



计算机科学

COMPUTER SCIENCE

灾害场景下基于SDN和无人机辅助的无线边缘存储系统性能优化方法

孙石泉, 叶苗, 朱成, 王勇, 蒋秋香

引用本文

孙石泉, 叶苗, 朱成, 王勇, 蒋秋香. 灾害场景下基于SDN和无人机辅助的无线边缘存储系统性能优化方法[J]. 计算机科学, 2025, 52(11): 306-319.

SUN Shiquan, YE Miao, ZHU Cheng, WANG Yong, JIANG Qiuxiang. Performance Optimization of Wireless Edge Storage System Based on SDN and Drone Assistance in Disaster Scenarios [J]. Computer Science, 2025, 52(11): 306-319.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于卷积双延迟深度确定性策略梯度的卫星网络多路径路由算法](#)

Multipath Routing Algorithm for Satellite Networks Based on Convolutional Twin Delay Deep Deterministic Policy Gradient

计算机科学, 2025, 52(11): 280-288. <https://doi.org/10.11896/jsjcx.240800161>

[基于申威编译器的并行调度策略优化技术研究](#)

Research on Parallel Scheduling Strategy Optimization Technology Based on Sunway Compiler

计算机科学, 2025, 52(9): 137-143. <https://doi.org/10.11896/jsjcx.241200072>

[基于SDN的通存一体化边缘在网存储节点选择方法](#)

SDN-based Integrated Communication and Storage Edge In-network Storage Node Selection Method

计算机科学, 2025, 52(8): 343-353. <https://doi.org/10.11896/jsjcx.240900174>

[基于深度强化学习的在线并行SDN路由优化算法研究](#)

Online Parallel SDN Routing Optimization Algorithm Based on Deep Reinforcement Learning

计算机科学, 2025, 52(6A): 240900018-9. <https://doi.org/10.11896/jsjcx.240900018>

[基于GMM的容器定制化调度策略](#)

Customized Container Scheduling Strategy Based on GMM

计算机科学, 2025, 52(6): 346-354. <https://doi.org/10.11896/jsjcx.240900154>

灾害场景下基于 SDN 和无人机辅助的无线边缘存储系统性能优化方法

孙石泉¹ 叶苗¹ 朱成^{2,3} 王勇³ 蒋秋香⁴

1 桂林电子科技大学信息与通信学院 广西 桂林 541004

2 桂林医科大学信息中心 广西 桂林 541001

3 桂林电子科技大学计算机与信息安全学院 广西 桂林 541004

4 桂林电子科技大学光电工程学院 广西 桂林 541004

(sunshiquan, stu@foxmail.com)

摘要 传统边缘分布式存储系统中网络配置繁琐,优化网络所需的网络状态信息测量操作开销大,当终端设备对数据存储和检索的业务需求处于高峰时,会导致网络链路负载过重从而影响数据转发传输的性能。此外,现有分布式存储系统在进行数据的存储节点选择时,只考虑了节点剩余存储空间,没有考虑网络状态和节点自身负载对系统存储性能的影响。为解决上述问题,设计和实现了一种基于软件定义网络(Software Defined Network,SDN)和无人机辅助的边缘分布式存储系统,利用 SDN 技术测量网络状态、网络节点自身负载和存储节点负载状态信息,通过无人机移动节点飞行到重负载网络节点的上方进行分流以平衡各条链路的流量负载;对于重负载网络节点和存储节点的选择,提出了一种基于多属性决策模型综合考虑网络状态和节点自身负载状态的节点选择算法,选择出重负载网络节点和合适的存储节点,然后通过对无人机的位置部署,实现网络链路流量的分流,平衡网络链路的流量负载。经实验测试,结果显示在无线 Mesh 网络拓扑中,所提无线边缘分布式存储系统的存储性能优于现有边缘分布式存储系统,存储时间明显缩短,在增加流量负载的情况下依然可以保持良好的存储性能,具有良好的负载均衡性能。

关键词: 边缘分布式存储;软件定义网络;节点选择;负载均衡;无线 Mesh

中图分类号 TP393

Performance Optimization of Wireless Edge Storage System Based on SDN and Drone Assistance in Disaster Scenarios

SUN Shiquan¹, YE Miao¹, ZHU Cheng^{2,3}, WANG Yong³ and JIANG Qiuxiang⁴

1 School of Information and Communications, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China

2 Network Information Center, Guilin Medical University, Guilin, Guangxi 541001, China

3 School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China

4 School of Optoelectronic Engineering, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China

Abstract Traditional edge distributed storage systems often suffer from cumbersome network configuration and high operational overhead in measuring network state information. During peak demand periods for data storage and retrieval by terminal devices, network links can become overloaded, adversely affecting data transmission performance. Furthermore, existing distributed storage systems typically consider only the remaining storage space of nodes when selecting storage nodes, neglecting the impact of network state and node load on system storage performance. To address these issues, this paper designs and implements an edge-distributed storage system assisted by software-defined network(SDN) and drones. The system uses SDN technology to measure network state, node load, and storage node load information. Drones fly above heavily loaded network nodes to offload traffic and balance the load across different links. For the selection of heavily loaded network nodes and storage nodes, this paper proposes a node selection algorithm based on a multi-attribute decision model that comprehensively considers network state and node load. The algorithm identifies heavily loaded network nodes and suitable storage nodes, and deployment of drones helps achieve traffic

到稿日期:2024-09-02 返修日期:2024-12-14

基金项目:国家自然科学基金(62161006,62172095);广西研究生教育创新计划(YCSW2023310);广西无线宽带通信与信号处理重点实验室主任基金项目(GXKL06220110)

This work was supported by the National Natural Science Foundation of China(62161006,62172095), Subsidization of Innovation Project of Guangxi Graduate Education(YCSW2023310) and Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing(GXKL06220110).

通信作者:朱成(zhucheng@glmu.edu.cn)

offloading and load balancing. Experimental tests on a wireless Mesh network topology demonstrate that the proposed wireless edge-distributed storage system outperforms existing edge-distributed storage systems in terms of storage performance. The proposed system significantly reduces storage time and maintains good performance even under increased traffic load, demonstrating excellent load-balancing capabilities.

Keywords Edge distributed storage, Software defined network, Node selection, Load balancing, Wireless Mesh

1 引言

随着物联网(Internet of Things, IoT)技术的迅猛发展,边缘设备的数量呈指数级增长。这些设备被广泛应用于智能家居^[1]、智能交通^[2]、工业自动化^[3]和智慧城市^[4]等多个领域。然而,随着边缘设备的激增,产生的数据量也在不断增加,这对数据的存储和管理提出了新的挑战。通过互联网将数据存储在远程服务器上的云存储方式存在传输时延高、数据不安全、隐私泄露和依赖网络连接等问题,不适用于边缘网络下的海量数据存储^[5]。而边缘存储旨在将数据存储和处理能力从集中式数据中心转移到网络边缘,使其更接近数据源头。这样可以降低数据传输的延迟,提升响应速度,减轻核心网络的负载,减少带宽消耗,同时也能减少数据传输的次数和范围,从而降低了数据泄露的风险^[6]。

在边缘网络中存在着大量的移动终端设备和路由转发设备,有线连接的方式虽然数据传输速率较快,但存在布线繁琐、有线连接成本高、设备移动性差及可扩展性差的问题。边缘网络通常需要覆盖特定区域,如城市、工厂等。当某个网络节点出现故障或失效时,该节点所覆盖地区的网络通信服务会严重受损,也会影响整个网络的性能^[7]。无线 Mesh 网络通过节点之间的多跳通信,可以扩展网络的覆盖范围。无线 Mesh 节点之间的相互连接,可以形成多条通信路径,降低单点故障对整个网络的影响^[8]。Mesh 节点之间的无线连接也有利于节点的灵活部署和扩展,无需有线连接的布线操作,因而降低了网络部署和运维的成本。此外,边缘网络往往涉及到移动设备和动态场景,如智慧城市、智能交通等。无线 Mesh 网络具有良好的移动性和动态性,可以灵活应对这些场景的需求,保障移动设备的连接和通信。

在灾害场景下,如地震、洪水等自然灾害,传统的有线网络可能会因为基础设施的损坏而无法正常工作,例如基站等设施受损,以及军事前沿阵地组网缺乏固定基站设置等。这时,无线 Mesh 组网网络的优势得以体现,其可以快速部署,保证网络的连通性和数据传输的稳定性。因此,研究和设计一种在灾害场景下具有高可靠性和灵活性的边缘存储网络系统具有重要的现实意义。在这些紧急情况下,常常会出现“三断”情况,即断路、断网和断电。边缘存储系统面临突发性和不确定性带来的网络拥堵和数据流量激增,可能导致传输延迟。同时,环境条件恶劣,如电力短缺、设备损坏和信号干扰,会严重影响存储节点的稳定性和可用性。此外,灾后快速恢复要求系统具备高效的数据冗余和恢复机制,以确保关键数据的安全性和可获取性,这些因素共同增加了边缘存储系统的复杂性,对系统设计的可靠性和灵活性提出了更高的要求。无线 Mesh 网络和边缘存储技术的结合可以提供更可靠的解决方案,有效支持灾害现场的数据处理和传输需求。

SDN 架构将控制平面和数据平面分离,控制器可以获取全局的网络信息,交换机则只负责数据包的转发。这种数控分离的架构有利于设备的灵活部署,不受限于设备的生产厂商和型号。传统网络在计算网络状态信息时步骤复杂、配置繁琐,且获取网络状态信息的开销大。SDN 架构中控制器通过 Openflow 协议可控制数据层面中的所有交换机,获取交换机的端口信息,从而计算出全局的网络状态信息^[9]。利用 SDN 技术可解决传统网络架构网络设备配置繁琐及网络状态测量步骤复杂的问题。

在边缘分布式存储中,网络链路的负载情况和存储节点的选择对存储和检索的性能至关重要。当边缘移动终端设备产生的数据量较大,对数据存储和检索的业务需求处于高峰时,会导致网络链路负载过重,从而影响数据转发传输的性能^[10]。在无线 Mesh 网络中,网络拓扑中的每个节点可灵活移动,具有较高的动态性。网络链路的负载均衡旨在平衡各条链路的流量负载,这可以借助 UAV 移动节点的灵活性,实现网络链路流量的分流^[11-13]。因此,设计一种重负载网络节点选择算法,对于网络链路的负载均衡有重要意义。在选择出负载较重的网络节点后,将无人机移动节点移动至重负载节点上方,可实现网络链路流量的分流。因此,如何找到网络拓扑中负载较重的节点是网络负载均衡的关键问题^[14-15]。此外,当存储节点与移动终端设备的距离过远,或是存储节点的剩余存储空间不足时,系统的存储和检索性能也会受到影响。因此,如何选择合适的存储节点也是优化边缘存储系统性能的问题之一^[16]。

