An International ICT Journal
Featuring Industry-University-Institute Cooperation

ISSN 1673-5188
CN 34-1294/TN

ZTE COMMUNICATIONS

中兴通讯技术(英文版)

http://zte.magtechjournal.com

June 2024, Vol. 22 No. 2

Special Topic:
Advancements in Web3 Infrastructure for the Metaverse

ISSN 1673-5188
9 771673 518246 06

The 9th Editorial Board of ZTE Communications

CONTENTS

ZTE COMMUNICATIONS
June 2024 Vol. 22 No. 2 (Issue 87)

Special Topic ▶

Advancements in Web3 Infrastructure for the Metaverse

Review ▶

Research Papers ▶

Serial parameters: CN 34−1294/TN*2003*q*16*106*en*P*¥20.00*2200*13*2024-06

Statement

This magazine is a free publication for you. If you do not want to receive it in the future, you can send the “TD unsubscribe” mail to magazine@zte.com.cn. We will not send you this magazine again after receiving your email. Thank you for your support.

ZTE Communications Guidelines for Authors

Remit of Journal

ZTE Communications publishes original theoretical papers, research findings, and surveys on a broad range of communications topics, including communications and information system design, optical fiber and electro-optical engineering, microwave technology, radio wave propagation, antenna engineering, electromagnetics, signal and image processing, and power engineering. The journal is designed to be an integrated forum for university academics and industry researchers from around the world.

Manuscript Preparation

Manuscripts must be typed in English and submitted electronically in MS Word (or compatible) format. The word length is approximately 3 000 to 8 000, and no more than 8 figures or tables should be included. Authors are requested to submit mathematical material and graphics in an editable format.

Abstract and Keywords

Each manuscript must include an abstract of approximately 150 words written as a single paragraph. The abstract should not include mathematics or references and should not be repeated verbatim in the introduction. The abstract should be a self-contained overview of the aims, methods, experimental results, and significance of research outlined in the paper. Three to eight carefully chosen keywords must be provided with the abstract.

References

Manuscripts must be referenced at a level that conforms to international academic standards. All references must be numbered sequentially intext and listed in corresponding order at the end of the paper. References that are not cited in-text should not be included in the reference list. References must be complete and formatted according to *ZTE Communications* Editorial Style. A minimum of 10 references should be provided. Footnotes should be avoided or kept to a minimum.

Copyright and Declaration

Authors are responsible for obtaining permission to reproduce any material for which they do not hold copyright. Permission to reproduce any part of this publication for commercial use must be obtained in advance from the editorial office of *ZTE Communications*. Authors agree that a) the manuscript is a product of research conducted by themselves and the stated co-authors; b) the manuscript has not been published elsewhere in its submitted form; c) the manuscript is not currently being considered for publication elsewhere. If the paper is an adaptation of a speech or presentation, acknowledgement of this is required within the paper. The number of co-authors should not exceed five.

Content and Structure

ZTE Communications seeks to publish original content that may build on existing literature in any field of communications. Authors should not dedicate a disproportionate amount of a paper to fundamental background, historical overviews, or chronologies that may be sufficiently dealt with by references. Authors are also requested to avoid the overuse of bullet points when structuring papers. The conclusion should include a commentary on the significance/future implications of the research as well as an overview of the material presented.

Peer Review and Editing

All manuscripts will be subject to a two-stage anonymous peer review as well as copyediting, and formatting. Authors may be asked to revise parts of a manuscript prior to publication.

Biographical Information

All authors are requested to provide a brief biography (approx. 100 words) that includes email address, educational background, career experience, research interests, awards, and publications.

Acknowledgements and Funding

A manuscript based on funded research must clearly state the program name, funding body, and grant number. Individuals who contributed to the manuscript should be acknowledged in a brief statement.

Address for Submission

http://mc03.manuscriptcentral.com/ztecom

Guest Editorial >>>

Special Topic on Advancements in Web3 Infrastructure for the Metaverse

Guest Editors

Victor C. M. LEUNG

CAI Wei

Web3, also known as Web 3.0, has recently been attracting increasing attention from industry and academia. Leveraging the potential of blockchain technologies, Web3 has emerged as a pivotal foundation in the realm of metaverse development, which is considered by many as the next-generation Internet. Specifically, Web3 technologies such as smart contracts and protocols like non-fungible tokens (NFTs) have supported the immersive and content-rich experience of current Web3 metaverse projects. In addition, the decentralized autonomous organization (DAO) based on Web3 technologies has built the prototype of the future virtual society. Besides, many advanced breakthroughs in Web3 infrastructure for the metaverse have become accessible to the general public. Despite its significance, Web3 encounters several technical challenges that require attention, such as enhancing network and communication capabilities, designing efficient consensus protocols, and addressing human-centric factors in system designs and evaluations. Moreover, as an infrastructure for the metaverse, there are many research topics that are imperative to be addressed. For example, the existing NFT standards cannot well satisfy the interoperability and scalability of digital assets in the Web3 metaverse; the tokenomics system in the Web3 metaverse needs better token liquidity and balancing; the virtual identity in the Web3 metaverse is facing challenges of privacy and security. Therefore, the current development of Web3 infrastructure for the metaverse is still in its early stage, and several research directions are waiting for further study. This special issue of *ZTE Communications* aims to explore the state-of-the-art development in Web3 infrastructure dedicated to empowering the metaverse.

In this editorial, we will navigate through the key themes and insights offered by the accepted papers, each contributing a unique perspective to the ongoing discourse surrounding Web3 infrastructure for the metaverse.

The first paper titled "Building a Stronger Foundation for Web3: Advantages of 5G Infrastructure" delves into the multifaceted advantages of integrating 5G into the fabric of Web3. By leveraging its advancements in network speed, edge computing, capacity, security, and power efficiency, the authors underscore how 5G offers a robust foundation for the decentralized future of the internet. Through a comprehensive technical review, they elucidate the symbiotic relationship between 5G infrastructure and the transformative potential of Web3 technologies, laying the groundwork for enhanced user experiences and seamless connectivity within the metaverse.

The second paper titled "MetaOracle: A High-Throughput Decentralized Oracle for Web3.0-Empowered Metaverse" presents a pioneering approach to addressing the challenges associated with decentralized oracles in the Web3 metaverse. This paper introduces MetaOracle, a novel architecture designed to provide high-throughput, reliable data for blockchain-based applications. By leveraging a multi-identifier network (MIN) framework, MetaOracle mitigates risks associated with data integrity, offering increased reliability and throughput crucial for the seamless operation of Web3 metaverse applications. Through experimental validation, the authors demonstrate the efficacy of their approach in enhancing the trustworthiness and efficiency of oracles, thereby catalyzing the development of robust decentralized ecosystems

DOI: 10.12142/ZTECOM.202402001
Citation: (Format 1): LEUNG V C M, CAI W. Editorial: advancements in Web3 infrastructure for the metaverse [J]. ZTE Communications, 2024, 22 (2): 1–2. DOI: 10.12142/ZTECOM.202402001
Citation (Format 2): V. C. M. Leung and W. Cai, "Editorial: advancements in Web3 infrastructure for the metaverse," *ZTE Communications*, vol. 22, no. 2, pp. 1–2, Jun. 2024. doi: 10.12142/ZTECOM.202402001.

within the metaverse.

In the third paper titled "Optimization of High-Concurrency Conflict Issues in Execute-Order-Validate Blockchain," the authors delve into the scalability challenges inherent in blockchain architectures, particularly pertinent in the context of Web3 metaverse applications. Their paper proposes a novel approach based on MIN to address high-concurrency conflict issues in Execute-Order-Validate (EOV) blockchains. Through a meticulous analysis of consensus protocols and transaction processing mechanisms, the authors highlight the scalability and reliability benefits of their proposed architecture. Through empirical evaluation, they demonstrate significant enhancements in throughput and reliability, laying the groundwork for a more efficient and scalable Web3 infrastructure capable of supporting the diverse demands of the metaverse ecosystem.

The fourth paper titled "Utilizing Certificateless Cryptography for IoT Device Identity Authentication Protocols in Web3" proposes a decentralized authentication protocol tailored for IoT devices within the Web3 metaverse. This paper introduces a novel approach that integrates certificateless cryptography and physically unclonable functions to ensure robust security and privacy in the Internet of Things (IoT) networks. By leveraging blockchain technology, the proposed protocol distributes authentication services to edge authentication gateways and servers, mitigating risks associated with centralization and single points of failure. Through a comprehensive analysis of security threats and attack vectors, the authors demonstrate the effectiveness of their protocol in safeguarding the integrity and confidentiality of IoT device identities within the evolving landscape of the metaverse.

The fifth paper titled "Hierarchical Federated Learning Architectures for the Metaverse" explores the paradigm of Hierarchical Federated Learning (HFL) as a distributed machine learning approach ideally suited for the metaverse environment. This paper presents a comprehensive analysis of existing federated learning architectures and proposes a three-layer client-edge-cloud architecture tailored for the unique requirements of the metaverse. By leveraging hierarchical organization and edge computing capabilities, HFL offers scalability and privacy-preserving properties crucial for collaborative learning within decentralized environments. Through empirical evaluation and case studies, the authors highlight the potential of HFL to enable collaborative learning and knowledge sharing within the metaverse, paving the way for innovative applications and services.

Collectively, these contributions epitomize the interdisciplinary nature of Web3 infrastructure research, spanning advanced networking technologies, decentralized architectures, security mechanisms, and machine learning paradigms. As we delve deeper into the possibilities of the metaverse, it becomes evident that collaboration and innovation are paramount in realizing its full potential. Looking ahead, the journey towards a fully realized Web3 metaverse is fraught with challenges and opportunities alike. From enhancing network scalability to fostering inclusive and privacy-preserving ecosystems, the road ahead demands concerted efforts from researchers, developers, and industry stakeholders.

As guest editors, we extend our gratitude to the authors for their invaluable contributions and insights, as well as to the reviewers for their meticulous evaluations. We are also grateful for the conscientious and timely support of the staff of the ZTE Communications editorial office. We hope that this special issue serves as a catalyst for further exploration and collaboration in the dynamic landscape of Web3 infrastructure for the metaverse. Together, let us embark on this transformative journey towards a decentralized, interconnected, and immersive digital future.

Biographies

Victor C.M. LEUNG received the BASc (Hons.) and PhD degrees in electrical engineering from The University of British Columbia (UBC), Canada in 1977 and 1981, respectively. Dr. LEUNG is the Dean of the Artificial Intelligence Research Institute and a Professor of Engineering at Shenzhen MSU-BIT University, China, a Distinguished Professor of Computer Science and Software Engineering at Shenzhen University, China, and also an Emeritus Professor of Electrical and Computer Engineering and Director of the Wireless Networks and Mobile Systems (WiNMoS) Laboratory at UBC, Canada. His research is in the broad areas of wireless networks and mobile systems, and he has published widely in these areas. Dr. LEUNG is serving on the editorial boards of the *IEEE Transactions on Green Communications and Networking*, *IEEE Transactions on Computational Social Systems*, and several other journals. He received the 1977 APEBC Gold Medal, 1977-1981 NSERC Postgraduate Scholarships, IEEE Vancouver Section Centennial Award, 2011 UBC Killam Research Prize, 2017 Canadian Award for Telecommunications Research, 2018 IEEE TCGCC Distinguished Technical Achievement Recognition Award, and 2018 ACM MSWiM Reginald Fessenden Award. He co-authored papers that won the 2017 IEEE ComSoc Fred W. Ellersick Prize, 2017 IEEE Systems Journal Best Paper Award, 2018 IEEE CSIM Best Journal Paper Award, and 2019 IEEE TCGCC Best Journal Paper Award. He is a Life Fellow of IEEE, and a Fellow of the Royal Society of Canada (Academy of Science), Canadian Academy of Engineering, and Engineering Institute of Canada. He is named in the Clarivate Analytics lists of "Highly Cited Researchers" in the last four years.

CAI Wei received his PhD, MSc and BEng from The University of British Columbia (UBC), Canada, Seoul National University, Korea, and Xiamen University, China in 2016, 2011 and 2008, respectively. Dr. CAI is an Assistant Professor of Computer Engineering in the School of Science and Engineering at The Chinese University of Hong Kong, Shenzhen, China. He currently serves as the Director of the Human-Crypto Society Laboratory and the CUHK(SZ)-White Matrix Joint Metaverse Laboratory. Prior to joining CUHK-Shenzhen, he was a postdoctoral research fellow in the Wireless Networks and Mobile Systems (WiNMoS) Laboratory at UBC. He has also completed research visits at Academia Sinica (Taiwan), China, The Hong Kong Polytechnic University, China, and National Institute of Informatics, Japan. Dr. CAI has co-authored more than 100 journal/conference papers and received six best paper awards. His recent research interests focus on decentralized computing, including decentralized mechanisms, decentralized social computing, decentralized multimedia, and decentralized applications. He is an associate editor for *ACM Transactions on Multimedia Computing, Communications, and Applications* (TOMM), *IEEE Transactions on Computational Social Systems* (TCSS), and *IEEE Transactions on Cloud Computing* (TCC). He is a senior member of the IEEE and a member of the ACM.

Building a Stronger Foundation for Web3: Advantages of 5G Infrastructure

FENG Jianxin[1], PAN Yi[2], WU Xiao[3]

(1. TripleLab, Shenzhen 518000, China;
2. Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China;
3. WhiteMatrix Inc., Nanjing 210031, China)

DOI: 10.12142/ZTECOM.202402002

https://kns.cnki.net/kcms/detail/34.1294.TN.20240621.1421.008.html, published online June 21, 2024

Manuscript received: 2024-01-10

Abstract: The emergence of Web3 technologies promises to revolutionize the Internet and redefine our interactions with digital assets and applications. This essay explores the pivotal role of 5G infrastructure in bolstering the growth and potential of Web3. By focusing on several crucial aspects—network speed, edge computing, network capacity, security and power consumption—we shed light on how 5G technology offers a robust and transformative foundation for the decentralized future of the Internet. Prior to delving into the specifics, we undertake a technical review of the historical progression and development of Internet and telecommunication technologies.

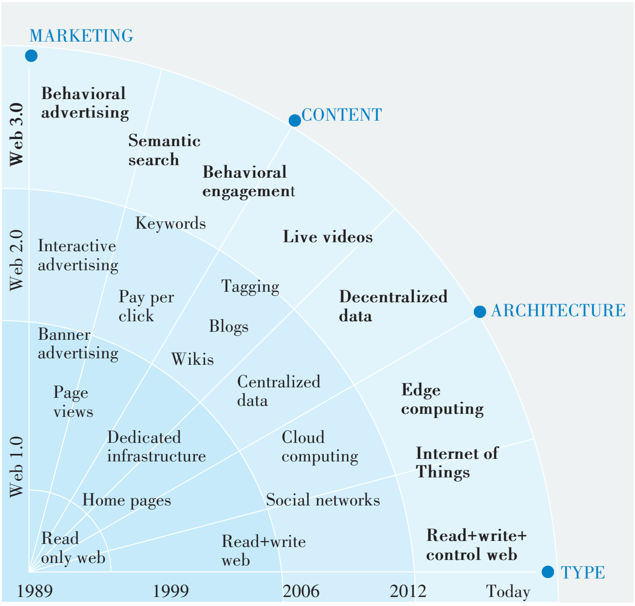
Keywords: Web3; 5G; blockchain; decentralized application; infrastructure

Citation (Format 1): FENG J X, PAN Y, WU X. Building a stronger foundation for Web3: advantages of 5G infrastructure [J]. ZTE Communications, 2024, 22(2): 3 – 10. DOI: 10.12142/ZTECOM.202402002

Citation (Format 2): J. X. Feng, Y. Pan, and X. Wu, “Building a stronger foundation for Web3: advantages of 5G infrastructure,” *ZTE Communications*, vol. 22, no. 2, pp. 3 – 10, Jun. 2024. doi: 10.12142/ZTECOM.202402002.

1 Introduction: The Web3 Revolution

Web3, also known as the decentralized web, presents a paradigm shift from its predecessor, Web2, by leveraging blockchain and other distributed ledger technologies (Fig. 1). This transformation envisions a more equitable, secure, and autonomous Internet where users regain control over their data, applications, and digital identities. However, achieving the full potential of Web3 requires a robust infrastructure, and herein lies the significance of 5G technology. The well-known Blockchain Trilemma emphasizes that within a decentralized system, attaining two out of the three core benefits—decentralization, security, and scalability—becomes feasible at any given point. In order to maintain uniform data coherence and the current state across all nodes within decentralized systems, diverse consensus algorithms are implemented, serving as the foundational support for both Layer 1 and Layer 2 infrastructures. When practical Web3 applications adopt a consistent consensus algorithm, their responsibility revolves around effectively managing the fundamental logic intrinsic to smart contracts and blockchains. The rapid advancements and widespread deployment of 5G are instrumental in overcoming critical challenges faced by Web3 applications, such as low transactions per second (TPS) and scalability. This essay delves into the advantages of 5G infrastructure in supporting the growth of Web3, focusing on network speed, edge computing, network capacity, security, and power consumption.[1–7]

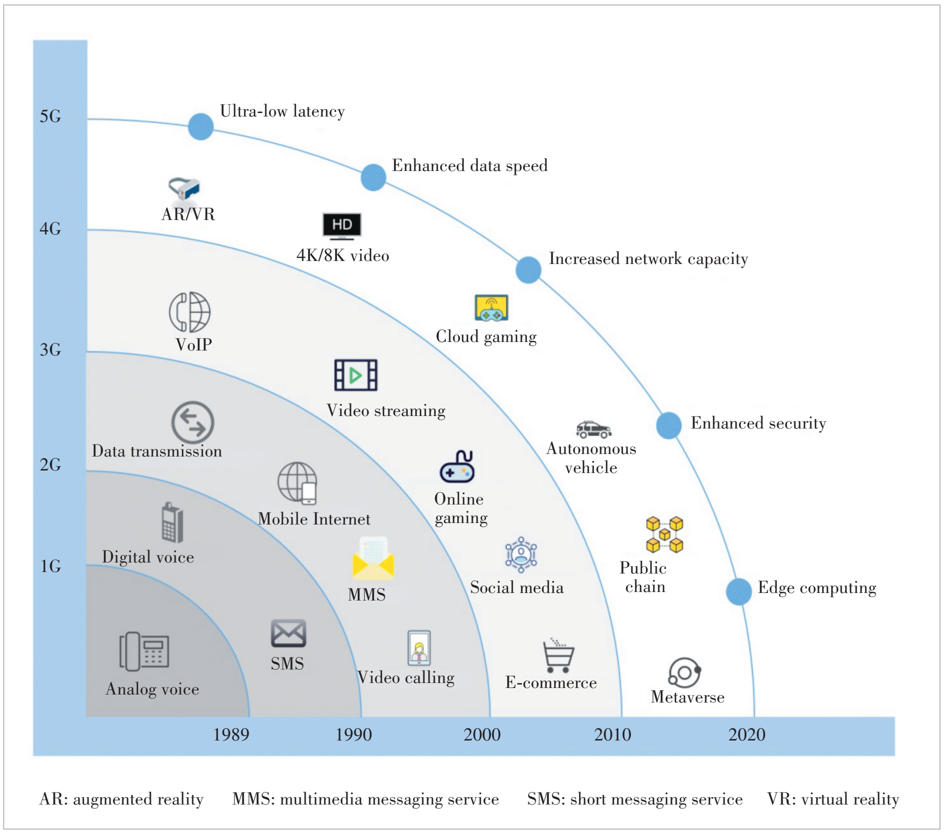


▲ **Figure 1. Evolution and technologies of Web 1.0, Web 2.0 and Web 3.0 scenarios**

This work was supported by the ZTE Industry-University-Institute Fund Project under Grant No. IA20221202011.

2 Network Speed: Paving the Way for Real-Time Interactions

Significant advancements in 5G technology have reduced latency from tens of milliseconds experienced in 4G to just a few milliseconds in 5G (Fig. 2). This reduction has enabled several novel application scenarios that were previously infeasible.[8]



AR: augmented reality MMS: multimedia messaging service SMS: short messaging service VR: virtual reality

▲ **Figure 2. Historical progression of telecommunication networks from the 1st to 5th generation, showcasing various technologies and scenarios**

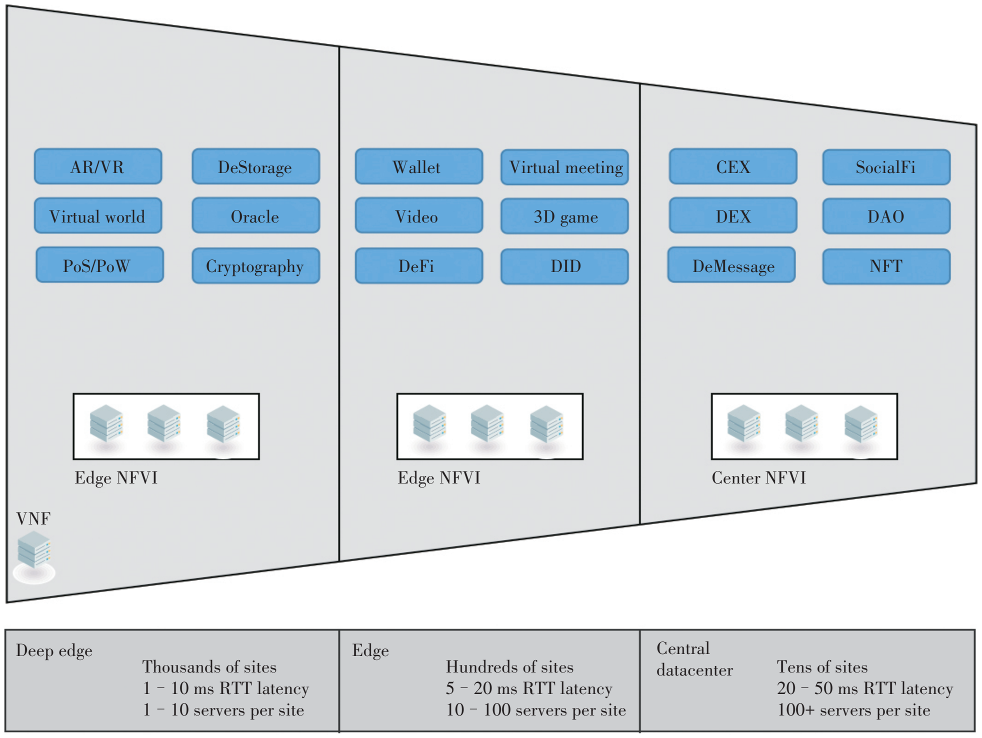
Web3 applications heavily rely on real-time interactions, where decentralized finance (DeFi), metaverse, and augmented reality/virtual reality (AR/VR) demand seamless and instant communications. The discourse surrounding Web3 applications can be effectively divided into three primary components: frontend, backend, and smart contracts. In the context of frontend and backend aspects, it is crucial to acknowledge that the functioning of all Web3 applications is in, herently reliant on the direct integration of 5G technology and its accompanying high-speed network capabilities. 5G technology exhibits unparalleled network speeds that significantly reduce latency and boost data transfer rates. On the smart contract side, the operational speed of decentralized applications (dApps) hinges upon the efficiency of the underlying blockchain system. The synchronization of data across distinct nodes demonstrates a direct correlation with the swiftness of the 5G network, although this calculation must also encompass the impact of the chosen consensus algorithm. While blockchain infrastructures based on Proof of Work (PoW) may experience modest enhancements, it is noteworthy that Proof of Stake (PoS) mechanisms showcase substantial improvements. Leveraging the ultra-low latency of 5G, dApps built upon PoS frameworks, and encompassing both Layer 1 and Layer 2 infrastructures are poised to execute smart contracts with heightened efficiency. This augmentation in operational efficiency not only bolsters the responsiveness of DeFi protocols but also facilitates seamless user experiences, removing unnecessary friction. Furthermore, the increased bandwidth of 5G networks enables higher quality AR/VR experiences, empowering users with immersive and engaging content. In essence, 5G lays the groundwork for a seamless real-time Web3 experience.

The data in Table 1 depicts selected Web3 scenarios that may gain advantages from leveraging the 5G network, contingent upon precise implementation and the maturity levels of the

▼**Table1. Comparison of network speed, latency and related scenarios between 4G and 5G**

Network Generation	Speed (real world)	Speed (theoretical)	Latency (real world)	Latency (theoretical)	Supported Scenarios
4G	10 – 50 Mbit/s	300 Mbit/s	30 – 50 ms	10 ms	Video streaming Real-time gaming IoT applications Smart homes
5G	100 Mbit/s – 1 Gbit/s	10 Gbit/s	10 – 20 ms	1 ms	8K video streaming AR/VR dApps More efficient public chain X-to-earn games 3D virtual world

AR: augmented reality dAPP: decentralized applications VR: virtual reality underlying technology and applications. As Web3 and 5G technologies continue to evolve, more innovative use cases may emerge that take full advantage of 5G's speed and low latency.



AR: augmented reality
CEX: centralized exchange
DAO: decentralized autonomous organizations
DeFi: decentralized finance
DEX: decentralized exchange
DID: decentralized identifier
NFT: non-fungible token
NFVI: network function virtualization infrastructure
PoS: Proof of Stake
PoW: Proof of Work
RTT: round-trip time
VNF: virtual network function
VR: virtual reality

▲Figure 3. Various scenarios operating at different positions and distances with the 5G network

3 Edge Computing: Empowering Distributed Applications

Multi-access edge computing (also known as mobile edge computing, MEC) is a critical feature of 5G networks. It is designed on a universal X86 platform, which transforms every 5G base station into a powerful edge computing node. Unlike today's cloud computing providers that deploy dozens of data centers around the world, 5G deployment will rapidly expand the scale of data centers to millions. Certain Web3 applications are constructed and evaluated within consortium or private blockchain settings. The comprehensive development lifecycle can be significantly fortified through the integration of edge computing. For instance, a cluster of MEC devices can swiftly establish a blockchain infrastructure. This approach offers expedient means to develop and test Web3 applications within this environment.

The decentralization of Web3 applications often means distributing computing power and resources across a network of nodes. Here, edge computing becomes a game-changer. 5G's edge computing capabilities facilitate the processing and storage of data closer to the end-users, reducing latency and enhancing the efficiency of distributed applications (Fig. 3). By leveraging edge servers, Web3 dApps can provide a smoother user experience, enhanced privacy, and reduced reliance on centralized data centers. This integration of 5G and edge computing fosters a more robust, scalable, and responsive Web3 ecosystem.[9 - 12]

4 Network Capacity: Facilitating Mass Adoption of Web3

The network capacity of 5G has been greatly improved, with over 20 times the capacity of 4G, thanks to the deployment of a large number of micro base stations.

The mass adoption of Web3 technologies necessitates a network capable of handling an ever-increasing number of connected devices and transactions. Given that numerous decentralized nodes operate on distinct cloud servers, the synchronization of data within the decentralized system is intricately linked to the capacity of the network. The decentralized and distributed nature of Web3 networks and applications presents very high requirements and challenges for the underlying basic communication network bandwidth and capacity. Hundreds of millions or even billions of terminals are constantly synchronizing and communicating with each other, and once network congestion occurs, it could lead to huge disasters. Traditional networks may struggle to cope with the projected surge in demand. 5G, with its unprecedented network capacity, emerges as the ideal solution. With its higher frequencies, smaller cells, and advanced beamforming techniques, 5G networks can accommodate a vast number of connected devices simultaneously. This scalability is crucial for ensuring the seamless functioning of Web3 applications, supporting the growth of the Internet of Things (IoT) and enabling widespread adoption.[13 - 14]

5 Security: Safeguarding the Decentralized Web

5G networks offer several advantages in terms of security, as shown in Table 2. With enhanced encryption, users can be assured of secure transmission of data. Additionally, 5G networks are designed to have a more secure architecture, making it more difficult for hackers to breach the network. The use of virtualized network functions in 5G networks also enables more advanced security features such as network slicing and isolation, which can help protect against cyber attacks.[15 – 16]

The decentralized nature of Web3 applications already brings inherent security benefits compared to centralized counterparts. Nevertheless, ensuring data integrity, user privacy, and protection against cyber threats remains paramount. 5G's enhanced security features, such as end-to-end encryption and network slicing, bolster the defenses of Web3 applications. Network slicing enables isolation and dedicated resources for specific services, reducing the attack surface and enhancing the resilience of decentralized networks. Additionally, the use of virtual private networks (VPNs) and enhanced authentication mechanisms in 5G further enhances the security and trustworthiness of Web3 systems.

6 Power Consumption: Enhancing Sustainability and Eco-Friendly Practices in Web3 Development

The power consumption dynamics of 5G networks exhibit a notable advantage over their 4G counterparts, primarily attributed to the incorporation of advanced radio access technologies, network virtualization, and sophisticated signal processing techniques.

Based on the research conducted by Nokia, the unit traffic power consumption for 5G technology exhibits significant efficiency gains over traditional technologies, reducing to as low as 10% of the previous energy consumption levels. This enhanced power efficiency becomes of paramount importance within the Web3 industry, as developers and practitioners grapple with the challenge of addressing criticisms regarding power consumption associated with decentralized applications. Particularly, blockchain-based Web3 applications have faced scrutiny due to the energy-intensive consensus mechanisms utilized for transaction validation. In this context, the low-power attributes of 5G technology offer a promising solution, optimizing the allocation of network resources and minimizing energy wastage. By reducing the energy requirements for data transmission and processing, 5G effectively mitigates the environmental impact of Web3 applications, fostering a more sustainable and eco-friendly digital ecosystem. This proactive, environmentally-conscious approach not only addresses concerns related to excessive power consumption but also positions Web3 at the forefront of advocating energy-efficient and eco-conscious technologies for the future.

The data presented in Table 3 indicates a substantial improvement in energy for data transmission (dozens of times), which is expected to considerably reduce energy consumption and mitigate the environment impact of Web3 industry.

7 Case Study: 5G Powers Web3 Application in Real World

The indispensability of graphics processing units (GPUs) across domains such as blockchain, artificial intelligence (AI),

▼Table 2. Comparison of security features and related scenarios between 4G and 5G

Network Generation	Security Features	Supported Scenarios
4G	Basic encryption (AED-128) Subscriber identity module authentication Authentication and Key Agreement (AKA)	Mobile broadband, web browsing Online gaming, basic applications Moderate user density
5G	Enhanced encryption (AES-256) Stronger mutual authentication Improved integrity protection Network slicing for isolated security domains Certificate-based device authentication Enhanced privacy protection (5G AKA) Improved authentication protocols (5G-EAP) Enhanced security for IoT devices (5G-SBA)	Augmented reality (AR) and virtual reality (VR) DeFi /GameFi / SocialFi / Wallet / NFT Public chain (PoS / PoW) Virtual world / Metaverse Massive IoT deployment Ultra-high user density Mission-critical communications

▼Table 3. Comparison of energy, network capacity, energy efficiency and related scenarios between 4G and 5G

Network Generation	MIMO	Energy/Watt	Capacity/(Mbit/s)	Energy Efficiency/(GB/kW·h)	Supported Scenarios
4G	2T2R	400	150	165	–
	4T4R	685	300	192	
5G	32T32R	500	5 000	4 395	Scenarios involving extensive wireless communications (Public chain, Metaverse, GameFi, ...)
	64T64R	810	10 000	5 425	

rendering, gaming, and data science is widely acknowledged. However, the contemporary GPU resource landscape is constrained by limitations in availability and exorbitant costs. In this context, we present a practical use case elucidating how the convergence of 5G technology and Web3 principles is poised to revolutionize GPU resource utilization within the rendering industry.

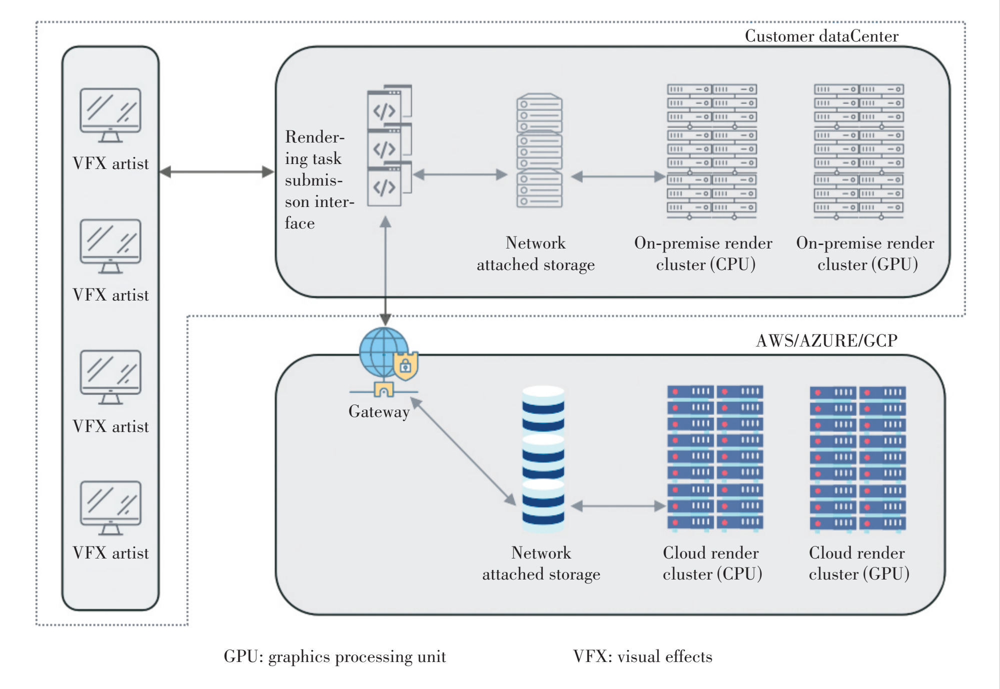
Launched in 2017, the Render Network serves as an intricate framework designed to cater to a diverse spectrum of computational tasks, ranging from fundamental rendering operations to intricate artificial intelligence processes. These tasks are expedited with remarkable speed and efficiency within a blockchain-driven peer-to-peer network, characterized by its resolute immunity to errors and delays, and fortified by airtight property rights protection.

