime和OTTTime分别来自计时器和OTT统计器。

3.5 分段延迟度量

由BITU模块的工作过程可知,在INT_CTL报文随路测量网络逐跳步时延过程中,当报文进入NRC芯片端口BITU模块时,为报文填入到达时间(ArrivalTime)信息;当报文离开NRC芯片端口BITU模块时,为报文填入离开时间(DepartTime)信息。ArrivalTime和DepartTime都是芯片内部计时器的瞬时值。为了易于实现,芯片内部为每个端口设置一个计时器,计时器在相同时钟频率(同源时钟)驱动下做累加操作。如果所有计时器的初始值都设置为0,则由于芯片各个端口复位差异和时钟树偏差(Skew)导致各个计时器的值在相同时刻可能存在偏差,所以需要为各个计时器调校合适的初始值,从而保证在相同的时刻所有端口计时器的计数值偏差不超过一个时钟周期。

如图5所示,如果将报文从进入NRC芯片端口BITU模块经过数据交换到离开本芯片端口BITU模块的时间定义为交换时延(SwitchLatency),再假设芯片本地时钟周期的频率为freq,则SwitchLatency的统计方法如式(1)所示。

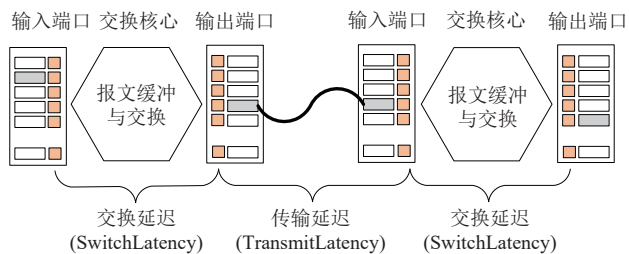


图5 分段延迟度量方法示意

$$\text{SwitchLatency} = (\text{DepartTime} - \text{ArrivalTime}) / \text{freq} \quad (1)$$

此外,由BITU模块工作过程可知,当报文离开NRC芯片端口BITU时,为报文填入单向传输时延(OTTTime)信息。OTTTime来自OTT统计器,该OTT统计器保存着从发送RTT_REQ报文到接收RTT_ACK报文的本地时钟周期数的一半,而且随着RTT_REQ和RTT_ACK报文的周期性发送与接收,OTT统计器中的值会周期性更新。尽管OTT统计器的值不能完全等于遥测报文的传输时延,但是由于影响传输时延的因素主要是物理介质性质和长度等,因此可以认为同一端口间不同时刻报文的传输时延是基本相同的。如图5所示,如果将报文从离开芯片端口BITU模块经过物理链路传输到下一跳网络芯片端口BITU模块的时间定义为传输时延(TransmitLatency),则TransmitLatency的计算方法如式(2)所示。

$$\text{TransmitLatency} = \text{OTTTime} / \text{freq} \quad (2)$$

互连网络的单跳步转发时延(OneHopLatency)等于该跳步的交换时延和传输时延之和。所以,根据遥测报文记录的信息,OneHopLatency的计算方法如式(3)所示。

$$\text{OneHopLatency} = [(\text{DepartTime} - \text{ArrivalTime}) + \text{OTTTime}] / \text{freq} \quad (3)$$

上述分段延迟遥测方法,采用了轻量级的本地时钟测量措施,避免了为遥测报文标记全局时钟戳而带来的设计复杂性。单次遥测任务能够准确测量互连网络的端到端逐跳步网络时延。与其他时延测量方案相比,本文提出的分段延迟度量方法实现代价较低。

3.6 带内遥测示例

如图6所示,遥测服务器INT-SVR运行遥测调度器,对从计算节点Cn26到Cn92的数据平面路径和时延情况进行单次带内遥测。

其基本过程为:遥测服务器INT-SVR上的带内遥测调度器启动单次遥测任务,向源计算节点Cn26发送单个遥测启动报文INT_STA。源计算节点NIC芯片(NIC26)端口Pa的BITU模块接收到INT_STA报文后,根据报文携带的信息产生遥测控

制报文INT_CTL,然后从端口Pa发送出去。产生的INT_CTL报文携带了<INF_OUT, NIC, NIC26, Pa, LS(NIC26. Pa), DT(NIC26), OTT(NIC26, NRC5)>信息。从Cn26计算节点NIC26芯片端口Pa输出的INT_CTL报文经过链路到达NRC5芯片的端口Pb,端口Pb内的BITU模块更新了INT_CTL报文,增加携带了<INF_IN, NRC, NRC5, Pb, LS(NRC5. Pb), AT(NRC5)>信息。从NRC5芯片端口Pb输入的INT_CTL报文被NRC5芯片交换到端口Pc输出,端口Pc内的BITU模块更新的INT_CTL报文增加携带<INF_OUT, NRC, NRC5, Pc, LS(NRC5. Pc), DT(NRC5), OTT(NRC5, NRC9)>信息。INT_CTL报文在逐跳步经过NRC9和NRC7输入端口和输出端口时,端口内的BITU模块不断在INT_CTL报文中增加了INF_IN和INF_OUT遥测信息。从NRC7芯片端口Pg输出的INT_CTL报文经过链路到达Cn92计算节点NIC92芯片端口Pa,端口Pa内的BITU模块产生的遥测报告报文INT_RPT携带INT_CTL报文包含的所有遥测信息,然后从Pa端口发送出去。从目的计算节点Cn92发送的INT_RPT报文,到达遥测服务器INT-SVR并最终上送给遥测调度器进行分析处理。

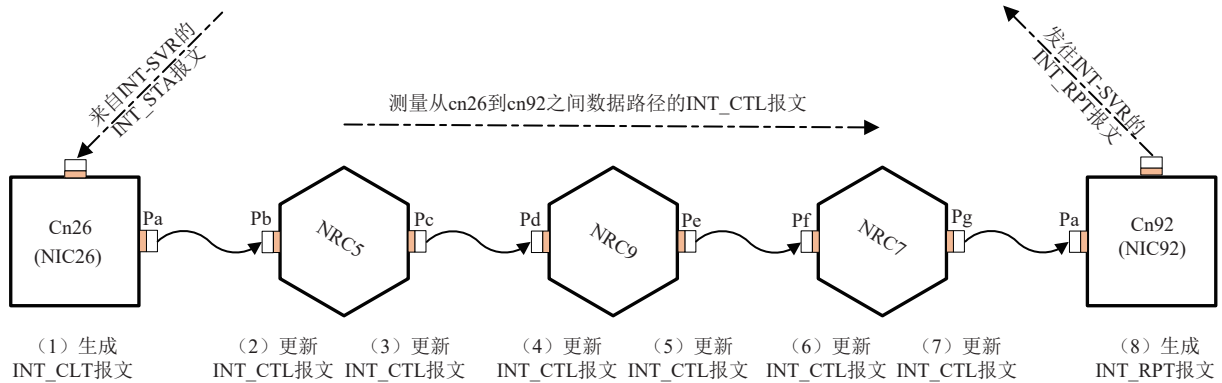


图6 带内网络遥测示例场景

通过上述带内遥测过程,运行在遥测服务器INT-SVR上的遥测调度器可获取的带内遥测信息如表1所示。从表中信息可以看出,从计算节点Cn26到Cn92之间的数据传输路径为: NIC26. Pa→NRC5. Pb→NRC5. Pc→NRC9. Pd→NRC9. Pe→NRC7. Pf→NRC7. Pg→NIC92. Pa;其逐跳步时延为: OTT(NIC26, NRC5)→ST(NRC5)→OTT(NRC5, NRC9)→ST(NRC9)→OTT(NRC9, NRC7)→ST(NRC7)→OTT(NRC7, NIC92),且路由路径上各个网络端口的链路状态为LS(NIC26.

Pa), LS(NRC5. Pb)等。其中,ST(Z)表示遥测路径上Z芯片的交换时延,具体是指从Z芯片输入端口BITU模块到Z芯片输出端口BITU模块之间的交换延迟,即[DT(Z)-AT(Z)]/freq。LS(Z. Px)表示遥测路径上Z芯片Px端口的链路状态。

3.7 架构特点分析

目前,支持带内遥测功能的网络主要包括RoCE[36]和IB[35]等RDMA网络,这些网络已经广泛应用于数据中心、超智中心、智算中心等场景。IB网络借助其上层UFM管理软件实现了带内遥

表1 示例场景中带内网络遥测获取的遥测信息

#	生成的带内遥测信息						
1	< INF_OUT,	NIC,	NIC26,	Pa,	LS(NIC26. Pa),	DT(NIC26),	OTT(NIC26, NRC5) >
2	< INF_IN,	NRC,	NRC5,	Pb,	LS(NRC5. Pb),	AT(NRC5) >	
3	< INF_OUT,	NRC,	NRC5,	Pc,	LS(NRC5. Pc),	DT(NRC5),	OTT(NRC5, NRC9) >
4	< INF_IN,	NRC,	NRC9,	Pd,	LS(NRC9. Pd),	AT(NRC9) >	
5	< INF_OUT,	NRC,	NRC9,	Pe,	LS(NRC9. Pe),	DT(NRC9),	OTT(NRC9, NRC7) >
6	< INF_IN,	NRC,	NRC7,	Pf,	LS(NRC7. Pf),	AT(NRC7) >	
7	< INF_OUT,	NRC,	NRC7,	Pg,	LS(NRC7. Pg),	DT(NRC7),	OTT(NRC7, NIC92) >
8	< INF_IN,	NIC,	NIC92,	Pa,	LS(NIC92. Pa),	AT(NIC92) >	

测,但其硬件实现细节尚未公开。与RoCE网络相比,本文提出的LW-INT架构具有如下特点:

(1)在实现复杂度上,LW-INT架构具有显著轻量化优势。RoCE网络遥测依赖IEEE 1588全局时钟同步,还需部署时间校准设备,实现技术复杂且硬件部署成本高。而LW-INT架构无需本地时钟同步,通过分段延迟度量方法,结合本地计时与RTT半值估算即可实现逐跳步时延测量,无需额外协议开销,实现难度大幅降低。

(2)在超算场景适配性方面,LW-INT架构针对性解决了HPC互连网络的核心需求。RoCE交换机带内遥测面向以太网设计,未考虑超算互连网络中虚通道(VC)的差异化传输特性,无法精准捕捉不同VC的时延差异。LW-INT架构支持VC差异化测量,可精准匹配超算互连网络的传输特性。同时支持多遥测服务器分布式部署,可适配超大规模HPC集群需求。