近年来,许多学者在提升边缘存储系统性能提升方面做出了创新的研究,研究方法主要集中在路由算法、无人机辅助边缘网络中无人机位置的部署,以及存储节点选择方法 3 方面。其中,路由算法通过优化数据传输的路径,来平衡网络链路的流量负载;无人机辅助的方式通过将无人机移动节点作为中继节点,优化无人机的位置部署来实现网络链路的分流;存储节点选择方法通过选择合适的存储节点进行分布式存储,提高数据存储效率。路由算法主要有基于群智能优化算法的路由算法和基于强化学习的智能路由算法^[17]。基于群智能优化算法的路由算法通过模拟自然界中生物群体行为的协作行为来寻找问题的最优解,通过算法的多次迭代,在网络拓扑中找到对多个目标优化的最佳路径,平衡各个网络节点的负载。基于强化学习的路由算法通过智能体与网络环境的交互学习,根据不同的状态选择最优的行动,从而得到最优路径。在无人机辅助的边缘网络中,无人机搭载边缘服务器作为移动基站,根据用户分布及其动态移动和网络性能的优化目标调整无人机的位置部署^[12],给终端用户提供更好的计算或存储服务。存储节点选择的方法旨在选择合适的存储节点,考虑存储节点的距离、传输数据的类型和存储节点的存储

空间等因素选择出合适的存储节点,优化存储性能^[18]。通过路由算法来进行负载均衡的方法可以动态调整数据流的路径,使网络资源得到更有效的利用,将流量均匀分布到多条链路上,避免单一链路成为瓶颈,从而提高网络的总吞吐量。但其无法适应动态的网络拓扑,当网络拓扑发生变化时,原有的群智能优化算法或强化学习算法需要重新进行训练学习,无法及时地更新路由。同时,在静态拓扑中,当业务处于高峰期、传输数据量暴增时,部分节点不可避免地会出现负载过重的情况,并且无法及时平衡网络链路的流量负载。无人机辅助的边缘存储网络利用无人机作为移动节点来增强边缘计算和存储能力。无人机可以根据业务需求灵活部署在不同的地理位置,提供临时或长期的网络覆盖和存储服务,使系统具有良好的灵活性和可扩展性。无人机还可以在网络中移动,根据实时需求进行动态负载均衡,优化资源利用和数据传输效率。但现有的无人机辅助的边缘网络往往是在仿真平台上进行研究,算法复杂度高,无法在实际硬件上进行部署,并且在无人机的位置部署中,通常考虑用户的分布和移动性、基站节点的无线功耗及频谱利用率等,没有考虑网络状态及节点的自身负载状态,而链路的网络状态和节点自身的负载状态指标对数据传输的影响是巨大的。对于存储节点的选择方法,现有方法通常考虑存储节点的距离、存储数据的数据类型和存储节点之间的负载均衡等因素,但忽视了网络状态及节点自身负载状态对存储性能的影响。当用户到存储节点之间的网络时延过高,或者存储节点正在运行复杂算法或磁盘 IO 读写操作时,也会严重影响分布式存储的效率。

针对上述研究存在的问题,本文设计了一种无人机(Unmanned Aerial Vehicle, UAV)辅助的基于 SDN 架构的无线边缘存储系统。该系统通过无线 Mesh 组网的方式构建网络拓扑,并利用 SDN 技术测量全局的网络状态信息和网络节点状态信息。对于重负载网络节点和存储节点的选择,本文设计了一种多属性决策模型,是综合考虑网络状态和节点自身负载状态的节点选择算法,选择出负载最重的网络节点及合适的存储节点。此外,在系统中引入无人机移动节点,通过其分流作用,减少网络链路的流量负载,实现动态负载均衡。该边缘存储的系统架构和系统原型适用于在灾害场景下进行临时组网和对采集的数据进行高效实时的存储。

本文的创新点总结如下:

1)设计和实现了一种基于 SDN 和无人机辅助的边缘分布式存储系统原型,适用于部署在无线 Mesh 组网的边缘网络中。

2)传统网络架构存在网络配置繁琐、网络状态信息测量操作开销大的缺点,而本文设计的边缘分布式存储系统原型利用 SDN 技术采集无线 Mesh 网络下的网络状态信息、网络节点的状态信息和存储节点的状态信息,能有效地解决传统网络架构获取网络状态信息困难的问题。利用 SDN 技术测量链路的网络属性,包括剩余带宽、时延和丢包率等,并巧妙地利用 Packet-in 机制获取网络节点和存储节点的状态信息。

3)相比于现有的分布式边缘存储系统通过改进路由方式或考虑用户分布因素来优化链路负载的方式,本文充分考虑了 AP 移动性带来的便利,通过无人机移动节点实现了一种

动态负载均衡的分流方式。通过建立对重负载节点的多属性决策选择模型(包括网络状态属性和节点自身负载属性),再将无人机移动节点定位飞行到重负载节点上方进行分流,有效平衡了网络链路的流量负载,提高了存储系统的网络传输效率。

4)相比当前边缘存储系统仅考虑存储节点剩余存储空间这个单一因素,影响了系统读写性能的问题,本文基于多属性决策模型,设计并实现了一种综合考虑网络实时状态和存储节点实时负载状态的边缘存储系统的存储节点选择算法,能有效提高边缘分布式存储系统的存储性能。

本文第 2 章介绍了相关工作;第 3 章介绍了本文工作的问题描述;第 4 章详细介绍了系统与算法的设计;第 5 章介绍了实验结果;最后总结全文并展望未来工作。

2 相关工作

对于提升边缘存储网络系统的性能,现有研究主要集中在路由算法、UAV 辅助的网络中的 UAV 位置部署和存储节点的选择方法上。本章将介绍网络性能优化的相关工作,并分析不同方法的优缺点。

2.1 路由算法

Bhowmik 等^[19]设计了一种基于遗传算法的路由算法,通过 SDN 技术感知数据平面的流量负载,然后利用遗传算法计算得到全局的最优路由,动态地给链路分配流量负载,并应用于美国国家科学基金会网络上。Sha 等^[20]提出了一种基于深度确定性策略梯度(Deep Deterministic Policy Gradient, DDPG)强化学习算法的面向任务的混合路由方法,通过综合考虑网络状态和任务需求来决策路径输出是单路径还是多路径,并通过 DDPG 算法逐跳计算每个节点的路由表的流量分割模型,然后通过混合路由算法得到单路径与多路径混合的传输路径。Pei 等^[21]提出了链路虚拟容量的概念,并通过近端策略优化(Proximal Policy Optimization, PPO)算法调整关键链路的虚拟容量,旨在减少用 k-path 方法获取的多条路径的最大链路利用率,进而平衡链路的负载。Zheng 等^[22]研究了改进蚁群算法在 SDN 中负载均衡的应用,把控制层间分为网络拓扑感知、状态采集、负载均衡算法核心和流表分布 4 个模块。改进的蚁群算法综合考虑链路和服务器的性能参数,能根据网络链路流量和服务器利用率的变化动态调整路由方案,有效地实现了全局的负载均衡。Younus 等^[23]研究了强化学习在无线传感器网络上的路由优化,通过 Q-learning 强化学习算法优化数据传输的能量消耗和服务质量需求,SDN 控制器根据之前的奖励情况选择最佳路径,有效地提高了节点的寿命和数据包传输率。Riyaz 等^[24]研究和分析了蚁群算法(Ant Colony Optimization, ACO)和粒子群优化算法(Particle Swarm Optimization, PSO)在 SDN 负载均衡中的应用,通过评估节点和链路的可靠性提出了 SDN 未来的负载均衡框架,在 ACO 和 PSO 算法中考虑使用来自交换机的直接和间接信息将数据包转发到下一个节点,该框架可以有效地平衡节点负载并提高 SDN 中的服务质量(Quality of Service, QoS)。Ye 等^[25]提出了一种集成优化方法,首先使用奇异谱分析从网络流量数据中提取信息,并将数据分为有效载荷数

据和噪声数据,然后分别使用深度学习模型和卷积神经网络模型来预测有效载荷数据和噪声数据,最后通过基于人工蜂群算法的启发式路径选择方法来计算最佳路径。该方法在吞吐量、抖动和负载均衡因子方面均优于现有的方法。Zheng 等^[26]提出了一种图神经网络(Graph Neural Network, GNN)与深度强化学习(Deep Reinforcement Learning, DRL)结合的路由策略,通过 GNN 感知动态的网络拓扑,结合 DRL 的自主学习能力寻找最优的路由,有效地降低了端到端时延及链路的最大利用率。Gunavathie 等^[27]提出一种基于机器学习的主动重新路由方案,通过机器学习技术对应用程序进行分类并分配流优先级,同时分析每条链路的重要性,当关键链路出现过载时,实时调整路由以避免拥塞。以上启发式或基于人工智能的路由算法虽然有效地平衡了网络节点的负载,但算法的复杂度过高,并且需要经过多次训练才能得到最佳路径,当网络拓扑发生变化时无法及时获取全局的最佳路径,不适合在实际网络中进行部署。Qi 等^[28]提出一种基于 SDN 的空间网络多路径传输的动态子流分配策略,利用 SDN 控制器对网络状态进行监控和分析,并根据路径的时延和带宽为子流选择多个最优和不相交的传输路径,以提高链路利用率和负载均衡。Qi 等^[29]提出一种多路径路由策略,综合考虑链路时延和节点负载计算链路的传输成本,及时调整端到端的路由,有效地平衡了各节点的带宽消耗。Dong 等^[30]研究了一种基于扩展路径拥塞避免的星座网络负载均衡路由算法,利用更多的全局链路拥塞信息、有效的拥塞避免机制,以及主动发现与自动检测相结合的状态通知机制,获得了更好的网络负载均衡性能。Gayatri 等^[31]提出了一种基于最小公用多路径的新型路由,用于在节点之间传输数据时平衡负载和多路径合并。Luan 等^[32]研究了 Dijkstra 算法,对 SDN 链路负载均衡算法多条路径进行优化,计算出 k 条可用路径。上述路由算法的计算量不大,且负载均衡效果良好,但没有考虑网络拓扑的动态性,也没有考虑节点的移动性或节点故障对整个网络的影响,并且在传统网络架构下对网络状态信息的测量开销大,配置繁琐,不利于网络的管理和维护。

2.2 无人机辅助的负载均衡

Latif 等^[33]提出一种无人机辅助的网络结构,将无人机节点作为中继节点分担链路的流量负载,设计了一种算法,将流量负载均匀地分配到多 UAV 系统的不同链路上。Zhai 等^[34]提出一种无人机辅助的无线网络,把无人机节点作为移动基站节点,考虑各个地面基站的用户关联数量及频谱率,设计遗传算法来优化无人机位置以平衡各个小区的流量负载,该方法有效提高了网络的吞吐量和效率。Tao 等^[35]针对地面车辆网络的连通性和通信性能,提出了一种基于模型的通信中继无人机运动控制方法,通过改进粒子群优化和非线性模型预测控制相结合的方式决策中继无人机运动控制。Chen 等^[36]针对移动边缘计算(Mobile Edge Computing, MEC)网络中 UAV 的位置部署,提出了一种基于深度强化学习的智能优化算法,以最小化传输延迟、计算延迟和系统能耗。Zhang 等^[37]研究了边缘存储网络中无人机存储节点对系统存储性能的影响,无人机搭载存储服务器作为移动节点,使用 k-means 聚类算法对用户进行分类,考虑了 3 个用户与