Pioneering an epochal advancement, the Render Network stands as the pioneering decentralized GPU rendering platform. This pioneering platform endows artists with the capability to dynamically scale GPU rendering workloads on-demand, orchestrating them across globally distributed high-performance GPU Nodes. By virtue of an intricately designed blockchain marketplace tailored to harness latent GPU compute capacities, artists are endowed with unparalleled prowess to amplify their next-generation rendering endeavors. This decentralized architectural paradigm fosters a transformative departure from the conventional centralized GPU cloud model, heralding a realm of cost-efficiency and an astronomical surge in computational acceleration, upending established norms.[17 – 20]

A comparative analysis between the traditional rendering methodology and the decentralized distributed approach is shown in Table 4, Figs. 4 and 5.

In addition to the paradigm illustrated by the Render Network, a plethora of analogous Web3 applications have demonstrated advantageous integration with the burgeoning 5G network infrastructure. Noteworthy examples encompass Caduceus, Portalverse, Ipolloverse, and DBChain, each exhibiting symbiotic relationships with 5G, thereby enhancing their operational efficiency and user experiences.

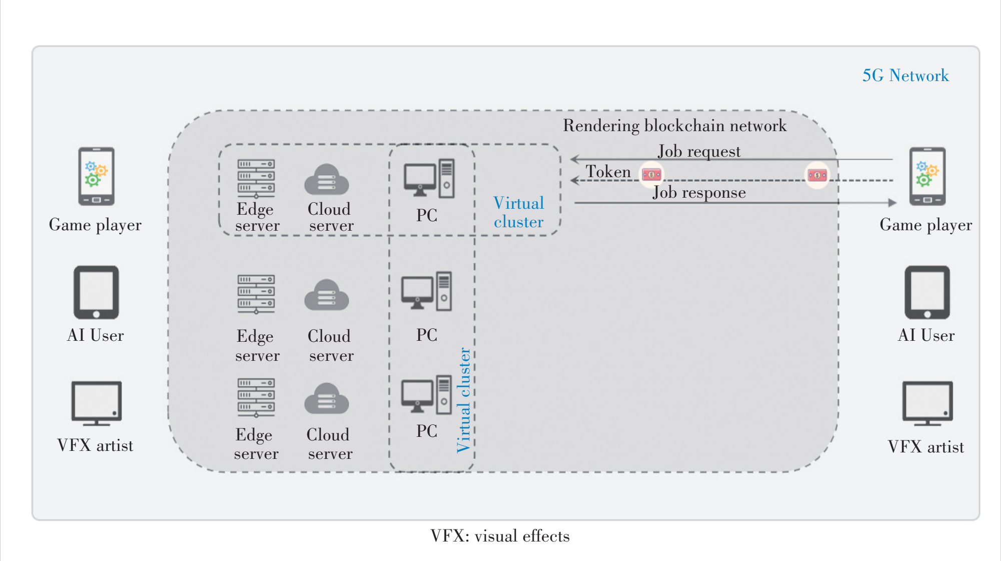
Concurrently, the synergy between blockchain and 5G unveils profound implications. Evidently, the Helium project emerges as an exemplar in leveraging blockchain technology to amplify the ambit of 5G coverage. By harnessing blockchain's innate attributes, Helium extends the perimeter of 5G accessibility, facilitating broader and more inclusive participation within the 5G ecosystem. This confluence exemplifies the potent interplay between blockchain's decentralized architecture and the transformative potential of 5G networks.


Customer dataCenter
VFX artist
VFX artist
VFX artist
VFX artist
Rendering task submisson interface
Network attached storage
On-premise render cluster (CPU)
On-premise render cluster (GPU)
AWS/AZURE/GCP
Gateway
Network attached storage
Cloud render cluster (CPU)
Cloud render cluster (GPU)
GPU: graphics processing unit
VFX: visual effects

▲**Figure 4. Conventional rendering approach involving on-premise data centers or remote centralized cloud services.**

▼**Table 4. What 5G network and technologies can bring to various rendering approaches**

Rendering Approach	5G Network & Technologies	Advantages	Disadvantages
Centralized on-premise cloud (AWS etc.)	Not needed	Complete authority over resource security is assured	Expensive resource Low utilization Less scalability Costly maintaining
Decentralized distributed	Ultra-low latency High speed transmission Network capacity Network security	Flexible scalability High utilization rate Efficient infrastructure Cost effective Almost unlimited resource	Security is uncertain uncontrolled resource



▲ Figure 5. Innovative paradigm facilitated by 5G networks, enabling end-users to seamlessly access available resources from any location and at any time

8 Future Challenges: Web3 Necessitating 5G and Upcoming Advancements

1) Latency and bandwidth requirements

Web3 applications relying on real-time data processing and interactions impose stringent demands for ultra-low latency and high bandwidth. While 5G edge computing mitigates latency, specific use cases may necessitate even lower latency and higher bandwidth. Emerging metaverse platforms like Decentraland, Sandbox, and Roblox witness rapid popularity and attract substantial user bases. Both Decentraland and Sandbox enjoy advantages stemming from their seamless frontend experiences and their utilization of smart contracts and NFTs to safeguard users' assets within decentralized systems. Moreover, multichain metaverses such as Matrix World stand to gain even greater benefits from the rapid expansion of 5G networks. Cloud-based metaverses offer players a seamless virtual world experience through their frictionless multichain connectivity, further augmented by the growth of 5G technology. According to a recent Gartner study, by 2026, 25% of the global population is projected to spend at least one hour daily in the metaverse, engaging in activities like work, shopping, education, social interactions, and entertainment. These virtual worlds simulate reality using 3D, 4K/8K video streaming, and real-time interactions, demanding heavy traffic and ultra-low latency, requirements that current 5G networks may not fully meet. To address these challenges, future technologies, such as Terahertz (THz) communication or advanced photonics, hold the potential to significantly enhance data transfer rates and minimize latency, thereby ensuring seamless user experiences in the metaverse.[21 – 22]

2) Security and data integrity

Decentralized applications are vulnerable to security breaches and data tampering. Blockchain consensus mechanisms involve a substantial number of participants, whether utilizing PoS or PoW. In theory, a higher number of participants in the blockchain network contribute to enhanced safety and security. This relies on the assumption that the participants are controlled by independent and separate entities. Presently, Ethereum has approximately half a million validators, with a significant portion running on cloud platforms like AWS, Azure, and GCP. This setup raises concerns, as cloud providers potentially hold the ability to influence the blockchain under specific conditions. With the advent of 5G, the abundance of edge devices surpasses the current cloud providers. In the near future, a considerable portion of nodes running on edge devices will be controlled by network operators. Overcoming and eliminating these security concerns presents a significant technical challenge in ensuring the integrity and trustworthiness of blockchain systems. To address these challenges, future technologies might include robust encryption methods, quantum-resistant cryptography, and blockchain-based consensus mechanisms that improve security and establish trust in the data exchanged between edge devices and the Web3 applications.

3) Interoperability and standards

Web3 applications often operate on multiple blockchain platforms and protocols, leading to challenges in achieving seamless interoperability among them. The increasing variety of blockchain types, present and future, deployed on 5G networks further adds to the complexity. Addressing seamless interoperability within the 5G network is another significant challenge due to the diverse technologies it encompasses, such as small cells, multiple-input multiple-output (MIMO), full duplex, and beamforming, each functioning differently. Interoperability holds a pivotal role within cross-chain technology. Effective data communication and the seamless transfer of assets across different chains necessitate the establishment of protocols and standards. These elements are crucial in unlocking the full potential of blockchain technology. Future technologies must prioritize establishing common standards and protocols to facilitate smooth communication and data exchange between various blockchains and networks. Emphasizing standardization will foster compatibility and enhance the overall Web3 ecosystem.

4) Edge infrastructure scalability

Edge computing relies on a distributed network of edge devices. Scalability becomes crucial as Web3 applications experience increased user adoption. Blockchain networks such as Bitcoin and Ethereum typically have the capacity to process dozens of TPS, whereas certain Layer-2 blockchain solutions like Polygon, Arbitrum, and Optimism can handle significantly higher TPS, reaching up to 65 000. The substantial TPS demands from numerous edge devices pose significant challenges to network scalability. Future technologies might involve advances in distributed computing, mesh networks, and software-defined infrastructures that can dynamically scale based on demand, ensuring optimal performance and availability.

5) Energy efficiency

In comparison to the previous generation, 5G technology demonstrates remarkable improvements in unit traffic power consumption, significantly lowering energy requirements. However, this efficiency gain comes at the cost of deploying a larger number of 5G base stations due to their limited coverage area. Additionally, the individual power consumption of 5G base stations is greater than that of 4G, leading to an overall increase in power consumption. Considering the projected growth of the blockchain industry and the potential for a significant number of blockchain nodes to run on 5G edge devices, network operators may face the challenge of deploying an even larger number of base stations to meet the burgeoning demand. To address this issue, solely focusing on reducing power consumption may not suffice, necessitating exploration of more feasible alternatives such as harnessing renewable energy sources like solar and wind energy. Integrating sustainable energy solutions can help mitigate the environmental impact and ensure the long-term sustainability of the 5G network infrastructure as it adapts to the evolving demands of the blockchain industry. Future technologies could include energy harvesting solutions, advanced power management techniques, and renewable energy integration to reduce the ecological footprint of edge computing operations.

6) Data governance and regulation

With the decentralized nature of Web3 applications and data processing at the edge, concerns about data governance and compliance with regulations might arise. 5G network operators retain sensitive customer information, encompassing personal and identification details like credit card data, address details, wallet information, transaction records, and payment history. In light of regulations like General Data Protection Regulation (GDPR), privacy demands have intensified, emphasizing robust user data protection and granting users greater control and ownership over their data. As data stored and recorded on a blockchain is immutable by design, it cannot be deleted or forgotten, ensuring data permanence and integrity. Future technologies could incorporate self-sovereign identity solutions, decentralized data marketplaces, and AI-driven compliance frameworks to address these challenges while empowering users with control over their data.

7) Reliability and fault tolerance

Token property holds a critical role within the Web3 ecosystem, where a plethora of applications such as DeFi, GameFi, and SocialFi heavily rely on token finance. Ensuring utmost reliability and robustness in the infrastructure network becomes imperative for these applications. While 5G networks have made significant strides in enhancing resilience through network slicing and dynamic rerouting, their susceptibility to signal degradation and limited coverage due to high-frequency signals poses challenges, leading to dropped connections and reduced reliability in specific regions. However, with the imminent arrival of 6G, the landscape of communication technology is set to witness groundbreaking advancements in ultra-reliable and fault-tolerant communications. Leveraging improved beamforming and massive MIMO technologies, 6G networks are poised to transcend the coverage limitations faced by 5G, promising more consistent connectivity across diverse environments. Furthermore, 6G is anticipated to implement sophisticated fault-tolerance mechanisms capable of autonomous self-healing and real-time network reconfiguration. By amalgamating resilient consensus mechanisms, redundant edge nodes, and decentralized storage solutions, a seamless and uninterrupted access to Web3 applications can be confidently expected.[23–24]

Addressing these challenges will require collaborative efforts from technology developers, industry stakeholders, and regulatory bodies. By leveraging future technologies that focus on performance, security, and user experience, Web3 applications can realize their full potential while leveraging the benefits of 5G advantages. It is essential to continuously innovate and adapt to the evolving landscape to build a robust and sustainable Web3 ecosystem.

9 Conclusion: Symbiotic Future of 5G and Web3, Unleashing a Full Potential of the Decentralized Web

In conclusion, the seamless integration of 5G infrastructure into the Web3 ecosystem holds the key to the success and widespread adoption of decentralized technologies. The remarkable combination of unparalleled network speed, advanced edge computing capabilities, expansive network capacity, robust security features, and energy-efficient low-power consumption provides a robust foundation for the decentralized web to flourish. As we embark on a path towards a more autonomous, equitable, and decentralized Internet, the convergence of 5G and Web3 forges a symbiotic relationship that unlocks the full potential of the decentralized web, fundamentally transforming the way we interact with digital assets and applications in the future. This transformation will revolutionize industries, redefine financial systems, empower individuals, and democratize access to information and opportunities.

The ongoing development and optimization of 5G networks are indispensable in building a stronger foundation for Web3 and fulfilling the promise of a decentralized, secure, sustainable, and user-centric digital world. As the world continues to embrace this symbiotic future, the decentralized web will stand as a beacon of innovation, trust, transparency, and inclusivity, paving the way for an unparalleled landscape of opportunities and a more equitable and transformative digital future.[25]

References

[1] MURRAY A, KIM D, COMBS J. The promise of a decentralized Internet: what is Web3 and how can firms prepare? [J]. Business horizons, 2023, 66 (2): 191 – 202. DOI: 10.1016/j.bushor.2022.06.002

[2] LI H, DANG R R, YAO Y, et al. A review of approaches for detecting vulnerabilities in smart contracts within web 3.0 applications [J]. Blockchains, 2023, 1(1): 3 – 18. DOI: 10.3390/blockchains1010002

[3] SADOWSKI J, BEEGLE K. Expansive and extractive network of Web3 [J]. Big data & Society, 2023, 10(1): 1 – 14. DOI: 10.1177/20539517231159629

[4] JIN L, PARROTT K. Web3 is our chance to make a better internet [J]. Harvard business review, 2022

[5] Goldman Sachs. Framing the Future of Web3.0 [R]. 2021

[6] A16z. The Web3 Landscape [R]. 2021

[7] CHAER A, SALAH K, LIMA C, et al. Blockchain for 5G: opportunities and challenges [C]//Globecom Workshops. IEEE, 2019. DOI: 10.1109/GCWkshps45667.2019.9024627

[8] NORP T. 5G requirements and key performance indicators [J]. Journal of ICT standardization, 2018, 6(1): 15 – 30. DOI: 10.13052/jicts2245-800x.612

[9] WANG H, LI H, SMAHI A, et al. MIS: a multi-identifier management and resolution system in the metaverse [J]. ACM transactions on multimedia computing, communications, and applications, 2024, 20(7): 191. DOI: 10.1145/3597641

[10] Ericsson. Edge computing and 5G [R]. 2020

[11] SIRIWARDHANA Y, PORAMBAGE P, LIYANAGE M, et al. A survey on mobile augmented reality with 5G mobile edge computing: architectures, applications, and technical aspects [J]. IEEE communications surveys & tutorials, 2021, 23(2): 1160 – 1192. DOI: 10.1109/comst.2021.3061981

[12] NABBEN K. Decentralised autonomous organisations (DAOs) as data trusts: a general-purpose data governance framework for decentralised data ownership, storage, and utilization [J]. Social science research network, 2021. DOI: 10.2139/ssrn.4009205

[13] China Mobile, China Telecom, China Unicom, et al. 5G-advanced technology evolution from a network perspective: towards a new era of intelligent connect X [R]. 2021

[14] MODESTA E E, FRANCIS A O, ANTHONY O O. A framework of 5G networks as the foundation for IoTs technology for improved future network [J]. International journal of physical sciences, 2019, 14(10): 97 – 107. DOI: 10.5897/ijps2018.4782

[15] GUPTA R K, ALMUZAINI K K, PATERIYA R K, et al. An improved secure key generation using enhanced identity-based encryption for cloud computing in large-scale 5G [J]. Wireless communications and mobile computing, 2022: 7291250. DOI: 10.1155/2022/7291250

[16] FROEHLICH M, WALTENBERGER F, TROTTER L, et al. Blockchain and cryptocurrency in human computer interaction: a systematic literature review and research agenda [C]//Proc. Designing Interactive Systems Conference. ACM, 2022: 155-177. DOI: 10.1145/3532106.3533478

[17] SUDHAMANI C, ROSLEE M, TIANG J J, et al. A survey on 5G coverage improvement techniques: issues and future challenges [J]. Sensors, 2023, 23(4): 2356. DOI: 10.3390/s23042356

[18] TAKYAR A. Web3 use cases and applications [EB/OL]. [2024-01-15]. https://www.leewayhertz.com/web3-use-cases-and-applications

[19] GOEL A K, BAKSSHI R, AGRAWA K K. Web3.0 and decentralized applications [C]//2nd International Conference on Innovative Research in Renewable Energy Technologies (IRRET 2022). IMPS, 2022. DOI: 10.3390/materproc2022010008

[20] ALLEN D, FRANKEL E, LIM W, et al. Ethics of decentralized social technologies: lessons from Web3, the fediverse, and beyond [R]. 2023

[21] MCCORMICK P. The value chain of the open metaverse [EB/OL]. (2021-01-25) [2024-01-15]. https://www. notboring. co/p/the-value-chain-of-the-open-metaverse

[22] Citi GPS. Metaverse and money: decrypting the future [R]. 2022

[23] Protocol Labs. Filecoin: a decentralized storage network [R]. 2018

[24] Storj. Storj: decentralized cloud storage [R]. 2017

[25] RAGNEDDA M, DESTEFANIS G. Blockchain and Web3.0: social, economic, and technological challenges [M]. UK: Taylor & Francis, 2019. DOI: 10.4324/9780429029530

Biographies

FENG Jianxin (fjx2000@gmail.com) is a serial entrepreneur in the cloud computing and blockchain industries. With years of experience working for several top-tier telecommunications companies both domestically and internationally, he has witnessed and contributed to the evolution from 3G to 4G and 5G, as well as the transition from Web2 to Web3. Currently, he operates a blockchain infrastructure company and a cloud supercomputing company. He has a BS in Math and EMBA from The University of Texas at Arlington (UTA), USA.

PAN Yi is currently a Chair Professor and the dean of College of Computer Science and Control Engineering at Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China and a Regents' Professor Emeritus at Georgia State University, USA. He is a Fellow of American Institute for Medical and Biological Engineering, Foreign Member of Russian Academy of Engineering, Foreign Member of Ukrainian Academy of Engineering Science, Member of European Academy of Sciences and Arts, European Academy of Natural Sciences, Fellow of the Royal Society for Public Health, Fellow of the Institute of Engineering and Technology, and Fellow of the Japan Society for the Promotion of Science. He received his BEng and MEng degrees in computer engineering from Tsinghua University, China in 1982 and 1984, respectively, and his PhD degree in computer science from the University of Pittsburgh, USA in 1991.

WU Xiao is a seasoned senior engineer with 15 years of expertise in software engineering, programming, and a decade of successful management across dynamic startup environments in Canada and China. Holding a BSc and MSc in computer science from the University of Alberta, Canada, he is the CEO of White Matrix Inc., a blockchain enterprise strategically invested in by Ant Group, and the visionary founder behind ChainIDE, the world's largest cloud-based blockchain integrated development environment. He has received accolades such as the first prize in the China Blockchain Development Contest (2020) and currently holds influential roles, serving as the Vice Director of The Chinese University of Hong Kong (Shenzhen)-White Matrix Joint Metaverse Laboratory. His global impact is underscored by co-organizing over 100 Web3 developer events worldwide, including ETH Riyadh and ETH Shanghai, and in 2023, ChainIDE's recognition and recommendation on Ethereum.org further solidified its position among developers globally.

MetaOracle: A High-Throughput Decentralized Oracle for Web 3.0-Empowered Metaverse

CHEN Rui[1], LI Hui[1], LI Wuyang[1], BAI He[1], WANG Han[1], WU Naixing[2], FAN Ping[2], KANG Jian[2], Selwyn DENG[2], ZHU Xiang[2]

(1. School of Electronic and Computer Engineering, Peking University, Shenzhen 518055, China;
2. China United Network Communications Co., Ltd., Shenzhen Branch, Shenzhen 518031, China)

DOI: 10.12142/ZTECOM.202402003

https://kns.cnki.net/kcms/detail/34.1294.TN.20240529.1356.002.html, published online May 30, 2024

Manuscript received: 2024-04-19

Abstract: Recent rapid advancements in communication technology have brought forth the era of Web 3.0, representing a substantial transformation in the Internet landscape. This shift has led to the emergence of various decentralized metaverse applications that leverage blockchain as their underlying technology to enable users to exchange value directly from point to point. However, blockchains are blind to the real world, and smart contracts cannot directly access data from the external world. To address this limitation, the technology of oracles has been introduced to provide real-world data for smart contracts and other blockchain applications. In this paper, we focus on mitigating the risks associated with oracles providing corrupt or incorrect data. We propose a novel Web 3.0 architecture for the Metaverse based on the multi-identifier network (MIN), and its decentralized blockchain oracle model called MetaOracle. The experimental results show that the proposed scheme can achieve minor time investment in return for significantly more reliable data and increased throughput.

Keywords: Web 3.0; metaverse; blockchain; smart contract; oracle

Citation (Format 1): CHEN R, LI H, LI W Y, et al. MetaOracle: a high-throughput decentralized oracle for Web3.0-empowered Metaverse [J]. ZTE Communications, 2024, 22(2): 11 – 18. DOI: 10.12142/ZTECOM.202402003

Citation (Format 2): R. Chen, H. Li, W. Y. Li, et al., "MetaOracle: a high-throughput decentralized oracle for Web3.0-empowered Metaverse," *ZTE Communications*, vol. 22, no. 2, pp. 11 – 18, Jun. 2024. doi: 10.12142/ZTECOM.202402003.

1 Introduction

Web 3.0 is currently one of the most trending topics, symbolizing the Internet revolution from a mere information exchange platform into a more open, decentralized, and secure ecosystem based on blockchain. In April 2014, WOOD[1] first introduced the concept of Web 3.0, and he believed that in the aftermath of Snowden's revelations, Internet users could no longer trust corporations, as these entities tend to exploit and manipulate user data for their financial gains. Subsequently, Web 3.0 has gained significant popularity since 2021 and is widely recognized as the future direction of Internet development.

Web 3.0 encompasses the seamless integration of the state-of-the-art technologies with the ultimate aim of creating a decentralized metaverse ecosystem. For instance, the digital twin technology creates a mirror image of the real world, while virtual reality (VR) and augmented reality (AR) provide immersive 3D experiences. The advancement in 5G and beyond offers ultra-high reliable and ultra-low latency connections for various metaverse devices. Wearable sensors and brain-computer interfaces (BCI) enable user-avatar interactions within the metaverse. Artificial intelligence (AI) facilitates the creation and rendering of large-scale metaverse environments. In addition, blockchain and smart contracts are vital in ensuring authentic rights for metaverse assets[2]. As a result, the metaverse holds the potential to revolutionize various industries and redefine the way we interact with digital content and each other.

To achieve direct and secure peer-to-peer value exchange within the metaverse, without the reliance on third-party trusted service intermediaries[3], the blockchain technology is leveraged. In essence, blockchain could be regarded as a public ledger where all committed transactions are stored in blocks with a chain-like structure[4]. Asymmetric cryptography and decentralized consensus algorithms are implemented for

This work is supported by Shenzhen Fundamental Research Programs under Grant Nos. JCYJ20220531093206015, GXWD20201231165807007-20200807164903001, JCYJ20210324122013036, and JCYJ20190808155607340, GuangDong Prov. R&D Key Programs under Grant Nos. 2019B010137001 and 2018B010124001, Basic Research Enhancement Program of China under Grant No. 2021-JCJQ-JJ-0483, National Key-stone R&D Program of China under Grant No. 2017YFB0803204, Natural Science Foundation of China under Grant No. 61671001, ZTE Industry-University-Institute Fund Project under Grant No. 2019ZTE03-01, and HuaWei Funding under Grant No. TC20201222002.

nodes to reach consensus and ensure transaction immutability. The transaction rules can be encoded in smart contracts. These contracts are executed automatically within the blockchain network when predefined conditions are satisfied[5]. Therefore, it is imperative to base on objective and trustworthy data to verify the execution conditions for smart contracts.

For security reasons, smart contracts are commonly executed within a secure sandbox environment, like the Ethereum Virtual Machine (EVM). They always depend on "oracles" to access external data[6]. Oracles can be centralized trusted third parties or decentralized entities that provide trustworthy off-chain data for verifying the conditions required to trigger smart contract execution. The centralized oracle relies on data from a single source, which may benefit high efficiency. Provable, also referred to as oraclize[7], is the leading oracle service built on Amazon Web Services (AWS). It is specifically designed to provide data feedback for smart contracts and it continues to maintain a large user base[8]. However, the utilization of centralized oracles may not only undermine the decentralization principle of the blockchain but also carry the risk of incorporating corrupt and inaccurate data into the blockchain[9]. On the other hand, by using consensus mechanisms to aggregate data from various independent sources, decentralized oracles resolve the singular data source problem. A few works have attempted to enhance the decentralized oracle models, and the details are presented in Section 2.

In the coming years, the metaverse is expected to undergo significant growth and expansion. In order to facilitate reliable and secure communication in the metaverse ecosystem, both individual and organizational users, as well as network devices, have a shared demand for trustworthy and privacy-enhancing identifiers. With the growing complexity and scale of the metaverse ecosystem, the co-governed multi-identifier network (MIN) can be a vital solution. MIN supports multiple identifiers such as identity, content, services, geographic information, and IP address in the network layer. The entire network is divided into hierarchical domains from top to bottom, with the top-level domain being multilaterally co-governed by countries that maintain a decentralized consortium blockchain. Regional organizations independently govern the other domains under the root domain[10]. This hierarchical multi-identifier network architecture allows for seamless integration and interoperability across diverse metaverse platforms, virtual worlds, and applications. Furthermore, MIN's large resolution capability enables the smooth handling of numerous identifiers and facilitates reliable and efficient communication among a wide range of metaverse entities. These features make MIN an indispensable component for supporting the evolving metaverse and its complex communication needs.

In this paper, we first propose a novel Web 3.0 architecture based on MIN for Web 3.0-empowered Metaverse, i.e., MIN-Web 3.0. It aims to enhance the connectivity and functionality of the metaverse by leveraging the capabilities of MIN and Web 3.0 technologies, providing a robust and scalable framework for metaverse entities to engage in secure and trusted communication. On the other hand, the risk of untrustworthy data input in the blockchain has gained significant attention. To address this issue, within the proposed MIN-Web 3.0 architecture, we further design a decentralized blockchain oracle model. With its decentralized design and consensus mechanism, MetaOracle ensures the availability of reliable data sources and enables high throughput for data processing. The performance of the proposed MetaOracle is compared with one of the most representative decentralized oracles, Chainlink, in terms of time overhead and throughput performance. The results demonstrate that MetaOracle outperforms Chainlink when a larger number of data request transactions need to be handled in the oracle. The main contributions of this paper are summarized as follows.

- We propose a novel Web 3.0 architecture based on MIN, MIN-Web 3.0, which provides a solid foundation for building the infrastructure and standards required for Web 3.0-empowered metaverse applications.
- Within the proposed MIN-Web 3.0 architecture, we further design a decentralized blockchain oracle model, MetaOracle, to enhance the reliability of the data from the outside world.
- To evaluate the functionality and performance of the proposed scheme, we compare MetaOracle with one of the most representative decentralized oracles, Chainlink. The experimental results demonstrate the advantages of MetaOracle, as it allows for a relatively low time overhead in return for remarkably higher throughput when handling a larger volume of requests for reliable data.

The rest of this paper is organized as follows. In Section 2, we introduce related works of the decentralized oracle models. Then, the design details of MIN-Web 3.0 architecture are formally described in Section 3. We present the core mechanism of our MetaOracle in Section 4. Section 5 demonstrates experimental results compared with another representative work in terms of time investment and throughput. Lastly, we conclude our work and the future outlook in Section 6.

2 Related Works

This section outlines the work related to decentralized oracle networks. Given the critical importance of trustworthy decentralized oracles for future blockchain applications, some studies have been conducted to enhance the trust and reliability of oracles. For example, LO et al.[11] developed a comprehensive framework to evaluate the reliability of diverse blockchain oracles based on trust. Their research provided valuable insights into the dependability of these oracles. Expanding on this topic, MA et al.[12] introduced an innovative decentralized oracle system that prioritizes reliability. Their proposal included specialized mechanisms designed to effectively verify and resolve disputes arising between blockchain smart contracts and oracles. By incorporating these mechanisms, they aimed to enhance the overall credibility of the decentralized oracle system. In a related study, HEISS et al.[13] delved into the concept of trust within decentralized oracles, specifically examining its significance in on-chain data interactions.

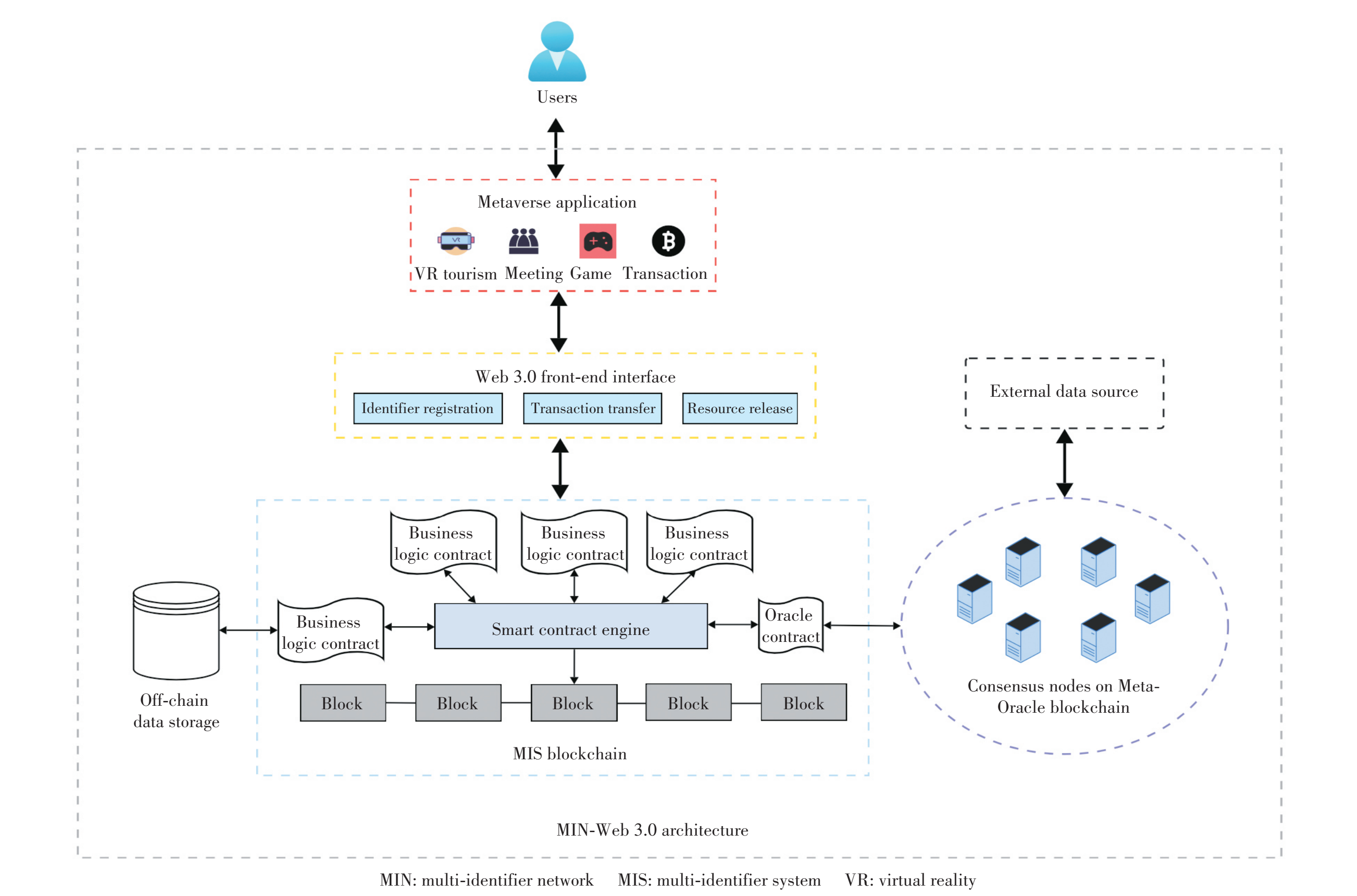
Moreover, a few oracle solutions implemented in the industry can effectively address the issue of data reliability. For instance, Chainlink serves as a decentralized oracle network which has a large market share in the decentralized oracle industry. In Chainlink, off-chain reporting (OCR) is adopted as the underlying mechanism for oracle nodes to collectively aggregate their observations into a single report of the blockchain[14]. Within the Chainlink network, a leader node is selected periodically. The leader node is responsible for regularly requesting follower nodes to provide their recently signed observations. These observations are then aggregated by the leader node to form a comprehensive report. To achieve consensus, a quorum of follower nodes must approve the report's validity by sending their own signed copies back to the leader node. Once the leader node receives the signed copies from a quorum of followers, it assembles a final report that includes the signatures of the approved quorum. This final report is then broadcast to all followers and reported to the smart contract on the blockchain. Notably, Chainlink completes the data aggregation off-chain and generates only one aggregated block in each round. As a result, individual node spends far less on gas costs.

In application, Chainlink emerges as one of the most representative oracles for connecting smart contracts with real-world data in the Web 3.0 ecosystem. For example, Chainlink provides Enjin, a virtual goods and gaming platform, with real-time game item prices and market data[15], thereby enabling fair and secure transactions within the Web 3.0 gaming ecosystem. However, challenges such as high aggregation time costs have been identified.