(3)在兼容性方面,LW-INT架构的适用范围更广。RoCE交换机采用的遥测技术仅适配于以太网环境,LW-INT架构所采用的随路测量方法及分段延迟度量方法具有普适性,无需绑定特定协议,可无缝适配超算系统高速互连网络,也可应用于其他无需依赖IEEE 1588全局时钟同步的网络场景,例如工业无线传感器网络(6TiSCH^[11])、低功耗广域网(LPWAN)等。

4 带内网络遥测调度算法

传统的带内网络管理技术^[15]以网络芯片为测量对象,管理服务器注入互连网络的单个管理报文所携带的指令只能访问单个网络芯片的配置或状态等信息。新型带内网络遥测技术以报文传输路径为测量对象,遥测服务器注入互连网络的单个遥测报文所携带的指令可以访问从源节点到目的节点的

路径上多个网络芯片的随路配置和状态信息。在本文所提出的LW-INT架构中,从源节点到目的节点路径上各个跳步的ChipID, PortID, ArrivalTime, OTTTime, DepartTime 等信息成为单次遥测任务的测量目标。

4.1 问题描述

实现对网络系统内所有链路状态的监控是带内遥测基础应用之一。因此,如何调度次数最少的遥测任务实现对互连网络系统内所有测量对象的覆盖,就成为带内网络遥测调度算法需要解决的重要问题。本文将其简称为最小路径遥测覆盖问题(Minimum Path Coverage Problem for INT, MPCP-T),该问题分步定义如下:

定义 1. 互连网络. 将互连网络建模为包含 S 个交换节点和 P 个主机节点的图 $G(V, E)$ 。节点集合 V 划分为 VH (主机节点集合)和 VS (交换节点集合)两个子集,且 $V=VH \cup VS$ 。 VH 中的每个主机节点都只与 VS 中的一个交换节点相连。 VS 中的每个交换节点可与 VH 中的一个或多个主机节点相连,也可与 VS 中的其他一个或多个交换节点相连。节点之间相连是指节点间存在无向边 $e \in E$, 两个节点之间可以存在多条边。

在实际互连网络结构中,主机节点是通过NIC芯片接入到互连网络的,NIC芯片通常只有一个或少数几个网络端口,NRC芯片通常包含许多个网络端口,而且由于互连网络结构中可能存在着冗余链路,因此上述互连网络模型中允许两个节点之间存在着多条边相连。

定义 2. 端口和链路. 将主机节点与交换节点,以及交换节点之间相连的连接点称为网络端口。假设节点 v 有 k 个网络端口,则节点 v 的网络端口集合表示为 $P(v)=\{v.p_0, v.p_1, \cdots, v.p_{k-1}\}$ 。假设节点 u 与 v 相连,且连接的网络端口分别为 p_m 和 p_n ,则将该条连接节点 u 与 v 的边称为链路,且表示为 $r=$

$(u.p_m, v.p_n)$, 其中 m 和 n 为端口编号。

定义 3. 路由路径. 互连网络的路由 R 集合由所有有序节点对 (u, v) 之间的 $P \cdot (P-1)$ 条路径组成, 其中 $(u, v) \in VH \times VH$. 从节点 u 到 v 的路由路径 $r(u, v)$ 可表示为 $t+1$ 条链路的向量 $r(u, v) = (r_0, r_1, \dots, r_t)$, 其中 $r_0 = (u.p_a, d_0.p_b)$, $r_1 = (d_0.p_c, d_1.p_d)$, $\dots$, $r_t = (d_{t-1}.p_e, v.p_f)$, t 表示路由路径的长度, $d_0, d_1, \dots, d_t$ 均为图 G 的节点, $r_0, r_1, \dots, r_t$ 均为图 G 的边, 且 $p_a \in P(u)$, $p_b \in P(d_0)$, $p_c \in P(d_0)$, $p_d \in P(d_1)$, $\dots$, $p_e \in P(d_t)$, $p_f \in P(v)$.

可见, 虽然图 G 中的边是没有方向性的, 但是路由路径是具有方向性的。通常称路由路径的起始节点为源节点, 路由路径的终止节点为目的节点。路由路径的源节点和目的节点只能是主机节点。

定义 4. 链路遥测任务. 本文所定义的互连网络的遥测任务 $\text{DoINT}(u, v)$ 是指对从源节点 u 到目的节点 v 之间的路由路径 $r(u, v)$ 进行随路测量, 从而获得路由路径 $r(u, v)$ 上各个节点的被路由经过的网络端口链路状态信息 (LinkStatus), 网络端口连接关系和传输时延 (TransmitLatency) 等信息。

定义 5. 最小路径遥测覆盖问题 MPCP-T. 本文所定义的 MPCP-T 问题是指对于给定的互连网络图 $G(V, E)$ 和路由集合 R (含所有主机节点对间的路由路径), 寻找元素数量最少的遥测任务集合 T , 使得 E 中所有被路由路径覆盖的链路 r , 均被 T 中至少一个遥测任务的遥测路径覆盖。

MPCP-T 问题的提出是为了以最小的遥测代价 (本文以遥测任务量作为衡量遥测代价的指标) 对互连网络中所有路由经过的链路进行带内遥测。尽管对互连网络中的 $P \cdot (P-1)$ 条路由路径全部进行带内遥测, 可以收集到数据平面内任意一条路由路径经过的所有链路的状态, 但是这并不是代价最小的遥测方案。例如, 在图 7 所示的由 3 个主机节点和 1 个交换节点构成的互连网络中, 路由路径共包括 6 条, 分别是 $r(A, B)$ 、 $r(B, C)$ 、 $r(C, A)$ 、 $r(A, C)$ 、 $r(C, B)$ 和 $r(B, A)$, 但只需要进行 3 次遥测任务就可以覆盖所有链路, 遥测任务分别是 $\text{DoINT}(A, B)$ 、 $\text{DoINT}(B, C)$ 和 $\text{DoINT}(C, A)$ 。随着互连网络规模的增大, 如果对网络所有路由路径全部进行周期性带内遥测, 这无论在时间开销还是带宽开销方面都是难以承受的。

MPCP-T 问题可被建模为一个整数线性规划 (Integer Linear Programming, ILP) 问题, 描述如式 (4) 所示:

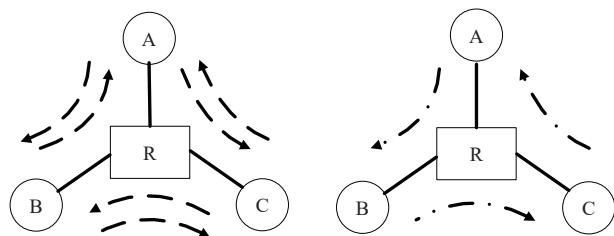


图 7 路由路径和最小遥测链路覆盖示意

$$\begin{aligned} & \text{Minimize } \sum_{u \in VP} \sum_{u \in VP, u \neq v} x_{uv} \\ & s.t. \sum_{(u, v) \in VP \times VP, l \in r(u, v)} x_{uv} \geq 1 \forall l \in E \quad (4) \\ & x_{uv} \in \{0, 1\} \forall u, v \in VP, u \neq v \end{aligned}$$

其中, x_{uv} 是一个二进制变量, 如果选择遥测任务 $\text{DoINT}(u, v)$ 进行遥测, 则其取值为 1; 否则为 0。目标函数是最小化选择的遥测任务数量。约束条件为如果链路被任何路由路径覆盖, 则它必须被至少一个选定的遥测任务覆盖。

本文选择遥测任务量作为核心代价指标, 原因如下: 遥测任务量直接决定了遥测报文的封装、解封、发送、接收的次数与传输路径覆盖范围, 任务量越大, 越容易抢占业务流带宽、增加遥测服务器 CPU 工作负载、增加交换芯片转发压力, 进而影响超算核心计算任务的通信性能; 同时, 遥测任务量是直观可量化的指标, 无需复杂资源建模即可精准统计, 能直接导向面向超算网络“少任务、全覆盖”的优化目标; 此外, 超算互连网络具有较低的最大跳步数, 无需考虑遥测报文容量不足及 MTU 限制等问题, 单个遥测任务足以覆盖网络中任何端到端的路由路径。因此, 以任务量作为遥测代价的衡量指标具有合理性与场景适配性。

4.2 A2A-CB 算法

为了求解 MPCP-T 问题, 本文首先提出了一种启发式求解算法 A2A-CB。该算法的基本思想: 首先对互连网络中所有路由路径 (All to All, A2A) 进行遥测, 然后尝试消减 (CutBack, CB) 长度较短的路由路径, 直到没有路由路径可以消减。那么, 剩余的所有路由路径即为选定的带内遥测任务路径。

A2A-CB 算法的伪代码描述如算法 1 所示。首先分别遍历所有源主机节点 u 和目的主机节点 v (Line3~Line4), 对所有 (Src, Dest) 节点对进行遥测 (Line6), 并记录遥测任务 (Line5)、每条路由路径长度 (Line7) 和经过每个网络端口的路由路径 (Line8)。然后, 按照长度对所有路由路径进行排序 (Line10), 优先选择长度最短的路由路径尝试消减 (Line11~Line22)。尝试消减方法是: 如果删除该

路径后,该路径所经过的所有网络端口仍然有其他路由路径经过,则删除该路由路径(Line21)。其中,Link.PassedBy用来记录经过链路Link的所有源和目的节点对的集合;TrySet用来暂时记录Link.PassedBy消减当前源和目的节点对的集合;BC_enable是一个布尔变量,用来记录是否消减当前源和目的节点对。同时,本文将没有尝试消减遥测任务的基础算法(Line3-Line5)称之为A2A算法。

算法1. A2A-CB 算法伪代码

输入:Network, $G=(VH \cup VS, E)$
输出:INT Task Set, T

1. $T \leftarrow \emptyset$;
2. Foreach Link in G ,
3. Link.PassedBy $\leftarrow \emptyset$;
4. Foreach $u \in VH$
5. Foreach $v \in VH$ and $v \neq u$
6. $T \leftarrow T \cup \{(u, v)\}$
7. DoINT(u, v) //执行带内网络遥测相关操作
8. $(u, v).PathLen \leftarrow |r(u, v)|$
9. Foreach Link in $r(u, v)$
10. Link.PassedBy $\leftarrow$ Link.PassedBy $\cup \{(u, v)\}$
11. Sort T by PathLen in Increasing order
12. Foreach $(u, v) \in T$ in order
13. BC_enable $\leftarrow True$
14. Foreach Link in $r(u, v)$
15. TrySet = Link.PassedBy - $\{(u, v)\}$
16. If TrySet == $\emptyset$ then
17. BC_enable $\leftarrow False$
18. Break
19. If BC_enable == True then
20. Foreach Link in $r(u, v)$ do
21. Link.PassedBy $\leftarrow$ Link.PassedBy - $\{(u, v)\}$
22. $T \leftarrow T - \{(u, v)\}$
23. Return T