无人机的关联标准,包括用户接收信噪比、用户偏好和延迟,根据无人机与用户群体的关联,将无人机部署到不同的位置上,有效提高了存储效率。Miao 等^[38]针对传统固定基站在复杂地形下的部署成本、信号传输损耗、覆盖范围有限等问题,研究了一种全局和局部路径规划的多无人机辅助的 MEC 卸载算法。该算法考虑监测区域优先次序、无人机剩余能量和到目标点的距离,以尽量减少 UAV 的飞行距离和能耗,同时根据用户的移动性计算 UAV 的最优通信覆盖率,在有效地平衡了地面基站节点的负载的同时,提供了更多的卸载服务。Wang 等^[39]设计了一种支持 UAV 的 MEC 网络的任务卸载框架,把问题分解为区域划分和无人机位置优化两个子问题,将无人机的位置部署优化子问题建模为旅行商问题,在平衡各个区域流量负载的同时,可以显著降低无人机的能耗。Elgendy 等^[40]针对多用户、多层无人机辅助 MEC 系统,提出了一种负载均衡、资源分配和任务卸载的集成模型,将地面基站交叉服务区域中的用户移交最合适服务器来优化地面 MEC 服务器之间的负载,并将无人机悬停在地面负载过重的基站服务器区域上空,最终将联合任务卸载、负载均衡和资源分配公式化为一个整数问题。Zhu 等^[41]研究了多无人机辅助应急通信,调度无人机为所有地面用户提供服务,提出了一种基于聚类和平等区域划分的负载均衡部署和路径规划算法,对无人机的位置进行联合优化。Pan 等^[42]提出了一种中继覆盖算法,不断优化无人机的位置和地面用户的小区划分,还研究了无人机间的中继消息转发方法,提出了一种以中继链路吞吐量为优化目标的中继选择方法,保证无人机间的全连通性,该方案有效地降低了数据传输的时延。Guo 等^[43]针对地面用户数量、位置和资源需求可能发生变化的问题,提出了一个具有自主预测特性的动态无人机边缘计算物联网网络框架,该框架不仅可以根据地面用户的实时需求支持无人机动态进入和退出,还可以根据地面用户的分布更新其位置部署。Shao 等^[44]提出了一种以节点选择和路径优化为重点的多任务协同计算机制,设计了一种基于多智能体近端策略优化的节点选择与路径优化算法,以优化传输时延的多任务协同计算的目标,有效平衡了网络节点的负载。Saif 等^[45]针对洪水、地震等灾害情况,设计了一种多无人机辅助的 B5G 网络以提高灾区的通信恢复能力,提出了一种考虑地面基站的能量水平、邻居节点数量和无人机的最大发射功率的聚类算法,通过聚类算法选择出簇头,并将无人机部署到簇头的位置以恢复通信,该方法能快速恢复灾区的网络通信服务。

以上设计或算法通过把无人机移动节点作为通信中继节点或移动服务器节点,有效地减少了网络节点的负载,减少了数据传输的时延,能够适应复杂且动态变化的网络环境,提升了网络性能。此外,无人机节点可以扩大网络信号的负载范围,对于地震、洪水等自然灾害情况可以提供紧急的通信服务。但这些算法都是在仿真平台以仿真的形式运行,并且算法通常考虑用户的分布及移动性、基站节点的无线信号能耗及频谱利用率,没有考虑网络节点的异构性和网络状态的多个属性,且高复杂度的算法难以在实际设备中部署。此外,在传统网络架构下对网络状态信息的测量开销巨大,不利于算法的高效运行和数据的高速传输。

2.3 存储节点的选择

Rashid 等^[18]使用 FUSE 接口在 Linux Ext4 文件系统上设计并部署了一种边缘分布式存储系统,该系统针对存储系统可用性不一致以及边缘设备的配置问题,引入了博弈论资源激励框架来动态分配并调节网络的 QoS,同时建立合作存储的共享方案来提高系统的可用性。但该系统在设计时并没有考虑移动边缘的复杂场景,同时在选择存储节点时仅考虑剩余存储空间单一因素进行节点博弈,存储节点的负载容易失衡。Sonbol 等^[46]设计了一种专为网络边缘设计的去中心化存储系统 EdgeKV,凭借位置透明和基于接口的设计,可以随异构边缘节点系统扩展。EdgeKV 在本地和全局数据访问方面均优于云存储,平均响应时间和吞吐量都有提升。但该系统没有考虑无线网络场景下移动节点对边缘存储性能的影响。Qiao 等^[47]提出了一种基于强化学习的边缘分布式可信存储架构,用于解决智能交通系统(Intelligent Transportation System, ITS)中产生的异构交通网络协调问题,使用基于可信度和流行度的强化学习动态存储数据,改善资源调度和存储空间分配。该方法有效地解决了 ITS 的数据传输效率问题,但是没有考虑到存储节点的异构性以及数据选择存储节点进行分布式存储时带来的负载不均衡问题。Kontodimas 等^[48]开发了一种将数据分配并存储在边缘或云资源商的机制,并利用纠错码技术提高了系统的可用性和寿命。该系统主要通过应用程序与边缘存储节点的位置来进行数据的分配和存储的决策,有效地降低了输出存储的时延。但该方法忽视了网络状态以及存储节点的负载状态信息,仅使用位置信息来选择存储节点及分配传输的数据量会导致存储节点之间的负载不平衡。Li 等^[49]针对边缘网络的存储和管理问题,提出了一种利用数据流行度和数据相似性来解决边缘分布式存储系统中数据访问性能和存储效率问题的算法。该算法使用开源分布式存储系统 Cassandra 进行原型设计,有效缩短了服务请求响应时间。但这种利用数据类型来选择存储节点的方法没有综合考虑存储节点的负载以及网络状态,不利于网络链路及存储节点之间的负载均衡。

文献^[50]设计了一种基于 SDN 和多属性决策的边缘分布式存储节点选择算法,考虑了存储节点的多个因素,并设计了一种存储节点状态信息自上报机制,但该系统在有线连接的网络拓扑下开展工作,没有考虑网络节点的移动性。本文提出了一种基于 SDN 和无人机辅助的无线边缘存储系统性能优化的方法,联合网络负载和存储节点的选择优化问题来提高系统的存储性能。利用 SDN 技术测量出网络状态信息,并设计了一种网络节点状态自上报机制。对于重负载网络节点和存储节点的选择,提出了一种基于多属性决策模型,综合考虑网络状态和节点自身负载状态的节点选择算法,选择出重负载网络节点和合适的存储节点。在数据传输过程中,无人机悬停在重负载节点上方以平衡网络链路的流量负载,并不断根据重负载节点编号更新自身位置,实现动态负载均衡,提升网络的性能。通过灵活部署无人机,可以迅速覆盖受灾区域,恢复通信服务,为救援和信息传递提供保障。

3 问题描述

本章将对 UAV 辅助的边缘存储网络负载均衡问题和存

储节点的选择问题进行建模,包括网络节点的网络状态评价、重负载节点的选择以及存储节点的选择。本文基于多属性决策模型,设计了一种网络节点负载和存储节点性能的评分模型,用于衡量网络各个节点负载的大小以及存储节点的优劣情况。客户端在进行数据的分布式存储时会考虑各个存储节点的性能评分以分配不同的待传输数据量,同时在数据的传输过程中,无人机移动节点会移动到负载最重的网络节点上方以分担链路的流量负载。

为了通过无人机移动节点实现节点流量负载的有效平衡,本文提出了一种节点选择策略。此策略首要关注的是负载最重的交换机节点的识别。在选择这些关键节点时,综合考虑了网络状态信息,特别是节点间连接链路的剩余带宽、时延和丢包率等关键属性。此外,充分考虑了网络节点的异构性,即在实际网络拓扑中,不同节点的硬件性能存在显著差异。具体来说,在相同的流量负载下,硬件性能较弱的节点往往在处理和转发数据方面表现不佳,而性能优越的交换机节点则能够更顺畅地处理并转发数据包。通过这一综合考量,能够更加精确地定位并优化网络中的瓶颈节点,从而提升整个网络的流量处理效率和稳定性。

将网络拓扑建模为无向图,用 $G=(V, E, W)$ 表示。其中, V 表示网络节点的集合,用 n 表示网络节点的数量, E 表示节点之间链路的集合, W 表示链路参数的集合, e 表示网络中的某一条链路, $e \in E$ 。

对于节点的网络状态的评价,本文考虑节点的多个网络状态信息,包括链路的剩余带宽、时延和丢包率属性,通过不同的加权因子加权计算节点的网络状态得分。在计算节点的网络状态得分前,需要将节点的网络状态属性进行归一化处理,归一化的计算式如下:

$$attr_norm = \frac{attr - \min_attr}{\max_attr - \min_attr} \quad (1)$$

其中, $attr_norm$ 为归一化的属性值, $attr$ 为某一具体的属性值(如链路的剩余带宽或时延), \max_attr 和 \min_attr 分别为网络链路集合 E 中该属性的最大值和最小值。

将节点与其邻居节点的链路的所有属性值进行归一化处理后计算该节点的网络状态得分,节点的网络状态评价如式(2)所示:

$$link_score = w_1 \cdot bandwidth_norm \cdot 100 - w_2 \cdot delay_norm \cdot 100 - w_3 \cdot loss_norm \cdot 100 - 10 \quad (2)$$

其中, w_1 , w_2 和 w_3 分别为链路剩余带宽、时延和丢包率属性的权重因子。在本文的实际部署中,数据平面的交换机与控制器采用 In-band 的带内方式进行连接,因此数据平面中的业务数据和控制命令数据是在同一物理链路上进行传输的,导致链路丢包率的准确度下降,与控制器直接连接的交换机的丢包率较大。经过多次实验测试,权重矩阵 $[w_1, w_2, w_3]$ 取值为 $[0.7, 0.2, 0.1]$ 。式(2)中剩余带宽属性为正指标,时延和丢包率属性为负指标, w_1 为正系数, w_2 和 w_3 为负系数。每项属性乘以加权因子后再乘以 100,因此每个属性的得分值为 $[-100, +100]$ 。再对该条链路得分进行减 10 的运算,防止有较多的邻居节点的节点少数链路负载过载,但存在其与邻居节点链路的网络状态得分过高的情况。数值 10 是经

多次实验测试选取的最优值。某些节点可能与邻居节点有较多的链路,少数链路负载过重,但其他链路负载很轻,导致该节点与其邻居节点链路的网络状态得分过高。

对节点与其邻居节点的所有链路计算网络状态得分,则可以计算出该节点的网络状态总得分,再把该总分除以邻居节点的数量取平均,得到该节点的网络状态得分,如式(3)所示:

$$network_score_i = \frac{\sum_{j=1}^n link_score_{ij}}{n'} \quad (3)$$

其中, n' 为节点 i 的邻居节点数量, $link_score_{ij}$ 为节点 i 与其邻居节点的第 j 条链路的网络状态得分, $network_score_i$ 为节点 i 的网络状态得分。