3 MIN-Web 3.0 Architecture for Web 3.0-Empowered Metaverse

This section provides a comprehensive overview of the MIN-Web 3.0 architecture, where the technologies of Web 3.0 are integrated into MIN. As shown in Fig. 1, the improved MIN architecture consists of five modules: metaverse application, front-end, blockchain, off-chain data storage, and the oracle. These modules work collaboratively to establish a robust and high-performing system for decentralized applications.

At the core of the MIN-Web 3.0 architecture, MIN serves as



▲**Figure 1. MIN-Web 3.0 architecture for Web 3.0-empowered Metaverse** the underlying network layer, enabling the parallel coexistence of multiple identifiers, including identity, content, and geographic information[16]. The MIN network facilitates the generation, management, and resolution services of these identifiers, which greatly supports the deployment of consortium blockchain technology to achieve decentralization. The combination of MIN and Web 3.0 principles creates a powerful foundation for building decentralized metaverse applications that promote user autonomy and data sovereignty.

3.1 Front-End Module

Web 3.0 front-end refers to Web applications built using a new generation of Web technologies, especially the front-end interface of decentralized applications (DApps). Compared with traditional Web 2.0 applications, the Web 3.0 front-end uses a more decentralized architecture, as well as more powerful blockchain and cryptocurrency technologies. The application of these technologies enables the front end to achieve higher security, transparency, and trustworthiness.

To facilitate interaction with distributed applications, Web 3.0 front-ends must establish connectivity with blockchain networks by employing JavaScript libraries of Web 3.0 version, mainly including Web3.js and ethers.js, for seamless communication. In general, the Web 3.0 front end represents a novel approach to metaverse application interfaces, which is based on blockchain and cryptocurrency technology, with higher security, decentralization, and transparency.

3.2 Blockchain Module

As previously stated, blockchain is a decentralized distributed ledger technology that can be used to record transactions, store data, and execute smart contracts, providing some of the foundational services for Web 3.0 applications.

3.2.1 Multiple Identifier System for Web 3.0-Empowered Metaverse

Under the MIN-Web 3.0 architecture, we adopt the multi-identifier system (MIS) blockchain as the underlying blockchain service provider. MIS consists of multiple nodes to constitute a blockchain system, and each node can be managed by an independent organization or individual. MIS records a global state of multi-identifiers, including identity, content, service, space, IP address, and domain names[17]. Only the nodes that have successfully registered an identity identifier in MIS are allowed to engage in the consensus process.

The consensus algorithm, also referred to as the consensus mechanism, is a collaborative process within a distributed system for achieving agreement among multiple nodes. Within the blockchain, the consensus algorithm plays a crucial role in ensuring the security and credibility of the blockchain network. MIS adopts the Parallel Proof of Vote (PPoV) algorithm[18], which is considered a novel Byzantine Fault Tolerance (BFT) consensus algorithm for consortium blockchains. Its underlying mechanism is that the transaction can be stored in the blockchain ledger only when the number of affirmative votes in each block exceeds two-thirds of all voters.

The PPoV consensus algorithm guarantees data consistency among various nodes and enables BFT within the system. It demonstrates the characteristics of decentralization, tamper-proof data, and reduced reliance on trust. Moreover, the system's data throughput capacity is enhanced as it permits multiple accounting nodes to generate blocks in parallel during a consensus cycle.

3.2.2 Contracts in Web 3.0 Blockchain

Within the MIN-Web 3.0 architecture, there are mainly two types of smart contracts running on the blockchain module: one is the business logic contract, and the other is the oracle contract.

At the technical level, the business logics of the metaverse application are commonly written in smart contracts and executed by the smart contract engine on the blockchain. In other words, the contracts for business logic serve as the underlying infrastructure for enforcing the predefined transaction logic within the metaverse. By leveraging the blockchain technology, the execution of business logic becomes decentralized, transparent, and secure. The design for this type of contract is driven by the specific business need. For simplification, smart contracts that retrieve off-chain data storage can be categorized as business logic contracts.

On the other hand, the oracle contracts are responsible for invoking the oracle services to bring external data onto the blockchain and make it accessible to the business logic contracts. By calling the oracle contract, the business logic contract can obtain trustworthy and up-to-date information for making informed decisions and executing transactions. The oracle contracts are categorized into three types, namely consumer contracts, proxy contracts, and aggregator contracts.

The consumer contract is exposed to the business logic contract when a user wants to request specific data from the oracle service, while the proxy contract serves as middleware between the consumer contract and the aggregator contract which will be introduced later. Proxy further points to the aggregator for a particular data feed. Using proxy enables the underlying aggregators to be upgraded without any service interruption to consumer contracts. As a result, the design of proxy provides remarkable flexibility in managing and upgrading the pre-oracle network, allowing for smooth integration of enhancements as required. The most underlying layer connecting to the oracle network is the aggregator contract. According to the pre-defined interfaces, it receives periodic aggregated data updates from oracle and stores the updated data on the MIS blockchain. It is worth noting that the aggregator contract cannot directly send a data request to oracle, while it can only periodically receive the data feed from oracle.

Regarding the frequency of the data updates from oracle, two types of thresholds are set, i.e., the value threshold and the time threshold. When a node in the oracle network identifies that the latest detected values from data providers deviate from the value on the MIS blockchain by more than the defined deviation threshold, which means when the condition of value threshold is satisfied, a new aggregation round starts. On the other hand, when a specified amount of time has elapsed since the last update, the updated data will be pushed to the MIS blockchain.

3.3 Off-Chain Data Storage Module

The off-chain data storage module in blockchain enables the storage of data outside the blockchain while leveraging the smart contract capabilities of the blockchain for data access and management. The module plays a crucial role in addressing the challenges of limited storage capacity and high storage costs in blockchain. With the inherent limitations of blockchain's storage capacity, storing an extensive number of data can result in a significant increase in storage expenses. Furthermore, the public nature of blockchain data raises concerns regarding privacy and security, as it allows unrestricted access.

The inter planetary file system (IPFS) is a peer-to-peer distributed file system that utilizes content addressing to identify and retrieve files[19]. When a file needs to be stored off-chain, its content is hashed using the SHA-256 cryptographic hash function to obtain a unique content identifier (CID). The smart contract can later retrieve the data using this CID.

Considering the advantages of greater storage capacity and improved privacy protection offered by IPFS, we integrate an IPFS-based off-chain data storage scheme into our MIN-Web 3.0 architecture. Through the invocation of smart contracts, developers can conveniently access and manage data stored in off-chain data storage modules, thereby facilitating more efficient data management and interaction within the system.

3.4 Metaverse Application Module

Based on the blockchain technology, Metaverse applications are virtual world applications that transform real-world people, objects, scenes, and other elements into digital forms through digital identity, digital assets, smart contracts, etc. These applications facilitate interaction within the virtual world. With the emergence of the blockchain technology, metaverse applications have evolved towards decentralization and openness, becoming a significant component of the digital economy. Under the MIN-Web 3.0 architecture, the properties of MIN's security protection and performance optimization capabilities, along with virtual reality and the blockchain technology, enable the creation of a more real and immersive metaverse experience. Moreover, it offers a secure, transparent, and efficient solution to asset trading and management in financial scenarios.

3.5 Oracle Module

Our MetaOracle is deployed on the blockchain to conduct the data consensus, which means there are two blockchains involved in the MIN-Web3.0 architecture: MIS blockchain and MetaOracle blockchain. Further details regarding the MetaOracle blockchain are provided in Section 4.

4 Core Mechanisms of MetaOracle

In this section, we present a decentralized blockchain oracle combined with the PPoV consensus mechanism, MetaOracle, which serves as a decentralized oracle for trustworthy data feeds in Web 3.0-empowered metaverse applications.

4.1 Roles of Participants

MetaOracle includes three types of roles: the aggregator, the bookkeeper, and the voter. They collaborate to acquire the ultimate trustworthy data from external data sources.

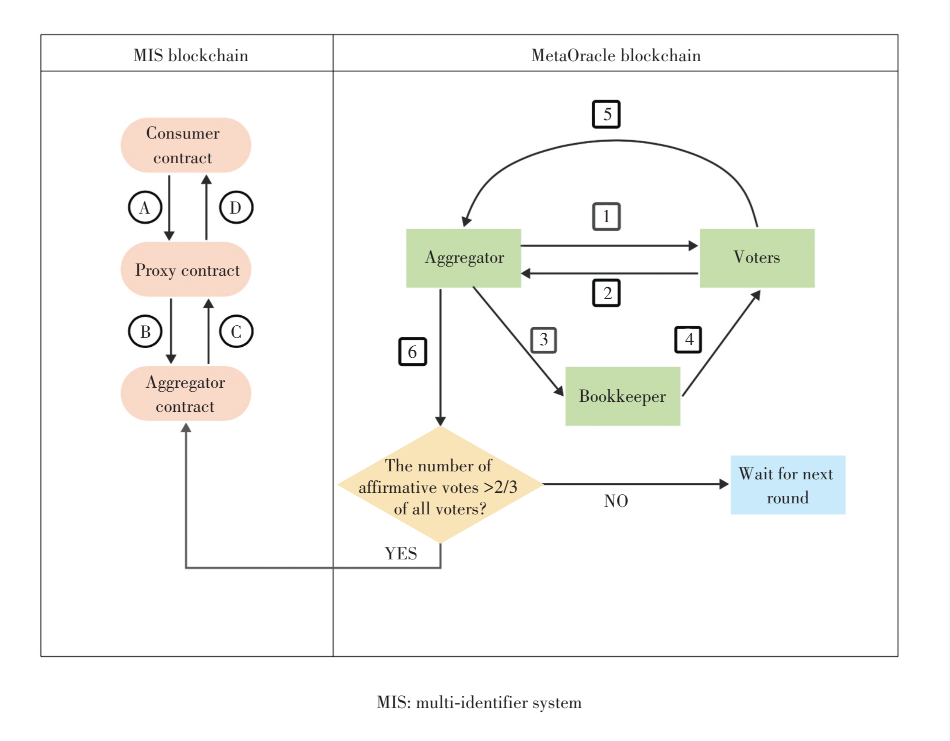
- Aggregator: An aggregator is responsible for aggregating the data collected by voter nodes on the MetaOracle blockchain. Each voter node can access various external data sources to retrieve specific data based on the predefined interfaces between the MIS blockchain and the MetaOracle blockchain. When a new round of data aggregation is triggered by meeting the threshold condition, the aggregator will request voter nodes to provide recently signed observations. These observations are then aggregated to the proper results by the aggregator according to a median or average principle. The aggregator then sends this result to the bookkeeper for generating a block.
- Bookkeeper: A bookkeeper is responsible for generating the blocks for transactions. After packaging a block for an aggregated result, the bookkeeper releases it to the network for votes.
- Voter: A voter node is responsible for two tasks within a round of data feed. One task is to fetch the information from the external world, and the other is to validate and vote on the block carrying aggregated results. The voting rule aligns with the aforementioned value threshold. In this rule, a positive vote is cast when the aggregated value deviates from the value that voters hold in this round by a margin smaller than the specified deviation threshold. The voting message contains the hash value of each block, an opinion indicating agreement or disagreement (−1, 0, 1), and the voter's signature information. By monitoring both the signature of the voter and their behavior, it is possible to detect and identify any malicious nodes within the network. Only when the number of affirmative votes in each block surpasses 2/3 of all voters, the block can be committed to the blockchain.

4.2 Whole Process of Retrieving Trustworthy Data

The overall process of retrieving trustworthy data from the MetaOracle blockchain to the MIS blockchain is described in Fig. 2.

4.2.1 MIS Blockchain Phase

A user initiates a request for external data through the con-



MIS: multi-identifier system

▲**Figure 2. Overflow of data feed**

sumer contract, and the request is first forwarded to the proxy contract before being sent to the aggregator contract. When the request reaches the aggregator contract, the aggregator responds with the up-to-date data stored on the MIS blockchain to the consumer contract through proxy. The request and response process is illustrated in Fig. 2 as Steps A to D.

4.2.2 MetaOracle Blockchain Phase

The MetaOracle blockchain is a crucial component for retrieving data from real-world data sources and reaching consensus on that data. If either the value threshold or the time threshold condition is met, a new round of data feed will be initiated. During a data feed round, requests for different data feeds may occur simultaneously. The aggregated result for each request can be regarded as a transaction. A substantial number of transactions can be packaged into a single block during a consensus round within the MetaOracle blockchain. MetaOracle can process these transactions together, thus reducing the overall processing time. On the other hand, the PPoV consensus algorithm is adopted to enable multiple bookkeepers to generate blocks concurrently, thereby enhancing the throughput of the MetaOracle blockchain.

Step 1: When conditions for different data feeds are met, the aggregator initially asks voters to retrieve the data from the external world.

Step 2: At certain intervals, the aggregator waits for and receives the data feeds from voters.

Step 3: The aggregator aggregates the data collected, and obtains proper results according to a median or average principle. Notably, the results may include the responses corresponding to different data feed requests. The aggregator then sends the results to bookkeepers for block generation.

Step 4: Each bookkeeper independently generates a block with a specific transaction allocation rule. The transaction allocation rule ensures that transaction pools of bookkeepers are distinct, resulting in unique blocks generated by each bookkeeper. The bookkeeper packages the aggregated results into a block and further broadcasts the block to the network for votes.

Step 5: After collecting the blocks from the bookkeepers, a voter casts individual votes for each block and creates a comprehensive voting message to send to the aggregator.

Step 6: The aggregator consistently waits for voting messages and counts the results. When the number of affirmative votes in each block surpasses 2/3 of all voters, the block can be confirmed, and the results will be pushed to the aggregator contract on the MIS blockchain. Otherwise, the consensus cannot be reached. The data feed can be carried over to the next round of aggregation until the threshold condition is met.

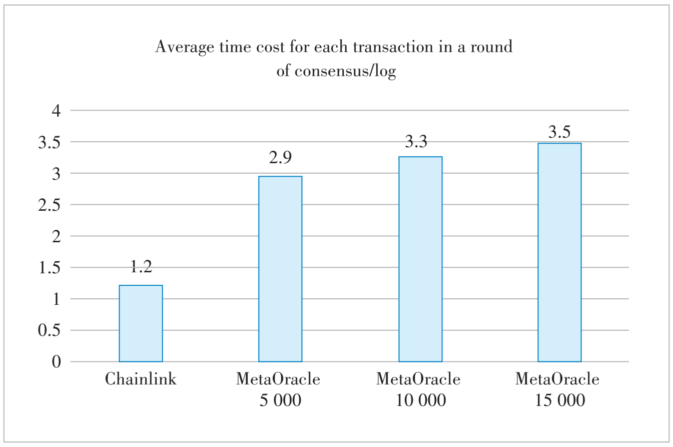
5 Experimental Analysis

In this section, we evaluate the performance of our proposed scheme. Our experiment uses 4 Linux physical machines and their operating systems are Ubuntu20.04. Each of them has a memory of 8G and 4 physical CPUs, and is interconnected through fiber optic Ethernet. The CPU is Inter(R) Core (TM) i5-8500 CPU@ 3.00 GHz. To adhere to the BFT mechanism, the total number of network nodes N must satisfy $N \geqslant 3f + 1$ where f refers to the number of faulty nodes. In our MetaOracle blockchain setup, we have established 28 nodes, consisting of three different roles: the aggregator, the bookkeeper, and the voter. In the experiment, we conducted tests on MetaOracle to measure its transaction processing capacity, represented as the number of transactions it can handle transactions per second (TPS), as well as the time required for the consensus process. We conducted these tests under three different conditions, specifically when executing 5 000, 10 000, and 15 000 transactions in a consensus round within our MetaOracle. For comparative analysis, we also collected corresponding data from Chainlink[14], a well-known decentralized oracle, as a reference point.

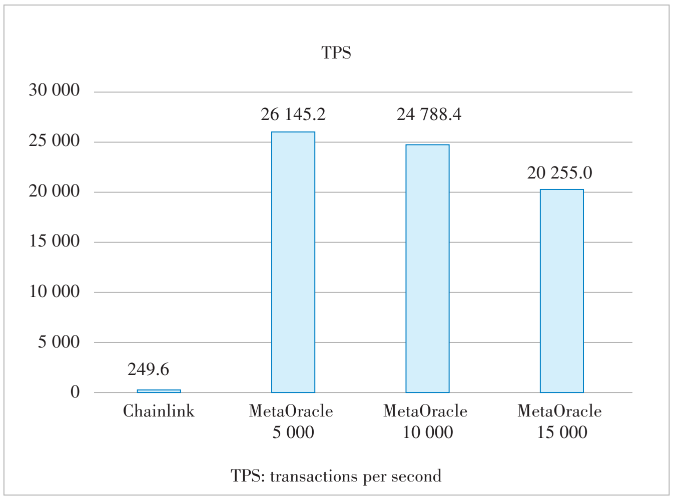
We initially compared the proposed MetaOracle with Chainlink in terms of the average time cost for each transaction in a round of consensus. Chainlink employs the Schnorr signature scheme to conduct off-chain consensus, eliminating the need for block generation during the consensus process. In other words, this approach reduces the time required for block production. To facilitate the analysis, we applied a logarithmic scale to the time cost in Fig. 3. In practice, the time cost for each transaction in a round of consensus in the Chainlink oracle is approximately 1 s lower than that of our proposed oracle.

However, the off-chain blockchain characteristic of Chainlink has a dual impact when it comes to a large number of transactions. If a large number of transactions are not packaged together in a block for transmission, the time required to transmit each transaction can be influenced by network latency. MetaOracle can package a bundle of transactions into a block and broadcast it on the blockchain to achieve consensus. This allows MetaOracle to handle a considerable number of transactions within a consensus round.

As it is depicted in Fig. 4, Chainlink's capability to handle TPS is limited to less than 250 due to the negative impact of the off-chain consensus mechanism mentioned earlier. In contrast, our MetaOracle achieves a minimum TPS of 20 000 when handling 1 5000 transactions within a round of consensus, surpassing Chainlink by a factor of at least 80. However, it is important to note that the TPS performance of MetaOracle may exhibit a peak value. Therefore, the TPS value of MetaOracle may vary when the block size changes from 5 000 to 15 000. The factors contributing to these phenomena can be considered as potential directions for future research in the field of oracles. As a result, our proposed MetaOracle achieves significantly higher throughput compared with Chainlink, albeit with a slight sacrifice of 1 s in latency.



▲**Figure 3. A comparison of the average time cost for each transaction in a round of consensus process between Chainlink and MetaOracle. The time cost data have been logarithmically scaled for ease of observation. MetaOracle 5 000 refers to the condition when MetaOracle handles 5 000 transactions within a round of consensus, and with a block size of 5 000. In contrast, since Chainlink does not generate blocks during consensus, we only compare Chainlink when it handles a single transaction**



▲**Figure 4. A comparison of TPS between Chainlink and MetaOracle**

6 Conclusions

In summary, this paper proposes a new MIN-Web 3.0 architecture for secure communications in Web 3.0-empowered metaverse applications, and a decentralized blockchain oracle called MetaOracle, which can enhance the reliability and security of data feed on the MIS blockchain while maintaining its low time overhead and high throughput. We also compare MetaOracle with Chainlink to demonstrate its effectiveness. In the future, we will undertake real-world case studies to examine the behavior and response strategies of blockchain oracles in diverse attack scenarios, such as the Sybil attack and collusion attack, to enhance the robustness of our scheme. Additionally, further research on the factors that influence the TPS performance of MetaOracle will be explored to further optimize its throughput and scalability.

References

[1] WOOD G. What is Web 3? Here's how future Polkadot founder Gavin Wood explained it in 2014 [EB/OL]. (2022-01-04) [2024-04-19]. https://cryptonews.net/news/altcoins/2974191/

[2] WANG Y T, SU Z, ZHANG N, et al. A survey on metaverse: fundamentals, security, and privacy [J]. IEEE communications surveys & tutorials, 2023, 25(1): 319 – 352. DOI: 10.1109/COMST.2022.3202047

[3] POTTS J, RENNIE E. Web3 and the creative industries: how blockchains are reshaping business models [M].A research agenda for creative industries. London: Edward Elgar Publishing, 2019. DOI: 10.4337/9781788118583.00013

[4] ZHENG Z B, XIE S A, DAI H N, et al. An overview of blockchain technology: architecture, consensus, and future trends [C]//International Congress on Big Data (BigData Congress). IEEE, 2017: 557 – 564. DOI: 10.1109/BigDataCongress.2017.85

[5] WANG S, YUAN Y, WANG X, et al. An overview of smart contract: architecture, applications, and future trends [C]//Intelligent Vehicles Sym-

posium (IV). IEEE, 2018: 108 – 113. DOI: 10.1109/IVS.2018.8500488

[6] CALDARELLI G. Understanding the blockchain Oracle problem: a call for action [J]. Information, 2020, 11(11): 509. DOI: 10.3390/info11110509

[7] Provable. Provable Documentation [EB/OL]. [2024-04-19]. https://docs.provable.xyz/

[8] SOBER M, SCAFFINO G, SPANRING C, et al. A voting-based blockchain interoperability Oracle [C]//International Conference on Blockchain (Blockchain). IEEE, 2021: 160 – 169. DOI: 10.1109/Blockchain53845.2021.00030

[9] AL-BREIKI H, REHMAN M H U, SALAH K, et al. Trustworthy blockchain Oracles: review, comparison, and open research challenges [J]. IEEE access, 2020, 8: 85675 – 85685. DOI: 10.1109/ACCESS.2020.2992698

[10] LI H, WU J X, YANG X, et al. MIN: Co-governing multi-identifier network architecture and its prototype on operator's network [J]. IEEE access, 2020, 8: 36569 – 36581. DOI: 10.1109/ACCESS.2020.2974327

[11] LO S K, XU X W, STAPLES M, et al. Reliability analysis for blockchain Oracles [J]. Computers & electrical engineering, 2020, 83: 106582. DOI: 10.1016/j.compeleceng.2020.106582

[12] MA L M, KANEKO K, SHARMA S, et al. Reliable decentralized Oracle with mechanisms for verification and disputation [C]//The 7th International Symposium on Computing and Networking Workshops (CANDARW). IEEE, 2019: 346 – 352. DOI: 10.1109/CANDARW.2019.00067

[13] HEISS J, EBERHARDT J, TAI S. From Oracles to trustworthy data on-chaining systems [C]//International Conference on Blockchain. IEEE, 2019: 496 – 503. DOI: 10.1109/Blockchain.2019.00075

[14] Chainlink. Chainlink 2.0 Whitepaper [EB/OL]. [2024-03-06]. https://naorib.ir/white-paper/chinlink-whitepaper.pdf

[15] Enjin. Enjin Coin [EB/OL]. [2024-03-06]. https://enjin.io/enjin-coin

[16] WANG Y M, LI H, HUANG T, et al. Scalable identifier system for industrial Internet based on multi-identifier network architecture [J]. IEEE Internet of Things journal, 2023, 10(3): 1919 – 1932. DOI: 10.1109/JIOT.2021.3137526

[17] LYU Q, LI H, LIN X N, et al. H-MIS: A hierarchical multi-identifier system based on blockchain [C]//International Conference on Big Data. IEEE, 2023: 2326 – 2333. DOI: 10.1109/BigData59044.2023.10386505

[18] WANG Z X, LI H, WANG H, et al. A data lightweight scheme for parallel proof of vote consensus [C]//IEEE International Conference on Big Data. IEEE, 2021: 3656 – 3662. DOI: 10.1109/BigData52589.2021.9671637

[19] CHEN Y L, LI H, LI K J, et al. An improved P2P file system scheme based on IPFS and Blockchain [C]//International Conference on Big Data. IEEE, 2017: 2652 – 2657. DOI: 10.1109/BigData.2017.8258226

Biographies

CHEN Rui is currently pursuing the master degree at the School of Electronic and Computer Engineering, Peking University, China. Her research interests focus on blockchain.

LI Hui (lih64@pkusz.edu.cn) is a professor of Shenzhen Graduate School, Peking University, China. He received his BE and MS degrees from School of Information Engineering, Tsinghua University, China in 1986 and 1989, respectively, and PhD degree from the Department of Information Engineering, The Chinese University of Hong Kong, China in 2000. He is the Director of Shenzhen Key Lab of Information theory & Future Internet Architecture, and Director of PKU Lab of China Environment for Network Innovations (CENI), National Major Research Infrastructure.

LI Wuyang is currently pursuing his master degree at the School of Electronic and Computer Engineering, Peking University,China. His research interests focus on blockchain.

BAI He received her BE degree from the School of Information Engineering, Zhengzhou University, China in 2019. She is currently pursuing her PhD degree at the School of Electronic and Computer Engineering, Peking University, China. Her research interests focus on congestion control and transport protocol.

WANG Han is currently pursuing her PhD degree at the School of Electronic and Computer Engineering, Peking University, China. Her research interests focus on blockchain and metaverse.

WU Naixing works as a professor-level senior engineer in China United Network Communications Co., Ltd. He received his PhD degree from the Department of Computer Application Technology, Beijing University of Posts and Telecommunications, China in 2000.

FAN Ping works as a senior engineer in China United Network Communications Co., Ltd.

KANG Jian works as a senior engineer in China United Network Communications Co., Ltd.

Selwyn DENG works as a senior engineer in China United Network Communications Co., Ltd.

ZHU Xiang works as a senior engineer in China United Network Communications Co., Ltd.

Optimization of High-Concurrency Conflict Issues in Execute-Order-Validate Blockchain

MA Qianli[1], ZHANG Shengli[1], WANG Taotao[1],
YANG Qing[1], WANG Jigang[2]
(1. Shenzhen University, Shenzhen 518000, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202402004

https://kns.cnki.net/kcms/detail/34.1294.TN.20240621.1419.006.html, published online June 21, 2024

Manuscript received: 2024-03-23

Abstract: With the maturation and advancement of blockchain technology, a novel execute-order-validate (EOV) architecture has been proposed, allowing transactions to be executed in parallel during the execution phase. However, parallel execution may lead to multi-version concurrency control (MVCC) conflicts during the validation phase, resulting in transaction invalidation. Based on different causes, we categorize conflicts in the EOV blockchain into two types: within-block conflicts and cross-block conflicts, and propose an optimization solution called FabricMan based on Fabric v2.4. For within-block conflicts, a reordering algorithm is designed to improve the transaction success rate and parallel validation is implemented based on the transaction conflict graph. We also merge transfer transactions to prevent triggering multiple version checks. For cross-block conflicts, a cache-based version validation mechanism is implemented to detect and terminate invalid transactions in advance. Experimental comparisons are conducted between FabricMan and two other systems, Fabric and Fabric++. The results show that FabricMan outperforms the other two systems in terms of throughput, transaction abort rate, algorithm execution time, and other experimental metrics.

Keywords: blockchain; MVCC conflict; reordering; parallel validation; transaction merging

Citation (Format 1): MA Q L, ZHANG S L, WANG T T, et al. Optimization of high-concurrency conflict issues in execute-order-validate blockchain [J]. ZTE Communications, 2024, 22(2): 19 – 29. DOI: 10.12142/ZTECOM.202402004
Citation (Format 2): Q. L. Ma, S. L. Zhang, T. T. Wang, et al., "Optimization of high-concurrency conflict issues in execute-order-validate blockchain," *ZTE Communications*, vol. 22, no. 2, pp. 19 – 29, Jun. 2024. doi: 10.12142/ZTECOM.202402004.

1 Introduction

Blockchain is essentially a form of distributed ledger technology, and the popularity of blockchain technology began with the emergence of Bitcoin[1] . The true reason for this popularity is that blockchain enables peer-to-peer transactions without the need for a trusted third party. With the advent of smart contracts in Ethereum[2], blockchain technology has been extensively researched and applied in various fields such as finance[3], healthcare[4 – 5], supply chain[6], and the Internet of Things[7], leveraging its characteristics of decentralization, immutability, and traceability.

From the perspective of participants, blockchain systems can be divided into permissioned chains and permissionless chains. Permissionless chains, also known as public chains, allow any node to anonymously participate. Due to the unknown identities of the nodes and mutual distrust, such blockchain systems often use proof of work or other consensus mechanisms to solve the Byzantine fault tolerance consensus problem[8]. On the other hand, permissioned chains consist of a group of identity-verified nodes. These systems are often only applied to specific scenarios where the nodes, although not entirely trusting each other, share common goals. Permissioned chains constrain participating nodes and can control the read and write permissions of different nodes, making them more suitable for enterprise-level applications.

However, whether they are permissionless chains like Bitcoin and Ethereum, or permissioned chains like Tendermint and Quorum, most mainstream blockchain systems use active replication[9]: First, transactions are sorted through consensus protocols or atomic broadcast and packaged into blocks for dissemination to nodes; then all nodes execute transactions in sequence, changing their ledger states. We call this system the order-execute (OE) architecture, and its limitation lies in the fact that all nodes must execute all transactions serially in order, which is undoubtedly a limitation on throughput. In order to achieve better parallelism in transaction execution, a new execute-order-validate (EOV) architecture has been proposed. In an EOV system, clients send transaction proposals to multiple nodes for endorsement during the execution phase. The endorsers are only a subset of the nodes in the blockchain network, and different endorsers can endorse different transactions at the same time, enabling the system to execute transactions in parallel. After collecting a sufficient number of endorsements, the client packages all response into a transaction and send it to orderers for block creation. Finally, the orderers send the blocks to all the nodes for validation and synchronization of ledger states. This model utilizes optimistic concurrency control techniques to ensure the consistency of data. However, it may lead to multi-version concurrency control (MVCC)[10] conflicts during the validation phase.

We categorize conflicts in EOV systems into two types: within-block conflicts and cross-block conflicts. Within-block conflicts occur within the same block, where the write set modifications of transactions alter the version numbers of read sets for later-executed transactions, resulting in the invalidation of the latter transactions during the validation phase. Cross-block conflicts occur when the value read by a transaction during the execution phase is invalidated before reaching the validation phase due to modifications made by the submission of other blocks. SHARMA et al.[11] proposed a system called Fabric++ to address within-block conflicts by reordering transactions. However, our testing showed that Fabric++ is inefficient when the transaction conflict rate is high. To address this issue, we made several optimizations to the EOV blockchain based on Fabric v2.4, naming FabricMan. The main contributions of this paper are as follows:

1) We design a reordering algorithm with stable time complexity to reduce within-block conflicts. Experimental results show that our algorithm performs better under high transaction conflict rates compared with Fabric++.

2) Based on the transaction conflict graph generated during reordering, we perform parallel validation of unrelated transactions in the validation phase to leverage the advantages of multi-core CPUs.

3) At the chaincode level, we analyze transactions and merge simple transfer transactions to maximize the validation pass rate.

4) We implement a cache-based version validation mechanism to detect and terminate invalid transactions during the ordering phase, reducing cross-block conflicts.

The rest of the paper is organized as follows: Section 2 introduces the structures of Fabric and Fabric++, as well as other related research. Section 3 provides a theoretical analysis of the problems in Fabric and proposes our findings. Section 4 describes the design of FabricMan. Section 5 presents experimental tests of FabricMan's optimizations and compares them with Fabric and Fabric++. Finally, Section 6 concludes our work.

2 Background and Related Work

2.1 EOV Architecture in Hyperledger Fabric

One of the representative blockchain platforms based on the EOV architecture is Hyperledger Fabric[12], abbreviated as Fabric. All nodes in Fabric are known and authorized at all times and are mainly divided into three types: 1) Clients are responsible for submitting transaction proposals and collecting endorsement responses; 2) peers are responsible for executing and validating transaction proposals, and then committing their write sets to maintaining local ledgers; 3) orderers are responsible for ordering transactions and packaging them into blocks according to predefined rules. The workflow of a transaction consists of three phases: execution, ordering, and validation.

1) Execution phase

During the execution phase, clients send the transaction proposal to a subset of peers (endorsers), according to a predefined policy. Endorsers simulate the execution of transactions in parallel based on the current ledger state, generating the corresponding read and write sets. The read set consists of (key, ver) tuples, and the write set consists of (key, val) tuples, where key is a unique name representing the entry, and ver and val are the latest version number and value of the entity, respectively. After execution, endorsers return the read and write set with their signatures to the client. When a client collects sufficient responses from different endorsers, it can package them into a transaction and send them to the ordering service to enter the next phase.

2) Ordering phase

During the ordering phase, different orderers continuously receive transactions from different clients. The ordering service needs to achieve two goals: a) reaching a consensus on transaction orders, and b) packaging ordered transactions into blocks according to rules and delivering them to all peers. In Fabric v2.4, the Raft protocol is used for achieving crash-fault-tolerant consensus in a). The block creation rules in b) are generally formed by the maximum block interval and the maximum number of transactions included in a block.

3) Validation phase

When a node receives a block from the orderers, it first checks for the presence of signatures and the legality of the block structure. If the check passes, the block is added to a validation queue to ensure it can be added to the blockchain. Then, it goes through the validating state-based endorsement check (VSCC) and MVCC validation stages. In the VSCC stage, the node checks if each transaction in the block meets the specific endorsement policy of the chaincode; if not, the transaction is marked as invalid but remains in the block. In the MVCC stage, all transactions are sequentially checked for multi-version concurrency control. If the version number of a key in the transaction's read set does not match the version number in the current local state, the transaction is marked as invalid. Finally, the node writes the block into its local ledger and modifies the ledger state according to the validity of each transaction.

2.2 Optimization of Fabric++

The vanilla Fabric sorts transactions based on the order in which they arrive at the orderers. While this approach allows for quick ordering, it may lead to unnecessary serialization conflicts. To address the aforementioned issue, SHARMA et al.[11] introduced a reordering algorithm in the ordering phase of Fabric. This algorithm terminates a small number of transactions based on the relationship between the read and write sets of transactions. Subsequently, it constructs a conflict-free ordering for the remaining transactions, thereby increasing the success rate of transactions within a block. The algorithm consists of five main steps as follows. 1) A conflict graph is built based on the read and write sets of all transactions to be sorted. 2) Tarjan's algorithm[13] is used to identify all strongly connected subgraphs and Johnson's algorithm[14] to identify all cycles within these subgraphs. 3) The cycles each transaction is part of are idenified, and the times of each transaction appearing in the cycles are counted. 4) The transactions that appear in the most cycles are sequentially terminated until the conflict graph has no cycles. 5) Finally, a serializable scheduling scheme is established using the remaining transactions.