由以上过程可见,A2A-CB算法通过首先生成所有可能的任务,再按路径长度排序并优先选择长度短的路径进行筛选,有效移除那些路径上所有链路都被其他任务遥测的任务,从而优化任务集合。

4.3 A2A-EP算法

为了求解MPCP-T问题,本文提出另外一种启发式求解算法A2A-EP。该算法的基本思想:首先通过遥测互连网络中所有路由路径(A2A),获得所

有路由路径长度信息。然后尝试从空集开始不断扩展(ExPend,EP)形成最终的遥测任务集合,其集合扩展方法是:迭代选择包含未覆盖链路数目最多的路由路径对应的遥测任务加入集合,同时更新计算当前所有未纳入集合的路由路径的未覆盖链路数目。当剩余的所有路由路径的未覆盖链路数目均为“0”时,则遥测任务集合构建完成。

算法2. A2A-EP算法伪代码

输入:Network, $G=(VH \cup VS, E)$
输出:INT Task Set, T

1. $T \leftarrow \emptyset$;
2. FlowSet $\leftarrow \{(u, v) | u \in VH, v \in VH, u \neq v\}$
3. Foreach $u \in VH$
4. Foreach $v \in VH$ and $v \neq u$
5. DoINT(u, v) //执行带内网络遥测相关操作
6. Set(u, v).PathLen $\leftarrow |r(u, v)|$
7. while FlowSet $\neq \emptyset$
8. Choose(u, v) in FlowSet with the largest LinkCnt
9. if LinkCnt $\neq 0$
10. $T \leftarrow T \cup \{(u, v)\}$
11. Flowset $\leftarrow$ Flowset - $\{(u, v)\}$
12. Foreach(o, p) in FlowSet
13. $(o, p).LinkCnt = (o, p).LinkCnt - |r(u, v) \cap r(o, p)|$
14. else if LinkCnt == 0
15. FlowSet $\leftarrow$ FlowSet - $\{(u, v)\}$
16. Return T

A2A-EP算法的伪代码描述如算法2所示。算法的开始部分(Line2~Line5)与算法A2A-CB类似,但不需要将遥测任务集合 T 初始化为所有可能路径。然后,优先选择未覆盖链路最多的路由路径尝试扩展到遥测任务集合(Line7~Line14),每扩展一个遥测任务,需要更新所有未被纳入集合 T 的路由路径的未覆盖链路数目(Line11~Line12)。如果路由路径的未覆盖链路数目更新后为“0”,则从未被纳入遥测任务集合的路径集合中删除该路径(Line13~14)。其中,FlowSet用来记录未被纳入遥测任务集合的路由路径,其初始值为网络拓扑中的所有路由路径;FlowSet中的所有路由路径使用一个LinkCnt变量记录未覆盖链路的数目,其初始值为路由路径的长度;每当有路由路径 $r(u, v)$ 被扩展到遥测任务集合时,FlowSet集合中所有路由路径的该数值都需要减去本路由路径与 $r(u, v)$ 的公共链路的数目。

由以上过程可见,A2A-EP算法侧重于根据路由路径的未遥测链路数量来动态调整 and 选择遥测任务,通过不断更新路由路径的未遥测链路计数值来决定遥测任务选择的优先级,从而构造优化的任务集合。

4.4 计算复杂度分析

本节将首先证明 MPCP-T 问题为 NP 完全问题,然后分析 A2A-CB 和 A2A-EP 算法的计算复杂度。

4.4.1 MPCP-T 问题难度

该小节将证明 MPCP-T 问题是 NP 完全问题。沿用本小节之前的符号表示,假设给定的互连网络图 $G(V, E)$, V 为节点集合, E 为链路集合, R 为路由集合, T 为最小规模遥测任务集合, P 为互连网络中主机节点数目;并假设图 $G(V, E)$ 的主机节点集合 VH 中任意两节点间路由路径包含的链路数目平均值为 D 。

引理 1. MPCP-T 问题属于 NP 类问题

证明:对于任意给定的候选遥测任务集合 T' , 可以通过以下步骤验证是否为 MPCP-T 问题的可行解。第一步,遍历 E 中所有被路由覆盖的链路,检查每条链路是否至少被 T' 中一条遥测路径包含。第二步,确认 T' 中每个遥测任务均对应 R 中的一条有效路由路径。在第一步中,遍历链路的时间复杂度为 $O(|E|)$,检查单条链路是否被覆盖的时间复杂度为 $O(|T'| \cdot D)$,因此总的复杂度为 $O(|E| \cdot |T'| \cdot D)$ 。第二步的时间复杂度为 $O(|T'|)$ 。所以,可以在多项式时间内验证 T' 是否为 MPCP-T 的可行解,即 MPCP-T 属于 NP 类问题。证毕。

引理 2. 任何 SCP 问题可在多项式时间内规约为 MPCP-T 问题

证明:集合覆盖问题(Set Covering Problem, SCP)是经典的 NP 完全问题^[37],其定义为:给定元素集合 $U = \{u_1, u_2, \dots, u_n\}$ 和 U 的子集族 $S = \{S_1, S_2, \dots, S_m\}$,寻找最小规模的子集族 $S' \subseteq S$,使得 $\bigcup_{S_j \in S'} S_j = U$ 。通过以下映射可将 SCP 实例 (U, S) 转化为 MPCP-T 实例 (G, R) :(1)将集合元素映射为链路。将 SCP 中的每个元素 $u_i \in U$ 映射为 MPCP-T 中的一条链路 $e_i \in E$,即 $E = \{e_1, e_2, \dots, e_n\}$,其中 $e_i \leftrightarrow u_i$ 。(2)将集合的子集映射为路由路径。将 SCP 中的每个子集 $S_j \in S$ 映射为 MPCP-T 中的一条路由路径 $r_j \in R$,使得 r_j 恰好包含与 S_j 中元素对应的所有链路。即若 $S_j = \{u_1, u_2, \dots, u_k\}$,则 $r_j = \{e_1, e_2, \dots, e_k\}$ 。(3)主机节点构造。设置两个主机节点 s 和 t ,所有路由路径 r_j 均定义为从 s 到 t 的单向路径,确保每

条路径独立对应一个子集。

下面说明 SCP 实例存在可行解 S' 当且仅当 MPCP-T 实例存在可行解 T' ,且两者规模一致。(1)必要性:若 S' 是 SCP 的可行解($\bigcup_{S_j \in S'} S_j = U$),则将 S' 中所有子集对应的路由路径构成遥测任务集合 T' 。由于 S' 覆盖所有元素 U ,则 T' 覆盖所有链路 E ,即 T' 是 MPCP-T 的可行解。(2)充分性:若 T' 是 MPCP-T 的可行解(覆盖所有链路 E),则将 T' 中所有遥测路径对应的子集构成 S' 。由于 T' 覆盖所有链路 E ,则 S' 覆盖所有元素 U ,即 S' 是 SCP 的可行解。(3) $|S'| = |T'|$,即 SCP 解的最小规模与 MPCP-T 解的最小规模完全等价。上述映射过程中,集合元素与链路、子集与路由路径的映射均为一一映射,构造时间为 $O(|U| + |S| \cdot k)$ (k 为 SCP 子集平均大小),属于多项式时间。证毕。

定理 1. MPCP-T 是 NP 完全问题

证明:根据引理 1 可知,MPCP 问题属于 NP 类问题。根据引理 2 可知,存在从 NP 完全问题 SCP 到 MPCP-T 的多项式时间规约,根据 NP 完全问题的定义,MPCP-T 是 NP 完全问题。

证毕。

此外,由于 MPCP-T 问题存在至少一个可行解(即 A2A 基础算法给出的遥测任务集合就是一个可行解),且其可行解的数量是有限的(遥测任务集合最大数量为 $2^{P(P-1)}$),而有限非空集合中必然存在最小值元素,所以 MPCP-T 问题一定存在最优解。但是,由于该问题是 NP 完全问题,在大规模网络场景下无法通过精确算法在多项式时间内求解,因此需要启发式算法以较低时间复杂度获得高质量的近似解。

4.4.2 A2A-CB 计算复杂度

A2A-CB 算法首先初始化数据结构,时间复杂度为 $O(|E|)$ 。随后,对每对顶点 u 和 v (共 $|VH|^2$ 对)计算路径并更新链路信息,该过程的时间复杂度为 $O(|VH|^2 \cdot D)$ 。接着,算法对路径按长度排序,复杂度为 $O(|VH|^2 \log |VH|^2)$ 。最后,算法逐个检查路径是否满足条件并移除不满足的路径,复杂度为 $O(|VH|^2 \cdot D)$ 。综合而言,A2A-CB 算法的总时间复杂度为 $O(|VH|^2 \cdot D)$ 。

4.4.3 A2A-EP 计算复杂度

A2A-EP 的计算复杂度主要由两部分组成,分别是路径计算和主循环中的更新操作。路径计算部分的复杂度为 $O(|VH|^2 \cdot D)$,主循环中的更新操作复

杂度为 $O(|VH|^2 \cdot D \cdot \log|VH|^2)$ 。综合而言,整个算法的计算复杂度为 $O(|VH|^2 \cdot D \cdot \log|VH|^2)$ 。

可见,相比于A2A-CB算法,A2A-EP算法的计算复杂度较高。

5 实验评估

为了系统性地评估上述带内遥测架构及调度算法的性能和有效性,本文设计并实施了三个层次的实验。首先,基于FPGA平台对LW-INT遥测架构进行了硬件实现,着重评估了其核心功能模块BITU的硬件资源占用情况,并对分段延迟度量机制的可行性进行了初步验证。其次,通过构造基于Python的实验环境,对A2A-CB/EP算法在遥测任务处理效率、时间复杂度等方面的性能指标进行了深入分析。最后,借助Mininet网络仿真平台,进一步分析评估了本文提出的带内遥测机制在网络资源占用(包括带宽消耗和CPU占用)以及分段延迟度量等关键性能和功能指标上的表现。