除了节点的网络状态得分,控制器还可以通过本文设计的节点状态自上报机制获取到网络节点的 CPU 利用率和内存利用率数据。在本文设计中,每项属性的值越大代表节点的性能越好,而对实际部署来说,设备的 CPU 利用率和内存利用率越小越好,因此需要对节点的 CPU 利用率和内存利用率取倒数。然后把节点的网络状态得分、CPU 利用率的倒数和内存利用率的倒数加入到矩阵中构成多属性决策模型的属性矩阵。属性矩阵如式(4)所示:

$$\begin{pmatrix} N_1 & \frac{1}{C_1} & \frac{1}{R_1} \\ N_2 & \frac{1}{C_2} & \frac{1}{R_2} \\ \vdots & \vdots & \vdots \\ N_n & \frac{1}{C_n} & \frac{1}{R_n} \end{pmatrix} \quad (4)$$

其中, N 是网络状态的得分, C 为 CPU 使用率, R 是内存使用率。

运行多属性决策算法后便可以得到每个节点的最终得分列表,如式(5)所示:

$$[score_1, score_2, \dots, score_n] = H(G) \quad (5)$$

其中, H 为多属性决策算法函数; $score$ 为节点的分数,即通过多属性决策算法计算的节点得分,该得分代表了节点的负载情况,得分越低,节点的负载越重。

排序该得分列表即可得到得分最低的节点编号,如式(6)所示:

$$node = \underset{score}{\operatorname{argmin}}([score_1, score_2, \dots, score_n]) \quad (6)$$

其中, $node$ 为负载最重的节点。在选择到负载最重的节点后,无人机移动节点飞行到该节点的上方以实现流量分流,平衡节点的负载,提升网络的性能。本文的优化目标之一是平衡节点的负载,通过无人机移动节点减少负载最重的节点的负载。优化目标如式(7)所示:

$$\operatorname{maximize} F(G) = \min([score_1, score_2, \dots, score_n]) \quad (7)$$

其中, G 表示整个网络拓扑的图数据; $score$ 为节点的分数,代表了节点的负载情况; F 为重负载节点选择算法函数。

存储节点的选择与上述重负载网络节点的选择类似,同样基于多属性决策模型,综合考虑网络状态和存储节点的自身负载状态信息来选择存储节点,并为每个已选择的存储节点分配不同的数据量,存储节点的性能越好,分配的数据量

越多。本文对存储节点的选择考虑了 5 个因素,分别为节点剩余存储空间 V 、路径的网络状态得分 S 、磁盘 IO 负载 L 、CPU 使用率 C 和内存使用率 R 。存储节点选择的建模问题与上述重负载节点选择的模型类似,这里不再做详细的介绍。通过多属性决策模型后,可以得到各个存储节点的分数,将该分数进行归一化处理,然后乘以客户端待传输的文件总大小,便可以得到各个存储节点分配到的数据量,不同性能的存储节点会分配到不同的待存储的数据量。本文的另一个优化目标如式(8)所示:

$$S_i = F \cdot score_i' \quad (8)$$

其中, F 为客户端传输文件的大小, $score_i'$ 为多属性决策算法计算的存储节点 i 的得分, S_i 为 i 号存储节点分配的待存储的数据量。

4 无线边缘分布式存储系统架构设计与算法实现

4.1 系统总体架构

本文方法的总体架构由数据平面、控制平面、数据处理平面、知识平面和边缘设备平面组成,如图 1 所示。

1)数据平面。该平面由 SDN 交换机组成,负责数据包的解析和转发的工作。将每个交换机作为一个 AP,以供移动终端设备的无线连接,同时 AP 之间通过无线 Mesh 组网的方式连接,每个 AP 都可以自由移动。此外,交换机会通过 Openflow 协议与控制平面通信回复控制器的请求,且会周期性地上报自身状态信息给控制器,包括 CPU 的使用率和内存使用率。

2)控制平面。SDN 架构将控制平面和数据平面分离,从而实现网络的集中控制和编程。控制器周期性地通过南向接口向数据平面发送请求,获取全局网络的状态信息,包括网络拓扑、链路的带宽、链路的时延和丢包率等。此外,控制器还负责交换机的流表项的下发以及 Packet-in 事件的处理。本文系统中控制平面的功能主要是:网络状态的测量、网络节点和存储节点的状态信息的测量、网络拓扑的图形化显示。

3)数据处理平面。控制平面获取到交换机的端口信息和节点的状态信息后将其交给数据处理平面,数据处理平面根据原始的端口的数据包信息计算得到网络链路的信息,如链路的剩余带宽、时延和丢包率等,再将全局的网络状态信息和节点状态信息进行保存。

4)知识平面。知识平面获取数据处理平面存储的网络状态信息和节点状态信息,然后通过多属性决策算法综合考虑多个属性计算得到重负载节点的编号,通过移动节点飞行到该重负载节点上方即可实现流量的分流。此外,Dijkstra 算法在计算最优路径时设置不同的权重因子,综合考虑链路的剩余带宽、时延和丢包率信息,最终得到最优传输路径。

5)边缘设备平面。该平面由移动终端设备和边缘存储的服务器组成。移动终端设备作为客户端,通过 Samba 协议向服务器存储文件或者从服务器中下载文件到本地。服务器则会通过 Packet-in 机制周期性地上报自己的状态信息给控制器,包括 IO 负载大小、CPU 使用率、内存使用率和磁盘空闲率。本文选择树莓派 4B 开发板作为存储节点。

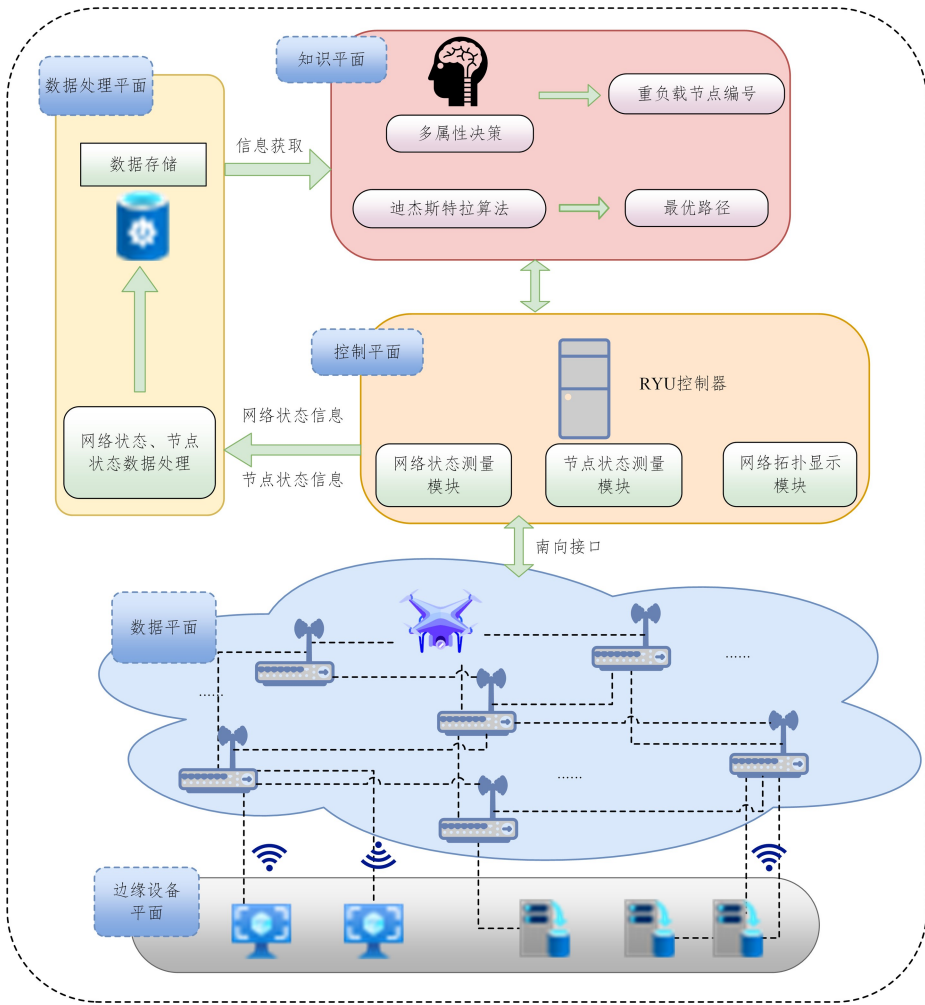


图1 系统总结架构

Fig.1 System summary architecture

4.2 网络状态信息的测量

本文的网络拓扑包括7台交换机、3个NAS服务器和1个客户端。选择6台交换机放在地面作为静止的节点,再选择1台交换机作为移动节点。受实验场所和条件的限制,在初步探索阶段,为了模拟无人机的动态行为,暂时采用了人工手持交换机的方式进行移动,以此作为无人机移动的一种近似模拟。这种模拟方法虽然简单,但为后续的深入研究提供了实验基础,并为进一步探索无人机搭载网络设备的实际运行状况提供了参考。本文的网络拓扑如图2所示,其中无人机节点会自动与附近的AP节点以无线Mesh组网的方式进行连接。

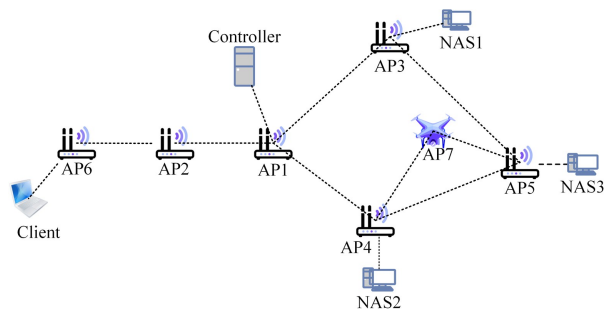


图2 网络拓扑

Fig.2 Network topology

图2中,AP1至AP6为地面的静止节点,AP7作为移动节点进行移动分流,控制器与AP1节点连接,客户端与AP6节点连接。AP3,AP4和AP5节点分别连接了一个NAS服务器,用于分布式存储客户端发送的文件数据,在实际部署中采用外接U盘的树莓派4B作为NAS服务器。

本文考虑的网络状态信息包括链路的剩余带宽、链路的时延和链路的丢包率。

控制器以 T 秒为周期发送统计交换机端口信息的请求,交换机收到该请求后,会将其所有端口的发送字节数 tx_b 、接收的字节数 rx_b 、发送的数据包数 tx_p 、接收的数据包数 rx_p 和该时刻的时间戳 t 作为回复发送给控制器,控制器在获取到所有交换机的端口统计信息后即可计算出链路的剩余带宽和丢包率属性。