2.3 Related Work

Currently, optimizations for the EOV blockchain can be broadly categorized into two types:

1) Improving the overall throughput of the system

THAKKAR et al.[15] conducted comprehensive tests on the performance of Fabric v1.0 by configuring parameters such as the block size (BS), endorsement policy, channel, resource allocation, and ledger database. They identified three main performance bottlenecks: endorsement policy validation, validation of the order of transactions in a block, and validation and submission of states in CouchDB. They proposed simple optimizations to address the following issues: a) using a hash map with serialized identities as keys to cache deserialized identities, reducing resource consumption for encryption operations; b) parallel validation of endorsements for multiple transactions to utilize idle CPU resources and improve overall performance; c) batch read and write optimization for CouchDB. These optimizations effectively increase the overall throughput of the system. GORENFLO et al.[16] reengineered Hyperledger Fabric v1.2 by a) passing only transaction IDs instead of entire transactions during ordering, b) actively caching unassembled blocks in committers, parallelizing as many verification steps as possible, c) redesigning the data management layer using an in-memory database instead of the original data storage, and d) separating roles responsible for endorsement and submission. These changes reduce computational and I/O overhead during transaction ordering and validation, increasing transaction throughput from 3 000 transactions per second (TPS) to 20 000 TPS.

2) Reducing read/write conflicts caused by parallel execution

RUAN et al.[17] studied the Fabric++ solution and found that it did not consider dependencies between transactions across blocks, limiting the effectiveness of reordering. They proposed a reordering algorithm based on a more granular concurrency control strategy and verified its safety, resulting in improved reordering effectiveness. SUN et al.[18] analyzed the reordering algorithm implemented in Fabric++ and found issues regarding trust. They proposed a trusted reordering algorithm grounded in a greedy approach.

3 Problem Analysis

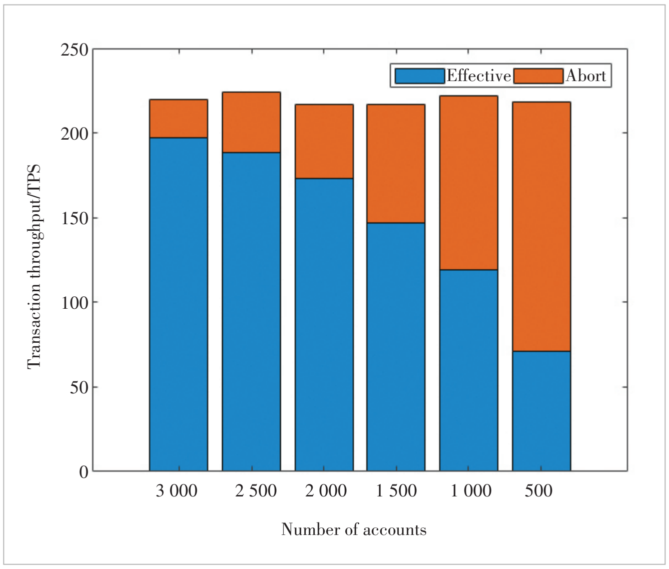
As mentioned in Section 2, Fabric generates read and write sets for transactions while execution. During the validation phase, nodes perform MVCC validation on the read sets based on the current state of the local database. If the versions do not match, the transaction is marked as invalid, and its write set cannot be used to update the ledger state, resulting in an MVCC conflict. To assess the impact of these conflicts on the system, we conducted tests on Fabric using the SmallBank smart contract under the configuration described in Section 5, as shown in Fig. 1.

In our experiments, each block contains 256 transactions. When the total number of accounts is 3 000, the conflict rate is relatively low, resulting in a high TPS for successful transactions, accounting for approximately 90%. However, as the number of accounts decreases, the conflict rate within blocks increases, resulting in a higher rate of transaction abortions. When the total number of accounts is 500, the TPS for successful transactions drops to only 30%. This demonstrates a significant performance decrease when the number of transaction conflicts within blocks increases.

In high-concurrency execution environments, we classify MVCC conflicts in the EOV blockchain into within-block and cross-block conflicts.

3.1 Within-Block Conflicts

Within-block conflicts occur when there are conflicts between different transactions within the same block. When mul-



▲Figure 1. Transaction throughput of Fabric tiple transactions that read or write the same key are grouped into the same block, it may lead to this type of conflict. In the example provided in Table 1, transactions T_1 and T_2 are sequentially packaged into a single block. During the validation phase, T_1 updates key k_1, changing its version number to v_1. Next, T_2 is validated, and its read set includes key k_1 with version v_0. During the MVCC validation, it is discovered that $v_0 \neq v_1$, resulting in T_2 being marked as invalid.

Observation 1: It is possible to reduce the number of conflicts within a block by modifying the validation order of transactions. The fundamental reason for within-block conflicts is that two different transactions perform a write-followed-by-read operation on the same key. By adjusting the order of transactions, we can ensure that they perform read operations before write operations. In the given example, if T_2 is validated before T_1, there would be no conflict.

As mentioned in Section 2.2, Fabric++ uses Johnson's algorithm during reordering, with a time complexity of $O((n+e)(c+1))$, where n is the number of nodes, e is the number of edges, and c is the number of cycles in the graph. While the number of nodes and edges in the conflict graph can be controlled to small values, the number of cycles may be very large. Ref. [19] highlighted a similar issue: when resolving cycles in Fabric++ , recalculating the occurrence count of individual transactions in a cycle results in a time complexity of $O(n^3)$ for the entire algorithm. Considering that topological sorting is an algorithm for ordering graph vertices with a stable time complexity of $O(n+e)$, we can propose a new reordering algorithm based on it. This algorithm can rapidly complete reordering even when transaction conflict rates are high.

Observation 2: The conflict graph generated by reordering can reflect dependency information between transactions, which can be utilized for parallel validation. In the Fabric, validation can be divided into two main stages: VSCC and MVCC. VSCC is used to evaluate whether endorsements in transactions comply with the endorsement policy, and this step is already parallelized in the system. MVCC, on the other hand, is executed sequentially. Ref. [15] pointed out that one of the performance bottlenecks of Fabric is the serial MVCC validation of all transactions within a block. If we parallelize the validation of unrelated transactions by leveraging transaction dependency relationships, we can fully harness the advantages of multi-core CPUs to enhance system performance.

Observation 3: Transfers are one of the primary transaction types, characterized by simple logic and fixed parameters, rendering them suitable for merging. As a permissioned blockchain, obtaining block data from Fabric is challenging. Ref. [20] analyzed transactions on Ethereum over a period of time and found that the main types of transactions leading to conflicts are ERC20 token transactions accounting for 60%, decentralized finance (DeFi) transactions accounting for 29%, and gaming transactions accounting for 10%. Therefore, we believe that the merging of transfer transactions holds significance.

▼Table 1. An example of within-block conflict

Order	Transaction	Read Set	Write Set	Validity
1	T_1	-	$(k_1, v_0 \rightarrow v_1)$	Valid
2	T_2	$(k_1, v_0), (k_2, v_0)$	$(k_2, v_0 \rightarrow v_1)$	Invalid

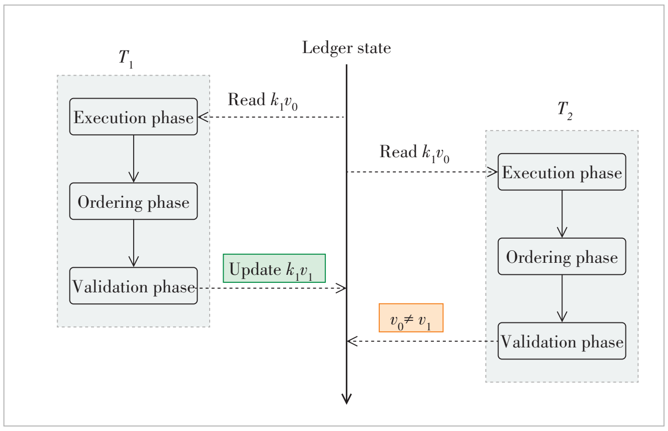
3.2 Cross-Block Conflicts

Due to the nature of the EOV structure, there is a certain delay between the execution and verification. If a transaction in a later block reads a key that was written by a transaction in an earlier block before the earlier block's verification, it can result in a dirty read in the later block, leading to a conflict. As shown in Fig. 2, T_1 and T_2 are two transactions in different blocks. During the execution phase, T_1 reads the current version number v_0 of key k_1. From the verification phase, it can be seen that T_1 modifies key k_1 in its write set, but since this step is a simulated execution, the database state is not altered. Therefore, T_2 still reads version v_0 of k_1. Subsequently, the block containing T_1 enters the verification phase and updates the version number of key k_1 to v_1, resulting in T_2 invalidated. Additionally, under special circumstances such as network congestion, cross-block conflicts can also occur.

Observation 4: Orderers have the opportunity to early abort invalid transactions caused by cross-block conflicts. All transactions arrive at the orderer for block generation. Since the version numbers of keys in the read sets are obtained from the ledger during execution, the version of a key in the ledger at this point must be no lower than the version in the read set. We can utilize a caching mechanism to store versions of keys, thereby filtering out invalid transactions.

4 Design of FabricMan

In the previous section, we have analyzed two types of conflicts in the EOV blockchain and identified four directions for optimization. In this section, we will first introduce our modifications to the ordering phase and then discuss the four modular designs for each direction: transaction reordering, parallel verification, transaction merging, and caching mechanism.



▲Figure 2. An example of cross-block conflict

Our optimization efforts primarily focus on the ordering phase. The orderer receives transactions from multiple clients and merges them into a batch until the conditions for block generation are fulfilled. The conditions consist of two parts: When the number of transactions in the batch reaches a predefined threshold, and then when the time taken to construct the batch reaches the maximum block generation time limit.

Once either condition is met, our system starts processing the batch. All transactions are first filtered through a version cache maintained by the orderers. During this process, the system extracts the read sets of transactions and compares them with the cache. Transactions that do not meet the filtering criteria are aborted and feedbacks are provided to the clients. Transactions are then checked to determine if they are transfer transactions. If so, they are moved to the transfer array and merged with other transfer transactions. The remaining transactions in the batch are non-transfer transactions that undergo reordering and subdivision into subgraphs based on dependency relationships. Finally, the system constructs a new block by incorporating merger transactions, transfer transactions, and the reordered batch. It then adds the transaction subgraphs to the block header and distributes the block to all peers.

4.1 Transaction Reordering

When reordering a transaction set S, it is necessary to identify the dependencies between the transactions to construct a transaction conflict graph. In the graph, if a transaction T_i points to another transaction T_j, it means that the write set of T_i intersects with the read set of T_j, denoted as $T_i \rightarrow T_j$. This indicates that during block ordering, T_j needs to precede T_i, otherwise T_j will be invalidated due to reading outdated versions. When constructing the conflict graph, each node represents a transaction. According to the aforementioned definition, the out-degree of a node indicates the number of other transactions whose validity it affects, while the in-degree indicates the number of other transactions that affect it. Additionally, an important issue to address is the presence of cycles in the graph. Circular graphs cannot be serialized, thus necessitating the use of algorithms to remove certain transactions and convert the original conflict graph into an acyclic graph.

Our algorithm is based on topological sorting, but due to the unsuitability of topological sorting for cyclic graphs, we make some improvements. From a high-level perspective, it mainly consists of five steps as follows. 1) The conflict graph is constructed based on the read-write sets of each transaction in S, and the in-degree and out-degree of each node are recorded. 2) A node n with the minimum in-degree is selected for processing. If multiple nodes meet this criterion, the one with the maximum out-degree is prioritized. If there are still multiple options, the one with the smallest index is chosen. 3) Other nodes pointing to n from the graph are removed until the in-degree of n is 0. 4) N is added to the result queue and removed from the graph. 5) Steps 2), 3), and 4) are repeated until there are no remaining nodes in the conflict graph. Finally, the result queue is reversed to obtain a conflict-free serialized ordering. The pseudo-code of Algorithm 1 implements these five steps.

Note that in Step 2), since we need to reduce the in-degree of n to 0, we prioritize selecting the node with the minimum in-degree to retain more transactions. Furthermore, since the out-degree of a node indicates the number of transactions it will affect after ordering, and the final ordering is the reverse of the ordering queue, i.e., transactions that enter the ordering queue first will be placed at later positions in the algorithm, we prioritize selecting transactions with the maximum out-degree for ordering.

Algorithm 1. Reordering algorithm

```
func ReorderSort(Transaction[ ] S) {
    // Step 1: Construct a transaction conflict graph and an exit and entry table
    Graph cg = buildConflictGraph(S)
    Graph incg = invert cg
    map[Transaction]int indegree = Calculate in-degrees using cg
    map[Transaction]int outdegree = Calculate out-degrees using cg
    // Step 2: Select nodes to be sorted
    while S is not empty:
        for each Transaction tx in S:
            if indegree[tx] < minIndegree:
                min = indegree[tx]
                nodeToSort = node
            else if indegree[tx] == min and outdegree[tx] > outdegree[nodeToSort]:
                nodeToSort = node
        // Step 3: Process nodeToSort so that their degree is 0
        for each nodeToRemove in incg[nodeToSort]:
            if nodeToRemove not in S:
                continue
            remove nodeToRemove from S
            for each tx in cg[nodeToRemove]:
                indegree[tx]--
            for each tx in incg[nodeToRemove]:
                outdegree[tx]--
        // Step 4: Add nodeToSort to the queue and remove from transaction graph
        append nodeToSort to result
        for each tx in cg[nodeToSort]:
            indegree[tx]--
        remove nodeToSort from S
    // Step 5: Return the reverse order of the ordering queue to obtain the ordering result
    return result.invert()
}
```

Here is an example to better understand the algorithm. We assume there are six transactions, T_0 to T_5, within a set S, and their read-write sets are as shown in Table 2. These six transactions access 10 different keys, k_0 to k_9, and 0 indicates that a transaction' s read or write set does not contain a certain key, while 1 indicates that it does. The reordering process is as follows:

1) The first step is constructing a conflict graph for transactions in S based on the read-write sets, as shown in Fig. 3. If T_i points to T_j, it indicates that the write set of T_i shares common keys with the read set of T_j. At this point, the in-degree and out-degree of each node are shown in Table 3.

2) In the first iteration, following step 2, the system selects T_5 as n because it currently has the smallest in-degree among the transactions in S. As the in-degree of T_5 is 0, Step 3 is skipped. Next, the system adds T_5 to the result, removes it from S, and then decrements the in-degree of the node pointed to by T_5, which is T_2.

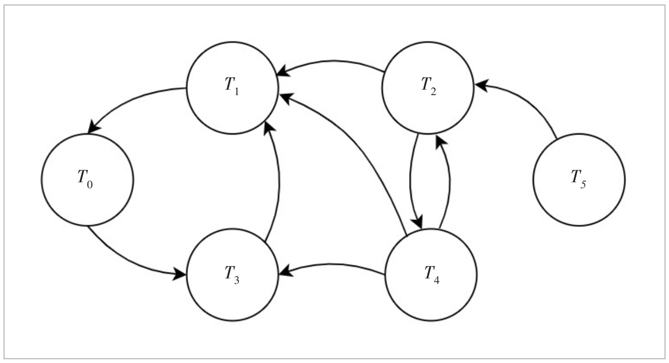
3) In the second iteration, since T_0, T_2 and T_4 all have an in-degree of 1, making them the nodes with the lowest in-degree, the system selects T_4 as n based on Step 2. Then, in Step 3, the system removes the node T_2 from S, reducing the in-degree of T_4 to 0. In Step 4, T_4 is added to the result and removed from S, and then the in-degrees of T_1 and T_3 are each decreased by one.

4) At this point, there remains a cycle consisting of T_0, T_1 and T_3 in the graph. Since their in-degree and out-degree are the same, the system chooses T_0 with the smallest index as n. It removes T_1, adds T_0 to the result, and then adds T_3 to the result. Reversing the result queue, we can get the final sorted result: $T_3 - T_0 - T_4 - T_5$.

The central part of this reordering algorithm exhibits a time complexity comparable to that of topological sorting, which is $O(n + e)$. Moreover, it is not affected by the number of cycles in the graph, eliminating this flaw presented in the Fabric++. It is worth noting that our algorithm does not guarantee the termination of the minimum number of transactions to make the graph acyclic, as this is an NP-hard problem. We merely provide a very lightweight way to terminate a small number of transactions, thereby generating a serializable ordering scheme.

▼Table 2. Read and write sets of the six transactions

Read Set										
T_x	k_0	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9
T_0	1	0	0	0	0	0	0	0	0	0
T_1	0	0	0	1	1	1	0	0	0	0
T_2	0	0	0	0	0	0	1	1	0	0
T_3	0	1	1	0	0	0	0	0	1	0
T_4	0	0	0	0	0	0	0	0	0	1
T_5	0	0	0	0	0	0	0	0	0	0
Wirte Set										
T_x	k_0	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9
T_0	0	0	1	0	0	0	0	0	0	0
T_1	1	0	0	0	0	0	0	0	0	0
T_2	0	0	0	1	0	0	0	0	0	1
T_3	0	1	0	0	1	0	0	0	0	0
T_4	0	0	0	0	0	1	1	0	1	0
T_5	0	0	0	0	0	0	0	1	0	0



▲ **Figure 3. Initial conflict graph formed by the six transactions in a block**

▼**Table 3. Initial in-degree and out-degree of the six transactions**

Transaction	In-Degree	Out-Degree
T_0	1	1
T_1	3	1
T_2	2	2
T_3	2	1
T_4	1	3
T_5	0	1

Another point that needs to be clarified is that the reordering algorithm proposed by RUAN et al.[17] resembles a greedy strategy. The system constructs a conflict graph for all pending transactions. When a new transaction is received, its dependencies are added to the conflict graph. If a cycle is detected, the new transaction is directly dropped, ensuring that transactions in the pending queue do not have circular dependencies. In contrast, Fabric++ and our algorithm are more modular. We construct a conflict graph for all the transactions within a block and then eliminate cycles, ultimately achieving a serializable ordering.

4.2 Parallel Verification

When conducting reordering, the algorithm generates a conflict graph among transactions, as indicated by lines 3 and 4 in Algorithm 1. Here, cg represents the conflict graph, depicted by a two-dimensional array. Each element cg[i] is a one-dimensional array, where the presence of an element j indicates that transaction T_j must occur before transaction T_i, namely $T_i \rightarrow T_j$.

After reordering, the aborted transactions are removed from the conflict graph. The system then performs a depth-first search (DFS) operation on cg to identify their connected components and further partition them into mutually disconnected subgraphs. When the block is generated, the information regarding these subgraphs will be serialized and appended to the block header. During the validation phase, nodes deserialize the information and utilize goroutines to perform MVCC validations on independent subgraphs in parallel.

4.3 Transaction Merging

During the execution phase, we analyze the parameters of chaincode transactions, which allows us to identify simple transfer transactions. Additionally, we include a value field in the structure of the transaction' s read set to represent its initial value.

In the ordering phase, nodes identify transfer transactions under different chaincodes and construct a transfer table for each chaincode. The node adds the initial balance of each account to the Moneymap table and their corresponding version numbers to the Versionmap table. If the sender' s balance is less than the transfer amount, the transaction is aborted. Otherwise, the sender' s balance is decreased by the transfer amount, and the receiver' s balance is increased by the same amount. After processing all transfer transactions, the system utilizes the keys and corresponding version numbers in the Versionmap to form the read set of the merger transaction. Similarly, it utilizes the keys and values in the Moneymap to create the write set of the merger transaction.

Once the merger transaction is constructed, it is positioned at the start of the block during formation, followed by all the merged transfer transactions. Retaining the merged transactions in the block serves two purposes: First, it enables the system to offer feedback to clients concerning the success or failure of the merger transactions during validation; second, it maintains transaction data on the blockchain for future auditing purposes.

4.4 Caching Mechanism

To mitigate the cross-block conflicts mentioned in Section 3.2, we deploy a cache utilizing a hash table at the orderers. This cache is employed to store the keys and version numbers extracted from the read and write sets of received transactions. The cache table consists of three fields: key, version, and updated flag. The key represents the unique entity name in the smart contract chaincode, while the version indicates the latest version number read for the corresponding key in the transaction. The updated flag denotes whether the key may have been updated by a new block.

Upon the arrival of transactions at the orderer, the read sets of each transaction are checked against the cache. If any key is found to have a version lower than that in the cache, the transaction is immediately aborted. This prevents outdated transactions from occupying system resources due to network delays. If a key has a version greater than or not present in the cache, the cache is updated with the latest version. If a key matches the version in the cache, the system checks if it has the updated flag; if so, the transaction is aborted.

After the completion of reordering, the orderer obtains a block containing transactions with no conflicts. Under normal operation, the write sets of this block are applied to update the ledger. At this point, the orderer checks the read sets of each transaction. If the cache contains keys identical to those in the read set, the corresponding values in the cache are marked with the updated flag, indicating that the version is no longer the latest for that key.

However, there is a potential issue with this cache mechanism. If transaction T_1 fails to pass the final validation phase for some unexpected reason (e.g., a signature issue), the version of k_1 in the node' s ledger remains v_0. However, in the orderer, the version of k_1 has already been altered to $v_0 - \mathit{updated}$, causing subsequent transactions reading $k_1 : v_0$ to fail. To address this issue, we introduce a timer in the cache. If a key is not updated within two block intervals, it is removed from the cache. This also ensures that the cache does not indefinitely increase in size.

5 Experimental Evaluation

To validate the improvements proposed in this paper, we conducted experimental evaluations of key metrics such as throughput, transaction abort rate, and algorithm execution time for Fabric, Fabric++, and FabricMan. Since the original Fabric++ code is implemented on Fabric v1.2 while FabricMan is based on Fabric v2.4, for comparison purposes, we also reimplemented Fabric++ based on Fabric v2.4.

5.1 Setup and Workload

The experimental setup involves a single-channel blockchain system comprising two organizations, each containing two peer nodes deployed via docker containers. The consensus mechanism used is Raft, and LevelDB serves as the state database. The experiments were conducted on a server with a 36-core CPU (Intel Core i9-10980XE 3.0 GHz), 256 GB RAM, running on Ubuntu 20.04.5 LTS.

Two types of workloads were employed in the experiments: Smallbank and custom chaincode, assessed through the caliper-benchmarks framework. The Smallbank contract creates a checking account and a savings account for each user and includes six functions. In the custom chaincode, we defined complex read-write transactions that read the balances of four accounts and modify two of them.

In Section 3, Smallbank is used to test Fabric for measuring the transaction throughput under different numbers of accounts. The transaction conflict rates corresponding to different numbers of accounts are shown in Table 4. In subsequent experiments, we continued to use these account numbers to measure the system' s performance under different transaction conflict rates.

5.2 Impact of Block Size

BS is one of the important factors affecting the throughput

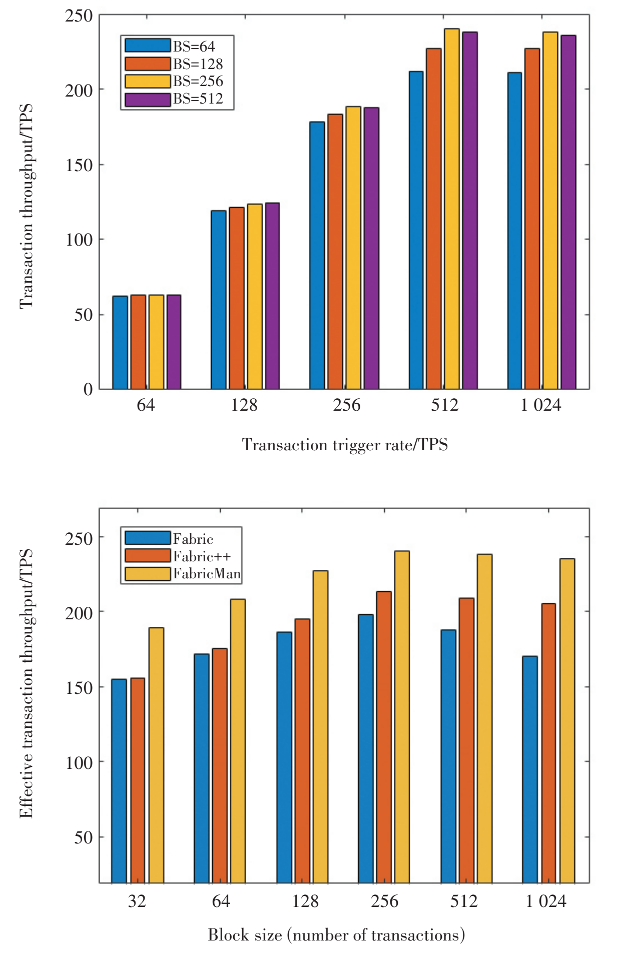
▼Table 4. Number of accounts and corresponding conflict rates

Number of Accounts	3 000	2 500	2 000	1 500	1 000	500
Conflict rate/%	10.5	14	20.3	32.3	46.4	67.8

and latency of a blockchain. We tested the impact of the transaction trigger rate on FabricMan, and the impact of changing the block size on the throughput of Fabric, Fabric++, and FabricMan. We used the Smallbank contract and tested the performance of the systems in a low-conflict environment when the number of accounts was set to 3 000. The results are shown in Fig. 4.

It is found that as the transaction trigger rate increases, the throughput of FabricMan also gradually increases, reaching saturation at around 240 TPS when the transaction trigger rate is 512 TPS. As the block size increases from 64 to 256, the system throughput also increases, but it decreases when the block size is set to 512. This is because increasing the number of transactions in a block leads to more conflicting transactions, reducing the throughput of successful transactions, and larger blocks also increase the transmission time in the system.

We can also observe that as the number of transactions in a block increases, the difference in throughput of successful transactions between FabricMan and Fabric++ compared to Fabric becomes larger. This is because as the number of transactions in the block increases, the probability of conflicts also increases, making the effect of reordering more pronounced. The throughput of all three systems reaches its peak when the block size is set to 256. Therefore, in subsequent experiments, we use 256 as the number of transactions included in a block, and set the maximum block interval to one second, and the transaction trigger rate to 512 TPS.



▲Figure 4. Impact of transaction trigger rate of FabricMan (up) and that of block size of three systems (down)

5.3 Comparison of Reordering Algorithms

To evaluate the performance of the reordering algorithms, we prepared multiple pre-packaged blocks (each containing 256 transactions) and applied the reordering algorithms of FabricMan and Fabric++ separately. Their execution time, the number of valid transactions in each block, and the throughput of valid transactions were compared. In this experiment, we used complex read-write transactions from custom chaincode to increase the conflict rate between transactions, aiming to better evaluate the performance of both algorithms.

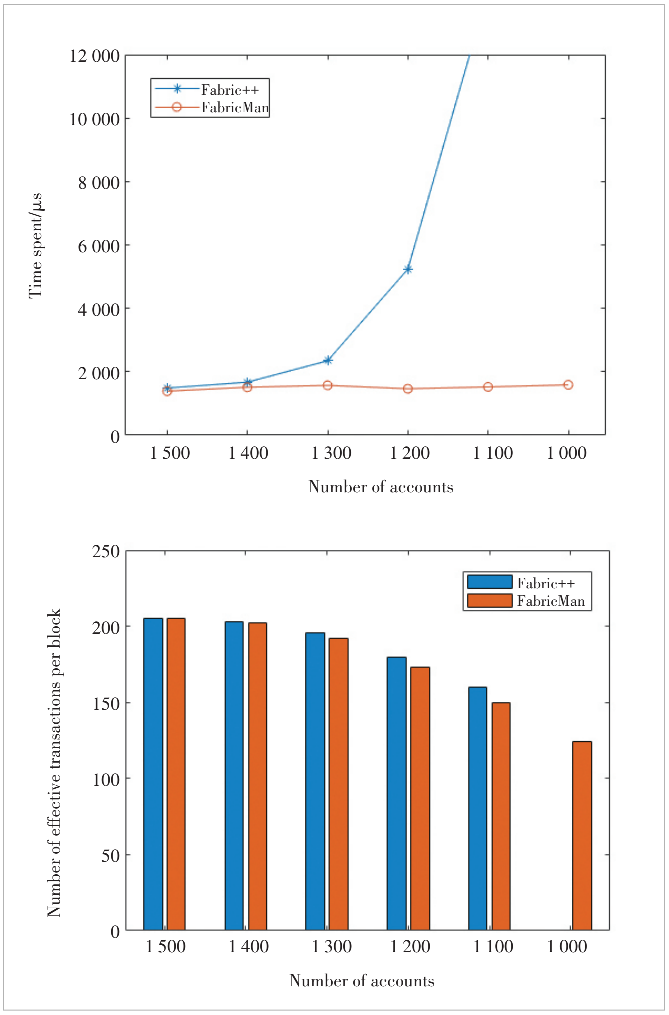
We controlled the number of accounts to gradually decrease from 1 500 to 1 000. Fig. 5 shows the time taken by both the algorithms and the number of valid transactions within each block. It can be observed that the time required by Fabric++ for reordering increases significantly as the conflict rate increases, and it becomes unable to generate blocks when the number of accounts reaches 1 000. In contrast, the algorithm of FabricMan remains stable at around 1 500 us. However, since FabricMan's reordering algorithm does not select nodes with the most cycles in each round like Fabric++, the final number of successful transactions is slightly lower than that in Fabric++. Nevertheless, its stable time complexity allows it to generate blocks normally even in high-conflict environments with 1 000 or fewer accounts.

The comparison of throughput between the two systems is shown in Fig. 6, where FabricMan only uses the optimization of the reordering. It can be seen that FabricMan consistently outperforms Fabric++ in throughput across different settings of the number of accounts. This is because, in the range of 1 500 to 1 400 accounts, where the conflict rate is relatively low, both reordering algorithms yield a comparable number of valid transactions within the same block. However, FabricMan's reordering algorithm has shorter execution times. As the number of accounts decreases below 1 400, the conflict rate increases significantly. While Fabric++'s algorithm can produce more valid transactions, the time it takes for execution increases significantly.

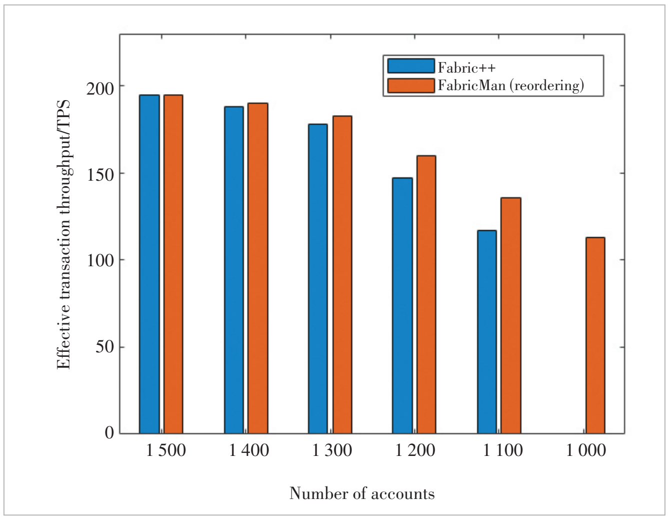
5.4 Effect of Parallel Verification

We also tested the impact of assigning different numbers of CPU cores to FabricMan on parallel verification time. The extensive use of CPU resources for identity encryption and decryption operations in permissioned blockchains could affect our experimental results. We used multiple pre-packaged blocks and conducted modular tests on MVCC verification time with validation nodes having CPU core counts ranging from 1 to 16.

The experiments employed the Smallbank contract and varied the number of accounts (AC) to represent different conflict rates. To ensure verification of the same number of transactions at different conflict rates, we only used the subgraph partitioning algorithm instead of reordering during block generation. The test results, as depicted in Fig. 7, indicate a significant reduction in verification time with an increase in the number of cores used. However, beyond 8 cores, the reduction in time becomes less pronounced, as this stage becomes bottlenecked by interactions with the database and the serial verification of the longest transaction conflict chain. Moreover, as the block conflict rate increases, resulting in longer conflict chains, the corresponding verification time also increases.