5.1 LW-INT遥测架构实验

基于FPGA平台,本节首先评估带内遥测架构核心单元BITU的资源消耗,然后验证分段延迟度量中传输延迟的可行性。

5.1.1 BITU资源消耗

基于3.4节所述的内建遥测单元BITU原理,本文采用Verilog硬件描述语言完成了BITU的功能模块设计。为验证BITU的轻量级特性,本文选择OpenCore平台^[38]提供的开源IEEE 1588模块作为基准模块。在FPGA测试环境中完成逻辑综合与实现后,对BITU和开源IEEE 1588模块的硬件资源使用情况进行了统计,具体结果如表2所示。

表2 BITU和IEEE 1588在FPGA上实现的资源使用情况

资源类型	BITU	IEEE 1588
CLB LUTs	546	1442
CLB Registers	1332	2038
CLB	424	343
LUT as Logic	546	1298
LUT as Memory	0	144
LUT Flip Flop Paires	202	551

资源评估结果显示,BITU在硬件资源使用上显著优于开源IEEE 1588模块。具体而言,BITU在CLB LUTs、CLB Registers、LUT as Logic和LUT Flip Flop Paires等关键资源上的占用均大幅低

于IEEE 1588模块,分别节省了62%、35%、58%和63%。同时,BITU未使用存储器(LUT as Memory)资源,进一步体现了其轻量级设计的优势。实验结果充分验证了本文提出的LW-INT是一种轻量级的遥测架构,其核心功能模块BITU具有硬件逻辑资源占用少且易于实现等优点。

5.1.2 分段延迟度量

根据3.5节分段延迟度量方法可知,网络延迟主要由两种类型延迟构成,分别是交换延迟(SwitchLatency)和传输延迟(TransmitLatency)。受限于在FPGA平台上仅实现了BITU模块、网络端口的链路层MAC和物理层PCS等模块,只能对两个被测端口通过光纤对接的网络单跳步的传输延迟进行测量。在该FPGA平台上,MAC、BITU、PCS逻辑的工作频率为80 MHz,网络端口速率设置为20 Gbps(4×5 Gbps)。实验进行了五次,每次均选取不同长度的光纤连接两个被测端口,待链路状态稳定后记录下OTT统计器的值(即OTTTime)。传输延迟测量及统计结果具体如表3所示。

表3 在不同光纤长度下传输延迟测量情况

光纤长度 (m)	光传输延迟 (ns)	OTTTime (周期数)	测量延迟 (ns)	延迟差值 (ns)
5	25	42	525	500
10	50	45	562.5	512.5
15	75	45	562.5	487.5
20	100	50	625	525
25	125	51	637.5	512.5

传输延迟测量结果显示,随着光纤长度的增加,通过BITU模块测量到的OTTTime周期数及其换算成的测量延迟也随之增加。根据光信号在光纤中的传输速率为 2×10^8 米/秒,可计算出光信号每传输1米所需时间为5纳秒。测量延迟减去光信号传输延迟后得到的延迟差值,主要包含网络端口物理层处理延迟及测量误差。假设网络端口物理层处理延迟为500纳秒,则测量误差的最大值为2个时钟周期,即25纳秒。由此可知,在未来ASIC实现中,BITU的工作频率和网络端口速率远高于FPGA实现,其传输延迟的测量误差将显著降低。该实验结果表明,本文提出的LW-INT架构是一种轻量级的遥测架构,其采用的分段延迟度量方法在测量传输延迟方面具有可行性和有效性。

5.2 A2A-CB/EP调度算法实验

为了分析评估A2A-CB/EP算法的性能,本文针

对超算系统三种典型网络拓扑结构(Fat-Tree^[39-40]、Torus^[41]、Dragonfly^[42])开展了实验。对于每种拓扑结构,不同的参数取值将会产生不同的拓扑实例。对于每个拓扑实例,本文对比了A2A-CB/EP算法和基准算法的性能。由于A2A-CB/EP算法是要近似解决MPCP-T问题,即寻找较少的遥测任务以覆盖网络的所有路由路径,因此本文将同一拓扑实例下的遥测任务数量作为各种不同遥测调度算法性能对比的指标。本文使用Python语言实现了所述实验涉及的遥测调度算法等,其中网络拓扑结构模型的创建和路由路径的计算等过程使用了Python语言的NetworkX库^[43]。

实验在一台商用服务器上进行,其基本配置为: Intel(R) Xeon(R) Gold 6126 CPU @ 2.60 GHz 处理器、528GB 内存、CentOS Linux 7 (Core)操作系统、Python 2.7.5 版本、NetworkX 2.2 版本。

5.2.1 基准算法

在性能评估实验中,本文选择了两种基准算法用于性能对比,简单描述如下:

(1)A2A 基础算法:如 4.2 节所述,该算法是 A2A-CB 算法去掉尝试消减遥测任务部分的基础算法。假设 P 表示主机节点数目,则 A2A 基础算法产生的遥测任务量为 $P \times (P-1)$ 。

(2)INTO-CH 算法:该算法是文献[21]提出的用于解决 INTO 网络遥测调度的集中优化(INTO Concentrate Optimization)问题的启发式算法。其最坏情况下的计算复杂度为 $O(n^2 \cdot m + n \cdot m^2)$,其中 n 为网络总端口数目, m 为网络中总的路由路径数目。

INTO 问题约束了每个遥测流的容量,即限制了可分配给流的收集和传输的遥测项数量。由于在本文所提出的 LW-INT 轻量级带内网络遥测架构中,设计的单个遥测项可控制在 12 个字节内,因此可假设遥测报文可以收集和传输实际网络中任意节点对间的遥测信息。为了确保性能对比的公平性,设定 INTO-CH 算法的流容量是无限的。

此外,由上文分析可知,INTO-CH 算法的计算复杂度显著高于 A2A-CB 和 A2A-EP 算法。

(3)SARG 算法:该算法是文献[24]提出的一种结合随机贪心策略与模拟退火机制的启发式算法,核心目标是网络功能虚拟化(Network Function Virtualization, NFV)服务功能链(Service Function Chain, SFC)生成最优监控流(Monitoring Flow, MF)集合,在满足所有链路监控需求的同时最小化遥测开销。该算法的复杂度为 $O(X \cdot L^2)$,其中 L 为被

SFC 覆盖的物理链路数, X 为模拟退火的迭代次数。

SARG 算法专为 NFV 网络切片监控设计,可为多租户的服务流生成最优监控流方案。为保证 SARG 的性能,对 SARG 算法核心参数设定如下:模拟退火的初始温度为 10^5 ,冷却率为 10^{-5} ,终止温度为 1。为了对比公平起见,还对 SARG 算法进行了如下设置:(1)将 MF 大小阈值(λ')设置为 $|E|$,避免因 MF 大小拆分任务;(2)将所有链路的频率约束(μ_{ij})统一设置为任意相同值,避免因频率差异额外拆分 MF;(3)将所有链路的遥测项约束(δ_{ij})统一设置为所有 SF 遥测项的并集,避免遥测项约束影响 MF 构建;(4)将链路物理链路容量约束(γ_{ij})设置为主机节点数 P ,避免因链路容量拒绝承载 MF。此外,将 SARG 算法运行的实际退出条件调整为:达到模拟退火终止条件或当链路覆盖率达到 100% 时,以便于与本文提出的算法进行性能比较。

5.2.2 基于 Fat-Tree 拓扑的对比

胖树(Fat-Tree)拓扑结构具有高等分带宽和低网络直径等优点,因此广泛应用于超算系统,例如部署在国家超算天津中心“(NSCC-TJ)”的“天河一号(TH-1A)^[44]”和国家超算广州中心(NSCC-GZ)的“天河二号(TH-2)^[45]”均采用胖树结构。本实验中的胖树拓扑描述采用 m -port- n -tree 模型($FT(m, n)$)^[39-40],其数据平面采用 OSRM 路由算法^[39]。

对于 $FT(m, n)$ 模型,交换节点和主机节点的网络端口数目分别为 m 和 1。 $FT(m, n)$ 可以支持 $m \times (m/2)^{n-1}$ 个主机节点互连,共使用 $(2n-1) \times (m/2)^{n-1}$ 个交换节点,需遥测 $n \times m^n / 2^{n-1}$ 条物理链路。表 4 所示为在六种参数取值下的 $FT(m, n)$ 拓扑结构情况。

表 4 在六种参数取值下的 $FT(m, n)$ 拓扑结构

拓扑实例 名称	主机节点数目 (个)	交换节点数目 (个)	链路数目 (条)
$FT(16, 2)$	128	24	256
$FT(24, 2)$	288	36	576
$FT(32, 2)$	512	48	1024
$FT(8, 3)$	128	80	384
$FT(16, 3)$	1024	320	3072
$FT(20, 3)$	2000	500	6000

本实验构建了六种类型 $FT(m, n)$ 模型的网络结构并实现了 OSRM 路由算法^[39]。分别以表 4 所述六种拓扑结构实例为输入,分别运行 A2A-CB/EP 和 INTO-CH 算法获得各自算法产生的遥测任务量,如图 8 所示。

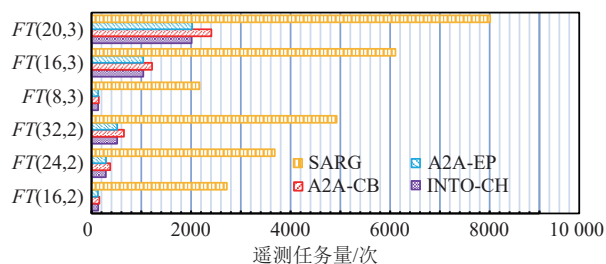


图8 六种Fat-Tree拓扑中不同算法产生遥测任务量比较

图8实验数据表明:(1)在上述六种 $FT(m,n)$ 拓朴实例中,A2A-EP算法与INTO-CH基准算法产生的任务量高度接近,其相对差值(即绝对差值占INTO-CH基准算法任务量的比值)最大约为1.04%。值得注意的是,在 $FT(16,2)$ 和 $FT(16,3)$ 两种拓朴中,A2A-EP算法分别优于INTO-CH基准算法2.99%和0.19%。(2)与INTO-CH基准算法相比,A2A-CB算法产生的遥测任务量的相对差值范围为9.77%~30.2%,平均值约为20.03%。(3)相较于A2A基础算法,A2A-CB/EP算法的遥测任务量实现了数量级的显著降低。例如,在 $P=2000$ 的 $FT(20,3)$ 拓朴中,A2A基础算法任务量为3998000(图中未显示),而A2A-EP算法和A2A-CB算法的任务量分别为2020和2480,分别缩减至A2A基础算法的0.05%和0.06%。(4)与SARG基准算法相比,A2A-EP和A2A-CB算法产生的遥测任务量显著更低。例如,在 $FT(16,2)$ 中,SARG算法的任务量为2719,而A2A-EP和A2A-CB的任务量分别为130和157,SARG算法的任务量分别是A2A-EP和A2A-CB算法的20.9倍和17.3倍,凸显了A2A系列算法在任务量优化上的显著优势。