交换机某个端口的已用带宽为周期 T 内该端口前后两次处理数据字节数的差。链路 $e_{(i,j)}$ 的已用带宽取 i 号交换机链路端口与 j 号交换机的链路端口的已用带宽的最大值。链路的剩余带宽为链路的最大带宽减去该链路的已用带宽。交换机端口的已用带宽的计算式如式(9)所示,链路的已用带宽和剩余带宽的计算式分别如式(10)和式(11)所示:

$$bw_{sw} = \frac{(tx_b_2 + rx_b_2) + (tx_b_1 + rx_b_1)}{T} \quad (9)$$

$$bw_used_{(ij)} = \max(bw_sw_{num}), num = i, j \quad (10)$$

$$bw_remain_{(ij)} = bw_{capmax} - bw_used_{(ij)}, (i, j) \in (1, 2, \dots, n), i \neq j \quad (11)$$

其中, bw_sw 为链路中某一端交换机的端口的已用带宽, tx_b_2 和 tx_b_1 分别为端口先后 2 次发送数据的字节数, T 为交换机上报端口统计信息的周期, $bw_used_{(ij)}$ 代表链路 $e_{(i,j)}$ 的已用带宽, bw_sw_{num} 为链路中编号为 num 的交换机的端口的已用带宽, $bw_remain_{(ij)}$ 为链路 $e_{(i,j)}$ 的剩余带宽, bw_{capmax} 为链路的最大带宽。

链路 $e_{(ij)}$ 的丢包率 $loss_{(ij)}$ 表示从 i 号交换机到 j 号交换机和从 j 号交换机到 i 号交换机两个方向的丢包率的最大值, 其计算式如下:

$$loss_{(ij)} = \max\left(1 - \frac{rx_p_j}{tx_p_i}, 1 - \frac{rx_p_i}{tx_p_j}\right) \quad (12)$$

链路的时延通过 echo-request 报文和 LLDP 报文获取到的时间数据来计算。控制器向交换机发送带时间戳的 echo-request 请求报文, 在收到交换机返回的 echo-reply 报文后记录该时刻的时间, 用该时间减去发送 echo-request 请求报文时的时间戳得到控制器到交换机 i 和交换机 j 的往返的时延 t_echo_{swi} 。控制器也会向交换机发送 LLDP 报文, 在接收到 LLDP 报文时记录该时刻的时间, 该时间减去发送报文时的时间即为控制器到链路 $e_{(ij)}$ 再到控制器这个环路的时延 t_lldp_i 和 t_lldp_j 。因此, 链路 $e_{(ij)}$ 的时延计算式如式(13)所示:

$$delay_{ij} = \frac{(t_lldp_i + t_lldp_j - t_echo_{swi} - t_echo_{swj})}{2} \quad (13)$$

4.3 节点状态信息的获取

文献[50]巧妙地利用 Packet-in 事件把存储节点的自身状态信息上报给控制器。本文对重负载网络节点的选择考虑节点的异构性, 与文献[50]类似, 把存储节点状态信息自上报机制部署到交换机上。数据平面的每个交换机获取自身的 CPU 利用率和内存利用率信息, 然后把这些数据封装在 TCP 包中, 并向一个不存在的目的 IP 地址发送数据包以触发 Packet-in 事件, 把数据包传输给控制器。控制器在接收到该数据包后解析其内容即可获得交换机的状态信息。

4.4 无线资源的资源分配

在本文设计的网络结构中, 控制器的 2.4G 无线信号主要用于控制器与数据平面的交换机之间的连接。数据平面的数据传输通过 5G 频段的 Mesh 网络进行传输。边缘设备平面的设备可以连接数据平面的交换机的 2.4G 或 5G 信号的 WIFI。系统无线资源的分配如图 3 所示。

将 2.4G 和 5G 信号用于不同功能的设计具有以下优点:

1) 优化资源分配。2.4G 网络专注于状态监控和控制信号传输, 避免了与大量数据传输的竞争, 保证了控制信号的稳定性和及时性, 而 5G 网络主要用于交换机之间的数据传输。这种通信信道分离的方法使控制链路和数据链路分离, 可以使流表下发等控制命令快速下发, 从而及时地安装流表以及获取交换机的状态信息。

2) 高效的数据传输。5G 网络提供高速数据传输的通道, 减少了数据传输的延迟, 提升了整体网络性能。

3) 减少无线信号的干扰。将 2.4G 和 5G 信号用于不同场景下的数据传输, 减少无线信号的相互干扰。此外, 每个 AP 的 2.4G 频段的 WIFI 信号设置在不同的信道, 可减少同频信号的干扰, 提高各自的传输效率和可靠性。

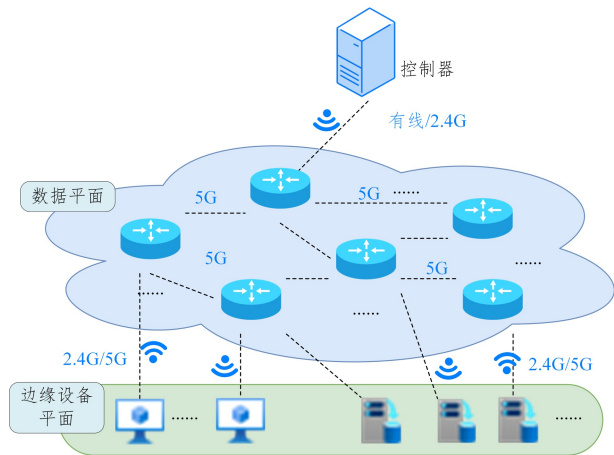


图 3 系统无线资源的分配

Fig. 3 Allocation of system wireless resources

4.5 重负载节点选择算法

本文多属性决策模型考虑的 3 个属性因子分别是节点的网络状的得分数值、节点的 CPU 使用率和内存使用率。其中, 节点的网络状态的得分是通过评价节点与该节点的邻居节点的链路情况得到的, 算法的具体流程如算法 1 所示。

算法 1 节点的网络状态评价算法

输入: 网络拓扑的图结构数据 graph、链路属性的加权因子 W_{bw} ,

W_{delay} , W_{loss}

输出: 节点的网络状态得分字典 network status

1. 初始化: 构建空字典 network status;
2. $max_bandwidth = \max(\text{获取边的带宽属性})$ for 边的属性字典 in graph 的边;
3. $max_delay = \max(\text{获取边的时延属性})$ for 边的属性字典 in graph 的边;
4. $max_loss = \max(\text{获取边的丢包率属性})$ for 边的属性字典 in graph 的边;
5. $min_bandwidth = \min(\text{获取边的带宽属性})$ for 边的属性字典 in graph 的边;
6. $min_delay = \min(\text{获取边的时延属性})$ for 边的属性字典 in graph 的边;
7. $min_loss = \min(\text{获取边的丢包率属性})$ for 边的属性字典 in graph 的边;
8. for 节点 in graph 的节点列表 do
9. 获取该节点的邻居节点列表;
10. for 节点 in 该节点的邻居节点列表 do
11. 获取节点与该邻居节点链路的剩余带宽 bandwidth、时延 delay 和丢包率 loss 属性;
12. $bandwidth_norm = \frac{(bandwidth - min_bandwidth)}{(max_bandwidth - min_bandwidth)}$;
13. $delay_norm = \frac{(delay - min_delay)}{(max_delay - min_delay)}$;
14. $loss_norm = \frac{(loss - min_loss)}{(max_loss - min_loss)}$;

15. 该条链路的得分 = $W_{bw} * bandwidth_norm * 100 - W_{delay} * delay_norm * 100 - W_{loss} * loss_norm * 100 - 10$;
16. 总分 += 该条链路的得分;
17. end
18. 在得分字典 network_status 中节点的最终得分 = 总分 ÷ 节点的邻居节点数量;
19. end

如算法 1 所示,将网络拓扑的图结构数据和权重因子作为算法的输入,输出为每个节点的网络状态得分。第 1 行初始化节点网络状态得分的字典;第 2—7 行获取网络链路的剩余带宽、时延和丢包率属性的最大值和最小值;第 8—18 行获取每个节点的网络状态的得分;第 12—14 行利用最大最小法对数据进行归一化处理;第 10—15 行计算节点与其每个邻居节点链路的网络状态得分,再对每条链路的得分求和得到总分;第 18 行是把总分除以邻居节点的数量取平均。其中,第 15 行中剩余带宽 $bandwidth_norm$ 为正指标,时延 $delay_norm$ 和丢包率 $loss_norm$ 为负指标,链路的得分越大代表该链路的网络状态越好。链路的网络状态得分如式(3)所示。

对于算法 1 的复杂度,由于控制器的内存是足够的,并且该算法所用到的内存空间并不大,因此本文只分析算法的时间复杂度。算法 1 的第 2—7 行分别是获取链路的剩余带宽、时延和丢包率属性的最大值和最小值,一共 6 个变量,每次获取时都需要循环遍历边的属性字典,因此该部分的时间复杂度为 $O(6n)$ 。第 8—19 行也是一个循环,并且在第 10 行中遍历每个节点的邻居节点。设节点的平均节点数为 m (m 为常数并小于 n),则该部分程序的时间复杂度为 $O(n \cdot m)$ 。算法 1 的时间复杂度为上述两部分程序的时间复杂度的和,因此算法 1 的时间复杂度为 $O((6+m) \cdot n)$ 。时间复杂度的计算通常只考虑阶数,因此算法 1 的时间复杂度可化简为 $O(n)$ 。

获取每个网络节点的网络状态得分后,则可把该值与节点的 CPU 利用率和内存利用率信息加入到属性矩阵中。多属性决策算法的主要计算流程为:首先对属性矩阵中的属性值进行归一化处理;其次根据加权因子进行加权处理,在处理后的属性矩阵中找到每个属性类型的正理想解和负理想解,计算每个节点的属性与正理想解和负理想解的欧几里得距离;然后计算每个节点的属性与正理想解的接近度,该值越大,节点的性能越好;最后排序所有节点的得分得到负载最重的节点。重负载节点选择算法如算法 2 所示,具体步骤如下。

步骤 1 构建属性矩阵 \mathbf{M} 并对其进行处理得到加权决策矩阵 \mathbf{Z} 。第 3—4 行构造属性矩阵 \mathbf{M} ,构造方式如式(14)所示,其中 N 是网络状态的得分, C 为 CPU 使用率, R 是内存使用率。第 6—9 行把属性矩阵模一化得到矩阵 \mathbf{M}' ,然后对标准化的矩阵 \mathbf{M}' 乘以加权矩阵得到决策矩阵 \mathbf{Z} 。模一化的计算式如式(15)所示,其中 f_{ij} 是矩阵 \mathbf{M} 中的元素, i 和 j 分别代表矩阵的行号和列号, n 为网络节点的个数。3 个决策因素对网络节点的负载影响的比重是不同的,网络状态比重相对较大,需选取合适的加权系数 W ,通过式(17)构造加权后决策矩阵 \mathbf{Z} 。式(16)中权重的值一般通过多次实验测试来选取。

$$\mathbf{M} = \begin{pmatrix} N_1 & \frac{1}{C_1} & \frac{1}{R_1} \\ N_2 & \frac{1}{C_2} & \frac{1}{R_2} \\ \vdots & \vdots & \vdots \\ N_n & \frac{1}{C_n} & \frac{1}{R_n} \end{pmatrix} \quad (14)$$

$$\mathbf{M}' = \frac{f_{ij}}{\sqrt{\sum_{i=1}^n f_{ij}^2}}, i=1,2,\dots,n; j=1,2,3 \quad (15)$$

$$\mathbf{W} = [W_N \quad W_C \quad W_R] \quad (16)$$

$$\mathbf{Z} = \mathbf{W}_j \times \mathbf{M}'_{ij}, i=1,2,\dots,n; j=1,2,3 \quad (17)$$