▲**Figure 5. Comparison of two algorithms in execution time and effective transaction quantity**

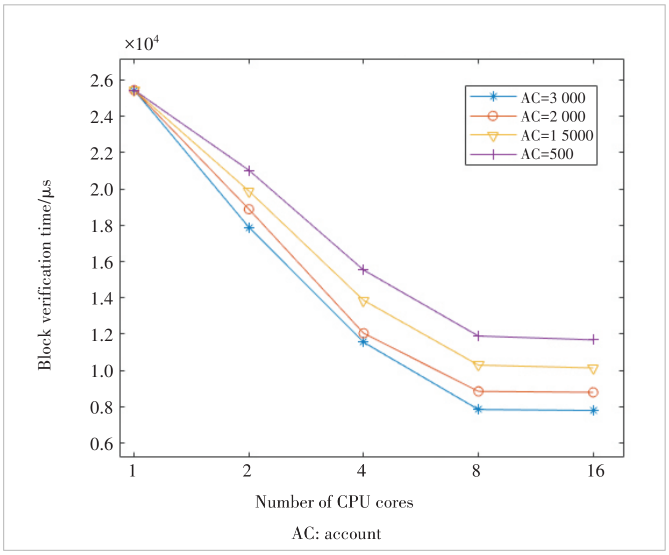


▲**Figure 6. Throughput of two systems under different account numbers**

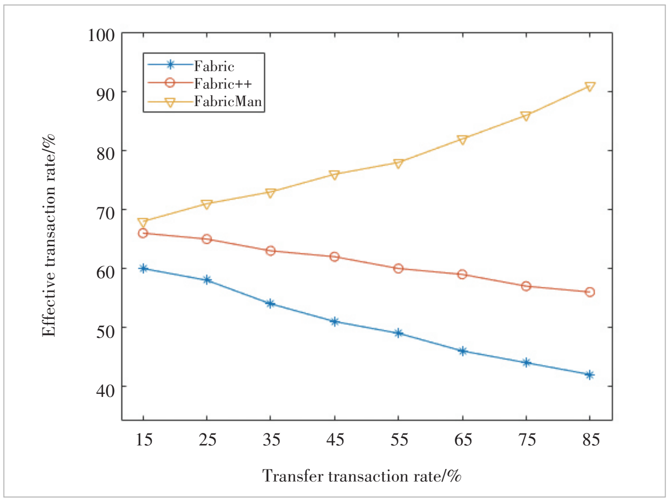
5.5 Effect of Transaction Merging

In this section, we used 1 000 accounts to send non-transfer or transfer transactions in the Smallbank contract. Non-transfer transactions cannot be merged, while transfer transactions can. We gradually increased the proportion of transfer transactions from 15% to 85%. The proportion of successful transactions in the Fabric, Fabric++, and FabricMan systems is shown in Fig. 8.

When the proportion of transfer transactions is low, Fabric++ and FabricMan have a higher successful transaction rate due to reordering. However, as the proportion of transfer transactions increases, the successful transaction rate in FabricMan increases because of the transaction merging mechanism. When the proportion of transfers reaches 85%, over 90% of transactions in FabricMan can be successfully submitted. In contrast, as the read-write set of transfers in Smallbank is more complex than others, the number of MVCC conflicts in Fabric and Fabric++ increases, leading to a decrease in the successful transactions rate.



▲**Figure 7. Block verification time under different numbers of CPU cores**



▲ **Figure 8. Effective transaction rates within the three systems with different transfer transaction rates**

5.6 Combinations of Optimizations

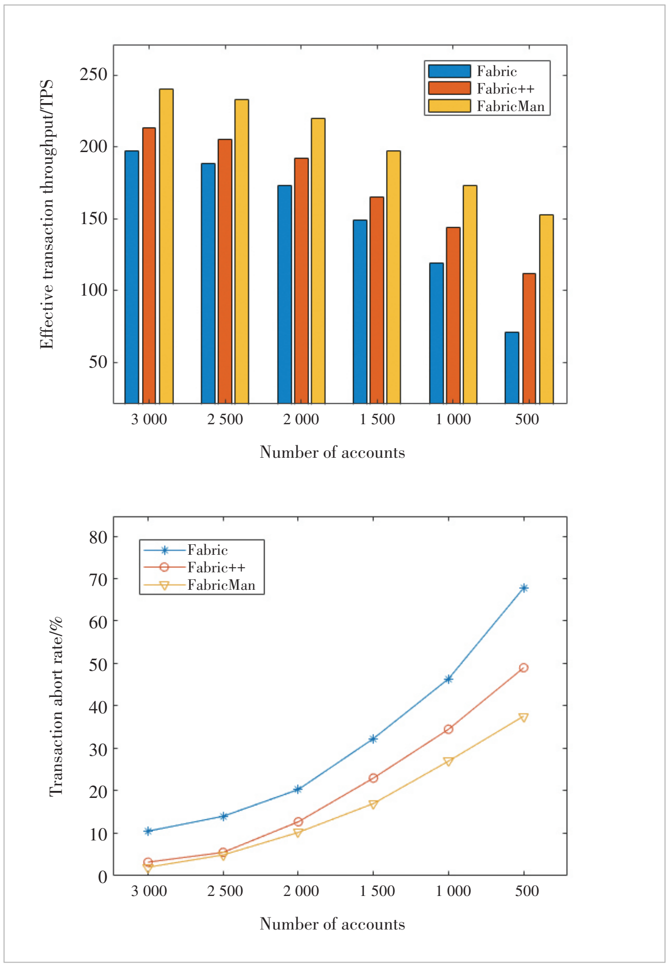
Finally, we conducted comprehensive performance testing of the systems. We used the Smallbank contract as the workload and applied all optimizations mentioned in Section 4 to FabricMan. We tested the system under different numbers of accounts and compared the throughput and transaction abort rate with Fabric and Fabric++, as shown in Fig. 9.

The results demonstrate that when the number of accounts is 3 000 and the transaction abort rate is low, Fabric exhibits the lowest throughput among the three systems, at less than 200 TPS. Fabric++, benefiting from reordering, achieves a slightly higher throughput of around 215 TPS. In contrast, FabricMan, leveraging both reordering and parallel validation along with the merging of transfer transactions, achieves the highest throughput of approximately 240 TPS.

As the number of accounts gradually decreases from 3 000 to 500, resulting in an increasing transaction abort rate, Fabric experiences a significant decline in effective transaction throughput, dropping to less than half of its initial level. Meanwhile, FabricMan experiences a slower decrease in effective transaction throughput, with the lowest transaction abort rate. Even in high-concurrency conflict environments, FabricMan can maintain relatively high throughput.

6 Conclusions and Future Work

We mitigate the performance impact of MVCC conflicts arising from concurrent execution in the innovative EOV blockchain by introducing a comprehensive blockchain architecture named FabricMan. This architecture addresses both within-block and cross-block conflicts while enabling parallel validation and transaction merging. Through testing, FabricMan has demonstrated superior performance in terms of throughput, transaction abort rate, and execution time compared to the baseline schemes. However, there are several areas for future improvement in our work. First, the reordering of transactions within a block may potentially compromise the fairness of the system. In future work, we plan to analyze this issue and introduce appropriate parameters into the algorithm to address it. Second, our experiments were conducted using Docker on a single server, which could not effectively simulate factors such as communication latency present in real networks. The problem of cross-block conflicts was not adequately addressed, so it was not discussed in the experimental phase. In future work, deploying the system in a multi-node environment can provide a better understanding of this issue and facilitate further discussion.



▲ **Figure 9. Comparison of throughput and transaction abortion rates of three systems**

References

[1] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system [EB/OL]. (2008-10-31)[2024-03-15]. https://nakamotoinstitute.org/library/bitcoin

[2] BUTERIN V. A next generation smart contract & decentralized application platform [R]. Ethereum white paper, 2014

[3] QIN K H, ZHOU L Y, GERVAIS A. Quantifying blockchain extractable value: how dark is the forest? [C]//Proc. IEEE Symposium on Security and Privacy (SP). IEEE, 2022: 198 – 214. DOI: 10.1109/SP46214.2022.9833734

[4] AZARIA A, EKBLAW A, VIEIRA T, et al. MedRec: using blockchain for medical data access and permission management [C]//Proc. 2nd International Conference on Open and Big Data (OBD). IEEE, 2016: 25 – 30. DOI: 10.1109/OBD.2016.11

[5] FAN K, WANG S Y, REN Y H, et al. MedBlock: efficient and secure medical data sharing via blockchain [J]. Journal of medical systems, 2018, 42(8): 136. DOI: 10.1007/s10916-018-0993-7

[6] ABEYRATNE S A, MONFARED R P. Blockchain ready manufacturing supply chain using distributed ledger [J]. International journal of research in engineering and technology, 2016, 5(9): 1 – 10. DOI: 10.15623/IJRET.2016.0509001

[7] DAI H N, ZHENG Z B, ZHANG Y. Blockchain for internet of things: a survey [J]. IEEE internet of things journal, 2019, 6(5): 8076 – 8094. DOI: 10.1109/JIOT.2019.2920987

[8] VUKOLIĆ M. The quest for scalable blockchain fabric: proof-of-work vs. BFT replication [M]//Open problems in network security. Cham: Springer International Publishing, 2016: 112 – 125. DOI: 10.1007/978-3-319-39028-4_9

[9] CHARRON-BOST B, PEDONE F, SCHIPER A. Replication: theory and practice [M]. Berlin Heidelberg: Springer, 2010

[10] PAPADIMITRIOU C H, KANELLAKIS P C. On concurrency control by multiple versions [J]. ACM transactions on database systems, 9(1): 89 – 99. DOI: 10.1145/348.318588

[11] SHARMA A, SCHUHKNECHT F M, AGRAWAL D, et al. Blurring the lines between blockchains and database systems: the case of hyperledger fabric [C]//Proc. 2019 International Conference on Management of Data. ACM, 2019: 105 – 122. DOI: 10.1145/3299869.3319883

[12] ANDROULAKI E, BARGER A, BORTNIKOV V, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains [C]//Proc. Thirteenth EuroSys Conference. ACM, 2018: 1 – 15. DOI: 10.1145/3190508.3190538

[13] TARJAN R. Depth-first search and linear graph algorithms [C]//Proc. 12th Annual Symposium on Switching and Automata Theory. IEEE, 1971: 114 – 121. DOI: 10.1109/SWAT.1971.10

[14] JOHNSON D B. Finding all the elementary circuits of a directed graph [J]. SIAM journal on computing, 1975, 4(1): 77 – 84. DOI: 10.1137/0204007

[15] THAKKAR P, NATHAN S, VISWANATHAN B. Performance benchmarking and optimizing hyperledger fabric blockchain platform [C]//Proc IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS). IEEE, 2018: 264 – 276. DOI: 10.1109/MASCOTS.2018.00034

[16] GORENFLO C, LEE S, GOLAB L, et al. FastFabric: scaling hyperledger fabric to 20000 transactions per second [J]. International journal of network management, 2020, 30(5): e2099. DOI: 10.1002/nem.2099

[17] RUAN P C, LOGHIN D, TA Q T, et al. A transactional perspective on execute-order-validate blockchains [C]//Proc. 2020 ACM SIGMOD International Conference on Management of Data. ACM, 2020: 543 – 557. DOI: 10.1145/3318464.3389693

[18] SUN Q C, YUAN Y Y, GUO T, et al. A trusted solution to hyperledger fabric reordering problem [C]//Proc. 8th International Conference on Dependable Systems and Their Applications (DSA). IEEE, 2021: 202 – 207. DOI: 10.1109/DSA52907.2021.00031

[19] WU H B, LIU H, LI J. FabricETP: a high-throughput blockchain optimization solution for resolving concurrent conflicting transactions [J]. Peer-to-peer networking and applications, 2023, 16(2): 858 – 875. DOI: 10.1007/s12083-022-01401-9

[20] GARAMVOLGYI P, LIU Y X, ZHOU D, et al. Utilizing parallelism in smart contracts on decentralized blockchains by taming application-inherent conflicts [C]//Proc. IEEE/ACM 44th International Conference on Software Engineering (ICSE). IEEE, 2022: 2315 – 2326. DOI: 10.1145/3510003.3510086

Biographies

MA Qianli (maqianli@foxmail.com) received his BE degree in software engineering from University of Electronic Science and Technology of China in 2020. He is currently working toward his ME degree in electronic information engineering from Shenzhen University, China. His research focuses on blockchain.

ZHANG Shengli received his BE degree in electronic engineering and ME degree in communication and information engineering from University of Science and Technology of China in 2002 and 2005, respectively, and PhD degree with the Department of Information Engineering, The Chinese University of Hong Kong, China in 2008. After that, he joined Communication Engineering Department, Shenzhen University, China, where he is currently a full professor. He has authored or coauthored more than 20 IEEE top journal papers and ACM top conference papers, including *IEEE Journal on Selected Areas in Communications*, *IEEE Transactions on Wireless Communications*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Communications*, and *ACM Mobicom*. His research interests include blockchain, physical layer network coding, and wireless networks.

WANG Taotao received his PhD degree in information engineering from The Chinese University of Hong Kong (CUHK), China in 2015, MS degree in information and signal processing from Beijing University of Posts and Telecommunications, China in 2011, and BS degree in electrical engineering from University of Electronic Science and Technology of China in 2008. He joined the College of Information Engineering, Shenzhen University, China, as a tenure-track assistant professor in 2016 and was promoted as a tenured associate professor in 2021.

YANG Qing received his BE degree (Hons.) from Huazhong University of Science and Technology, China and PhD degree from The Chinese University of Hong Kong, China. In 2018, he joined as an assistant professor at the College of Electronics and Information Engineering, Shenzhen University, China and the Principal Researcher at the Blockchain Technology Research Center, Shenzhen University.

WANG Jigang received his PhD degree in computer science from Harbin Engineering University, China in 2007. From May 2007 to June 2009, he held a postdoctoral position in Institute of Computer Science, Tsinghua University. From August 2009, Dr. WANG has been with Cyber Security Product Line, ZTE Corporation as general manager. His recent research interests include operating systems, network and information security, and artificial intelligence.

Utilizing Certificateless Cryptography for IoT Device Identity Authentication Protocols in Web3

WU Zhihui[1,2], HONG Yuxuan[1], ZHOU Enyuan[3], LIU Lei[1], PEI Qingqi[1]

(1. Guangzhou Institute of Technology, Xidian University, Guangzhou 510700, China;
2. Guangzhou Lianrong Information Technology Co. Ltd., Guangzhou 510700, China;
3. The Hong Kong Polytechnic University, Hong Kong 999077, China)

DOI: 10.12142/ZTECOM.202402005

https://kns.cnki.net/kcms/detail/34.1294.TN.20240621.1417.004.html, published online June 21, 2024

Manuscript received: 2024-03-25

Abstract: Traditional methods of identity authentication often rely on centralized architectures, which poses risks of computational overload and single points of failure. We propose a protocol that offers a decentralized approach by distributing authentication services to edge authentication gateways and servers, facilitated by blockchain technology, thus aligning with the decentralized ethos of Web3 infrastructure. Additionally, we enhance device security against physical and cloning attacks by integrating physical unclonable functions with certificateless cryptography, bolstering the integrity of Internet of Thins (IoT) devices within the evolving landscape of the metaverse. To achieve dynamic anonymity and ensure privacy within Web3 environments, we employ fuzzy extractor technology, allowing for updates to pseudonymous identity identifiers while maintaining key consistency. The proposed protocol ensures continuous and secure identity authentication for IoT devices in practical applications, effectively addressing the pressing security concerns inherent in IoT network environments and contributing to the development of robust security infrastructure essential for the proliferation of IoT devices across diverse settings.

Keywords: blockchain; certificateless cryptography; identity authentication; IoT

Citation (Format 1): WU Z H, HONG Y X, ZHOU E Y, et al. Utilizing certificateless cryptography for IoT device identity authentication protocols in Web3 [J]. ZTE Communications, 2024, 22(2): 30 – 38. DOI: 10.12142/ZTECOM.202402005

Citation (Format 2): Z. H. Wu, Y. X. Hong, E. Y. Zhou, et al., "Utilizing certificateless cryptography for IoT device identity authentication protocols," *ZTE Communications*, vol. 22, no. 2, pp. 30 – 38, Jun. 2024. doi: 10.12142/ZTECOM.202402005.

1 Introduction

In the era of Web3 and the metaverse, the Internet of Things (IoT) plays a pivotal role in shaping the landscape of digital connectivity and immersive experiences[1 – 3]. As decentralized networks and blockchain-based technologies redefine the way we interact with digital platforms, the IoT acts as a fundamental enabler, bridging the physical and digital realms[4 – 5]. By seamlessly integrating a myriad of interconnected devices, sensors, and actuators into the digital fabric, the IoT facilitates real-time data exchange, automation, and smart decision-making within the metaverse environment. Moreover, in the context of Web3, where user sovereignty and data ownership are paramount, the IoT empowers individuals to leverage their connected devices to assert control over their digital identities and assets securely. Whether it is enhancing virtual experiences through augmented reality devices or enabling smart environments that adapt to users' preferences in the metaverse, the IoT emerges as a cornerstone technology, driving innovation and connectivity in the Web3 and metaverse era.

Among the various measures to ensure IoT security, identity authentication is crucial as it lays the foundation for the rapid and healthy development of IoT applications and is key to maintaining network security. Identity authentication in IoT applications primarily ensures the legitimacy of devices through effective verification methods, establishing trust relationships between devices and ensuring secure data

This work was supported by the National Key Research and Development Program of China under Grant No. 2021YFB2700600, the National Natural Science Foundation of China under Grant No. 62132013, the Key Research and Development Programs of Shaanxi under Grant Nos. S2024-YF-YBGY-1540 and 2021ZDLGY06-03, the Basic Strengthening Plan Program under Grant No. 2023-JCJQ-JJ-0772, the Key-Area Research and Development Program of Guangdong Province under Grant No. 2021B0101400003, Hong Kong RGC Research Impact Fund under Grant Nos. R5060-19 and R5034-18, Areas of Excellence Scheme under Grant No. AoE/E-601/22-R, and General Research Fund under Grant Nos. 152203/20E, 152244/21E, 152169/22E and152228/23E.

communication. Moreover, it effectively prevents malicious devices from accessing IoT systems, averting potential security incidents and ensuring the safe and reliable operation of IoT systems. Therefore, strengthening research on IoT identity authentication technology is of significant practical importance for safeguarding IoT security and promoting its healthy development.

To ensure communication security, certificateless cryptography technology has been widely applied in the design of identity authentication schemes. This cryptographic system, with its notable advantages of not requiring key management and simplifying certificate management processes, has garnered significant attention and active research from academia and industry worldwide. Furthermore, the combination of blockchain technology with the IoT is considered a crucial development trend. The distributed nature of blockchain is highly suitable for meeting the network access requirements of IoT devices in dynamic environments. Additionally, blockchain' s traceability provides potential avenues for privacy protection and accountability of IoT devices. These characteristics of blockchain technology, particularly its data storage and distributed architecture, provide a technical foundation for achieving efficient, secure identity authentication, and trusted access for IoT devices.

However, existing identity authentication schemes based on certificateless cryptography and blockchain technology still have shortcomings that prevent them from meeting authentication requirements in IoT scenarios. These include significant computational and communication overheads, making them unsuitable for resource-constrained IoT scenarios, inadequate defense against common malicious attacks such as physical/cloning attacks, and insufficient support for key security features such as dynamic anonymity.

This paper addresses the security and efficiency issues facing current IoT device identity authentication. For single-device authentication scenarios, a novel trusted IoT device identity authentication protocol is proposed, combining certificateless cryptography for secure and efficient identity authentication between IoT devices and introducing blockchain technology for trustworthy data storage and traceability within the authentication system. Initially, authentication services are shifted from centralized trusted authorities to edge devices, with edge authentication gateways and servers assuming identity authentication responsibilities, thereby decentralizing the authentication architecture. Furthermore, the protocol integrates physical unclonable functions with certificateless cryptography to safeguard device secret values against malicious attacks, ensuring the integrity of device signatures. Finally, dynamic anonymity in the authentication process is achieved through fuzzy extractor technology, enabling the continuous change of users' pseudonymous identities while maintaining the consistency and security of digital signatures, thus enhancing user anonymity.

2 Related Work

2.1 Certificate Based Identity Authentication Schemes

Identity authentication schemes can be categorized into centralized Public Key Infrastructure (PKI)-based schemes and decentralized schemes based on Distributed Public Key Infrastructure (DPKI). Traditional PKI relies on centralized authorization and authentication, often leading to single points of failure, particularly in IoT environments where numerous certificate issuance requests might overwhelm central CA servers, impacting service availability. DPKI-based schemes address these shortcomings by leveraging distributed infrastructures like blockchain, granting users complete control over their digital identities while ensuring privacy protection. For instance, SAMIR et al.[6] proposed Decentralized Trustworthy-Self-Sovereign Identity Management (DT-SSIM), a framework utilizing Shamir' s secret sharing scheme, blockchain, and smart contracts for identity sharing management, integrity checks, and user identity verification. Similarly, YIN et al.[7] presented a distributed IoT identity scheme integrating IoT devices as lightweight blockchain nodes and incorporating a dual-certificate model based on commitments and Bullet-Proofs for privacy protection. BAO et al. [8] proposed a blockchain-based identity management scheme for industrial IoT, ensuring identity authenticity, blindness, unlinkability, traceability, revocability, and public verifiability. Moreover, VERMA et al.[9] introduced an efficient aggregation signature scheme for industrial IoT, improving performance, especially in computational overhead and execution time, crucial for resource-constrained IoT devices. Despite DPKI' s advancements in enhancing user key security and reducing private key transmission needs, certificate authentication and management remain challenges due to resource-intensive operations like revocation, storage, distribution, and authentication. Further research and optimization are required to enhance DPKI' s applicability, especially in resource-constrained IoT scenarios such as smart homes and vehicular networks.

2.2 Identity Authentication Schemes Based on Certificateless Cryptography

AL-RIYAMI and PATERSON introduced Certificateless Public Key Cryptography (CL-PKC) to address key management issues in identity-based cryptographic systems[10]. CL-PKC eliminates the need for certificate authentication while resolving key management problems by involving a Key Generation Center (KGC) that generates partial private keys for users. DING et al. designed an anonymous identity authentication scheme based on an elliptic curve and certificateless signature technology, suitable for resource-constrained IoT devices, effectively resisting impersonation attacks[11]. WANG et al. developed a reliable and efficient certificateless signature scheme using blockchain technology and smart contracts to address potential issues such as man-in-the-middle attacks and KGC compromise[12]. LI et al. proposed a lightweight authentication scheme for Information-Physical Energy Systems (IPES) combining elliptic curve cryptography and certificateless cryptography[13]. In vehicular ad hoc networks (VANETs), WANG et al. introduced a certificateless anonymous revocable authentication protocol for vehicle-to-vehicle communication[14]. ZHOU et al. presented a privacy-preserving identity authentication protocol based on certificateless aggregate signature schemes, while ALI et al. proposed an Enhanced Lightweight and Secure Certificateless Authentication Scheme (ELWSCAS)[15-16]. IQBAL et al. proposed a Certificateless Aggregate Signature (CLAS) scheme based on super elliptic curve cryptography[17]. These schemes balance authentication efficiency and security, offering alternatives suitable for scenarios where key management burdens are intolerable.

3 Methodology

3.1 System Design

3.1.1 System Model

As is shown in Fig. 1, the identity authentication system proposed in this paper mainly consists of the following three types of entities: Trusted Authority (TA), Edge Authentication Gateway (EAG), and IoT Terminal Device (TD). Each entity is described as follows:

TA, as an authoritative entity with abundant computational and storage resources, is the highest authority node in the system, responsible for generating and publishing the system’s public parameters. TA also manages the identities of TD, including generating pseudonym identity markers, enabling the traceability of real identities, and functionalities such as identity revocation.

EAS, as the honest node, has stronger computing and storage capabilities than EAG and is mainly responsible for verifying the legitimacy of the aggregated signatures forwarded by EAG.

TD is generally considered as an untrusted node with relatively limited computational and storage resources. TD requires identity authentication when accessing networks or data.

3.1.2 System Assumption

Based on reasonable assumptions, this paper proposes an authentication scheme for IoT devices based on a distributed architecture. The assumptions are as follows:

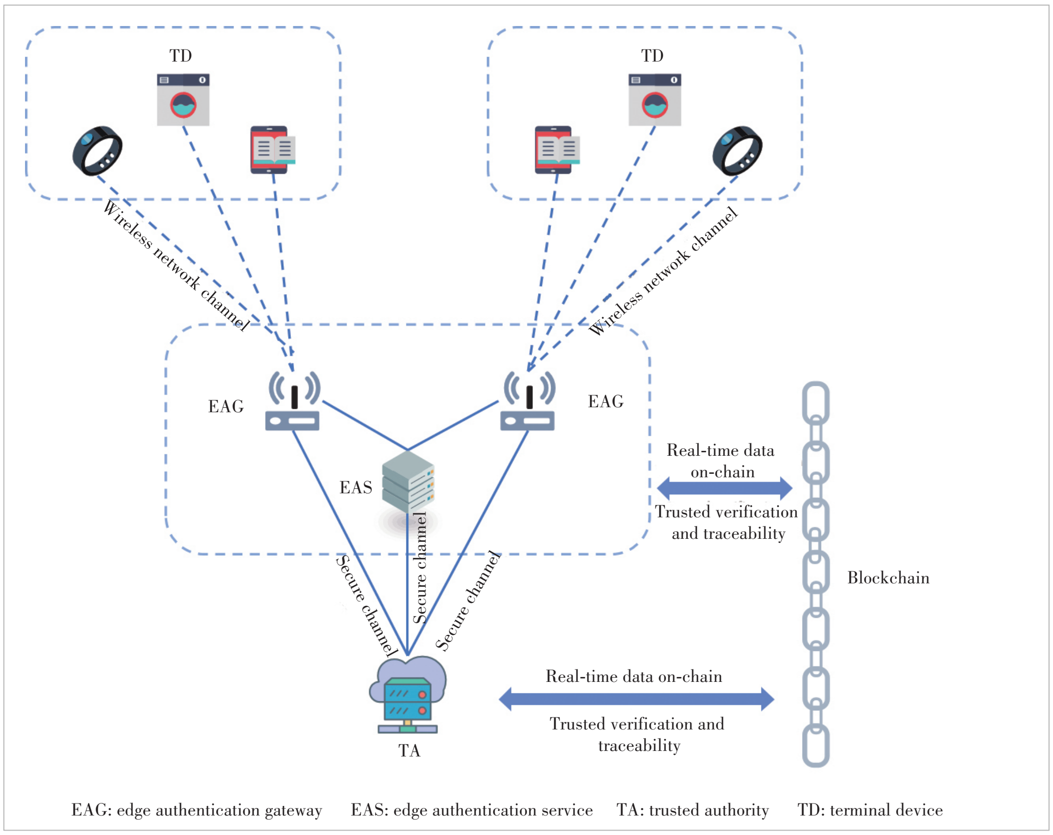
1) Trusted authority organizations are legitimate and absolutely trustworthy.

2) System initialization and key distribution phases are conducted in a secure communication environment, preventing malicious attackers from stealing relevant communication data during these phases. However, during the authentication phase, malicious attackers might still be able to eavesdrop, forge, or tamper with the transmitted messages.

3) All IoT devices are embedded with Physical Uncloneable Function (PUF) chips, and there is no need to employ error correction mechanisms to ensure the unclonability and tamper-proof nature of PUFs.

3.2 Scheme Description

The identity authentication protocol proposed in this paper mainly includes ten parts: system initialization, secret value generation, pseudonym identity generation, partial private key generation, signature generation, signature verification, batch signature



▲Figure 1. System architecture

generation, batch signature verification, pseudonym identity update, and identity revocation. The following section details the specific algorithmic implementations of these components.

3.2.1 System Initialization Phase

The system initialization phase is carried out by TA, mainly used for generating public parameters and a master key for the system. When inputting the security parameters, TA randomly selects a large prime number q that satisfies $q > 2^{\lambda}$ and an elliptic additive cyclic group G of order q over the finite field F_q. P is the generator of the group G. TA chooses six secure hash functions $H_i: \{0,1\}^* \rightarrow Z_q (i = 1, 2, \cdots, 6)$ and randomly chooses $msk \in Z_q$ as the system master secret key and calculates the corresponding public key $MPK = msk \cdot P$. Then, TA chooses a fuzzy extractor $F_{EXT} = (Gen, Rep)$, where Gen and Rep respectively represent the key generation algorithm and the key reproduction algorithm of the fuzzy extractor. Finally, TA obtains the system public parameters $Params = (q, G, P, MPK_{TA}, H_1, H_2, H_3, H_4, H_5, H_6, F_{EXT})$ and broadcasts them within the system.

3.2.2 Secret Value Generation Phase

In the secret value generation phase, TD generates a secret incentive value and its public key based on the secret incentive value and the PUF function. TD determines the secret challenge value c_{TD} and generates the response value $r_{TD} = PUF(c_{TD})$ based on PUF. Then, TD computes $s_{TD} = H_1(r_{TD})$ as its secret key and calculates the corresponding public key $S_{TD} = s_{TD} \cdot P$.

Different from existing solutions, TD does not directly store the private key s_{TD}, but instead store a secret challenge value c_{TD} determined by themselves. Even if an attacker successfully steals this secret value, they cannot generate an identical private key based on this secret value due to the unclonable and tamper-proof characteristics of PUF. Therefore, this mechanism can effectively resist physical/cloning attacks by attackers on IoT terminal devices.

3.2.3 Pseudonym Identity Generation Phase

During the pseudonym identity generation phase, TD collaborates with TA to generate a pseudonym identity, which is used for subsequent anonymous communication of TD. Assuming TD's real identity marker is RID_{TD}, TD sends its public key S_{TD} along with RID_{TD} to TA to apply for the generation of a pseudonym identity. Upon receiving the pseudonym identity generation request message from TD, TA first verifies the legitimacy of the request from TD: TA checks if it exists in the malicious device list $List_{\text{malicious}}$ maintained by TA. If it is on the list, the request is denied. Otherwise, TA proceeds to calculate the pseudonym identity $PID_{TD} = H_2(S_{TD}, msk_{TA}) \oplus RID_{TD}$ and sends it to TD.

3.2.4 Partial Private Key Generation Phase

The partial private key generation phase mainly completes the generation of partial public/private keys of TD and fuzzy extractor key.

1) Extract Partial Private Key

Upon receiving (S_{TD}, PID_{TD}) from TD, TA chooses $v_{TD} \in Z_q$ and calculates the corresponding public key $V_{TD} = v_{TD} \cdot P$. TA computes the partial private key $d_{TD} = v_{TD} + msk_{TA} \cdot h_1$, where $h_1 = H_3(MPK_{TA}, V_{TD}, PID_{TD})$.

2) Extract Fuzzy Extractor Key

TA executes the following formula to obtain the fuzzy extractor key δ_{TD} and helper string η_{TD} of TD: $< \delta_{TD}, \eta_{TD} >= Gen(PID_{TD})$, where $Gen(\cdot)$ is fuzzy extractor's key generation algorithm. TA calculates the TD's on-chain index $idx_{TD} = H_4(\delta_{TD})$ and uses it as an input parameter to trigger the smart contract, which uploads TD's public key pair (S_{TD}, V_{TD}) and pseudonym identity PID_{TD} to the blockchain for certification, and set a certain validity period for PID_{TD}. TA secretly keeps TD's original pseudonym identity $PID_{\text{origin}} = PID_{TD}$ and helper string η_{TD}. At the same time, to ensure that TD's partial private key and helper string are not leaked to other nodes in the network, TA sends them to TD through a secure channel, and similarly sends η_{TD} to EAG within the domain through a secure channel. After receiving them, TD can verify the legitimacy of the partial private key through the following equation. Ultimately, TD completes the identity registration process in the system, obtaining the secret value pair $(c_{TD}, d_{TD}, \delta_{TD})$ and the public key pair (S_{TD}, V_{TD}).

3.2.5 Signature Generation Phase

1) Offline Signature Generation

TD chooses $e_{TD} \in Z_q$ and computes $E_{TD} = e_{TD} \cdot P$. TD computes $\vartheta_{\text{offline}} = e + d_{TD} \cdot h_2$ and obtains offline signature $\sigma_{\text{offline}} = (E_{TD}, \vartheta_{\text{offline}})$, where $h_2 = H_5(E_{TD}, S_{TD}, V_{TD}, PID_{TD})$.

2) Online Signature Generation

After comfirming the message M, TD obtains the latest timestamp T_{send} and recovers its secret key $s_{TM} = H_1(PUF(c_{TM}))$. TD computes $\vartheta_{TM} = \vartheta_{\text{offline}} + s_{TM} h_3$ and obtains signature $\sigma_{TM} = (E_{TM}, \vartheta_{TM})$, where $h_3 = H_6(E_{TD}, M, T_{\text{send}}, \delta_{TD})$. Finally, TD initiates an authentication request and sends $(\sigma_{TD}, M, T_{\text{send}}, PID_{TD})$ to EAG.

3.2.6 Signature Verification Phase

EAG first checks the legitimacy of T_{send} and PID_{TD}. If they are illegal, the authentication message is rejected; otherwise, EAG restores the fuzzy extractor key $\delta_{TD} = Rep(PID_{TD}, \eta_{TD})$ of TD and verifies the legitimacy of the signature through the following equation:

$$\vartheta_{TD} \cdot P = E_{TD} + h_2' \cdot (V_{TD} + h_1' \cdot MPK_{TA}) + h_3' \cdot S_{TD}, \quad (1)$$

where $h_1' = H_3(MPK_{TA}, V_{TD}, PID_{TD})$, $h_2' = H_5(E_{TD}, S_{TD}, V_{TD}, PID_{TD})$, and $h_3' = H_6(E_{TD}, M, T_{\text{send}}, \delta_{TD})$. If the equation holds, the received σ_{TD} is a legal signature and the TD's identity authentication is successful. Otherwise, EAG refuses to receive messages from TD, and TD identity authentication fails.

The following equation proves the correctness of Eq. (1):

$$\vartheta_{TD} \cdot P = (e_{TD} + d_{TD} \cdot h'_2 + s_{TD} \cdot h'_3) \cdot P =$$
$$e_{TD} \cdot P + (v_{TD} + msk_{TA} \cdot h'_1) \cdot h'_2 \cdot P + s_{TD} \cdot h'_3 \cdot P =$$
$$E_{TD} + h'_2 \cdot (V_{TD} + h'_1 \cdot MPK_{TA}) + h'_3 \cdot S_{TD} . \tag{2}$$

3.2.7 Batch Signature Generation Phase

When EAG receives multiple authentication request messages from different devices in a short period of time, EAG first checks the validity of the timeliness of these messages. If the timestamp of a message has expired, the authentication message is invalid. After that, EAG calculates $\vartheta_{\text{agg}} = \sum_{i=1}^{n} \vartheta_i$ and obtains an aggregate signature $\sigma_{\text{agg}} = (\vartheta_{\text{agg}}, E_1, E_2, \cdots, E_n)$. Eventually, EAG forwards $(\sigma_{\text{agg}}, M_i, T_i)_{i \in 1,2,\cdots,n}$ to EAS.