概括而言,A2A-EP算法与INTO-CH基准算法产生的遥测任务量数目约为 $O(P)$;A2A-CB算法比INTO-CH基准算法计算出的遥测任务量平均高出约20.03%,比SARG算法计算出的遥测任务量低出约85.69%;A2A基础算法构建的遥测任务量数目约为 $O(P^2)$ 。

5.2.3 基于Torus拓扑的对比

超算系统发展的初期,以低维Torus为代表的 k -ary- n -cube是互连网络的主流拓扑。随着硬件条件的发展成熟,高维Tours结构已经被许多超算系统使用。例如Fujitsu公司推出的K computer采用的Tofu互连网络是6D Torus结构^[41],该公司推出的Fugaku超算系统采用的TofuD互连网络沿用了Torus拓朴^[46]。本实验中的Torus拓朴采用 $Torus(k_1, k_2, \dots, k_n)$ 描述,其中 n 为总维度, k_i 为第 i 个维度上

的节点数。其数据平面的路由采用维序路由DoR(Dimension order Routing)算法^[47]。

不同于间接网络的Fat-Tree和Dragonfly拓扑,Torus通常设计为直接网络,主机节点与交换节点实现紧耦合,主机节点无需通过单端口网络接口卡,而直接连接网络交换机。 $Torus(k_1, k_2, \dots, k_n)$ 拓朴结构可以支持 $k_1 \times k_2 \times \dots \times k_n$ 个主机节点互连,共使用 $k_1 \times k_2 \times \dots \times k_n$ 个交换节点,需遥测 $(k_1 \times k_2 \times \dots \times k_n) \times (n/2)$ 条物理链路。表5所示为在六种参数取值下的 $Torus(k_1, k_2, \dots, k_n)$ 拓朴结构情况。

表5 在六种参数取值下的 $Torus(k_1, k_2, \dots, k_n)$ 拓朴结构

拓朴实例 名称	主机节点数目 (个)	交换节点数目 (个)	链路数目 (条)
$Torus(12, 12)$	144	144	144
$Torus(16, 16)$	256	256	256
$Torus(8, 8, 8)$	512	512	768
$Torus(7, 6, 5, 4)$	840	840	1680
$Torus(6, 5, 5, 3, 3)$	1350	1350	3375
$Torus(6, 5, 5, 3, 3)$	2160	2160	6480

本实验构建了六种类型 $Torus(k_1, k_2, \dots, k_n)$ 模型的网络结构并实现了DoR算法^[48]。实验方法与5.2节一致。图9统计了表5所述六种拓朴结构实例下,三种算法的遥测任务量对比结果。

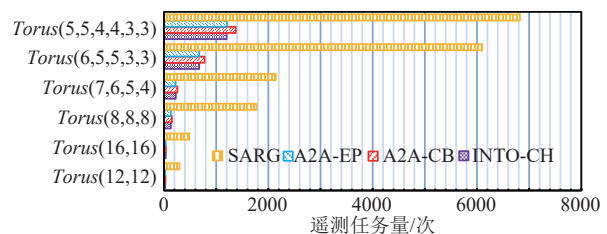


图9 六种Torus拓朴中不同算法产生遥测任务量比较

从图9实验数据可见:(1)在Torus拓朴中,A2A-EP算法与INTO-CH基准算法的性能一致性更为突出。具体而言,在 $Torus(12, 12)$ 、 $Torus(16, 16)$ 、 $Torus(8, 8, 8)$ 和 $Torus(6, 5, 5, 3, 3, 3)$ 四种拓朴中,两者产生的遥测任务量完全一致;仅在其余两种拓朴中存在微小差异,且平均相对差值不足1.29%。(2)A2A-CB算法的遥测任务量高于INTO-CH基准算法,其相对差值范围为12.50%~16.67%,平均值约为14.57%。(3)与A2A基础算法相比,A2A-CB/EP算法的遥测任务量实现了数量级的显著降低。例如,在 $P=2160$ 的 $Torus(5, 4, 4, 3, 3, 3)$ 拓朴中,A2A基础算法任务量为4663440

(图中未显示),而 A2A-EP 算法和 A2A-CB 算法的任务量分别为 1211 和 1376,均缩减至 A2A 基础算法的约 0.03%。(4)与 SARG 基准算法相比,A2A-CB/EP 算法的遥测任务量实现了显著降低。例如,在 $Torus(5,5,4,4,3,3)$ 拓扑中,A2A-CB 算法和 A2A-EP 算法的任务量分别比 SARG 算法降低了约 79.56% 和 82.33%;(5)值得注意的是,Torus 拓扑的显著特征是除了 SARG 外的三种算法产生的遥测任务量均小于对应主机节点数(例如 $P=2160$ 时,最大遥测任务量为 $1376<2160$),这一特征与 Fat-Tree 和 Dragonfly 两种间接网络形成了明显差异。

5.2.4 基于 Dragonfly 拓扑的对比

Dragonfly 拓扑^[42]突破了 k -ary- n -cube 拓扑中的维的概念,用层次化的方法进行互连,是大规模低直径互连网络拓扑结构的典型代表。Cray 公司设计的基于 Cascade 系统^[48]的 XC30 超计算机采用的就是 Dragonfly 拓扑结构。本实验中的 Dragonfly 拓扑描述采用 $DF(p,a,g,h)$ 描述,其中 p 表示每个交换机连接的主机数, a 表示每个全互连组(Group)内的交换机数目, g 表示组的数目, h 表示每个路由器用于连接到其他组的全局通道数。其数据平面采用基础的最小路由(Minimal Routing)算法^[42]。

对于 $DF(p,a,g,h)$ 模型,交换节点和主机节点的网络端口数目分别为 $p+(a-1)+h$ 和 1。 $DF(p,a,g,h)$ 可以支持 $p \times a \times g$ 个主机节点互连,共使用 $a \times g$ 个交换节点,需遥测 $[a \times g \times (a+2p+h-1)]/2$ 条物理链路。表 6 所示为在六种参数取值下的 $DF(p,a,g,h)$ 拓扑结构情况。

表 6 在六种参数取值下的 $DF(p,a,g,h)$ 拓扑结构

拓扑实例 名称	主机节点数目 (个)	交换节点数目 (个)	链路数目 (条)
$DF(8,2,5,4)$	200	25	300
$DF(8,5,9,4)$	360	45	540
$DF(8,8,8,7)$	512	64	960
$DF(12,9,9,4)$	972	81	1458
$DF(16,9,9,8)$	1296	81	1944
$DF(16,8,17,8)$	2176	136	3196

本实验构建了六种类型 $DF(p,a,g,h)$ 模型的网络结构并实现了最小路由算法^[42]。实验方法与 5.2 节一致。图 10 数据统计了表 6 所述六种拓扑结构实例下,三种算法的遥测任务量对比结果。

从图 10 实验数据可以看出:(1)在上述六种 Dragonfly 拓扑实例中,A2A-EP 算法与 INTO-CH

基准算法的任务量高度一致,其相对差值(即绝对差值占 INTO-CH 基准算法任务量的比值)最大约为 1.39%。值得注意的是,在 $DF(12,9,9,4)$ 和 $DF(16,9,9,8)$ 两种拓扑中,A2A-EP 算法的任务量甚至低于 INTO-CH 基准算法,分别低出 0.20% 和 0.08%,这表明 A2A-EP 算法在 Dragonfly 拓扑中的高效性。(2)与 INTO-CH 基准算法相比,A2A-CB 算法产生的遥测任务量显著增加,其相对差值范围为 25.50%~46.32%,平均值约为 35.73%。这一结果表明,Dragonfly 层次化拓扑结构对 A2A-CB 算法的性能产生了较大影响,其任务量增长幅度高于 Fat-Tree 和 Torus 拓扑场景。(3)与 A2A 基础算法相比,A2A-CB/EP 算法的遥测任务量也实现了数量级降低。例如,在 $P=2176$ 的 $DF(16,8,17,8)$ 拓扑中,A2A 基础算法任务量为 4,732,800(图中未显示),而 A2A-EP 算法和 A2A-CB 算法的任务量分别为 2189 和 2855,分别缩减至 A2A 基础算法的 0.05% 和 0.06%。(4)与 SARG 基准算法相比,A2A-CB/EP 算法的遥测任务量依然实现了显著降低。

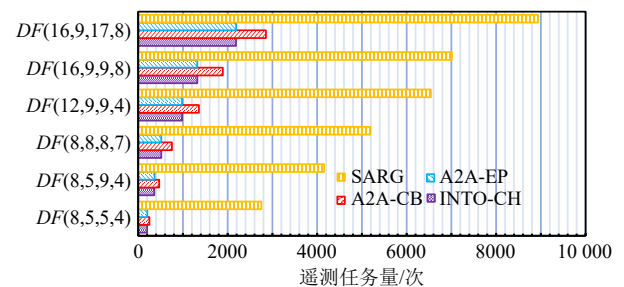


图 10 六种 Dragonfly 拓扑中不同算法产生遥测任务量比较

可见,在六种 Dragonfly 拓扑中,A2A-CB 算法与 INTO-CH 基准算法的任务量相对差值的平均值为 35.73%,显著高于 Fat-Tree 拓扑中的 20.03% 和 Torus 拓扑中的 14.57%。这与 Dragonfly 拓扑层次化结构特征相关。

5.2.5 各种拓扑下的综合比较

为了进一步分析 A2A-CB/EP 算法的拓扑结构敏感性,本文选取上述各种拓扑结构中支持主机节点数目均为 512 的拓扑实例的实验测试数据进行了统计,数据可视化如图 11 所示。

由图 11 可见,即使网络的规模(最大支持的主机节点数目)相同,不同网络拓扑结构下,遥测调度算法产生的遥测任务量也存在差异。首先,A2A-EP 算法在性能上优于 A2A-CB 算法,与 INTO-CH 基准算法相当,显著优于 SARG 算法,展现出较优