步骤 2 计算加权矩阵 \mathbf{M}' 的正负理想解,并计算出每个节点的属性到正负理想解的欧氏距离。第 10—11 行计算加权决策矩阵 \mathbf{Z} 的正理想解和负理想解,正负理想解的求法分别如式(18)和式(19)所示:

$$\mathbf{Z}^+ = (Z_1^+, Z_2^+, Z_3^+) = \max\{Z_{ij} \mid j=1,2,3\}, i=1,2,\dots,n \quad (18)$$

$$\mathbf{Z}^- = (Z_1^-, Z_2^-, Z_3^-) = \min\{Z_{ij} \mid j=1,2,3\}, i=1,2,\dots,n \quad (19)$$

第 13—15 行计算每个网络节点到正理想解和负理想解的欧氏距离 D^+ 和 D^- ,计算式分别如式(20)和式(21)所示:

$$D_i^+ = \sqrt{\sum_{j=1}^n (Z_{ij} - Z_j^+)^2}, i=1,2,\dots,n \quad (20)$$

$$D_i^- = \sqrt{\sum_{j=1}^n (Z_{ij} - Z_j^-)^2}, i=1,2,\dots,n \quad (21)$$

步骤 3 计算每个网络节点的属性与正理想解的接近度 C_i^+ 作为该节点的最终得分,该值越大代表对应的网络节点性能越好。通过排序每个节点的最终得分可选择出负载最重的节点。计算每个网络节点的属性与正理想解的接近度的计算式如式(22)所示:

$$C_i^+ = \frac{D_i^-}{D_i^+ + D_i^-}, i=1,2,\dots,n \quad (22)$$

算法 2 重负载节点选择算法

输入:网络拓扑的图数据 graph,节点的网络状态得分 N 、节点的 CPU 使用率 V 和内存使用率 R ,加权系数 W_N, W_C, W_R

输出:负载最重的网络节点编号

1. 初始化,构建节点负载排序列表 l ;
2. 初始化,构建属性矩阵 \mathbf{M} ;
3. for 节点 in 节点列表 do
4. 属性矩阵分别加入每个节点的 $N, \frac{1}{C}, \frac{1}{R}$;
5. end
6. for 行号 in range(属性矩阵的列的维度) do
7. 列中的每个属性 $M'_{ij} = \frac{M_{ij}}{\sqrt{\sum M_{ij}^2}}$
8. $M'_{ij} = M'_{ij} * W_i$
9. end
10. 正理想解列表 \mathbf{Z}^+ = 每一列中的最大值 $\max(\mathbf{M}')$;
11. 负理想解列表 \mathbf{Z}^- = 每一列中的最小值 $\min(\mathbf{M}')$;
12. 创建空的一维矩阵 scores;
13. for 节点序号 in range(属性矩阵的行的维度) do

14. $D^+ = \sqrt{\sum_1^n (Z_{ij} - Z_i^+)^2}$;
15. $D^- = \sqrt{\sum_1^n (Z_{ij} - Z_i^-)^2}$;
16. 节点的得分 $C_i^+ = \frac{D_i^-}{D_i^+ + D_i^-}$;
17. 在 scores 列表中加入该节点的得分 C_i^+ ;
18. end
19. 重负载节点编号 num=scores 列表中排序找到得分最小的节点的编号。

在算法 2 中,第 2—4 行遍历每个节点并把节点的相关属性放入属性矩阵中,时间复杂度为 $O(n)$ 。第 6—9 行同样遍历每个节点,对属性矩阵的元素进行归一化操作,其时间复杂度为 $O(n)$ 。第 13—18 行也遍历每个节点,计算每个节点的最终得分,其时间复杂度为 $O(n)$ 。只考虑阶数的情况下,算法 2 的时间复杂度为 $O(n)$ 。

在选择完重负载节点后,通过无人机移动节点的分流作用,减少重负载节点的流量负载,实现动态负载均衡。无人机的引入不仅增强了系统的动态适应能力,还特别适用于灾害场景下的应急通信。通过无人机的灵活部署,可以迅速覆盖受灾区域,恢复通信服务,为救援和信息传递提供保障。

4.6 存储节点选择算法

与边缘网络中重负载网络节点的选择方法类似,在考虑边缘存储的边缘节点选择时,同样利用多属性决策算法计算各个存储节点的得分,根据该得分计算各个存储节点需分配的带存储的数据量。控制器在获得这些存储节点的负载信息后,使用多属性决策方法选择出合适的存储节点的位置,并为其分配不同的待存储的数据量,设计的算法与重负载节点选择的算法 2 类似,这里不再具体叙述。

5 实验及结果分析

5.1 实验环境及硬件设计

本系统中包含 1 台控制器、3 台存储节点、7 台交换机及 1 台移动节点。系统的软件环境如表 1 所列。

表 1 系统软件版本

Table 1 System software version

软件类型	软件版本
OpenWrt	23.10.1
Open vSwitch	2.15.0
OpenFlow	1.3
Samba	1.0
控制器操作系统	Ubuntu20.04 LTS
存储节点操作系统	Raspbian
RYU	4.8

选择畅网微控公司的 N5105 型号的软路由作为 RYU 控制器。该软路由含有 6 个 2.5G 有线网口、Intel 四核处理器和 DDR4 双内存,且具有无线 WiFi 功能,体积小巧,适合边缘存储系统设计。

选择小米 AX6000 路由器作为 SDN 交换机。该路由器的 CPU 频率高达 2GHz, RAM 大小为 512MB, Flash 大小为 128MB, 体积小巧。

SDN 交换机在 OSI 网络五层协议中的数据链路层工作,用于局域网中的数据转发。路由器通常用于跨越域的网络层

的数据转发,但路由器也有数据转发的功能,可以把路由器配置为交换机的功能。把路由器配置为本文的无线 SDN 交换机功能的步骤如下。

步骤 1 编译 OpenWrt 系统固件。从 Github 下载 LEDE 版的 OpenWrt 系统的源码到 Ubuntu 系统中,下载系统所需的额外软件包的源码,并在编译选项中选择把软件编译到固件。此外,在编译选项中选择与设备匹配的相关的驱动和通信协议编译到固件中。下载编译环境所需软件包然后开始编译。

步骤 2 向路由器烧录 Uboot 固件和 OpenWrt 系统固件。下载并烧录路由器型号对应的 Uboot 固件,使路由器可以进入 Uboot 模型,烧录任意的 OpenWrt 固件。进入 Uboot 模式后烧录步骤 1 编译的系统固件。

步骤 3 设置 Open vSwitch 相关的功能。通过 SSH 远程连接路由器,对 OVS 进行 SDN 相关的设置,如添加网桥、设置控制器的 IP 和端口号、设置交换机的 dpid 号等,把路由器配置成 SDN 交换机的功能。

步骤 4 设置无线 Mesh 组网的功能。在 2.4G 或 5G 无线频段中新建 802.11s 协议的 Mesh 节点网络,并把该 Mesh 网络对应的网卡加入到 OVS 的网桥中。

本文选择树莓派 4B 开发板作为存储节点,其 CPU 频率高达 1.5GHz, RAM 类型为 DDR4, RAM 大小为 4GB, Flash 大小由外置的 SD 卡决定,可根据不同场景自由配置 Flash 大小。

5.2 实验结果及分析

本文分别使用了 2GB、1.5GB 和 1000MB 这 3 种不同大小的文件进行存储测试。首先在传统边缘分布式存储系统 (Traditional Edge Distributed Storage, TEDS) 和不带移动节点的边缘分布式存储系统 (Edge Distributed Wireless Storage, EDWS) 以及本文设计的带移动节点的边缘分布式存储系统 (Mobile Edge Distributed Wireless Storage, MEDWS) 中进行测试,对比存储节点选择的优化作用。然后在 EDWS 和 MEDWS 中测试存储文件的传输时间。最后将文献[34]的考虑基站的主机关联数量及频谱利用率的无人机辅助的方法 (Hybrid-aware Mobile Edge Distributed Wireless Storage, HMEDWS) 与 MEDWS 进行对比。本实验在实物上进行部署,无法随时更改 SDN 交换机的无线发射功率,实验中的 SDN 交换机的无线信号功率都是相同的,因此本实验中 UMEDWS 只考虑基站的用户关联数量。此外,在指定的关键链路上增加随机的负载,然后在 EDWS 和 MEDWS 中测试存储文件的传输时间。通过不同条件下的文件传输时间,对比移动节点的负载均衡作用。此外还利用 Python 的 Qt 工具包设计了网络拓扑的图形化显示界面,可直观地显示出网络拓扑的结构和变化。

每个交换机都有 2.4G 和 5G 的 WIFI 信号供移动终端设备连接,首先在 2.4G 和 5G 的信号下分别在 TEDS 和 EDWS 以及 MEDWS 中存储一个 1000MB 大小的文件,并随机选择一个树莓派存储节点不断进行复制文件和删除文件的磁盘读写操作,以测试不同系统以及不同的无线频段下的文件传输时间。文件的传输时间曲线如图 4 所示。

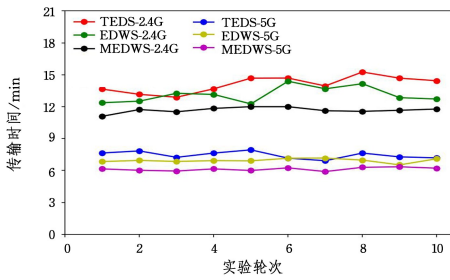


图4 文件在不同系统下的传输时间

Fig. 4 File transfer time in different systems

从中可以看出,在 2.4G 和 5G 无线信号下,文件在 EDWS 中的传输时间基本小于 TDES,而在 MEDWS 中的传输时间均小于 EDWS 和 TDES。在 TEDS 中,存储节点的选择仅考虑磁盘剩余存储空间大小,而 EDWS 还考虑了文件在不同系统下的传输时间存储节点的 IO 负载、CPU 利用率和内存利用率等信息。当存储节点的剩余存储空间较大,但正在进行磁盘的读写操作,或正在运行复杂度高的算法时,CPU 利用率和内存利用率很大,此时会对节点的存储性能造成很大影响。因此,EDWS 的存储效率比 TEDS 高,传输时间更短。TEDS 和 EDWS 的网络拓扑都是静态的,没有考虑节点的移动性对系统性能的影响。MEDWS 的优势在于移动节点的分流作用,当网络中某条链路负载过重时,移动节点会飞行到重负载节点上方进行分流,平衡链路的流量负载,因此数据传输的效率会更高,传输时间明显缩短。