3.2.8 Batch Signature Verification Phase

After receiving $(\sigma_{\text{agg}}, M_i, T_i)_{i \in 1,2,\cdots,n}$ from EAG, EAS calculates $\delta_i = Rep(PID_i, \eta_i)$, $h_1^i = H_3(MPK_{TA}, V_i, PID_i)$, $h_2^i = H_5(E_i, S_i, V_i, PID_i)$, and $h_3^i = H_6(E_i, M_i, T_i, \delta_i)$. If Eq. (3) holds, all related devices are successfully authenticated. Otherwise, EAS rejects all the authentication messages.

$$\vartheta_{agg} \cdot P = \sum_{i=1}^{n} \vartheta_i \cdot P =$$
$$\sum_{i=1}^{n} (e_i + d_i \cdot h_2^i + s_i \cdot h_3^i) =$$
$$\sum_{i=1}^{n} E_i + h_2^i (V_i + h_1^i \cdot MPK_{TA}) + h_3^i \cdot S_i . \tag{3}$$

3.2.9 Pseudonym Identity Update Phase

TA obtains TD's original pseudonymous identity PID_{origin} from the security database and executes Algorithm 1 to generate a new pseudonymous identity PID_{TD}^{new}. Then, TA executes the relevant smart contract and updates TD's pseudonymous identity on the chain.

Algorithm 1: Pseudonym Identity Update Algorithm

```
input: original pseudonym identity $PID_{\text{origin}}$.
output: updated pseudonym identity $PID_{TD}^{\text{new}}$.
TA performs the following steps:
    TA converts $PID_{\text{origin}}$ to binary string binary_pid;
    TA randomly selects $d$ different bits in binary_pid ($d$ is
      the maximum distance tolerated by the fuzzy extractor);
    For each selected bit on binary_pid:
      if the corresponding bit is 0 then
        flip this bit to 1;
    else
      flip this bit to 0;
    end
TA converts the updated binary_pid into an integer form
  and obtains a new pseudonym identity $PID_{TD}^{\text{new}}$.
```

3.2.10 Identity Revocation Phase

In the identity revocation phase, TA performs the following steps to revoke TD's identity:

1) When TD is detected as a malicious node, TA first obtains TD's original pseudonym identity PID_{origin} and calculates TD's real identity $RID_{TD} = H_2(S_{TD}, msk_{TA}) \oplus PID_{\text{origin}}$.

2) TA triggers the smart contract to update the TD pseudonym identity status to "revoked".

3) TA adds TD's real identity to the malicious node blacklist $List_{\text{malicious}}$.

4 Evaluation

4.1 Setup

Regarding the identity authentication protocol designed in this paper, relevant experimental simulations are conducted in this section. All simulations in this section were performed on a personal laptop configured with an AMD Ryzen 75800H with Radeon Graphics 3.20 GHz 16.0 GB RAM, running the Windows 10 operating system. The simulations were implemented using the C programming language and simulated relevant cryptographic operations through the MIRACL cryptographic library. For the evaluation of computational and communication overheads, we selected a super singular elliptic curve defined over a finite field, where p and q are large prime numbers with 160 bits each. To obtain more accurate experimental results, each experiment was repeated 50 times, and the average of all test results was taken as the final experimental result. In the experiments, the proposed scheme was compared with identity authentication schemes from Refs [18 – 21], considering computational overheads, communication overheads, and security features for scheme comparison. The measured time cost of different operations is shown in Table 1.

4.2 Computation Cost Comparison

As shown in Table 2, in the authentication scheme proposed in this paper, the computational costs of signature gen-

▼Table 1. Running time of cryptographic operations

Notations	Operation	Execution Time/ms
T_{bp}	Bilinear paring operation	3.642 5
T_{bm}	Scalar multiplication in bilinear pairing	0.233 9
T_{ba}	Addition in bilinear pairing	0.165 8
T_{em}	Scalar multiplication in ECC	0.137 3
T_{ea}	Addition in ECC	0.096 0
T_{pth}	Map to point hash operation	3.813 3

ECC: elliptic curve cryptography

▼Table 2. Comparison of computation cost of schemes

Scheme	Single Signature	Single Verification	Aggregate Verification
SHEN et al.[18]	$(3T_{bm} + 1T_{pth} + 1T_{bp})_{\mathrm{on}}$	$3T_{bp} + 2T_{pth}$	$nT_{bp} + 1T_{bm} + (n - 1)T_{ba}$
KUMAR et al.[19]	$(3T_{bm} + 1T_{pth} + 1T_{bp})_{\mathrm{on}}$	$3T_{bp} + 2T_{pth} + 2T_{bm} + 1T_{ba}$	$3T_{bp} + (n + 1)T_{pth} + nT_{bm} + (3n - 2)T_{ba}$
KAMIL et al.[20]	$(3T_{em} + 1T_{ba})_{\mathrm{on}}$	$3T_{bp} + 3T_{em} + 1T_{ba}$	$3T_{bp} + (2n + 1)T_{em} + (2n - 1)T_{ba}$
ZHU et al.[21]	$(3T_{em} + 1T_{ba})_{\mathrm{on}}$	$4T_{em} + 3T_{ea}$	$(5n + 2)T_{em} + (7n + 4)T_{ea}$
Ours	$(T_{em})_{\mathrm{off}}$	$4T_{em} + 3T_{ea}$	$(3n + 1)T_{em} + (4n - 1)T_{ea}$

$(\cdot)_{\mathrm{on}}$: The signature algorithm is executed online. $(\cdot)_{\mathrm{off}}$: The signature algorithm is executed offline.

eration and signature verification algorithms are $(T_{em})_{\mathrm{off}} \approx 0.14$ ms and $4T_{em} + 3T_{ea} \approx 0.84$ ms, respectively. Since the authentication schemes in Refs [18], [19] and [20] require complex operations like bilinear pairings or point-to-hash operations, the computational overhead of these schemes is considerable. In comparison, the computational costs in both signature generation and verification stages of the proposed scheme are lower. Moreover, considering the utilization of online/offline signature aggregation techniques in this paper, the primary time overhead in the signature generation stage occurs during the offline signing phase. In the online stage, TD only needs to perform very minimal modular multiplication and addition operations, enabling a more efficient execution of the signature algorithm compared to the scheme in Ref. [21]. Consequently, this scheme can better meet the real-time requirements of current IoT scenarios.

Regarding the computational costs of signature verification, the time required for the authentication schemes in Refs. [18 – 21], and the proposed scheme in this paper are 18.55 ms, 19.19 ms, 11.50 ms, 0.84 ms, and 0.84 ms respectively. By employing a reduced number of modular multiplication operations instead of bilinear pairing computations, the computational overhead of the proposed scheme is also superior to those in Refs. [18 – 20].

Besides, the computation time cost of aggregate verification in each scheme increases linearly with the number of aggregated signatures. Specifically, the scheme in Ref. [19] requires the highest computation cost while the proposed scheme has the best computational efficiency.

4.3 Communication Cost Comparison

To evaluate the signature length and communication overhead of the proposed scheme, this experiment introduces the following metrics: $|G|$, $|G_1|$ and $|Z_q|$, which represent the size of group elements based on elliptic curve, bilinear pairing, and integer field, respectively. In this experiment, the specifications for each metric are defined as follows: $|G|$ is 320 bits, $|G_1|$ is 1 024 bits, and $|Z_q|$ is 160 bits.

As shown in Table 3, compared to the authentication schemes in Refs. [18 – 21], the single signature of the proposed scheme is reduced by 76.57%, 76.57%, 76.57% and 50% respectively. Although the communication overhead of the scheme in Ref. [21] is the same as that of the proposed scheme, it is difficult for this scheme to meet the security requirements of dynamic anonymity and resistance to physical/cloning attacks in IoT scenarios. Also, we can find that the length of aggregate signature in our scheme is smaller than that of schemes in Refs [18], [20], and [21]. Besides, although the aggregate signature length in Ref. [19] is less than that in the proposed scheme, the scheme in Ref. [19] has less than ideal computational efficiency and cannot provide user anonymity protection.

In conclusion, in the IoT environment, there are numerous devices that often have limited resources, such as restricted storage space and computing power. Compared to traditional authentication schemes, the proposed scheme does not require the storage and management of a large number of certificates and has low computational and communication overhead. This can significantly reduce the demand for storage resources, simplify the hardware requirements of devices, and thereby lower the overall system costs.

4.4 Security Analysis

No adversary can forge any user's identity authentication message, even if he/she has access to the public message. Also, the privacy information of non-target users can be obtained. Hence, the unforgeability ensures that the validity of the authentication message represents the identity legality of the message sender.

In the security model of certificateless public key cryptography, there are mainly two types of attackers:

1) Type I attacker A_1: This type of attacker is an external attacker that can obtain the user's private key, but does not have the ability to obtain the system's master key;

2) Type II attacker A_2: This type of attacker simulates an honest but passive KGC, capable of obtaining the system's master key, but without the ability to replace any user's pub-

▼Table 3. Comparison of communication cost of schemes

Scheme	Single Signature Length/bits	Aggregate Signature Length/bits
SHEN et al.[18]	$2\|G_1\| = 2\,048$	$(n + 1)\|G_1\| = (n + 1)2\,048$
KUMAR et al.[19]	$2\|G_1\| = 2\,048$	$(n + 1)\|G_1\| = (n + 1)2\,048$
KAMIL et al.[20]	$2\|G_1\| = 2\,048$	$2\|G_1\| = 2\,048$
ZHU et al.[21]	$2\|G\| + 2\|Z_q\| = 960$	$2n\|G\| + 2\|Z_q\| = 640n + 320$
Ours	$1\|G\| + 1\|Z_q\| = 480$	$n\|G\| + 1\|Z_q\| = 320n + 160$

lic key.

The security (unforgeability) of the certificateless signature algorithm is mainly proved by the games between the attacker $A \in \{A_1, A_2\}$ and the challenger $C \in \{C_1, C_2\}$.

Lemma 1: If there exists an adversary A_1 who can forge a valid signature with a non-negligible advantage ε_1, we can build a challenger C_1 who can solve the hardness of ECDL problem with an obvious advantage $\varepsilon_1' \geqslant \left(1 - \frac{1}{e}\right)\left(\frac{\varepsilon_1}{e(q_1 + q_2 + 1)q_{H_2}}\right)$, where q_1, q_2 and q_{H_2} denote the number of Partial-Private-Key-Extract query, Private-Key-Extract query, and H_2 oracle query, respectively.

Proof: In the beginning, the challenger C_1 takes an instance (P, aP) of the ECDL problem as input, and its purpose is to compute a.

Setup: In this stage, C_1 executes the system initialization algorithm and obtains the public parameters of the system $Params = (q, G, P, MPK, H_1, H_2, H_3, F_{EXT})$. $H_i(i = 1,\cdots,6)$ are six random prediction machines. C_1 then picks PID^* as the target user.

Query: In this stage, the challenger C_1 can query the following oracles adaptively and polynomially.

- *Create-User Query*: When C_1 receives *Create-User Query* with PID_i as input, it first checks whether there is a tuple $(PID_i, S_i, V_i, s_i, d_i, \delta_i, flag)$ in the list L_{user}. If so, C_1 sends $PK_i = (S_i, V_i)$ directly to A_1. If no, C_1 performs the following operations: 1) If $PID_i \neq PID^*$, C_1 randomly selects s_i, d_i, $h_1^i \in Z_q$, calculates $V_i = d_i P - h_1^i MPK$, $S_i = s_i P$, and $< \delta_i, \eta_i > = Gen(PID_i)$, and sets $flag = False$=; 2) If $PID_i = PID^*$, C_1 randomly selects v_i, s_i, $h_1^i \in Z_q$ and calculates $V_i = v_i \cdot P$ and $S_i = s_i \cdot P$, letting $d_i = \perp$ and $flag = False$. Finally, C_1 returns $PK_i = (S_i, V_i)$ to A_1 and adds $(PID_i, S_i, V_i, s_i, d_i, \delta_i, flag)$ and (MPK, V_i, PID_i, h_1^i) to the lists L_{user} and L_1, respectively.
- H_1 *Query*: When C_1 receives an H_1 *Query* from A_1 with input (MPK, V_i, PID_i), it first checks for the existence of tuples (MPK, V_i, PID_i, h_1^i) in list L_1. If so, C_1 will directly send h_1^i to A_1. If not, C_1 will submit the corresponding *Create−User* query with PID_i as input, then find h_1^i from list L_1, and send it to A_1.
- H_2 *Query*: When C_1 receives an H_2 *Query* from A_1 with input (E_i, S_i, V_i, PID_i), it first checks whether a tuple $(E_i, S_i, V_i, PID_i, h_2^i)$ exists in list L_2. If so, C_1 sends h_2^i directly to A_1. If it does not exist, C_1 randomly selects $h_2^i \in Z_q$ and sends it to A_1, adding $(E_i, S_i, V_i, PID_i, h_2^i)$ to list L_2.
- H_3 *Query*: When C_1 receives an H_3 *Query* from A_1 with input $(E_i, M_i, t_i, \delta_i)$, it first checks whether a tuple $(E_i, M_i, t_i, \delta_i, h_3^i)$ exists in list L_3. If so, C_1 sends h_3^i directly to A_1. If it does not exist, C_1 randomly selects $h_3^i \in Z_q$ and sends it to A_1, adding $(E_i, M_i, t_i, \delta_i, h_3^i)$ to list L_3.
- *Partial-Private-Key-Extract*: When C_1 receives *Partial-Private-Key-Extract Query* with input PID_i from A_1, it determines whether PID_i and PID^* are equal. If they are equal, the C_1 query is terminated. Otherwise, C_1 checks whether a tuples $(PID_i, S_i, V_i, s_i, d_i, \delta_i, flag)$ exists in list L_{user}. If so, C_1 directly sends (d_i, δ_i, V_i) obtained by list L_{user} to A_1. If not, C_1 submits PID_i as input to the corresponding *Create-User Query*, then finds (d_i, δ_i, V_i) from the list L_{user}, and sends it to A_1
- *Secret-Value-Extract*: When C_1 receives a *Secret-Value-Extract Query* from A_1 with PID_i as input, C_1 checks whether a tuple $(PID_i, S_i, V_i, s_i, d_i, \delta_i, flag)$ exists in list L_{user}. If so, C_1 sends (s_i, S_i) obtained by list L_{user} to A_1. Otherwise, C_1 submits PID_i as input to the corresponding *Create−User Query*, and then finds (s_i, S_i) from the list L_{user} and sends it to A_1.
- *Private-Key-Extract Query*: When C_1 receives a *Private-Key-Extract Query* with PID_i as input from A_1, it determines whether PID_i and PID^* are equal. If equal, the inquiry is terminated; otherwise, C_1 checks whether a tuple $(PID_i, S_i, V_i, s_i, d_i, \delta_i, flag)$ exists in the list L_{user}. If so, C_1 directly sends $SK_i = (s_i, d_i, \delta_i)$ obtained by list L_{user} to A_1. Otherwise, C_1 submits PID_i as input to the corresponding *Create-User Query*, and then finds $SK_i = (s_i, d_i, \delta_i)$ from list L_{user}, and sends it to A_1.
- *Replace-Public-Key Query*: When C_1 receives A_1 *Replace-Public-Key Query* with (PID_i, S_i', V_i') as input from A_1, it determines whether PID_i and PID^* are equal. If equal, the inquiry is terminated; otherwise, C_1 gets $(PID_i, S_i, V_i, s_i, d_i, \delta_i, flag)$ from list L_{user} and replaces (S_i, V_i) in the list with (S_i', V_i').
- *Sign Query*: When C_1 receives the *Sign Query* with (PID_i, M_i) as input from A_1, it determines whether PID_i and PID^* are equal. If $PID_i \neq PID^*$ and $s_i \neq \perp$, C_1 selects e_i, h_2^i and $h_3^i \in Z_q$ at random and calculates $E_i = e_i \cdot P$ and $\vartheta_i = e_i + d_i \cdot h_2^i + s_i \cdot h_3^i$, where $h_2^i = H_2(MPK, V_i, PID_i)$ and $h_3^i = H_3(E_i, S_i, V_i, PID_i)$. C_1 then sends the signature $\sigma_i = (\vartheta_i, E_i)$ to A_1 and adds $(E_i, S_i, V_i, PID_i, h_2^i)$ and $(E_i, M_i, t_i, \delta_i, h_3^i)$ to the lists L_2 and L_3, respectively. If $PID_i = PID^*$, C_1 gets $(PID_i, S_i, V_i, s_i, d_i, \delta_i, flag)$ and (MPK, V_i, PID_i, h_1^i) from the lists L_{user} and L_1, where: $s_i = \perp$ and $d_i = \perp$. ϑ_i, h_2^i and $h_3^i \in Z_q$ are randomly selected and $E_i = \vartheta_i \cdot P - h_2^i(V_{TM} + h_1^i \cdot MPK) - h_3^i \cdot S_i$ is calculated. Finally, C_1 sends the signature $\sigma_i = (\vartheta_i, E_i)$ to A_1 and adds $(E_i, S_i, V_i, PID_i, h_2^i)$ and $(E_i, M_i, t_i, \delta_i, h_3^i)$ to the list L_2 and L_3, respectively.

Forgery: When C_1 receives forged signature $\sigma^* = (E^*, \vartheta^*)$ from A_1 about (PID^*, M^*), it determines whether PID_i and PID^* are equal. If not, C_1 terminates the game. Otherwise, C_1 replays A_1 and gets a new forged signature $\sigma^{*(2)} = (E^*, \vartheta^*)$ about (PID^*, M^*). From this, C_1 can obtain:

$$\begin{cases} \vartheta^* = e^* + h_2^*(v^* + a \cdot h_1^*) + s^* \cdot h_3^* \\ \vartheta^{*(2)} = e^* + h_2^{*(2)}(v^* + a \cdot h_1^*) + s^* \cdot h_3^*. \end{cases} \tag{4}$$

In turn, C_1 outputs $a = \frac{1}{h_1^*}\left(\frac{\vartheta^* - \vartheta^{*(2)}}{h_2^* - h_2^{*(2)}} - v^*\right)$ as the solution to the elliptic curve discrete logarithm problem.

Next, we define the following events:

Event E_1: E_1 indicates that C_1 does not terminate the game during the query phase.

Event E_2: E_2 indicates that C_1 does not terminate the game during the Forgery phase.

Event E_3: E_3 indicates that the forged signature σ^* and $\sigma^{*(2)}$ about (PID^*, M^*) are valid signatures

Based on game analysis, we can calculate:

The probability of E_1 is $Pr(E_1) \geqslant \left(1 - \frac{1}{q_1 + q_2 + 1}\right)^{q_1 + q_2}$.

The probability of E_2 is $Pr(E_2) = \frac{1}{q_1 + q_2 + 1}$.

According to the Forking lemma, if one legitimate signature can be output by an advantage ε_1, the probability of two legitimate signatures being output is $Pr(E_3) \geqslant (1 - \frac{1}{e}) \frac{\varepsilon_1}{q_{H_2}}$.

From this, we can get the advantages of solving ECDLP problems:

$$\varepsilon_1' = Pr(E_1 \wedge E_2 \wedge E_3) \geqslant \left(1 - \frac{1}{e}\right)\left(\frac{\varepsilon_1}{e(q_1 + q_2 + 1)q_{H_2}}\right). \quad (5)$$

If malicious attacker A_1 can successfully forge two signatures with the probability of ε_1', it can be inferred that C_1 has the ability to solve the elliptic curve discrete logarithm problem. However, the existence of this capability is in apparent contradiction with the fact that the elliptic curve discrete logarithm problem is considered to be difficult to solve. Therefore, it can be inferred that the probability of A_1 forging a signature successfully is negligible. Therefore, we can conclude that this scheme can effectively defend against the threat of Class I attackers. Proof completes.

5 Conclusions

This paper proposes a blockchain-based identity authentication protocol for IoT devices in Web3, addressing security vulnerabilities in current centralized authentication methods. Based on blockchain and certificateless cryptography, the authentication process between IoT terminal devices and authentication nodes is designed, incorporating technologies such as physical unclonable functions and fuzzy extractors into the authentication protocol to achieve security features lacking in current identity authentication protocols, such as resistance to physical/cloning attacks and dynamic anonymity. Simulation results demonstrate that compared to existing solutions, the proposed identity authentication protocol has the advantages of low computational/communication overhead. Additionally, the security analysis of the protocol shows its excellent performance against various malicious attacks.

References

[1] LIU Z T, XIANG Y X, SHI J, et al. Make Web3.0 connected [J]. IEEE transactions on dependable and secure computing, 2022, 19(5): 2965 – 2981. DOI: 10.1109/TDSC.2021.3079315

[2] RAY P P. Web3: A comprehensive review on background, technologies, applications, zero-trust architectures, challenges and future directions [J]. Internet of Things and cyber-physical systems, 2023, 3: 213 – 248. DOI: 10.1016/j.iotcps.2023.05.003

[3] CHEN C, ZHANG L, LI Y H, et al. When digital economy meets Web3.0: applications and challenges [J]. IEEE open journal of the computer society, 2022, 3: 233 – 245. DOI: 10.1109/OJCS.2022.3217565

[4] GUPTA M. Integration of IoT and Blockchain for user Authentication [J]. Scientific journal of metaverse and blockchain technologies, 2023, 1(1): 72 – 84. DOI: 10.36676/sjmbt.v1i1.10

[5] GAO H M, DUAN P F, PAN X F, et al. Blockchain-enabled supervised secure data sharing and delegation scheme in Web3.0 [J]. Journal of cloud computing, 2024, 13(1): 21. DOI: 10.1186/s13677-023-00575-8

[6] SAMIR E, WU H Y, AZAB M, et al. DT-SSIM: A decentralized trustworthy self-sovereign identity management framework [J]. IEEE Internet of Things journal, 2022, 9(11): 7972 – 7988. DOI: 10.1109/JIOT.2021.3112537

[7] YIN J, XIAO Y, PEI Q Q, et al. SmartDID: A novel privacy-preserving identity based on blockchain for IoT [J]. IEEE Internet of Things journal, 2023, 10(8): 6718 – 6732. DOI: 10.1109/JIOT.2022.3145089

[8] BAO Z J, HE D B, KHAN M K, et al. PBidm: privacy-preserving blockchain-based identity management system for industrial Internet of Things [J]. IEEE transactions on industrial informatics, 2023, 19(2): 1524 – 1534. DOI: 10.1109/TII.2022.3206798

[9] VERMA G K, KUMAR N, GOPE P, et al. SCBS: a short certificate-based signature scheme with efficient aggregation for industrial-internet-of-things environment [J]. IEEE Internet of Things journal, 2021, 8(11): 9305 – 9316. DOI: 10.1109/JIOT.2021.3055843

[10] AL-RIYAMI S S, PATERSON K G. Certificateless public key cryptography [C]//Advances in Cryptology: ASIACRYPT 2003. Berlin, Heidelberg: Springer, 2003, 2894: 452 – 473. DOI: 10.1007/978-3-540-40061-5_29

[11] DING X Y, WANG X X, XIE Y, et al. A lightweight anonymous authentication protocol for resource-constrained devices in Internet of Things [J]. IEEE Internet of Things journal, 2022, 9(3): 1818 – 1829. DOI: 10.1109/JIOT.2021.3088641

[12] WANG W Z, XU H, ALAZAB M, et al. Blockchain-based reliable and efficient certificateless signature for IIoT devices [J]. IEEE transactions on industrial informatics, 2022, 18(10): 7059 – 7067. DOI: 10.1109/TII.2021.3084753

[13] LI X, JIANG C, DU D J, et al. A novel revocable lightweight authentication scheme for resource-constrained devices in cyber – physical power systems [J]. IEEE Internet of Things journal, 2023, 10(6): 5280 – 5292. DOI: 10.1109/JIOT.2022.3221943

[14] WANG Z L, ZHOU Y W, QIAO Z R, et al. An anonymous and revocable authentication protocol for vehicle-to-vehicle communications [J]. IEEE Internet of Things journal, 2023, 10(6): 5114 – 5127. DOI: 10.1109/JIOT.2022.3222469

[15] ZHOU Y W, CAO L, QIAO Z R, et al. An efficient identity authentication scheme with dynamic anonymity for VANETs [J]. IEEE Internet of Things journal, 2023, 10(11): 10052 – 10065. DOI: 10.1109/JIOT.2023.3236699

[16] ALI U, BIN IDRIS M Y I, FRNDA J, et al. Enhanced lightweight and secure certificateless authentication scheme (ELWSCAS) for Internet of Things environment [J]. Internet of Things, 2023, 24: 100923. DOI: 10.1016/j.iot.2023.100923

[17] IQBAL A, ZUBAIR M, KHAN M A, et al. An efficient and secure certificateless aggregate signature scheme for vehicular ad hoc networks [J]. Future Internet, 2023, 15(8): 266. DOI: 10.3390/fi15080266

[18] SHEN L M, MA J F, LIU X M, et al. A secure and efficient ID-based aggregate signature scheme for wireless sensor networks [J]. IEEE Internet of Things journal, 2017, 4(2): 546 – 554. DOI: 10.1109/JIOT.2016.2557487

[19] KUMAR P, KUMARI S, SHARMA V, et al. A certificateless aggregate

signature scheme for healthcare wireless sensor network [J]. Sustainable computing: Informatics and systems, 2018, 18: 80 – 89. DOI: 10.1016/j.suscom.2017.09.002

[20] KAMIL I A, OGUNDOYIN S O. On the security of privacy-preserving authentication scheme with full aggregation in vehicular ad hoc network [J]. Security and privacy, 2020, 3(1): e104. DOI: 10.1002/spy2.104

[21] ZHU F, YI X, ABUADBBA A, et al. Certificate-based anonymous authentication with efficient aggregation for wireless medical sensor networks [J]. IEEE Internet of Things journal, 2022, 9(14): 12209 – 12218. DOI: 10.1109/JIOT.2021.3134693

Biographies

WU Zhihui received his master's degree from Xidian University, China. He is the Deputy General Manager of Guangzhou Lianrong Information Technology Co. He has been responsible for project management and technical development in the fields of data security, privacy computing and blockchain technology for many years. He has led or participated in more than ten research projects, and published two papers and 11 invention patents. He received 2023 Blockchain Innovator of the Year Award.

HONG Yuxuan received his BE degree in information security from Xidian University, China in 2021. He is currently pursuing his ME degree with College of Guangzhou Institute of Technology, Xidian University. His research interests include identity authentication and blockchain.

ZHOU Enyuan is currently pursuing his PhD degree in Department of Computing in The Hong Kong Polytechnic University, China. He received his BE degree in information security from Northeastern University, China and MSc degree in cyberspace security (supervised by Prof. PEI Qingqi) from Xidian University, China. His current research interests include Blockchain, Database, and knowledge graph. He has published several papers in prestigious journals and conferences in data management field such as VLDB and IEEE TKDE.

LIU Lei received his BEng degree in electronic information engineering from Zhengzhou University, China in 2010, and his MSc and PhD degrees in communication and information systems from Xidian University, China in 2013 and 2019, respectively. He is currently an associate professor with the Guangzhou Institute of Technology, Xidian University. His research interests include vehicular ad hoc networks, edge intelligence and distributed computing.

PEI Qingqi (qqpei@mail.xidian.edu.cn) is a full professor and PhD supervisor of Xidian University, China. He serves as the director of the Blockchain Application and Evaluation Research Center of Xidian University and the executive director of the Shaanxi Key Laboratory of Blockchain and Secure Computing. His research interests focus on cognitive networks, data security, and blockchain. He has led or participated in more than 30 national, provincial and ministerial projects. He has published more than 100 journal or conference papers and obtained more than 60 patents (including five international PCT patents) and 40 registered software copyrights. He was awarded one second prize of national technology invention awards and three first prizes of provincial or ministerial scientific and technological awards.

Hierarchical Federated Learning Architectures for the Metaverse

GU Cheng, LI Baochun

(Department of Electrical and Computer Engineering,University of Toronto, Toronto M5S2E8, Canada)

DOI: 10.12142/ZTECOM.202402006

https://kns.cnki.net/kcms/detail/34.1294.TN.20240624.1615.002.html, published online June 24, 2024

Manuscript received: 2024-05-31

Abstract: In the context of edge computing environments in general and the metaverse in particular, federated learning (FL) has emerged as a distributed machine learning paradigm that allows multiple users to collaborate on training a shared machine learning model locally, eliminating the need for uploading raw data to a central server. It is perhaps the only training paradigm that preserves the privacy of user data, which is essential for computing environments as personal as the metaverse. However, the original FL architecture proposed is not scalable to a large number of user devices in the metaverse community. To mitigate this problem, hierarchical federated learning (HFL) has been introduced as a general distributed learning paradigm, inspiring a number of research works. In this paper, we present several types of HFL architectures, with a special focus on the three-layer client-edge-cloud HFL architecture, which is most pertinent to the metaverse due to its delay-sensitive nature. We also examine works that take advantage of the natural layered organization of three-layer client-edge-cloud HFL to tackle some of the most challenging problems in FL within the metaverse. Finally, we outline some future research directions of HFL in the metaverse.

Keywords: federated learning; hierarchical federated learning; metaverse

Citation (Format 1): GU C, LI B C. Hierarchical federated learning architectures for the metaverse [J]. ZTE Communications, 2024, 22 (2): 39 – 48. DOI: 10.12142/ZTECOM.202402006

Citation (Format 2): C. Gu and B. C. Li, "Hierarchical federated learning architectures for the metaverse [J]. *ZTE Communications*, vol. 22, no. 2, pp. 39 – 48, Jun. 2024. doi: 10.12142/ZTECOM.202402006.

1 Introduction

With the advent of spatial computing and head-mounted devices such as the Apple Vision Pro, the metaverse computing environments have quickly become a reality and will become an everyday routine in the future. On the other hand, with the breakneck speed at which both computation power and efficiency in deep learning have advanced in the past decade due to its dramatic success, we almost always need to rely upon the power of deep learning models in any distributed computing environments, and in the metaverse in particular. One major bottleneck in using deep learning in the metaverse today, however, is the availability of raw data that we can use to train a new model or fine-tune a pretrained model.

One of the key contributing factors to the high popularity of deep learning lies in its ability to enable the data-driven paradigm in algorithm design. Unlike the traditional algorithm design paradigm, which relies on human expertise to produce hand-crafted logic and heuristics, the data-driven paradigm instead focuses on training a model, often as a black box, to learn features from a large dataset in order to produce desired outputs. This method excels in tasks where the desired function is too complex to understand or to construct manually; however, at the same time, it often requires large volumes of training datasets to work effectively, especially when in metaverse computing environments. This becomes a significant issue in traditional centralized deep learning which requires a server to access data directly, yet the data may contain highly sensitive or private information that makes it difficult to do so.

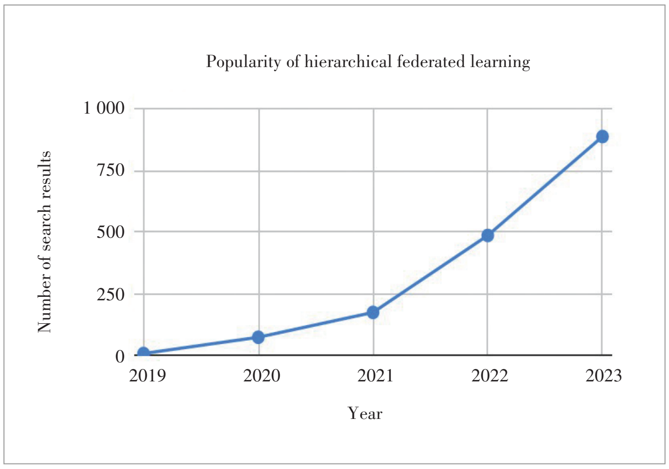
Federated learning (FL) is one of the methods that has emerged in recent years that aims to solve this exact problem. The term was first introduced in 2017[1] to describe a distributed machine learning paradigm that typically involves a group of clients and a central server, where each client trains a deep learning model with its local data, and then uploads the model parameters to the central server to aggregate into a single global model. This allows the clients to retain sensitive data locally, while still contributing to the global model by uploading its local model parameters via each round of communication with the server. The intrinsic protection of privacy offered by this architecture has earned immense popularity in the past few years.

However, the original two-layer client-server architecture for federated learning is hardly scalable in the metaverse, where scalability is needed the most. One of the key reasons why the original architecture is not suitable for the metaverse environments is the high bandwidth demand in each communication round between the server and clients. As the complexity and size of local models increase, the number of data each client needs to send to the server also increases proportionally. Some popular modern models like large language models often have from millions to billions of parameters, making the bandwidth consumption of each communication round extremely high. The most apparent effect of this is an increased communication time between the server and the clients. Sometimes the communication time becomes significant enough to warrant a reduction in total communication rounds by increasing the number of local training rounds. However, this can result in a variety of negative effects, including an increase in the total number of rounds for the global model to converge or a drop in the final model's accuracy. Moreover, these negative effects often get amplified when the data are non-independently and identically distributed (non-IID), which is common in real-world data[2].