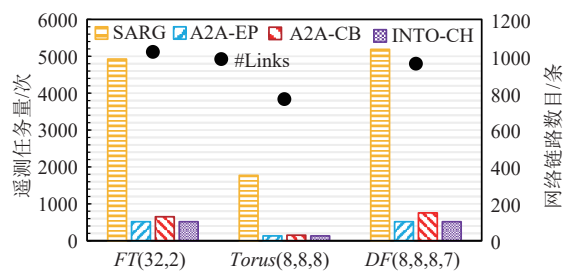


图 11 三种相同规模拓扑中不同算法产生遥测任务量比较

的性能。此外，在 $FT(32,2)$ 和 $DF(12,5,9,8)$ 拓扑结构中，各种算法产生的遥测任务量基本相同，但远高于算法在 $Torus(8,8,8)$ 拓扑结构中产生的遥测任务量。这不但与拓扑类型（直接网络或间接网络）相关，而且与拓扑所包含的链路数目相关。

表 7 在六种拓扑结构中各种算法运行完成时间和内存占用情况

拓扑实例名称	主机节点数目(个)	算法完成时间(秒)				峰值内存占用(MB)			
		A2A-EP	A2A-CB	INTO-CH	SARG	A2A-EP	A2A-CB	INTO-CH	SARG
$FT(16,2)$	128	16.1	2.1	38.2	52.9	45.2	55.5	55.8	54.7
$Torus(12,12)$	144	20.5	4.7	40.1	60.5	60.7	65.9	71.1	65.1
$DF(8,2,5,4)$	200	19.9	4.5	38.6	71.2	40.6	42	43.8	42.4
$FT(32,2)$	512	299.6	84.6	472.6	921.1	333.6	367.4	480.2	368.9
$Torus(8,8,8)$	512	316.2	82.9	453.1	816.2	346.5	451.2	490.1	394.9
$DF(12,5,9,8)$	512	313.6	93.8	429.6	813.6	263.7	354.3	409.9	322.4

由表 7 数据可见，本文提出的 A2A-EP 和 A2A-CB 算法在完成时间上优于现有的 INTO-CH 和 SARG 算法。具体而言，在相同拓扑结构下，A2A-CB 算法的完成时间为 A2A-EP 的 1/8 到 1/4，且远低于 INTO-CH 和 SARG 算法。例如，在小规模的 $FT(16,2)$ 拓扑中，A2A-CB 的完成时间为 2.1 秒，而 INTO-CH 和 SARG 的完成时间分别为 38.2 秒和 52.9 秒，A2A-CB 的效率提升 17 倍以上。类似地，在较大规模的拓扑（如 $FT(32,2)$ 和 $Torus(8,8,8)$ ）中，A2A-CB 的完成时间虽有所增加，但仍显著低于其他算法，且完成时间的增幅远小于 INTO-CH 和 SARG 算法。

在内存占用方面，A2A-EP 在所有拓扑实例中均表现最优，峰值内存占用显著低于 A2A-CB、INTO-CH 和 SARG 算法。A2A-CB 的表现优于 INTO-CH，但在某些情况下略高于 SARG。整体而言，A2A-EP 在内存占用方面具有显著优势，适用于不同规模和类型的拓扑结构；INTO-CH 的表现最差，峰值内存占用在各种情况下均最高。此外，随着拓扑规模的增加，内存占用整体呈上升趋势，但不同

此外，为了比较各种算法实际运行的时间和消耗内存开销，本文采用以下步骤获取算法的开销数据：每次执行各种算法时，首先利用 psutil (Python System and Process Utilities) 获取进程初始占用的内存，并使用 time 标准库记录算法执行起始时间；然后在算法开始执行时启动线程实时监测并更新峰值内存；最后在算法执行完毕时停止监测线程，并返回算法的执行时间和峰值内存等信息。表 7 统计了在六种拓扑结构中各种算法运行完成时间和峰值内存占用的情况。实验结果表明，在相同的拓扑结构下，同一算法多次运行时，其完成时间与峰值内存占用量表现出高度的一致性和稳定性。鉴于此，未进一步呈现各算法完成时间和内存占用的极值和均值。

算法的增长速率存在差异。A2A-EP 的增长速率最低，INTO-CH 的增长速率最快。

综合上述遥测任务量、算法完成时间、峰值内存占用等方面的性能对比实验结果，可见本文提出的 A2A-EP 和 A2A-CB 算法在遥测任务调度中展现出显著优势。A2A-EP 峰值内存占用显著低于其他算法；A2A-CB 在完成时间上表现最为优异，运行效率远高于 INTO-CH 和 SARG 算法。两种算法在多种拓扑结构中均表现出高效性和适应性，适用于实际网络中的遥测任务调度场景。

5.3 遥测仿真实验

本实验在一台商用服务器上进行，其硬件配置与 5.2 节所述服务器硬件配置相同。在服务器上使用 Mininet 网络仿真平台^[49]模拟主机和 P4 可编程交换机，并搭建了如图 12 所示的 $FT(4,3)$ 三层胖树拓扑，图中 □ 表示 P4 交换机节点 s，○ 表示主机节点 h。其中，P4 交换机的数据平面处理逻辑使用 P4₁₆ 标准定义的语法和语义格式进行编写，仿真实现了 BITU 模块的功能。数据平面逻辑运行在开源的软件交换机模型 BMv2 (Behavioral Model version2)^[50]

上。在定义拓扑脚本中,对每条链路随机指定1~5毫秒的传输延迟,并通过tc为每个交换机端口指定3~9毫秒的延迟规则。

另外,实验选择主机节点 $h(0,0,0)$ 作为遥测服务器(INT-SVR),负责遥测任务的周期性发起和遥测数据的收集。实验设置遥测周期为10秒,即每隔

10秒遍历执行遥测任务集合中的所有遥测任务。实验首先使用本文提出的A2A-EP算法为该拓扑构造遥测任务集合。经过运行该算法的程序,构造的遥测任务集合共包含16个遥测任务,例如 $h(0,0,0) \rightarrow h(1,0,0)$, $h(0,0,1) \rightarrow h(1,0,1)$, $h(0,0,1) \rightarrow h(0,0,0)$ 等。经过检查,其链路接口覆盖率达到100%。

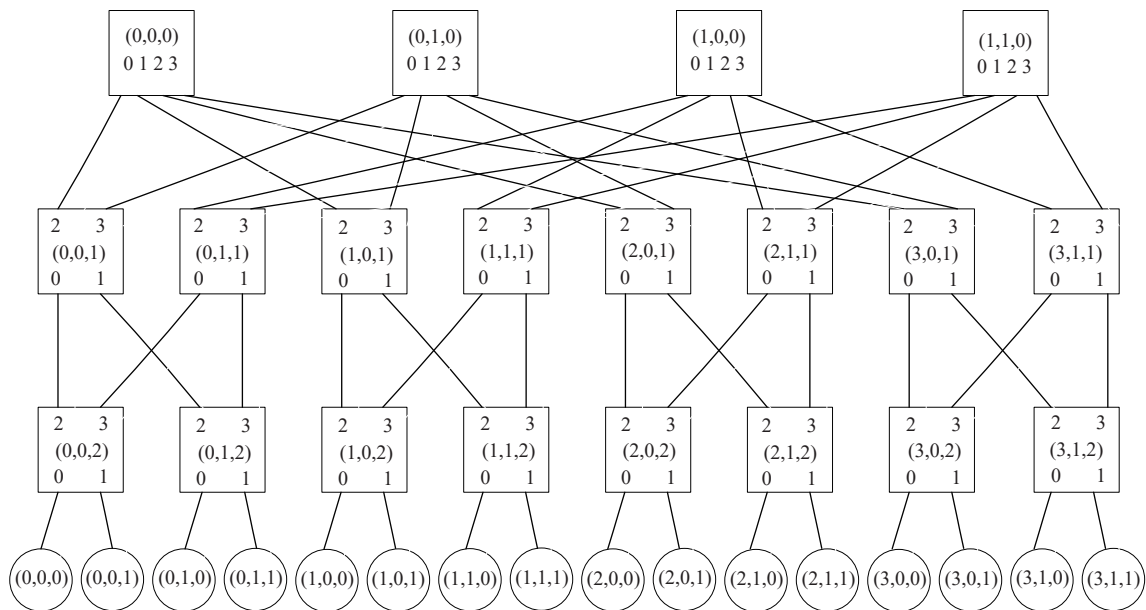


图12 实验使用的FT(4,3)拓扑结构

5.3.1 额外带宽消耗

根据本文3.3节所定义的遥测报文格式,设定支持的网络最大跳步数为10跳步(即遥测控制报文和遥测报告报文均包含20组遥测信息),并考虑到遥测报文封装到链路层帧后所额外增加的链路层开销,遥测启动报文(INT_STA)、遥测控制报文(INT_CTL)和遥测报告报文(INT_RPT)实现的长度分别为50字节、354字节、354字节。

为了分析带内遥测对网络性能的影响,需要计算出单个遥测周期内执行所有遥测任务所产生的数据量。对于图12所示的拓扑结构而言,进行一次全网络带内遥测,共需要执行16次遥测任务,需要传输的遥测报文数据量最多约为 $354 \times 16 = 5664$ 字节。对于超大规模的计算系统而言,假设包含的节点数目为20 000个,单次全网络带内遥测大约需要执行20 000次遥测任务,需要传输的遥测报文数据量大约为6.75 MB($20\,000 \times 354$ 字节)。这对于目前网络链路速率普遍高于200 Gbps的超算系统互连网络而言,即使执行连续的网络状态扫描,带内遥测的额外带宽消耗也微乎其微。

在仿真实验环境中,遥测服务器在周期性执行遥测任务的同时,本文使用iperf工具对 $h(0,0,0) \rightarrow h(1,0,0)$ 的带宽进行了测试,其带宽值未有明显的周期性波动。这也说明,带内遥测流量对普通数据流量的带宽影响可忽略不计。

5.3.2 CPU资源消耗

在本文提出的LW-INT架构中,除了遥测服务器上的软件组件(如遥测应用、遥测调度器和遥测主机接口)占用一定CPU资源外,其余的遥测报文传输与处理均由网卡NIC芯片和交换机NRC芯片完成,无需遥测路径的源和目的主机节点CPU资源介入。在该实验平台上,遥测应用通过命令行界面接收用户的遥测请求,遥测调度器依据遥测请求周期性地启动遥测任务。各遥测任务通过遥测主机接口向网络注入遥测报文,而遥测应用则接收遥测报告报文,并显示主要信息。