分别在 2.4G 和 5G 的信号下分布式存储 2GB 大小的文件到 3 个存储节点中,对比 2.4G 和 5G 信号的传输效率。两种信号下的传输时间曲线如图 5 所示。系统的网络拓扑如图 6 所示,其中 7 号节点为无人机移动节点,无人机节点只存在于本文设计的 MEDWS 中,对比实验的 EDWS 中不存在无人机节点。

从图 5 中可以看出,5G 信号下的传输时间明显短于 2.4G 信号。5G 频段的信号传输得更快,但信号衰弱得也快,容易受到空间障碍物的影响,从而导致传输距离受限。2.4G 频段的信号传输速度较慢,但穿墙能力较强,传输的距离较远。5G 和 2.4G 频段的无线信号各有优劣,在实际部署中可以根据需求选择对应的无线信号连接。在本文中,客户端和存储节点都距离所连接的交换机较近,因此在 5G 频段同时用于 Mesh 网络的情况下,5G 信号的传输速度依然快于 2.4G。

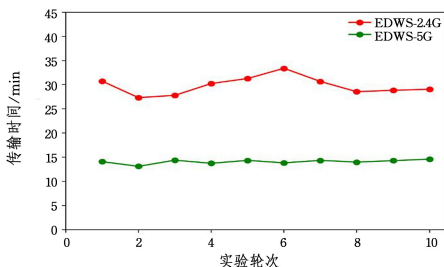


图5 文件在 EDWS 中 2.4G 和 5G 信号下的传输时间

Fig. 5 Transmission time of the file under 2.4G and 5G signals in EDWS

在 2.4G 无线信号下分布式存储 1000 MB 大小的文件到 3 个存储节点,分别在 EDWS 和 MEDWS 中测试文件的传输

时间。传输时间的曲线如图 7 所示。然后在 AP2 与 AP1 的链路上增加负载,AP1 无线连接一台树莓派作为服务器,AP2 连接树莓派作为客户端并不断随机向服务器发送流量数据,流量大小在 10 kB,200 kB,600 kB,1 MB,2 MB 和 5 MB 中随机选择并持续一分钟。在增加负载的情况下,2.4G 信号下的传输时间如图 8 所示。

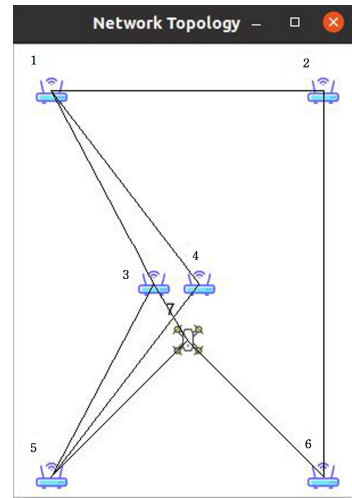


图6 网络拓扑的图形化显示

Fig. 6 Graphical display of the network topology

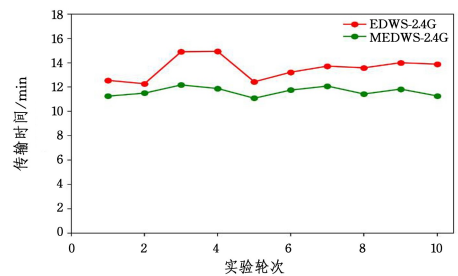


图7 2.4G 信号下 EDWS 和 MEDWS 的文件传输时间

Fig. 7 File transfer time of EDWS and MEDWS under 2.4G signal

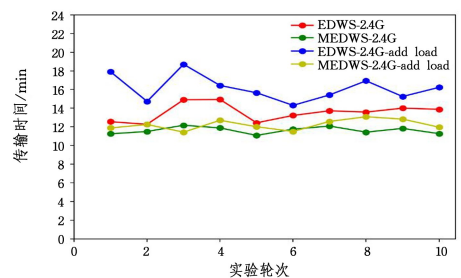


图8 2.4G 信号下增加负载时 EDWS 和 MEDWS 的文件传输时间(电子版为彩图)

Fig. 8 File transfer times of EDWS and MEDWS with increased load under 2.4G signal

从图 7 可以看出,在 2.4G 信号下,MEDWS 的传输时间明显少于 EDWS。在传输文件过程中,部分网络链路难免会出现负载过重的情况,数据包丢包率大,导致传输速率降低。通过无人机的移动分流,可以及时减轻重负载节点的负载,把部分数据流分流到无人机移动节点中,平衡节点的负载,避免出现网络拥塞的情况。

从图 8 可以看出,在 2.4G 信号下,在随机增加 AP2 与 AP1 之间的链路负载的情况下,MEDWS 的传输速率受到的影响很小(黄色曲线),传输时间只略微增加,而 EDWS 的传输时间显著增加(蓝色曲线),并且传输速率不稳定,传输时间波动较大。在增加负载的情况下,EDWS 在传输文件时会出现部分链路负载过重的情况,并且无法平衡网络中的链路负载,导致数据传输速率低下。由于增加的负载是在指定范围随机选择的,因此 EDWS 的传输时间波动较大。MEDWS 的移动节点具有分流的作用,在链路负载过重的情况下可以及时飞行到重负载节点上方进行分流,因此在增加链路负载的情况下 MEDWS 的传输时间只略微增加。

保持上述实验设置,在 2.4G 信号下,分别在 HMEDWS 和 MEDWS 中测试文件的传输时间,文件传输的时间曲线如图 9 所示。

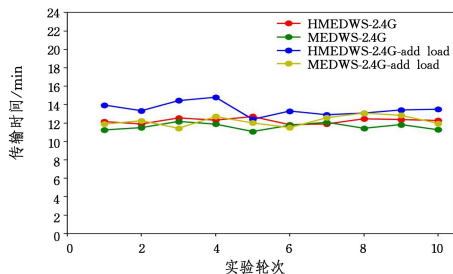


图 9 2.4G 信号下 HMEDWS 和 MEDWS 的文件传输时间

Fig. 9 File transfer time of HMEDWS and MEDWS under 2.4G signal

从图 9 可以看出,在 2.4G 信号下,在不增加以及增加链路负载的情况下,文件在 MEDWS 的传输时间都要略短于 HMEDWS。HMEDWS 只考虑交换机所连接的用户数量,移动节点实时移动到连接终端数量最多的网络节点上方。而 MEDWS 考虑了网络状态信息及网络节点的自身负载情况。数据传输过程中,当某条链路流量负载过重时,并不一定是连接终端数量最多的交换机出现网络拥塞,也有可能是路由中某个节点数据转发出现拥塞。MEDWS 充分考虑了网络状态信息,包括链路的剩余带宽、时延和丢包率属性,同时也考虑了各个网络节点自身的状态,包括 CPU 利用率和内存利用率,网络节点可能是异构的,因此文件在 MEDWS 中的传输时间更短。

将客户端和 NAS 服务器与交换机连接的无线信号改为 5G 信号,文件大小改为 1.5GB,其他实验条件与上述 2.4G 信号下的实验条件相同,测试 EDWS 和 MEDWS 存储文件的传输时间。传输时间的曲线分别如图 10 和 11 所示。

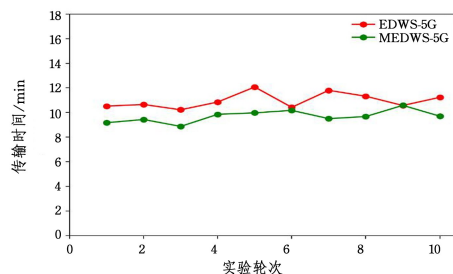


图 10 5G 信号下 EDWS 和 MEDWS 的文件传输时间

Fig. 10 File transfer time of EDWS and MEDWS under 5G signal

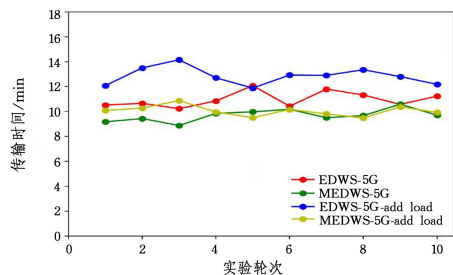


图 11 5G 信号下增加负载时 EDWS 和 MEDWS 的文件传输时间

Fig. 11 File transfer time of EDWS and MEDWS when the load is increased under 5G signal

从图 10 和图 11 可以看出,在 5G 信号下,MEDWS 的传输时间均短于 EDWS 的传输时间,在增加链路负载的情况下,EDWS 的传输时间波动较大,而 MEDWS 的传输时间略微增加,再次验证了移动节点的分流效果,实现了动态的负载均衡,提升了网络的性能。

保持上述实验设置,在 5G 的信号下,分别在 HMEDWS 和 MEDWS 中测试文件的传输时间,文件传输的时间曲线如图 12 所示。

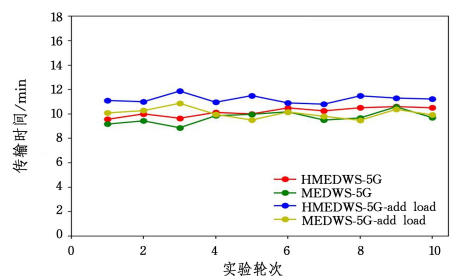


图 12 5G 信号下 HMEDWS 和 MEDWS 的文件传输时间

Fig. 12 File transfer time of HMEDWS and MEDWS under 5G signal

从图 12 可以看出,在 5G 信号下,在不增加以及增加链路负载的情况下,文件在 MEDWS 中的传输时间略小于 HMEDWS。在 5G 信号下的结果与 2.4G 信号下的结果类似,再次验证了移动节点的位置部署中网络状态和网络节点自身状态的重要性。网络链路负载更多体现在网络状态以及网络节点自身状态中,相比于 HMEDWS, MEDWS 利用了 SDN 技术测量网络状态信息及网络节点自身状态信息,通过多属性决策算法计算出重负载网络节点,而不只是考虑网络节点的终端连接数量。

结束语 随着边缘网络中移动终端设备越来越多,庞大的数据传输量给边缘存储网络的链路带来了巨大的负载压力。本文设计的 UAV 辅助的无线边缘分布式存储系统考虑网络的多个属性,包括网络状态信息和节点自身的状态信息,通过多属性决策模型选择出重负载节点和存储节点,之后通过移动节点进行链路流量的分流。与现有的边缘分布式存储网络相比,本系统有效平衡了网络链路的负载,存储文件的时间明显缩短,提升了网络的性能。此外,把 2.4G 和 5G 的无线信号用于不同场景下的数据传输,减少无线信道的干扰,优化了无线信号的资源分配。实际的网络拓扑会比本文的更加庞大,数据传输量更大,并且终端设备的位置是动态的。此

外,单控制器的设计也难以控制复杂的网络。由于实验条件的限制,本文暂时只对单无人机移动节点进行部署。未来考虑多控制器的部署以及多无人机移动节点的部署,以减少控制器的负载并平衡更加复杂多变的网络。