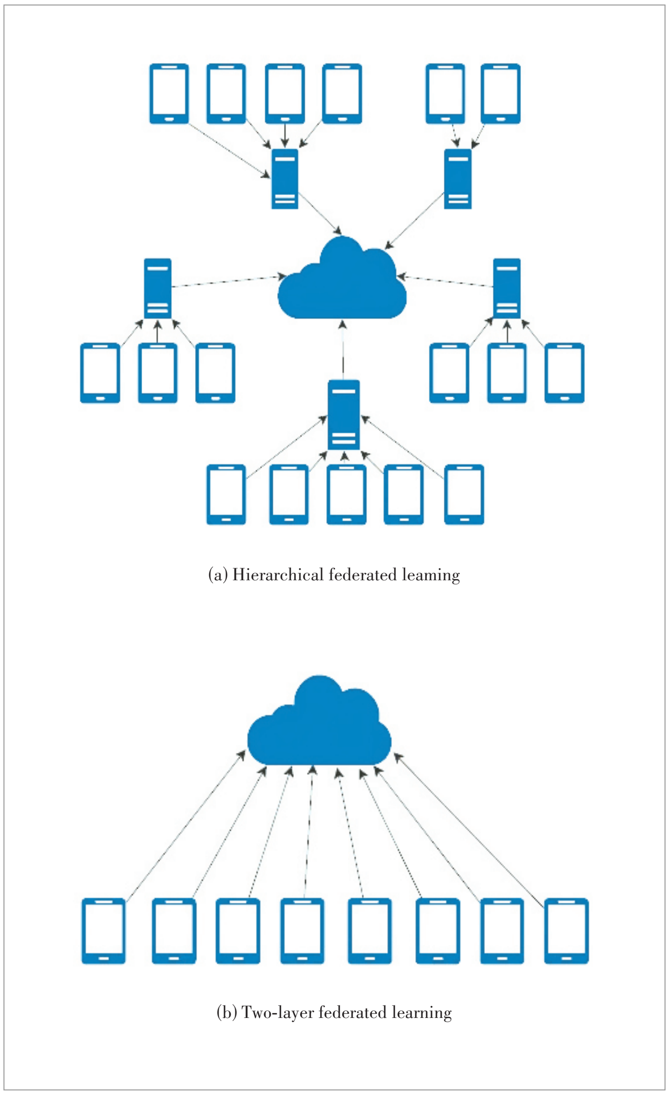
A common approach to increasing scalability in distributed systems in general and the metaverse in particular is to add edge servers, especially when the central server needs to serve a large number of clients. This type of hierarchical organization pattern can be very frequently observed in distributed computing architecture design, such as edge computing systems[3] and software-defined networks (SDN)[4]. We argue that it should be the preferred paradigm of distributed learning in the metaverse computing environment as well.

Driven by this intuition, hierarchical federated learning (HFL) was introduced around late 2019 to early 2020[5 – 6], along with a modified version of a federated averaging (FedAvg) algorithm, called the hierarchical stochastic gradient descent (HSGD) algorithm. Since then, HFL has gained significant popularity, as shown in Fig. 1. In a three-layer HFL, each edge server is assigned to a group of clients local to its service area. The clients only communicate with their assigned edge server, which aggregates the client parameter updates to an edge-local model. Each edge server then sends the aggregated edge-local model to the central cloud server, which further aggregates them to a single global model. Fig. 2 shows the architecture of the three-layer HFL versus traditional two-layer federated learning. Since the original introduction, a number of research works[6 – 13] have emerged to further build on the broad concept of HFL.

However, not all works agree on a three-layer hierarchical structure. Refs. [14 – 16] argue that using a cloud server exposes the system to a single point of failure; instead, they



▲ **Figure 1. Approximate numbers of publications with "hierarchical federated learning" in the title, with results obtained through Google Scholar Advanced Search API and may count duplicated publications**


(a) Hierarchical federated leaming
(b) Two-layer federated learning

▲ **Figure 2. Architecture of three-layer hierarchical federated learning versus two-layer traditional federated learning** propose a two-layer system with a group of edge servers communicating with each other. Refs.[17 – 18] argue that three layers might not be enough in FL and instead propose a multi-layered HFL architecture. In this paper, we examine all of those variations and propose that the three-layer architecture is typically the best choice in most scenarios, including the metaverse, compared with other alternatives. In addition, besides scalability, we will also examine some other intrinsic advantages of HFL over traditional federated learning in the metaverse. Ref. [10] uses knowledge distillation and meta-learning to take advantage of the naturally clustered clients in HFL to improve the training performance on non-IID training data problems. Other researchers propose to capitalize on edge computing power to apply data pruning and quantization mechanisms[7].

The overarching objective of this paper is to provide a general overview of the current landscape of HFL in edge computing environments in general and the metaverse computing environments in particular, and to examine the different architectures for achieving better scalability with HFL. We will also provide some insights on why HFL has the potential to become the mainstream framework for next-generation federated learning systems in the metaverse and outline potential research directions.

2 Background

2.1 Federated Averaging Algorithm

The term federated learning was first introduced by MCMAHAN et al. in 2017[1], along with the Federated Averaging (FedAvg) algorithm which would become the foundation for the majority of federated learning algorithms in the years that ensue. The intrinsic privacy protection offered by FL's nature of not requiring end devices to directly upload data has incentivized heavy research investments and has inspired a large number of works, including the HFL. We begin by examining the FedAvg algorithm to build a foundation for later discussions on the hierarchical stochastic gradient descent algorithm.

The FedAvg algorithm can be defined as the following. Let us assume that we have a training dataset $D = \left\{x_i\right\}_{i=1}^{|D|}$ that is divided among K clients, each owning a subset of the training data $D_k \in D$. Each client k also owns a local model denoted as f_k, which is fully parameterized as a set of weights w_k. We can calculate the loss value of the local model with a loss function l_k.

Training is performed in a total of T rounds. In each round, the FedAvg algorithm can be divided into the local training stage and the global aggregation stage. During round t, each client k performs m local iterations to minimize some average local loss:

$$\min_w L_k\left(w_k^t, D_\mathrm{k}\right),$$

$$L_k\left(w_k^t, D_\mathrm{k}\right) = \frac{\sum_{i=1}^{D_\mathrm{k}} l_k\left(x_i, w_k^t\right)}{\left|D_k\right|}. \quad (1)$$

Once the local update is completed, the clients send the weights w_k^t to the central server for an average aggregation via the following equation:

$$w^{t+1} = \frac{\sum_{k=1}^{K}\left|D_k\right| w_k^t}{|D|}. \quad (2)$$

After aggregation, the global server then broadcasts the model to all clients to be used for the next round of training. The algorithm terminates after T rounds.

2.2 Definition of Hierarchical Federated Learning

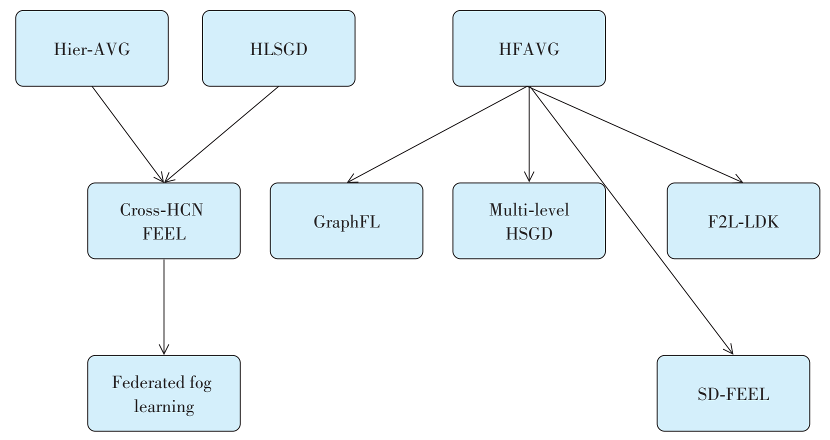
Before we start, it is necessary to define the exact scope of our discussion. To do so, we must first find a suitable definition for what is HFL. However, to our best knowledge, despite the growing popularity of the topic, there is no existing work that adequately defines what HFL is. Fortunately, the nomenclature of the term itself is fairly self-explanatory and provides a good foundation on which we can easily build our own definition. To begin with, as the name indicates, HFL is a subset of FL. Hence, it inherits the same set of definitions. Same as FL, HFL is also a distributed machine learning paradigm that involves multiple data owners and a model owner to train a single model, whereby the training data are kept strictly private to each data owner. Typically, in traditional FL, each data owner trains its own local model and sends the update to a single model owner entity, which aggregates all updates to update the target global model. An example would be the FedAvg algorithm discussed in the previous section. It is worth noting that, from a system organization perspective, this direct communication between data owners and model owners naturally partitions the system into two layers.

Instead of directly aggregating all updates from the data owners into a global model update, HFL employs multiple intermediate model aggregators to first aggregate the client updates into intermediate updates, and then aggregate these intermediate updates into the final global update. From a system organization perspective, these intermediate model aggregators typically form one or more extra layers in the system and introduce a hierarchy to the communication structure, which is responsible for the "hierarchical" part in HFL. An example of this would be the Hierarchical Stochastic Gradient Descent algorithm introduced in Ref. [5].

To summarize, HFL can be broadly defined as a subset of FL that consists of multiple intermediate aggregators between the data owners and the global server.

3 Progenitors of Hierarchical Federated Learning

Although hierarchical federated learning has only recently started to gain popularity, the concept was first introduced between 2018 and 2019. Specifically, Refs. [5 – 6, 19 – 20] contributed to building the foundation for the HFL field, as shown in the dependency graph in Fig. 3. We will discuss these works in this section. A comparison between these works and the rest of the works discussed in this paper can be found in Table 1.



F2L-LDK: full stack federated learning with label-driven knowledge
FEEL: federated edge learning
FL: federated learning
HCN: heterogeneous cellular network
HFAVG: hierarchical federated averaging algorithm
Hier-AVG: hierarchical averaging stochastic gradient descent
HLSGD: hierarchical local stochastic gradient descent
HSGD: hierarchical stochastic gradient descent
SD: semi-decentralized

▲ **Figure 3. Citation graph for all the HFL architectures discussed in this paper. An arrow pointing from work A to work B means that A appears in B's citations**

1) Hierarchical local stochastic gradient descent (HLSGD) and hierarchical averaging stochastic gradient descent (Hier-AVG). The very first mentioning of an algorithm that can be classified as an example of HFL can be found in Ref. [19] by LIN et al., which was first available as a preprint in 2018 and subsequently published in 2020. The focus of the paper was comparing the performance of FedAvg (referred to as local-SGD in the paper) against traditional SGD with large mini-batches, and HLSGD was only briefly introduced in the appendix.

A simplified version of the HLSGD algorithm is shown in Algorithm 1, where we can observe that the algorithm follows our definition of HFL well. Each node in a graphic processing unit (GPU) block is a data owner, as local data are not shared between any two nodes horizontally and are not passed to outer loops. On top of the data owners, each "block" serves as an intermediate model aggregator, since each of them hosts a local-SGD or the FedAvg algorithm among the GPU nodes on the block. Finally, a global model is obtained by combining the models from the blocks. This basic anatomy of the algorithm can be observed across almost all HFL implementations.

The authors in Ref. [20] proposed an almost identical algorithm in early 2019, which is Hier-AVG. The paper was theory-focused and provided little consideration for real work application scenarios, but it nonetheless contributed to building the foundation for HFL by offering a formal math-

▼**Table 1. Comparison of different hierarchical federated learning architectures**

Architecture	Number of Layers	Client-Edge-Cloud	Scalability
HLSGD	3	Unspecified (but can be)	Unlimited
Hier-AVG	3	Unspecified (but can be)	Unlimited
HFAVG	3	Yes	Unlimited
Cross-HCN FEEL	3	No (client-edge-edge)	Limited by the coverage of the macro-cell base station
Graph-FL	2	No (client-edge)	Unlimited, but communication costs between servers exhibit quadratic growth
SD-FEEL	2	No (client-edge)	Unlimited, but may be limited by max network span in spare edge networks
Federated fog learning	$N \geqslant 2$	No (client-edge*N-cloud)	Unlimited
Multi-level HSGD	$N \geqslant 2$	No (client-edge * N-cloud)	Unlimited
F2L-LDK	3	Yes	Unlimited (supports dynamic edge participation)

F2L-LDK: full stack federated learning with label-driven knowledge
FEEL: federated edge learning
FL: federated learning
HCN: heterogeneous cellular network
HFAVG: hierarchical federated averaging algorithm
Hier-AVG: hierarchical averaging stochastic gradient descent
HLSGD: hierarchical local stochastic gradient descent
HSGD: hierarchical stochastic gradient descent
SD: semi-decentralized

ematical convergence proof for the algorithm.

Algorithm 1. The hierarchical local-SGD algorithm

```
Require: $K$ ←total numbers of nodes
Require: $K'$ ←numbers of nodes per block
Require: $T$←rounds of global updates
Require: $N$←rounds of block updates
Require: $M$←rounds of local updates
Initialize all models to $w_0$
for all $K$ in parallel do:
    for $t$ in 0, 1, 2, 3...$T$ do: (Global aggregation)
        for $n$ in 0, 1, 2, 3....$N$ do: (Block aggregation)
            for $m$ in 0, 1, 2, 3...$M$ do: (local update)
                Sample minibatch of data;
                Compute the gradient;
                update the local model $w_l$;
            end for
            enter synchronized mode:
            $w_b$←Aggregate all $w_l$ in the block;
            $w_l$←$w_b$
            end synchronized mode
        end for
        enter synchronized mode:
        $w_g$←Aggregate all block models $w_b$
        $w_g$←$w_g$
        end synchronized mode.
    end for
end for
```

2) The hierarchical federated averaging algorithm (HFAVG) is introduced by one of the most highly cited papers in the field of HFL, "Client-edge-cloud hierarchical federated learning" by LIU et al.,[5] which was first available as a preprint in late 2019, and was later published in 2020. Although the algorithm itself does not deviate from the HLSGD and Hier-AVG algorithms, it is the first to structure the algorithm explicitly under a three-layer client-edge-cloud setting. The main contribution of the paper is proposing HFL to solve one of the most significant challenges in the field of FL, especifically federated edge learning (FEEL): scalability.

One of the major bottlenecks in federated learning is the communication latency between a server and its clients. A solution to this problem is to use a server that is physically close to the clients. Following this reasoning, FEEL[21] emerged in early 2018. Instead of using a central cloud server to facilitate model weight aggregation, FEEL uses a single server at the edge of the network, for example, a cellular base station (CBS), to reduce latency. However, this naturally limits the scalability of the system as a CBS can only serve clients in its physical vicinity, which reduces the total size of the available client pool.

HFAVG offers an intuitive solution to bypassing this limitation by coordinating multiple edge servers through the cloud. The algorithm describes a process of performing FedAvg with each edge server independently for a set number of rounds, and then aggregating all the edge models in the cloud to obtain a global model. This structure of client-edge-cloud is intuitive and effective, and makes the application of an HFL algorithm a convincing case. This is perhaps the key reason why this paper received more popularity despite being released almost a year later than the previously mentioned papers.

3) Cross heterogeneous cellular network (HCN) FEEL. Following the footsteps of HFAVG, another paper "Hierarchical federated learning across heterogeneous cellular networks"[6] was released as a preprint in late 2019 and published in 2020. The paper also aims to solve the problem of scalability in federated edge learning. However, instead of defining a general framework for using the cloud to orchestrate edge servers, the paper suggests a more specific architecture of using small-cell base stations (SBS) as edge servers, and a dedicated macro-cell base station covering the physical area of the SBS as the central server.

The main advantage of this architecture over the more general client-edge-cloud federated learning is the low communication latency, which is further reflected in the algorithm design. The main difference between the HFL algorithms in cross HCN FEEL compared with the previous work is the lack of the innermost local update loop—each client only performs training on one minibatch of data before synchronizing with the small-cell base stations. This results in a very high rate of synchronization between clients in the same group (referred to as local averaging). Moreover, since the communication latency between the sub-cell and macro-cell base stations is also relatively small compared with a faraway cloud server, the rate at which the global model can be updated per edge model update (referred to as global averaging) can also be quite high. Studies on the convergence rate of HFL algorithms[5–6, 12, 20] have shown that the rate of local averaging for a small number of client groups is the dominating term in the overall convergence rate, calculated as the number of local iterations required. Coupled with the already low communication latency from MBS to the clients, in theory, cross HCN FEEL offers an extremely fast convergence speed compared with client-edge-cloud.

Although this conclusion may seem impressive, it holds limited practical value because convergence speed is usually not a major concern in federated learning. On the other hand, the trade-off is that the scalability of the overall framework is now limited by the coverage of MBS, which can be a major limitation. Moreover, using a cloud central coordinator has the advantage of easy management and deployment, and offers access to vast and flexible computation resources, all of which are important benefits in large-scale machine learning that cannot be easily enjoyed on a macro-cell base station.

4 Exploring the Organization in Hierarchical Federated Learning

Although three-layer HFL is the simplest and most intuitive way of organizing an HFL system, a number of works have also proposed fewer or more layers to construct an HFL system. We study some examples of such architectures and compare them with the three-layer alternatives.

4.1 Two-Layer Hierarchical Federated Learning

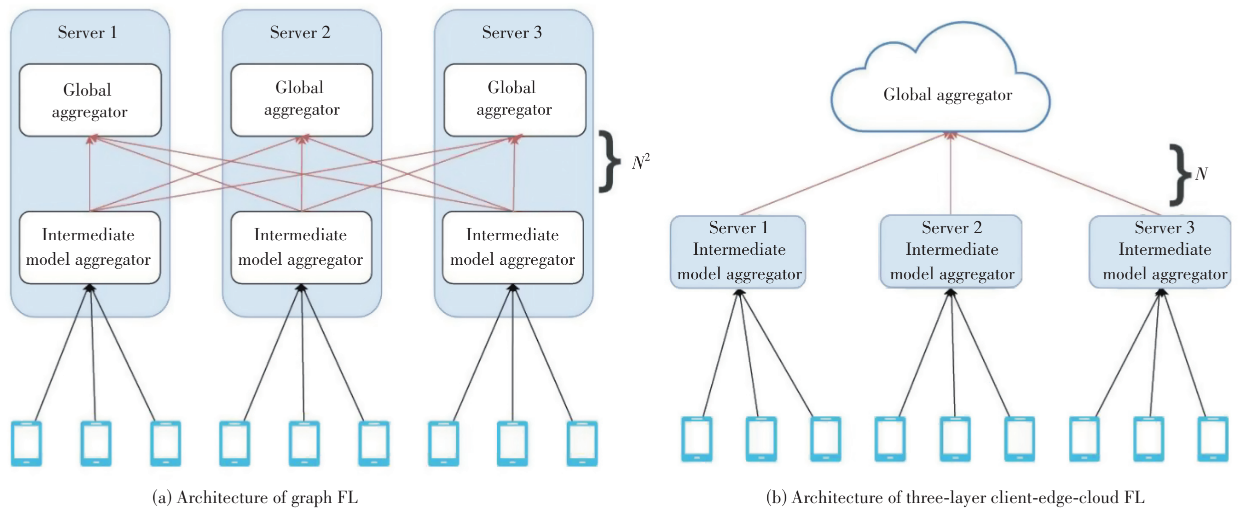
Sometimes there may not exist a dedicated global model aggregator in an HFL architecture that effectively renders the corresponding system organization into two layers. In this section, we discuss two such architectures and offer an argument that such designs can generally be replaced by a standard three-layer architecture, which would likely improves the system performance.

Introduced by RIZK et al. [14], GraphFL is a special privacy-focused HFL architecture. The main goal of GraphFL is to minimize the differential privacy risks between groups of clients. To achieve this, the clients are partitioned into small groups, each group connected to a dedicated intermediate model aggregator server running the FedAvg algorithm. So far, this is identical to standard HFL. However, instead of using a global server, the intermediate model aggregators run consensus algorithms among themselves to output a set of global models collectively, such that each server obtains its own version of the “global model”. The consensus algorithm involves each server sending all the other servers the weights of its model along with a small noise, and then performing an average aggregation with all the received models.

Discussion regarding differential privacy is out of the scope of this survey. However, GraphFL nonetheless offers a new perspective: how an HFL architecture can be structured. The differential privacy setting does provide a motivation for this graph-like organization. Nevertheless, it is debatable whether such an architecture is necessary, as we can easily employ a centralized cloud server to collect all the models from the intermediate model aggregator server, perform the consensus algorithm, and then send the result models back. Not only will this architecture allow the benefit of powerful cloud computing resources, but it can also potentially reduce communication latency as client-server communication typically enjoys higher bandwidth compared with peer-to-peer communication. Moreover, as we illustrate in Fig. 4, it can also eliminate the communication step between servers to exchange models, which requires N^2 operations between N servers, as opposed to N operations by sending all the models to a central server.

Another similar implementation of the two-level HFL algorithm is semi-decentralized federated edge learning (SD-FEEL)[15] by SUN et al. Unlike GraphFL, SD-FEEL is designed as a general-purpose federated learning algorithm. In this approach, multiple edge servers coordinate clusters of client nodes to perform local model updates and intra-cluster model aggregation. The edge servers then periodically share their updated models with neighboring (one-hop) edge servers for inter-cluster model aggregation.

The paper claims that this semi-decentralized training protocol leverages the low communication latency between edge servers to facilitate efficient model exchanges, and allows for a large number of client nodes to collaborate with minimal


Server 1
Server 2
Server 3
Global aggregator
Global aggregator
Global aggregator
N^2
Intermediate model aggregator
Intermediate model aggregator
Intermediate model aggregator
Global aggregator
N
Server 1 Intermediate model aggregator
Server 2 Intermediate model aggregator
Server 3 Intermediate model aggregator

(a) Architecture of graph FL

(b) Architecture of three-layer client-edge-cloud FL

FL: federated learning

▲Figure 4. Architecture of graph federated learning versus that of standard three-layer hierarchical federated learning. As we can observe, graph federated learning requires N^2 communications between the servers in the upper layer, while three-layer hierarchical federated learning only requires N

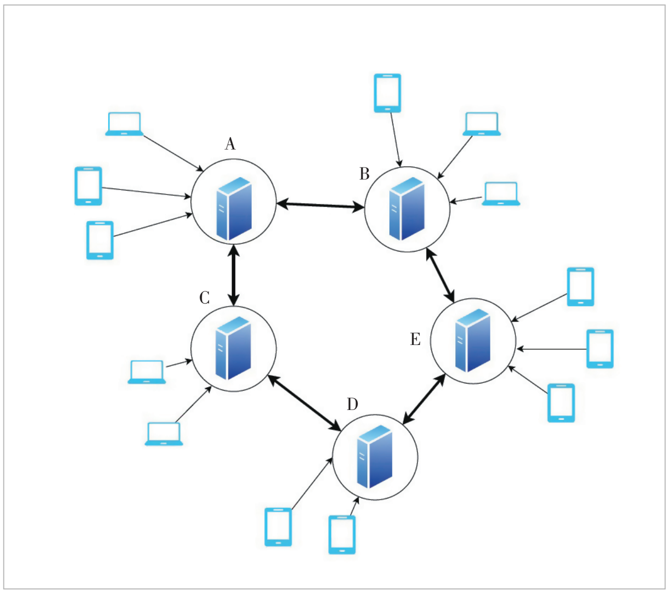
communication cost. However, there are two flaws in this claim. First, the low communication costs between edge servers are only true when the edge servers are close to each other or densely connected by a dedicated network. Second, it is very obvious that this algorithm will encounter issues when the number of edge servers grows, or when the network connecting the edge servers is sparse due to the information propagation delay. For a network where the maximum hop between two servers i, j is N, it would take at least N edge server aggregation rounds before i receives j's update from N rounds before, and vice versa, while at that time the update may already become stale. An example is shown in Fig. 5. There is no clear solution to this problem, which makes it questionable if the slightly lower communication cost offered by this method is worth trading off the reliability and scalability benefits of using a simple three-layer client-edge-cloud architecture.

4.2 Multi-Layer Hierarchical Federated Learning

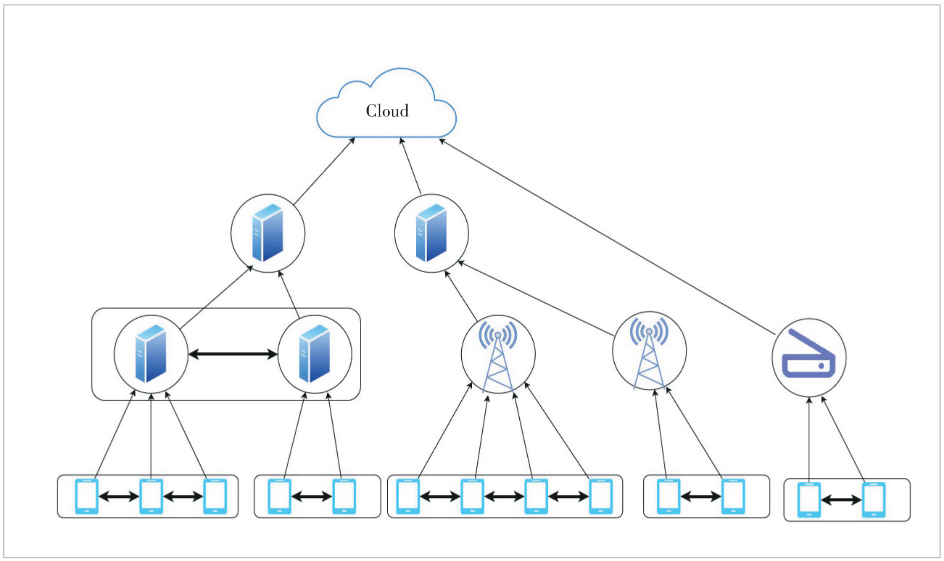
On the other end of the spectrum, we have HFL systems with multiple layers. They are typically constructed by using more than one layer of intermediate aggregators. At first glance, this kind of architecture may be intuitive, since oftentimes networks are multi-layered. However, we present two examples and argue that, typically, a three-layer HFL system is enough as an alternative to multi-layer HFL, if not better, from both a system design perspective and a theoretical perspective.

1) Federated fog learning. An example of an HFL framework that involves more than three layers is federated fog learning, introduced in Ref. [18]. Fog computing, also known as fog networking or fogging, is a distributed computing paradigm that brings computing and storage resources closer to the edge of the network in order to address the limitations of cloud computing in certain scenarios. It involves the deployment of small and low-power devices at the edge of the network, which are capable of performing local computing and storage tasks and communicating with each other and with the cloud. Due to this nature, applying HFL in fog scenarios would likely involve a multi-layered architecture design naturally. Indeed, federated fog learning introduces a generalized and multi-layered HFL architecture with the support of device-to-device collaborative training. An example is shown in Fig. 6. The local models trained on the bottom-layer IoT devices are passed through multiple layers of intermediate model aggregator devices before reaching the cloud for global aggregation. At each layer, models from the same groups in the lower layer are aggregated into a model that is reduced in dimensions (sizes) and then passed upwards, saving communication costs. Sometimes, depending on the use case, devices in the same layer may exchange models to perform a horizontal aggregation before sending the models to the upper layer.

The benefit of this architecture cannot counteract the complexity it brings. The paper does not offer any concrete algo-



▲ **Figure 5. An example of a semi-distributed federated edge learning system in a ring shaped edge network. The maximum number of hops in this network between any two servers is two, which means that, for example, it would take at least two rounds of local averaging before A receives the model from E, and vice versa**



▲ **Figure 6. Architecture of federated fog learning. The nodes enclosed in horizontal boxes perform peer-to-peer horizontal aggregation between each other before uploading the model to the upper layer** rithms, nor does it provide proof of convergence for this architecture, rendering it difficult to properly evaluate this new class of multi-layered HFL. However, it is well-known that one of the key motivations for FL is data privacy requirements, that is, training data cannot leave the devices of the data owners. However, it is debatable if such a requirement applies to small and low-power devices such as IoT sensors in a fog computing setting. Most often, a large group of IoT devices belong to a single silo (e.g. a factory or a warehouse), where the data generated can be collected and processed in a single silo server.

Generally, each silo can be seen as one data owner, hence there is usually little motivation to keep data private to each IoT device within the silo. It is much simpler and easier to train a silo model instead and perform cross-silo federated learning between different silos. On the other hand, in situations where keeping data private for IoT devices is a hard requirement, this federated fog learning system may become useful. However, one may also argue that instead of performing intermediate aggregation steps, it can be simpler (and often faster) to just relay the end device models to the closest edge server instead.

2) Multi-level HSGD. A more detailed example of multi-layer HFL is Multi-level HSGD introduced in Ref. [12] by WANG et al. This theory-focused paper extends the existing HSGD algorithm (the same algorithm as HLSGD, HFAVG, etc.) for three-layer HFL to multiple layers, and provides a convergence-bound analysis. The multi-level HSGD algorithm introduced is the same as the standard three-layer HFL algorithms such as HLSGD, except that instead of one single layer of intermediate model aggregators, there can now be multiple layers.

The paper presents an interesting analysis result, which shows that the convergence bounds of multi-level HSGD and regular three-layer HSGD are bounded by the same upper and lower bounds. In other words, adding more layers cannot improve the worst-case or the best-case convergence values of the multi-level HSGD algorithm from that of the three-layer version. What adding more layers does allow is more freedom in choosing the hyper parameters controlling the upstream and downstream aggregation rates for each layer, where the upstream aggregation rate refers to the number of updates the server in the current layer performs before sending the models to the server in the layer above, and downstream refers to the number of updates the lower layer performs before sending the update to the current server. However, whether this increased degree of freedom is beneficial is hard to determine, because it is difficult enough to tune the single pair of upstream and downstream aggregation rates in a three-layer system (i.e. the local aggregation and global aggregation rates). This is because these values are typically found empirically, requiring a full federated learning session to run from beginning to convergence repeatedly. It is easy to imagine that increasing the degree of freedom will make it extremely tasking to find an optimal operation point for the system.

Hence, one may argue that the fact multi-level HSGD only adds more degrees of freedom in choosing upstream and downstream aggregation rates without changing the upper and lower bounds is an argument against using multi-level HSGD as opposed to three-level, in the interest of keeping the system simple without significant performance detriments.

5 Unique Applications of Client-Edge-Cloud Hierarchical Federated Learning

Ever since client-edge-cloud three-layer federated learning design rose to popularity, there have been a number of works taking advantage of this architecture and presenting interesting solutions to unique challenges in federated learning[8–10]. In this section, we briefly discuss one of these works as an example to showcase the advantage of a client-edge-cloud federated learning architecture.

Full stack federated learning with label-driven knowledge distillation (F2L-LDK) is a federated learning method introduced in Ref. [10]. The method consists of two parts: a scalable HFL framework, dubbed full-stack federated learning, and a knowledge distillation-based model training scheme aimed at solving the non-indentically and independently-distributed (non-IID) data problem, which is a major challenge in federated learning.

The HFL algorithm of F2L-LDK is almost the same as the standard HLSGD or HFAVG algorithm with the exception that instead of running another round of FedAvg at the global server for all the edge models. It performs a multi-teacher knowledge distillation instead, where the “teachers” are the edge models and the “student” is the output global model. On top of offering good performance against non-IID data, this design makes the system very flexible to the number of edge servers participating in each round, even allowing adding or removing edge servers halfway through the training process without significant detriments to the training efficiency and final converged accuracy.

F2L-LDK is a prime example of the advantages of a client-edge-cloud architecture, as it leverages the flexibility and the powerful computing resources available in the cloud to perform knowledge distillation, while preserving great scalability thanks to the edge-based intermediate-model aggregators.

6 Open Challenges and Future Research

Despite its recent rise in popularity, HFL is still a young field, especially compared with traditional non-hierarchical federated learning. Hence, many research directions mature in non-hierarchical settings have yet to be studied under the hierarchical scenario. One example is asynchronous federated learning, which is yet to be extended to a multi-layered federated learning setting. Another example is quantization and model pruning, which are great methods for reducing communication costs in federated learning but have not been applied to HFL at the time of writing this paper. Besides the two examples, there are many more exciting research topics in the field of federated learning that can be further explored with HFL architectures.

7 Related Work

7.1 Cross-Silo Federated Learning

There exists a line of work called cross-silo federated learning[22], which may bear some similarities to HFL. On the surface level, cross-silo learning also tends to follow a three-layer organization structure, where the clients are divided into groups, each group assigned a server, often called a silo or an institution. However, there is a key distinction between cross-silo FL and HFL. In cross-silo FL, it is assumed that the silos or institutions are capable of accessing client data. Moreover, each silo or institution is autonomous, often independent of the cloud service provider. In other words, each silo or institution can be viewed as a single client in the sense of traditional two-layer FL. Hence, in this work, we do not consider cross-silo FL as HFL.

7.2 Federated Edge Learning

Similar to traditional centralized federated learning, FEEL is an approach to machine learning that allows multiple devices to collaborate and learn a shared model. The two are different in the location of the model training and the data being used.

In a typical centralized federated learning setting, the global model aggregator is located in the cloud. In each round of communication, clients must directly communicate with the cloud, which may result in high communication strains. On the other hand, federated edge learning moves the global aggregator to the edge server, greatly reducing the latency and bandwidth constraints for communication between the clients and the server. However, in doing so, federated edge learning trades off scalability, since an edge server is limited to serving clients within its service range. In some situations, it may also lose the benefits of flexibility and access to an abundance of computing resources, which are both features enjoyed by cloud-based FL.

8 Conclusions

In this paper, we provided an original definition for HFL in edge computing environments in general, and the metaverse in particular. We then presented four pieces of early prototypes of HFL architectures that initialized this field of study, and compared client-edge-edge with client-edge-cloud architectures from both communication and scalability perspectives. The latter architecture would be the most fitting alternative for the metaverse. We then explored different types of HFL based on the number of layers and also presented an argument that these architectures could generally be replaced by three-layer client-edge-cloud for better performance and simplicity. Next, we demonstrated the advantages of the client-edge-cloud architecture in the metaverse, showing one example work that studied the utilization of multi-teacher knowledge distillation in FL. Finally, we outlined some potential future research directions in the field of HFL, based on existing research in traditional federated learning.