本文统计了以用户命令行方式启动遥测应用运行后,1分钟内遥测服务器(即 $h(0,0,0)$ 主机节点)的CPU利用率情况,统计结果如图13所示。可见,CPU利用率呈现周期性特征,其周期与遥测周期完

全一致。在每个遥测周期内,CPU利用率的平均峰值大约为60%,其活跃期大约为3秒,其余7秒处于空闲状态。可以推测,遥测服务器的CPU利用率不仅与CPU本身的性能有关,而且与遥测应用程序、网络规模等因素密切相关。鉴于此非本文工作的重点,不再赘述。

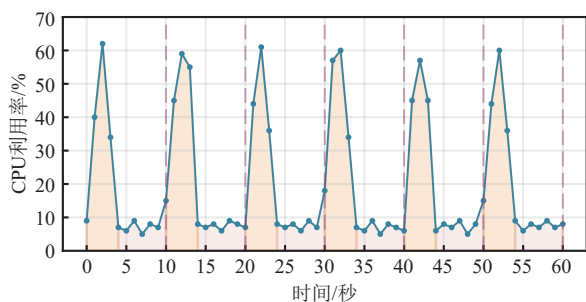


图13 遥测服务器的CPU资源消耗情况

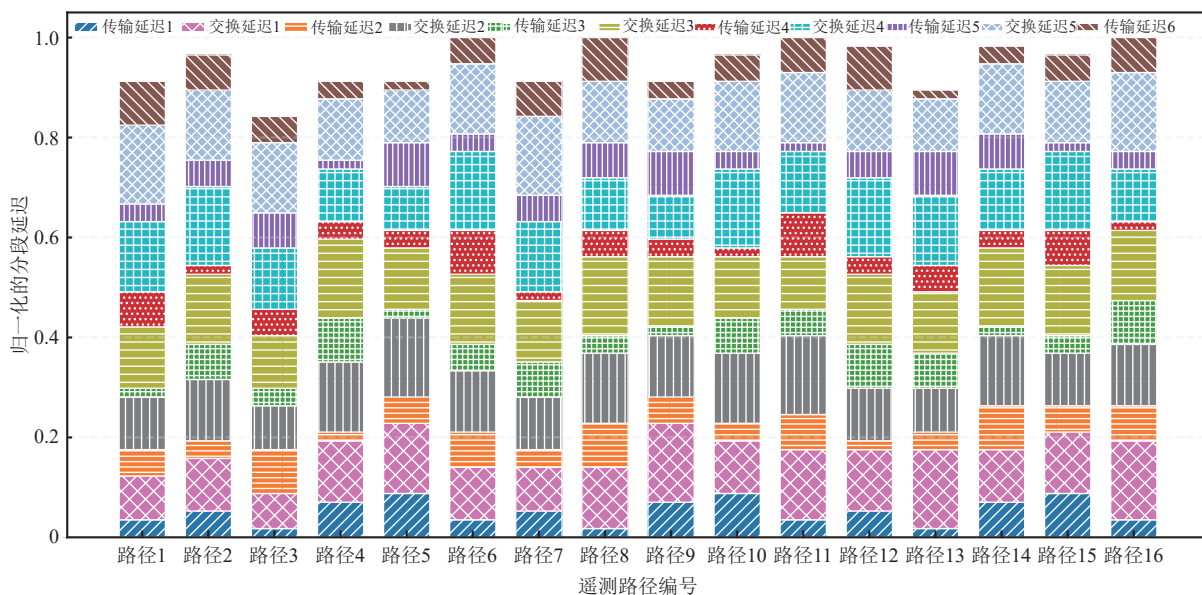


图14 各条遥测路径的分段延迟统计

6 总结与展望

针对超算系统互连网络数据平面高效率测量的需求,本文提出一种轻量级带内网络遥测架构(LW-INT),并基于该架构设计了两种启发式遥测调度算法(A2A-CB/EP)。LW-INT架构无需IEEE 1588协议提供全局同步时钟服务,可以实现对网络的端到端路由路径、链路状态和逐跳步网络时延等信息的准确测量,该架构具有硬件逻辑资源占用少且易于实现等优点,能契合HPC系统互连网络的遥测需求。通过较少的遥测调度次数,

5.3.3 分段延迟度量

在本实验中,遥测应用会在每个遥测周期内接收到16个遥测报告报文。每个报文携带了从源主机节点到目的主机节点的路由路径信息和各跳步网络延迟信息(包括SwitchLatency和TransmitLatency)。

图14呈现了16个遥测报告报文所携带的16条遥测路径上的网络分段延迟信息,每条路径由6段传输延迟和5段交换延迟交替堆叠构成,直观反映了各路径的延迟构成特征。交换延迟整体数值大于传输延迟,是各条路径延迟的主要贡献部分,且不同遥测路径的总延迟存在差异,归一化后最高的值为1。该仿真结果验证了本文提出的LW-INT架构分段延迟度量方法的有效性。结合遥测报告报文所携带的路由路径信息,分段延迟度量结果可为获取网络的精准状态提供了重要技术支撑。

A2A-CB/EP算法能够实现对互连网络系统内所有链路端口测量对象的覆盖。理论分析和实验结果表明,两种算法产生的遥测任务量均显著低于A2A基础算法,且其计算复杂度均低于INTO-CH基准算法。其中,A2A-EP算法与INTO-CH基准算法在遥测任务量性能上表现相当。尽管A2A-CB算法的遥测任务量性能低于A2A-EP算法,但其计算复杂度低于A2A-EP算法。A2A-EP和A2A-CB算法均显著优于SARG基准算法。可见,在实际应用场景中,可以根据超算系统的规模选择合适的算法。具体而言,在中小规模(节点数目<1000)的超算系统中,建议优先采用以遥测任务量为优化目标的

A2A-EP 算法, 以确保构建的遥测任务集合对网络资源的占用最小化; 而在大规模超算系统中, 则建议选择以更低计算复杂度为优化目标的 A2A-CB 算法, 以避免在构建遥测任务集合时因计算量过大而难以在合理时间内完成计算。

未来工作将进一步深入研究超算系统高性能互连网络的带内网络遥测技术。可以开展的工作包括如下两个方面: (1) 研究探索基于遥测数据的 HPC 应用通信性能优化问题。将结合 LW-INT 架构所收集的端到端路由路径、链路状态和逐跳步网络时延等详细遥测数据, 分析建立遥测数据与应用通信性能指标之间的时空对应关系。通过分析不同应用场景下网络状态参数对应用运行的性能影响规律, 构建面向应用视角的网络状态评估模型。例如, 针对高性能计算应用中常见的大规模数据传输、同步通信等场景, 提取遥测数据中与应用通信性能密切相关的关键特征, 进而实现对应用视角下网络状态精准拟合与动态预测, 为应用的通信优化、计算与网络资源的动态调度提供更具有科学性的依据, 提高 HPC 应用的通信性能, 降低 HPC 作业任务的完成时间。(2) 研究基于遥测数据的互连网络故障诊断与故障恢复技术, 提升网络的可用性与鲁棒性。

参 考 文 献

- [1] Tan L, Su W, Zhang W, et al. In-band network telemetry: A survey. *Computer Networks*, 2021, 186-107763.
- [2] Kim C, Sivaraman A, Katta N, et al. In-band network telemetry via programmable dataplanes//*Proceedings of the 2015 ACM Conference on SIGCOMM*. London, UK, 2015: 1-12
- [3] The P4.org Applications Working Group. In-band network telemetry (INT) dataplane specification version 2.1. San Francisco, USA: P4.org, 2020
- [4] Tang Y, Wu Y, Cheng G, Xu Z. Intelligence enabled SDN fault localization via programmable in-band network telemetry//*Proceedings of the 20th IEEE Int'l Conf. on High Performance Switching and Routing (HPSR)*. Xi'an, China, 2019. 1-6
- [5] Li Y, Miao R, Liu H, et al. HPCC: high precision congestion control//*Proceedings of the 2019 ACM Conference on Sigcomm*. Beijing, China, 2019:421-434
- [6] Turkovic B, Kuipers F, van Adrichem N, et al. Fast network congestion detection and avoidance using P4//*Proceedings of the 2018 Workshop on Networking for Emerging Applications and Technologies*. Budapest, Hungary, 2018: 45-51
- [7] Katta N, Hira M, Kim C, et al. Hula: scalable load balancing using programmable data planes//*Proceedings of the Symposium on SDN Research*. Santa Clara, USA, 2016: 1-12
- [8] Jeyakumar V, Alizadeh M, Geng Y, et al. Millions of little minions: using packets for low latency network programming and visibility. *ACM SIGCOMM Comp Commun Review*, 2014, 44(3): 3-14
- [9] Niu B, Kong J, Tang S, Li Y, Zhu Z. Visualize your IP-over-optical network in realtime: A P4-based flexible multilayer in-band network telemetry (ML-INT) system. *IEEE Access*, 2019, 7: 82413-82423
- [10] Tammana P, Agarwal R, Lee M. Cherrypick: Tracing packet trajectory in software-defined datacenter networks//*Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*. Santa Clara, USA, 2015:1-7
- [11] Karaagac A, De Poorter E, Hoebeke J. In-band network telemetry in industrial wireless sensor networks. *IEEE Transactions on Network and Service Management*, 2020, 17(1): 517-531
- [12] Haxhibeqiri J, Isolani P H, Marquez-Barja J M, Moerman I, Hoebeke J. In-band network monitoring technique to support SDN-based wireless networks. *IEEE Transactions on Network and Service Management*, 2021, 18(1): 627-641
- [13] Anand M, Subrahmaniam R, Valiveti R. POINT: An intent-driven framework for integrated packet-optical in-band network telemetry//*Proceedings of the 2018 IEEE Int'l Conf. on Communications (ICC)*. Kansas City, USA, 2018. 1-6
- [14] Zaballa E O, Franco D, Thomsen S E, et al. Towards monitoring hybrid next-generation software-defined and service provider MPLS networks. *Computer Networks*, 2021, 191: 107960
- [15] Cao J J, Xiao L Q, Wang K F, et al. The implementation and evaluation of in-band network management in supercomputing system. *Chinese Journal of Computers*, 2016, 39(9): 1717-1731 (in Chinese)
(曹继军, 肖立权, 王克非, 庞征斌. 超级计算系统互连网络带内管理的实现与评测. *计算机学报*. 2016, 39(9): 1717-1732)
- [16] Fioccola G, Capello A, Cociglio M, et al. Alternate-Marking Method for Passive and Hybrid Performance Monitoring. Fremont, USA: IETF, 2018
- [17] Mizrahi T, Navon G, Fioccola G, et al. AM-PM: Efficient network telemetry using alternate marking. *IEEE Network*, 2019, 33(4): 155-161
- [18] Van Tu N, Hyun J, Hong J W K. Towards onos-based sdn monitoring using in-band network telemetry.//*Proceedings of the 19th Asia-Pacific Network Operations and Management Symposium*. Seoul, Republic of Korea, 2017: 76-81
- [19] Tand S F, Li D Y, Niu B, et al. Sel-INT: A runtime-programmable selective in-band network telemetry system. *IEEE Transactions on Network and Service Management*, 2020, 17(2): 1018-1031
- [20] Tammana P, Agarwal R, Lee M. Simplifying datacenter network debugging with pathdump//*Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*. Savannah, USA, 2016: 233-248
- [21] Marques J A, Luizelli M C, Costa Filho R I T, et al. An optimization-based approach for efficient network monitoring using in-band network telemetry. *Journal of Internet Services and Applications*, 2019, 10(1):12