参考文献

- [1] HUDA N U, AHMED I, ADNAN M, et al. Experts and intelligent systems for smart homes' Transformation to Sustainable Smart Cities: A comprehensive review[J]. *Expert Systems with Applications*, 2024, 238: 122380.
- [2] MALIK H, ANEES T, FAHEEM M, et al. Blockchain and Internet of Things in smart cities and drug supply management: Open issues, opportunities, and future directions[J]. *Internet of Things*, 2023, 23: 100860.
- [3] CHAWLA D, MEHRA P S. A roadmap from classical cryptography to post-quantum resistant cryptography for 5G-enabled IoT: Challenges, opportunities and solutions [J]. *Internet of Things*, 2023, 24: 100950.
- [4] The big picture on the internet of things and the smart city: a review of what we know and what we need to know-ScienceDirect [EB/OL]. <https://www.sciencedirect.com/science/article/pii/S2542660522000609>.
- [5] HAZRA A, RANA P, ADHIKARI M, et al. Fog computing for next-generation Internet of Things: Fundamental, state-of-the-art and research challenges[J]. *Computer Science Review*, 2023, 48: 100549.
- [6] FENG C, HAN P, ZHANG X, et al. Computation offloading in mobile edge computing networks: A survey[J]. *Journal of Network and Computer Applications*, 2022, 202: 103366.
- [7] An application of meta-heuristic and nature-inspired algorithms for designing reliable networks based on the Internet of things: A systematic literature review-Gong-2023-International Journal of Communication Systems-Wiley Online Library[EB/OL]. <https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.5416>.
- [8] CHAI Y, ZENG X J, LIU Z. The future of wireless mesh network in next-generation communication: a perspective overview [J]. *Evolving Systems*, 2024, 15(4): 1635-1648.
- [9] NARWARIA A, MAZUMDAR A P. Software-Defined Wireless Sensor Network: A Comprehensive Survey[J]. *Journal of Network and Computer Applications*, 2023, 215: 103636.
- [10] Symmetry Dynamic Load Balancing Techniques in the IoT: A Review[EB/OL]. <https://www.mdpi.com/2073-8994/14/12/2554>.
- [11] SONG Z, QIN X, HAO Y, et al. A comprehensive survey on aerial mobile edge computing: Challenges, state-of-the-art, and future directions[J]. *Computer Communications*, 2022, 191: 233-256.
- [12] A Survey on UAV-Enabled Edge Computing: Resource Management Perspective [EB/OL]. <https://dl.acm.org/doi/full/10.1145/3626566>.
- [13] ALSUHLI G, FAHIM A, GADALLAH Y. A survey on the role of UAVs in the communication process: A technological perspective[J]. *Computer Communications*, 2022, 194: 86-123.
- [14] ADNAN M H, ZUKARNAIN Z A, AMODU O A. Fundamental design aspects of UAV-enabled MEC systems: A review on models, challenges, and future opportunities[J]. *Computer Science Review*, 2024, 51: 100615.
- [15] GU X, ZHANG G. A survey on UAV-assisted wireless communications: Recent advances and future trends [J]. *Computer Communications*, 2023, 208: 44-78.
- [16] A Storage Resource Collaboration Model Among Edge Nodes in Edge Federation Service [EB/OL]. <https://ieeexplore.ieee.org/abstract/document/9786648>.
- [17] ISYAKU B, BAKAR K, YUSUF N M, et al. Software defined wireless sensor load balancing routing for internet of things applications: Review of approaches[J]. *Heliyon*, Elsevier, 2024, 10(9): e29965.
- [18] RASHID M T, ZHANG D, WANG D. EdgeStore: Towards an Edge-Based Distributed Storage System for Emergency Response [C]//2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems, 2019: 2543-2550.
- [19] BHOWMIK C D, GAYEN T. Traffic aware dynamic load distribution in the Data Plane of SDN using Genetic Algorithm: A case study on NSF network[J]. *Pervasive and Mobile Computing*, 2023, 88: 101723.
- [20] SHA Z, HUO R, SUN C, et al. A Task-Oriented Hybrid Routing Approach based on Deep Deterministic Policy Gradient[J]. *Computer Communications*, 2023, 210: 183-193.
- [21] PEI X, SUN P, HU Y, et al. Enabling efficient routing for traffic engineering in SDN with Deep Reinforcement Learning [J]. *Computer Networks*, 2024, 241: 110220.
- [22] ZHENG H, GUO J, ZHOU Q, et al. Application of improved ant colony algorithm in load balancing of software-defined networks [J]. *The Journal of Supercomputing*, 2023, 79(7): 7438-7460.
- [23] YOUNUS M U, KHAN M K, BHATTI A R. Improving the Software-Defined Wireless Sensor Networks Routing Performance Using Reinforcement Learning [J]. *IEEE Internet of Things Journal*, 2022, 9(5): 3495-3508.
- [24] RIYAZ B M, ALI F, ALANSARI Z, et al. Artificial Intelligence Based Reliable Load Balancing Framework in Software-Defined Networks[J]. *Computers, Materials and Continua*, 2021, 70: 251-256.
- [25] ILBPS: An Integrated Optimization Approach Based on Adaptive Load-Balancing and Heuristic Path Selection in SDN[EB/OL]. <https://ieeexplore.ieee.org/abstract/document/10233939>.
- [26] Electronics Research on Generalized Intelligent Routing Technology Based on Graph Neural Network[EB/OL]. <https://www.mdpi.com/2079-9292/11/18/2952>.
- [27] GUNAVATHIE M A, UMAMAHESWARI S. MLPRS: A Machine Learning-Based Proactive Re-Routing Scheme for flow classification and priority assignment[J]. *Journal of Engineering Research*, 2023, 11(3): 114-122.
- [28] QI H, SI J, HOU J, et al. Subflow scheduling strategy for multipath transmission in SDN-based spatial network[J]. *Wireless Networks*, 2023, 29(2): 941-953.
- [29] QI H, GUO Y, HOU D, et al. SDN-based dynamic multi-path routing strategy for satellite networks[J]. *Future Generation Computer Systems*, 2022, 133: 254-265.
- [30] DONG C, XU X, LIU A, et al. Load balancing routing algorithm

- based on extended link states in LEO constellation network[J]. *China Communications*, 2022, 19(2): 247-260.
- [31] GAYATRI V, SENTHIL K M. Efficient Load Balancing with MANET Propagation of Least Common Multiple Routing and Fuzzy Logic [J]. *Computers, Materials & Continua*, 2022, 72(1): 1831-1845.
- [32] LUAN F, YANG J, ZHANG H, et al. Optimization of load-balancing strategy by self-powered sensor and digital twins in software-defined networks [J]. *IEEE Sensors Journal*, 2022, 23(18): 20782-20793.
- [33] LATIF Z, LEE C, SHARIF K, et al. An SDN-based framework for load balancing and flight control in UAV networks[J]. *IEEE Consumer Electronics Magazine*, 2022, 12(1): 43-51.
- [34] ZHAI D, LI H, TANG X, et al. Joint position optimization, user association, and resource allocation for load balancing in UAV-assisted wireless networks[J]. *Digital Communications and Networks*, 2022, 10(1): 25-37.
- [35] TAO C C, ZHOU R. A method of UAV motion control to optimize air-ground relay network[J]. *Systems Engineering & Electronics*, 2024, 46(5): 1712-1723.
- [36] CHEN X, MA R. Intelligent UAV planning for task-offloading with limited buffer and multiple computing servers[J]. *Physical Communication*, 2024, 62: 102240.
- [37] ZHANG M, MOHAMMED E H, NG S X. Intelligent caching in UAV-aided networks [J]. *IEEE Transactions on Vehicular Technology*, 2021, 71(1): 739-752.
- [38] MIAO Y, HWANG K, WU D, et al. Drone swarm path planning for mobile edge computing in industrial internet of things[J]. *IEEE Transactions on Industrial Informatics*, 2022, 19(5): 6836-6848.
- [39] WANG D, TIAN J, ZHANG H, et al. Task offloading and trajectory scheduling for UAV-enabled MEC networks: An optimal transport theory perspective[J]. *IEEE Wireless Communications Letters*, 2021, 11(1): 150-154.
- [40] ELGENDY I A, MESHOU S, HAMMAD M. Joint task offloading, resource allocation, and load-balancing optimization in multi-UAV-aided MEC systems [J]. *Applied Sciences*, 2023, 13(4): 2625.
- [41] ZHU Y, WANG S. Joint Deployment and Trajectory Planning of Multiple UAVs for Emergency Communications [C] // 2023 IEEE Global Communications Conference. *IEEE*, 2023: 1854-1859.
- [42] Multi-UAV Relay Connectivity Optimization for Heterogeneous Users Based on Load Balancing and Throughput Maximization [EB/OL]. <https://ieeexplore.ieee.org/abstract/document/10103874>.
- [43] GUO H, ZHOU X, WANG Y, et al. Achieve load balancing in multi-UAV edge computing IoT networks: A dynamic entry and exit mechanism [J]. *IEEE Internet of Things Journal*, 2022, 9(19): 18725-18736.
- [44] SHAO S J, SU L L, GUO S Y, et al. Multi-Agent Cooperative Game Based Task Computing Mechanism for UAV-Assisted 6G NTN[J]. *Mobile Networks and Applications*, 2023, 28(4): 1510-1518.
- [45] SAIF A, DIMYATI K, NOORDIN K A, et al. Skyward bound: Empowering disaster resilience with multi-UAV-assisted B5G networks for enhanced connectivity and energy efficiency[J]. *Internet of Things*, 2023, 23: 100885.
- [46] SONBOL K, ÖZKASAP Ö, ALOQILY I, et al. EdgeKV: Decentralized, scalable, and consistent storage for the edge[J]. *Journal of Parallel and Distributed Computing*, 2020, 144: 28-40.
- [47] QIAO F, WU J, LI J, et al. Trustworthy edge storage orchestration in intelligent transportation systems using reinforcement learning [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2020, 22(7): 4443-4456.
- [48] KONTODIMAS K, SOUMLIS P, KRETSIS A, et al. Secure distributed storage on cloud-edge infrastructures [C] // 2021 IEEE 10th International Conference on Cloud Networking (CloudNet). *IEEE*, 2021: 127-132.
- [49] LI S, LAN T. HotDedup: Managing hot data storage at network edge through optimal distributed deduplication [C] // IEEE Conference on Computer Communications. *IEEE*, 2020: 247-256.
- [50] YANG Y, YE M, JIANG Q, et al. A novel node selection method for wireless distributed edge storage based on SDN and a maldistributed decision model [J]. *Electronic Research Archive*, 2024, 32(2): 1160-1190.



SUN Shiquan, born in 1999, postgraduate. His main research interests include software-defined networking and embedded systems.



ZHU Cheng, born in 1973, professor, master's supervisor. His main research interests include artificial intelligence and big data, network security management, and smart library construction.

(责任编辑:何杨)