- [22] Hohemberger R, Gastro A G, Vogt F G, et al. Orchestrating in-band data plane telemetry with machine learning. *IEEE Communications Letters*, 2019, 23(12):2218-2221
- [23] Bhamare D, Kassler A, Vestin J, et al. IntOpt: in-band network telemetry optimization for NFV service chain monitoring// *Proceedings of the 2019 IEEE International Conference on Communications (ICC)*. Shanghai, China, 2019. 1-7
- [24] Bhamare D, Kassler A, Vestin J, et al. IntOpt: in-band network telemetry optimization framework to monitor network slices using P4. *Computer Networks*, 2022, 2016: 109214
- [25] Ben Basat R, Ramanathan S, Li Y, et al. PINT: Probabilistic in-band network telemetry//*Proceedings of the 2020 ACM Conference on SIGCOMM*. Virtual, USA, 2020:541-554
- [26] Chen W, Tian Y, et al. Task scheduling for probabilistic in-band network telemetry. *IEEE/ACM Transactions on Networking*, 2022, 30(6):2664-2677
- [27] Yuan P Y, Wang M, Wang L H, et al. Adaptive detection path configuration for in-band network telemetry in SDN based on graph segmentation. *Chinese Journal of Software*, 2023, 34(4): 2865-2877 (in Chinese)
(原鹏翼, 王淼, 王凌豪, 等. SDN中基于图分割的自适应带内网络遥测探测路径配置. *软件学报*, 2023, 34(4): 2865-2877)
- [28] Zhao Y, Yang K, Liu Z, et al. LightGuardian: A full-visibility, lightweight, in-band telemetry system using sketchlets// *Proceedings of the 18th Usenix Symposium on Networked System Design and Implementation (NSDI)*. Virtual, USA, 2021:65-80
- [29] Kousha P, Kamal Raj S D, et al. INAM: cross-stack profiling and analysis of communication in MPI-based applications// *Proceeding of the practice and experience in advanced research computing (PEARC'21)*. Boston, USA, 2021: 1-10
- [30] Langlet J, Ben Basat R, Oliaro G, et al. Direct telemetry access//*Proceedings of the 2023 ACM Conference on SIGCOMM*. New York, USA, 2023:351-364
- [31] Adamson R, Osborne T, Lester C, et al. STREAM: A scalable federated HPC telemetry platform//*Proceedings of the Cray User Group*. Helsinki, Finland, 2023: 1-8
- [32] Sun H, Li J, He J, et al. OmniWindow: a general and efficient window mechanism framework for network telemetry// *Proceedings of the 2023 ACM Conference on SIGCOMM*. New York, USA, 2024: 411-424
- [33] Ding R, Liu X, Yang S, et al. RD-Probe: Scalable monitoring with sufficient coverage in complex datacenter networks// *Proceedings of the 2024 ACM Conference on SIGCOMM*. Chicago, USA, 2024: 411-424
- [34] Guo S, Liu K, Lai Y, et al. ByteTracker: an agentless and real-time path-aware network probing system//*Proceedings of the 2025 ACM Conference on SIGCOMM*. Los Angeles, USA, 2025: 501-514
- [35] InfiniBandSM Trade Association. InfiniBand™ Architecture Specification Volume 1 Release 1.4. Beaverton, USA: InfiniBand Trade Association, 2020
- [36] Guo C, Wu H, Deng Z, et al. RDMA over commodity ethernet at scale//*Proceedings of the 2016 ACM Conference on SIGCOMM*. Florianopolis, Brazil, 2016: 202-215
- [37] Karp R M. Reducibility among combinatorial problems. Miller R E, Thatcher J W, Bohlinger J Deds. *Complexity of computer computations*. New York, USA: Plenum Press, 1972: 85-103
- [38] OpenCores. The reference community for free and open source gateway IP cores. <https://opencores.org>. 2026-01-10
- [39] Yuan X, Nienaber W, Duan Z, et al. Oblivious routing for fat-tree based system area networks with uncertain traffic demands // *Proceedings of the ACM SIGMETRICS'07*. San Diego, USA, 2007: 12-16
- [40] Lin X, Chung Y, Huang T. A multiple LID routing scheme for fat-tree-based Infiniband network//*Proceedings of the 18th IEEE International Parallel and Distributed Processing Symposium (IPDPS'04)*. SanaFe, USA, 2024: 1-8
- [41] Ajima Y, Sumimoto S, Shimizu T. Tofu: A 6D mesh/torus interconnect for exascale computers. *Computer*, 2009, 42(11): 36-40
- [42] Kim J, Dally W J, Scott S, et al. Technology-driven, highly-scalable dragonfly topology//*Proceedings of the 35th Annual International Symposium on Computer Architecture (ISCA)*. Beijing, China, 2008: 77-88
- [43] Hagberg A, Swart P J, Schult D A. Exploring Network Structure, Dynamics, and Function Using NetworkX. Los Alamos, USA: Los Alamos National Laboratory (LANL), 2008: LA-UR-08-05495
- [44] Yang X J, Liao X K, Lu K, et al. The TianHe-1A supercomputer: its hardware and software. *Journal of Computer Science and Technology*, 2011, 26(2): 344-351
- [45] LiaoX, Pang Z, Wang K, et al. High performance interconnect network for Tianhe system. *Journal of Computer Science and Technology*, 2015, 30(2): 259-272
- [46] Ajima Y, Kawashima T, Okamoto T, et al. The Tofu Interconnect D//*Proceedings of the 20th IEEE International Conference on Cluster Computing (CLUSTER)*. Belfast, UK, 2018: 646-654
- [47] Dally W J, Towles B. Principles and Practices of Interconnection Networks. San Francisco, USA: Morgan Kaufmann, 2004. 112-321.
- [48] Faanes G, Bataineh A, Roweth D, et al. Cray cascade: a scalable HPC system based on a dragonfly network// *Proceedings of the International Conference on SuperComputing*. Salt Lake City, USA, 2012: 1-9
- [49] Elbediwy M, Pontikakis B, David J P, et al. Enabling rank-based P4 programmable schedulers: requirements, implementation, and evaluation on BMv2 switches. *IEEE/ACM Transactions on Networking*, 2024, 32(5): 2101-2114
- [50] Gupta, N, Maashi M S, Tanwar S, et al. A comparative study of software defined networking controllers using Mininet. *Electronics*, 2022, 11(17): 2715



CAO Ji-Jun, Ph. D., associate professor. His main research directions include high-performance interconnection networks, network measurement and management, and complex coding circuits.

XU Jia-Qing, Ph. D., associate professor. His main research interests include high-performance computing and high performance interconnection networks.

LI Jin-Wen, Ph. D., professor. His main research interests include computer system architecture and high-speed circuit design.

YANG Han-Zhi, bachelor, engineer. His main research interests include network measurement and management technology

KONG De-Yong, Ph. D., associate professor. His main research interests include computer networks and communications.

DAI Yi, Ph. D., associate professor. Her main research interests include high-performance interconnection networks and collective communication.

Background

This paper investigates a problem within the field of High-Performance Computing (HPC), specifically focusing on in-band network telemetry methods and scheduling optimization in interconnect networks. The core challenge addressed is the Minimum Path Coverage Problem for Telemetry (MPCP-T) in high-speed interconnect networks of supercomputers, particularly concerning data plane measurements and telemetry scheduling.

Internationally, in-band network telemetry (INT) has become a focal point in network measurement technology research. Relevant studies have established INT specifications, proposed telemetry architectures such as INT and AM-PM, and developed scheduling algorithms like INTO and INTOPT, with applications extended to Ethernet, optical networks, and wireless networks. However, existing solutions struggle to adapt to the unique characteristics of supercomputing interconnect networks, such as the lack of IEEE 1588 protocol support and variations in virtual channel latencies. Additionally, current scheduling algorithms face issues of high computational complexity, failing to meet the telemetry requirements of supercomputing systems.

In response to this situation, this paper proposes a lightweight in-band telemetry architecture (LW-INT), enabling accurate end-to-end routing path, link state, and hop-by-hop delay measurements without the need for global synchronized clocks. Simultaneously, it defines the MPCP-T problem model

and designs two heuristic algorithms, A2A-CB and A2A-EP, to approximate solutions for this NP-hard problem.

The research project under this study is funded by the National Key Research and Development Program of China, titled “Key Technology Verification for the Tianhe 20E-Level Supercomputer” (Project No. 2025YFB3003100), with a sub-project focusing on “High-Performance Interconnect Network Technology inter Nodes” (Sub-project No. 2025YFB3003102). This project is a priority special project within the field of high-performance computing, aiming to achieve breakthroughs in critical technologies such as the architecture of next-generation indigenous supercomputing systems, novel processor structures, high-speed interconnect networks, system-level infrastructure, software environments, application-oriented co-design, large-scale system management, and fault tolerance. The research team (Thianhe team) has made significant achievements in supercomputing system development, key technology breakthroughs, and application implementations, with the Tianhe supercomputing system and related research outcomes previously earning the National Special Award for Scientific and Technological Progress.

The research findings of this paper can address the lightweight and precise measurement of the data plane in the high-speed interconnect networks of next-generation indigenous supercomputing systems, making it of great significance.