References

[1] MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [EB/OL]. (2016-02-17) [2024-04-16]. http://arxiv.org/abs/1602.05629

[2] ZHAO Y, LI M, LAI L Z, et al. Federated learning with non-IID data [EB/OL]. (2018-06-02) [2024-04-16]. http://arxiv.org/abs/1806.00582

[3] MAO Y Y, YOU C S, ZHANG J, et al. A survey on mobile edge computing: The communication perspective [J]. IEEE communications surveys & tutorials, 2017, 19(4): 2322 – 2358. DOI: 10.1109/COMST.2017.2745201

[4] KREUTZ D, RAMOS F M V, ESTEVES VERISSIMO P, et al. Software-defined networking: A comprehensive survey [J]. Proceedings of the IEEE, 2015, 103(1): 14 – 76. DOI: 10.1109/jproc.2014.2371999

[5] LIU L M, ZHANG J, SONG S H, et al. Client-edge-cloud hierarchical federated learning [C]//IEEE International Conference on Communications (ICC). IEEE, 2020: 1 – 6. DOI: 10.1109/ICC40277.2020.9148862

[6] ABAD M S H, OZFATURA E, GUNDUZ D, et al. Hierarchical federated learning ACROSS heterogeneous cellular networks [C]//Proceedings of ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 8866 – 8870. DOI: 10.1109/ICASSP40776.2020.9054634

[7] LIU L M, ZHANG J, SONG S H, et al. Hierarchical federated learning with quantization: Convergence analysis and system design [EB/OL]. (2021-03-26) [2024-04-16]. http://arxiv.org/abs/2103.14272

[8] LIU C, CHUA T J, ZHAO J. Time minimization in hierarchical federated learning [EB/OL]. (2022-10-07) [2024-04-16]. http://arxiv. org/abs/2210.04689

[9] LIU X F, WANG Q, SHAO Y F, et al. Sparse federated learning with hierarchical personalized models [EB/OL]. (2023-09-25) [2024-04-16]. http://arxiv.org/abs/2203.13517

[10] NGUYEN M D, PHAM Q V, HOANG D T, et al. Label driven Knowledge Distillation for Federated Learning with non-IID Data [EB/OL]. (2022-09-30) [2024-04-16]. http://arxiv.org/abs/2209.14520

[11] Wang X, Wang Y J. Asynchronous Hierarchical Federated Learning [EB/OL]. (2022-05-31) [2024-04-16]. https://arxiv.org/abs/2206.00054

[12] WANG J Y, WANG S Q, CHEN R R, et al. Demystifying why local aggregation helps: convergence analysis of hierarchical SGD [J]. Proceedings of the AAAI conference on artificial intelligence, 2022, 36(8): 8548 – 8556. DOI: 10.1609/aaai.v36i8.20832

[13] WU W T, HE L G, LIN W W, et al. Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems [J]. IEEE transactions on parallel and distributed systems, 2021, 32(7):

1539 – 1551. DOI: 10.1109/TPDS.2020.3040867

[14] RIZK E, SAYED A H. A graph federated architecture with privacy preserving learning [C]//The 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). IEEE, 2021: 131 – 135. DOI: 10.1109/SPAWC51858.2021.9593148

[15] SUN Y C, SHAO J W, MAO Y Y, et al. Semi-decentralized federated edge learning for fast convergence on non-IID data [C]//Proceedings of IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2022: 1898 – 1903. DOI: 10.1109/WCNC51071.2022.9771904

[16] ZHONG Z C, ZHOU Y P, WU D, et al. P-FedAvg: Parallelizing federated learning with theoretical guarantees [C]//IEEE Conference on Computer Communications. IEEE, 2021: 1 – 10. DOI: 10.1109/INFOCOM42981.2021.9488877

[17] DAS A, PATTERSON S. Multi-tier federated learning for vertically partitioned data [C]//IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021: 3100 – 3104. DOI: 10.1109/ICASSP39728.2021.9415026

[18] HOSSEINALIPOUR S, BRINTON C G, AGGARWAL V, et al. From federated to fog learning: distributed machine learning over heterogeneous wireless networks [J]. IEEE communications magazine, 58(12): 41 – 47, 2020. DOI: 10.1109/MCOM.001.2000410

[19] LIN T, STICH S U, PATEL K K, et al. Don't use large mini-batches, use local SGD [EB/OL]. (2018-08-22) [2024-04-16]. http://arxiv.org/abs/1808.07217

[20] ZHOU F, CONG G J. A distributed hierarchical SGD algorithm with sparse global reduction [EB/OL]. (2022-02-17) [2024-04-16]. http://arxiv.org/abs/1903.05133

[21] ZHU G, LIU D, DU Y, et al. Toward an intelligent edge: wireless communication meets machine learning [J]. IEEE communications magazine, 58(1): 19 – 25, 2020. DOI: 10.1109/MCOM.001.1900103

[22] KAIROUZ P, MCMAHAN H B, AVENT B, et al. Advances and Open Problems in Federated Learning [J]. Foundations and trends® in machine learning, 14(1 – 2): 1 – 210, 2021

Biographies

GU Cheng received his BASc Degree of Honours in Computer Engineering Cooperative Program with Distinction in 2022, and his MASc degree from the Department of Electrical and Computer Engineering in May 2024, both from University of Waterloo, Canada. His research interests focus on building next-generation AI assisted distributed systems.

LI Baochun (bli@ece.toronto.edu) received his BE degree from Tsinghua University, China in 1995 and his MS and PhD degrees from the University of Illinois at Urbana-Champaign, USA in 1997 and 2000, respectively. Since 2000, he has been with the Department of Electrical and Computer Engineering, the University of Toronto, Canada, where he is currently a Professor. Since August 2005, he has been holding the Bell Canada Endowed Chair in computer engineering. He was the recipient of IEEE Communications Society Leonard G. Abraham Award in the field of communications systems in 2000, the Multimedia Communications Best Paper Award from the IEEE Communications Society in 2009, the University of Toronto McLean Award in 2009, the Best Paper Award from IEEE INFOCOM in 2023, and the IEEE INFOCOM Achievement Award in 2024. He is a Fellow of the Canadian Academy of Engineering, the Engineering Institute of Canada, and IEEE. His current research interests include cloud computing, security and privacy, distributed machine learning, federated learning, and networking.

Learned Distributed Query Optimizer: Architecture and Challenges

GAO Jun[1], HAN Yinjun[2], LIN Yang[2], MIAO Hao[1], XU Mo[2]

(1. Peking University, Beijing 100871, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202402007

https://kns.cnki.net/kcms/detail/34.1294.TN.20240521.1932.002.html, published online May 23, 2024

Manuscript received: 2023-07-28

Abstract: The query processing in distributed database management systems (DBMS) faces more challenges, such as more operators, and more factors in cost models and meta-data, than that in a single-node DMBS, in which query optimization is already an NP-hard problem. Learned query optimizers (mainly in the single-node DBMS) receive attention due to its capability to capture data distributions and flexible ways to avoid hard-craft rules in refinement and adaptation to new hardware. In this paper, we focus on extensions of learned query optimizers to distributed DBMSs. Specifically, we propose one possible but general architecture of the learned query optimizer in the distributed context and highlight differences from the learned optimizer in the single-node ones. In addition, we discuss the challenges and possible solutions.

Keywords: distributed query processing; query optimization; learned query optimizer

Citation (Format 1): GAO J, HAN Y J, LIN Y, et al. Learned distributed query optimizer: architecture and challenges [J]. ZTE Communications, 2024, 22(2): 49 – 54. DOI: 10.12142/ZTECOM.202402007
Citation (Format 2): J. Gao, Y. J. Han, Y. Lin, et al., "Learned distributed query optimizer: architecture and challenges," *ZTE Communications*, vol. 22, no. 2, pp. 49 – 54, Jun. 2024. doi: 10.12142/ZTECOM.202402007.

1 Introduction

Distributed database management systems (DBMSs) serve as a key infrastructure to manage data when the storage and processing of data exceed the capability limitation of a single-node computer's node. The data are usually partitioned into different computer nodes according to different strategies, such as hash, range, and round-robin, and then the query is processed over the partitioned data transparently. Different kinds of distributed DBMS have emerged recently, including the distributed version of traditional databases[1 – 2] and new products in the distributed context[3 – 4]. In this paper, we assume that the distributed DBMS runs on homogenous hardware and software, which is more like parallel databases. We ignore the differences between the two terms in this paper.

Query processing is essential to the distributed DBMS. As in a single-node DBMS, the query processing in distributed DBMS hides its implementation details. When the end users submit a query, a distributed DBMS makes an evaluation plan according to its meta-data and statistics maintained, retrieves data from the different partitions, and shuffles data when needed. End users only need to express their query needs without considering the data placement or the detailed evaluation plans.

We can see that distributed query processing faces more challenges than its single-node version. First, the data are organized as partitions across the different computing nodes. Different partition strategies will impact the following query performances, and thus some kinds of statistics should be maintained to guide the query optimization at the granularity of partitions. Second, more operators are introduced in the distributed DMBS, including the operators to move the data across nodes, distributed versions for traditional operators, etc. These operators offer much larger space for optimization. Third, the cost model in the single-node DBMS should be substantially extended to consider other factors in the distributed context, like the communication cost and the computation skew. In fact, the evaluation cost in distributed DBMSs is largely determined by the slowest computer node[4].

The search space in the query optimization of the distributed DBMS includes the join order search, the physical operator selection, the movement of data, and the placement of operators on the computer nodes. The first two aspects, which are also present in a single-node DBMS query optimization, are recognized as NP-hard problems[5]. The latter two are related to the multiple computer nodes in distributed DBMSs. The data stored in the partitions have movement strategies different from other computer nodes, each of which is for various communication and computation costs[6]. Additionally, the placement of operators on computation nodes should be considered, as some pushing-down operators can reduce the inter-

This work was partially supported by NSFC under Grant Nos. 61832001 and 62272008, and ZTE Industry-University-Institute Fund Project .

mediate results.

One simple but usually effective method in query optimization of the distributed DMBS is the heuristic rule. That is, the plan is generated as in the single-node DMBS in the first phase. Then in the second phase, the distributed DMBS chooses the distributed version for the physical operators in their evaluation plan, in which the data are first shuffled to other nodes, the computation is performed in nodes, and the final results are collected from nodes. The heuristic optimizer can lower the cost in the optimization phase and always exploit the power of different computer nodes, which results in a competitive performance.

However, the heuristic optimizer is too rough for distributed query processing. Ref. [7] points out that the plan should be constructed as a whole, while the partitions are totally ignored in the heuristic plan generation. In addition, all physical operators involve the data movement in the first phase, whose communication cost may be reduced if one data movement can be shared by multiple operators. Moreover, the cost model should be extended to consider the communication and computation skewness. Finally, it is hard for these heuristic rules to adapt to new hardware in distributed DBMSs.

Learned query plan generation[8 – 14] has been studied and shows its advantages in the single-node DBMS. Most learned optimizers[9 – 13] view plan generation as a sequence of action decisions. They introduce encoders to represent a query expression, capture data statistics, and then rely on a reinforcement learning (RL) framework to learn the action policy or the value of states, which can be trained with the evaluation latency or cost of the generated query evaluation plan.

We also notice that generative models like GPT have achieved remarkable success in various tasks, which also brings important inspiration to the learned query plan generation. However, we cannot directly apply the GPT models, at least the current version, to generate the query plan. As a language model, GPT cannot yield the tree-structured plan directly. In addition, GPT lacks statistics data and meta data in distributed DBMSs, which is highly needed in efficient plan generation.

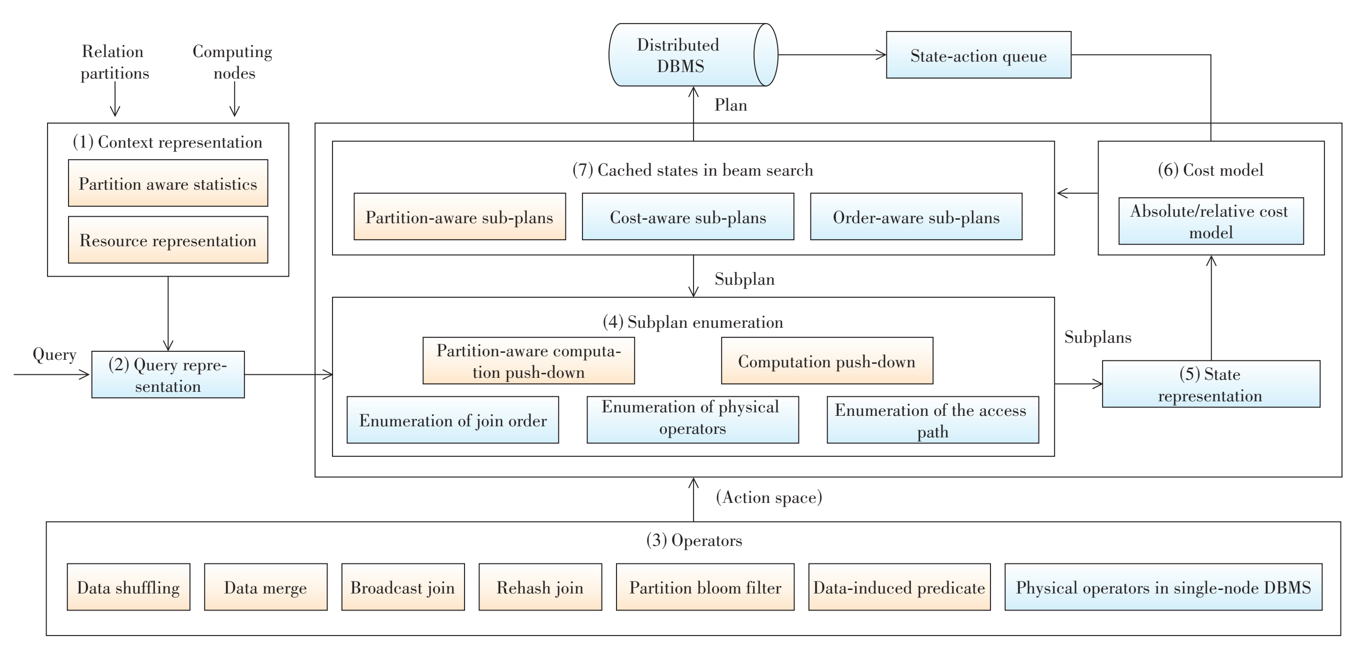
This paper mainly focuses on how to extend the learned optimizer in single-node DBMSs to distributed DBMSs. We first propose a possible but general architecture, then discuss challenges as well as possible solutions. We notice that there exist surveys[15 – 19] for the combination of AI and databases, ranging from learned index, learned data layout, database tuning, and plan generation, and there are some works[20 – 22] accelerating the query processing using new hardware in distributed DBMSs. Different from existing works, this paper focuses on the learned plan generation in the distributed context.

2 Architecture of Learned Distributed Query Optimizer

In this section, we first show the overall architecture of a learned query optimizer in the distributed DBMS, and then detail each component especially in aspects of distributed extensions compared with those in the single-node DBMS.

2.1 Overview

Our proposed architecture is similar to the ones presented in works like ROTS[9], LOGER[13], etc. As shown in Fig. 1, the



DBMS: database management system

▲**Figure 1. Architecture of a learned query optimizer in distributed DBMS** boxes with grey color indicate the related techniques that have been discussed in the single-node DBMS. It follows the RL framework to generate the final plan evaluation tree. The architecture fits the value-based RL models, and most of the components can be reused in the policy-based RL models. The states, actions, and rewards can be described as follows.

- The states are the intermediate forest of the sub-plan trees in plan generation. The initial state contains all table nodes, each for one sub-plan tree. The table embeddings come from (2) component. The table nodes also capture the data statistics modeled by (1) component. The state transition indicates that two sub-plan trees are merged. When the generation terminates, an entire evaluation tree is constructed. Due to the different order of tables in join operations and different placement of operators, the space of states is huge. Each state is encoded into a state representation in (5) and is further fed into the value model in (6) to measure which states have the potential to achieve high-performance results.
- The actions in RL indicate choosing two sub-plan trees and selecting one operator node to merge two sub-plan trees in each step to plan generation. The internal nodes in the evaluation tree are operators in relational algebra. Besides physical operators in the single-node DMBS, the distributed DBMS introduces operators to optimize the communication cost, skip data partitions, and operate among partitions in (3). As the action space is huge, the value-based RL can take the heuristic plan enumeration rules in (4) into consideration, which only produces the potential promising states for evaluation. The plan search can take a beam search strategy to keep more candidates in the plan enumeration in (7), even though the candidates are not with the minimal estimated cost at that time[12 – 13]. Such a strategy is in the same line with dynamic programming in SystemR, a well-known classic query optimizer.
- The rewards come from the feedback of distributed DBMS. As Structured Query Language (SQL) hints can specify the join order and physical operator, the final generated plan can be expressed using SQL hint first and then is submitted to the distributed DBMS for evaluation. The latency of the query plan can be collected as the rewards in the training of the value model. The rewards can take the form of the absolute query latency or the relative speedup compared with the latency of the plan using the default optimizer from the database.

2.2 Context Representation

The context representation component enables the query representation to be aware of the computing resource in different nodes and data distribution. The computing resource modeling and the statistics among the partitions are rarely studied in the single-node DBMS.

It is necessary to model computing resources as the slowest computing nodes that dominate the entire cost in distributed computing. The computing nodes in the distributed DBMS are interconnected by the network. Thus, the computing resource can be modeled as a graph, where each computing node has various computing and storage features, and the link is for the communication between nodes. The computing resource may be modeled with a graph neural network (GNN), in which the embedding of the computing nodes is generated.

For the modeling of intra-partition statistics, it can leverage the advanced techniques for the extensively studied cardinality estimation[23 – 28] in the single-node DBMS, which is roughly categorized into the data-driven[23 – 25] and query-driven[26 – 28] with different training signals. Note that, besides the machine learning based statistics, distributed DBMSs can build histograms or bloom filters in partitions, and these exact statistics enable distributed DBMSs to develop new operators, like data-induced predicates[29], to skip partitions in query evaluation. For the modeling of inter-partition statistics, the intra-partition data distribution representation can be attached to the computing nodes as one kind of feature and then the computing resource graph can learn the data distributions among the computing nodes.

2.3 Query Representation

The query representation component converts a query expression into a representation. Roughly, we have two methods to generate the query representation, namely based on the logical query graph and based on the plan from the database. Such a component is necessary to the distributed query optimizer, but shows a few differences with that in the single-node DBMS.

The logical query graph-based method is to build the query representation from a logical query graph, which is constructed from query expression. A logical query graph can contain table nodes with predicates as their features and edges with the join predicates, or takes the form of a heterogonous graph with table nodes, column nodes, predict nodes, etc., which is adopted by the Real Time Operating System (RTOS), LOGER. Then, a GNN is employed to learn the table embeddings, which will be fed into the following query generation components.

The plan-based method takes the plan generated by the default optimizer in the DBMS[30, 40]. QueryFormer[30] is a representative work that uses a transformer to model the plan. As the plan tree is deep, a virtual node is introduced to enable fast information exchanges among nodes in the plan. The plan from the database contains rich and easily-exploited features. However, the plan takes the tree form, and two tables that can be joined directly may be far in the tree, which may lead to some kinds of information loss compared with the logical query graph.

2.4 Physical Operators (Action Space)

The actions in RL indicate selecting one operator as an internal node and extending one or two sub-plan trees. The action space is related to the number of the physical nodes and the join orders. Obviously, the more operators in the distributed DBMS, the larger the search space. Here, we mainly list some of the operators introduced for the distributed DBMS, including the communication-related operators, the join operators, and partition-specific operators.

The communication-related operator can be one class of operators. Recall that the distributed version for physical operators includes the data movement first and then computation next. We can extract the communication operators as the first-class operators, which enables more flexibility in the plan construction. For example, Ref. [31] devises a method to exchange the order of communication and computation. In addition, the data merger operator can move the data from one partition to another, which can then reduce the communication cost in the following join operations. In other words, opposite to the data partition, a data merge operator lowers the communication overhead at the cost of the computation skew.

The join operators can be divided into the intra-partition and inter-partition operators. The intra-partition join operators can take a similar way as those in the single-node DBMS, like by hash-based and nested loop, and merge join according to the size and statistics distribution of input data. The inter-partition join is also extensively studied in Ref. [32], including the hash join to handle the equal join predicates, fragment-and-replicate join[33] for general join predicates, and asymmetric fragment-and-replicate join when two join tables are skewed.

The partition-specific operators work on partitions. The partitions are units in the data storage, and the query performance can be improved if the partitions are skipped. The partition bloom filter is one operator which can quickly check whether the partition contains the data meeting the query conditions or not. Another operator is the data-induced predicate operator[29], which can derive new predicates on partitions from the query and partition-related statistics.

2.5 Plan Enumeration

The action space in the distributed DBMS is huge. We can then introduce the heuristic rules in the plan enumeration, which only produces the promising candidates and then improves the stability of the reinforcement learning. Note that the heuristic rules are easy to incorporate in the value-based RL, which is taken by most of the learned optimizers. The following heuristic strategies recently studied in the literature can be considered in the learned optimizer.

The placement of the data movement operation is considered in Ref. [31]. As mentioned above, the data movement (or shuffling) becomes an independent operator and can be commutative with other operators in the plan. For example, with the consideration of the existing partition scheme, the data movement can be placed earlier, which can then reduce the large intermediate results.

Computation push-down[34] is also considered in the plan generation. That is, instead of collecting data from partitions and then performing operations like aggregating/distinction over the collected data, we can perform operators first at the granularity of partitions, and then transfer the results to the next phase. The predicates evaluated early at the granularity of partitions can reduce the intermediate results, which usually reduces the communication cost in the following.

2.6 State Representation

The state representation component captures different features of a constructed sub-plan forest and converts it into an embedding state that will feed into the following value model to help choose the candidate actions. This component shares similar functionalities with that of a single-node DBMS, but also displays several differences.

The first difference is the form of the evaluation plan. The plan in the single-node DBMS always takes the form of a left-deep tree, which can allow index-loop join and support the pipeline evaluation. However, neither the index-loop join nor the pipeline is the key improvement in the distributed DMBS. In other words, there are more bushy structure plans in the distributed DBMS. Such a change may impact the design of the underlying representation model. For example, QueryFormer[30] introduces a virtual node to handle the deep tree, while this issue is not serious in the bushy form.

The second difference is the heterogeneous nature of operators. As we mentioned before, distributed DBMSs support more operators than single-node DBMSs, while different operators are with different features, like input/output (IO), CPU, and communication cost. Most of the plan representation models, like Tree-CNN[35], Tree-LSTM[36], and Transformer[37], handle the nodes with the shared features space, which should be extended to capture the heterogeneous nature of operators.

2.7 Value Model

The value model plays a crucial role in the value-based RL and has a similar functionality to the critic model in the policy-based RL, which can be used to determine which states can result in a better performance. For example, Oceanbase[38] points out that in some cases the computation push-down is not beneficial, which can be detected with the aid of the value model. The value model on the final plan is actually the cost model for the query evaluation plan. Although the cost model is more complex, the learned value model in the distributed DBMS is similar to the corresponding one in the single-node DBMS.

The cost model in distributed DBMS should consider more factors, including the IO cost, CPU cost, communication cost, and the data skew. Fortunately, the cost model[39] in the learned optimizer need not explicitly express the weights of different factors. The value model will be trained with the signals like the query latency. With more data trained, the weights of different factors can be learned in the cost model automatically.

One possible extension of the cost model is the choice of the absolute or relative cost model. The absolute value model tries to mimic the latency for one query, and the relative cost model compares the values of two candidate plans. As a plan contains multiple nodes, and the relative cost model can capture the differences between two plans more easily, we guess that the relative cost model might be more suitable in the distributed DBMS.

2.8 Cached Sub-Plans in Plan Search

The cached plans are the candidates which need to be explored during the plan search. To cache one plan with the potential minimal cost is similar to the greedy search, which can be implemented efficiently at the cost of sub-optimal results. Thus, it needs to cache more candidates to gain performance improvement. The existing learned optimizers, like BALSA and LOGER, incorporate more cached sub-plans in beam search.

The progress in the distributed DBMS shows the different criteria in selecting sub-plans to cache. Besides the sub-plans with the minimal cost or preserving tuple orders, some distributed DBMSs, like Oceanbase[3], cache the sub-plans with interesting partitioning, which may avoid data partitioning in the following operators.

These plan caching strategies can be combined into the learned query optimizer easily using the beam search. The value model can help choose the sub-plan with the possible minimal cost. In addition, we can apply similar heuristic rules in distributed DBMSs to select the sub-plans with orders on specific attributes which could be useful for later operations. These plans are cached for future exploration using the beam search in learned query optimizers.

3 Challenges of Learned Distributed Query Optimizer

We identify two key challenges associated with the learned distributed query optimizer and discuss potential solutions to these issues.

3.1 Instability of Learned Query Optimizer

The learned query optimizer has a high chance to produce the plans with performance improvement when queries in the test set share the similarity with those in the training set. However, when the test query differs, the learned query optimizer may produce sub-optimal or bad plans. The stability of the learned query optimizer should be substantially enhanced before it can be deployed in real-life applications.

There are two possible solutions to the issue. BAO[10] mainly generates the candidate plans using existing DBMS optimizers with learned global parameters, and the bad performance is avoided with the help of the existing DBMSs. In fact, BAO has been extended to distributed DBMS[11]. However, such a method cannot produce plans from scratch and also is restricted by the capability of DBMSs.

An alternative solution is to extend the learned optimizer to produce both the candidate plan and its confidence in producing such a plan. When the confidence is lower than a given threshold, it directly relies on the DBMS to produce the final plan. In this way, plans with poor performance can be avoided.

3.2 High Training Cost of Learned Query Optimizer

The training of a learned query optimizer needs to search candidate plans from a huge plan space, in which each candidate is evaluated against the DBMS to obtain its latency. In addition, RL is notoriously hard to converge. In all, the training of a learning query optimizer is expensive.

Besides the transfer learning and meta-learning RL to improve the sample efficiency, an evolutionary algorithm (EA) could be one possible solution. In fact, EA methods have been adopted by PostgreSQL and can produce high-quality plans when the number of tables exceeds a given threshold. The existing research study, like learned concurrency control, also shows that EA algorithms could produce more effective results with less training cost than reinforcement learning[41].

4 Conclusions

The advance of distributed DBMS is required to incorporate the promising learned query optimizer. This paper outlines a possible architecture of the learned optimizer, mainly highlighting the differences from the learned optimizer in the single-node DBMS. In addition, this paper lists two major challenges and discusses their possible solutions.

References

[1] MUKHERJEE N, CHAVAN S, COLGAN M, et al. Distributed architecture of Oracle database in-memory [J]. Proceedings of the VLDB endowment, 2015, 8 (12): 1630 – 1641. DOI: 10.14778/2824032.2824061

[2] BLAKELEY J A, CUNNINGHAM C, ELLIS N, et al. Distributed/heterogeneous query processing in Microsoft SQL server [C]//The 21st International Conference on Data Engineering (ICDE'05). IEEE, 2005: 1001 – 1012. DOI: 10.1109/ICDE.2005.51

[3] YANG Z K, YANG C H, HAN F S, et al. OceanBase [J]. Proceedings of the VLDB endowment, 2022, 15(12): 3385 – 3397. DOI: 10.14778/3554821.3554830

[4] CHANG L, WANG Z W, MA T, et al. HAWQ: a massively parallel processing SQL engine in hadoop [C]//The 2014 ACM SIGMOD International Conference on Management of Data. ACM, 2014: 1223 – 1234. DOI: 10.1145/2588555.2595636

[5] IBARAKI T, KAMEDA T. On the optimal nesting order for computing *N*-relational joins [J]. ACM transactions on database systems, 9(3): 482 – 502. DOI: 10.1145/1270.1498

[6] RUPPRECHT L, CULHANE W, PIETZUCH P. SquirrelJoin: network-aware distributed join processing with lazy partitioning [J]. Proceedings of the VLDB endowment, 2017, 10(11): 1250 – 1261. DOI: 10.14778/3137628.3137636

[7] WANG G P. The optimization of query processing in Oceanbase 4.0. [EB/OL]. (2022-11-23) [2023-08-01]. https: //zhuanlan.zhihu.com/p/586113453

[8] MARCUS R, NEGI P, MAO H Z, et al. Neo: a learned query optimizer [J]. Proceedings of the VLDB endowment, 2019, 12(11): 1705 – 1718. DOI: 10.14778/3342263.3342644

[9] YU X, LI G L, CHAI C L, et al. Reinforcement learning with tree-LSTM for join order selection [C]//The 36th International Conference on Data Engineering

(ICDE). IEEE, 2020: 1297 – 1308. DOI: 10.1109/ICDE48307.2020.00116

[10] MARCUS R, NEGI P, MAO H Z, et al. BAO: making learned query optimization practical [C]//The 2021 International Conference on Management of Data. ACM, 2021: 1275 – 1288. DOI: 10.1145/3448016.3452838

[11] NEGI P, INTERLANDI M, MARCUS R, et al. Steering query optimizers: a practical take on big data workloads [C]//The 2021 International Conference on Management of Data. ACM, 2021: 2557 – 2569. DOI: 10.1145/3448016.3457568

[12] YANG Z H, CHIANG W L, LUAN S F, et al. Balsa: learning a query optimizer without expert demonstrations [C]//The 2022 International Conference on Management of Data. ACM, 2022: 931 – 944. DOI: 10.1145/3514221.3517885

[13] CHEN T Y, GAO J, CHEN H D, et al. LOGER: a learned optimizer towards generating efficient and robust query execution plans [J]. Proceedings of the VLDB endowment, 2023, 16(7): 1777 – 1789. DOI: 10.14778/3587136.3587150

[14] DOSHI L, ZHUANG V, JAIN G, et al. Kepler: robust learning for faster parametric query optimization [EB/OL]. [2023-08-01]. https://arxiv. org/pdf/2306.06798v2

[15] WANG W, ZHANG M H, CHEN G, et al. Database meets deep learning [J]. ACM SIGMOD record, 2016, 45(2): 17 – 22. DOI: 10.1145/3003665.3003669

[16] ZHOU X H, CHAI C L, LI G L, et al. Database meets artificial intelligence: a survey [J]. IEEE transactions on knowledge and data engineering, 2022, 34(3): 1096 – 1116. DOI: 10.1109/TKDE.2020.2994641

[17] LAN H, BAO Z F, PENG Y W. A survey on advancing the DBMS query optimizer: cardinality estimation, cost model, and plan enumeration [J]. Data science and engineering, 2021, 6(1): 86 – 101. DOI: 10.1007/s41019-020-00149-7

[18] CAI Q P, CUI C, XIONG Y Y, et al. A survey on deep reinforcement learning for data processing and analytics [J]. IEEE transactions on knowledge and data engineering, 2023, 35(5): 4446 – 4465. DOI: 10.1109/TKDE.2022.3155196

[19] ZHAO X Y, ZHOU X H, LI G L. Automatic database knob tuning: a survey [J]. IEEE transactions on knowledge and data engineering, 2023, 35 (12): 12470 – 12490. DOI: 10.1109/TKDE.2023.3266893

[20] GUO C X, CHEN H, ZHANG F, et al. Distributed join algorithms on multi-CPU clusters with GPUDirect RDMA [C]//The 48th International Conference on Parallel Processing. ACM, 2019: 1 – 10. DOI: 10.1145/3337821.3337862

[21] GAO H, SAKHARNYKH N. Scaling joins to a thousand GPUs. [EB/OL]. [2023-08-01]. https://adms-conf.org/2021-camera-ready/gao_presentation.pdf

[22] PAUL J, LU S L, HE B S, et al. MG-join: a scalable join for massively parallel multi-GPU architectures [C]//International Conference on Management of Data. ACM, 2021: 1413 – 1425. DOI: 10.1145/3448016.3457254

[23] YANG Z H, LIANG E, KAMSETTY A, et al. Deep unsupervised cardinality estimation [EB/OL]. (2019-11-21) [2023-08-01]. http://arxiv. org/abs/1905.04278

[24] HILPRECHT B, SCHMIDT A, KULESSA M, et al. DeepDB: learn from data, not from queries! [EB/OL]. (2019-09-02) [2023-08-01]. http://arxiv. org/abs/1909.00607

[25] WANG J Y, CHAI C L, LIU J B, et al. FACE [J]. Proceedings of the VLDB endowment, 2021, 15(1): 72 – 84. DOI: 10.14778/3485450.3485458

[26] DUTT A, WANG C, NAZI A, et al. Selectivity estimation for range predicates using lightweight models [J]. Proceedings of the VLDB endowment, 2019, 12(9): 1044 – 1057. DOI: 10.14778/3329772.3329780

[27] LI B B, LU Y, KANDULA S. Warper: efficiently adapting learned cardinality estimators to data and workload drifts [C]//International Conference on Management of Data. ACM, 2022: 1920 – 1933. DOI: 10.1145/3514221.3526179

[28] NEGI P, WU Z N, KIPF A, et al. Robust query driven cardinality estimation under changing workloads [J]. Proceedings of the VLDB endowment, 2023, 16 (6): 1520 – 1533. DOI: 10.14778/3583140.3583164

[29] KANDULA S, ORR L, CHAUDHURI S. Pushing data-induced predicates through joins in big-data clusters [J]. Proceedings of the VLDB endowment, 2019, 13(3): 252 – 265. DOI: 10.14778/3368289.3368292

[30] ZHAO Y, CONG G, SHI J C, et al. QueryFormer [J]. Proceedings of the VLDB endowment, 2022, 15(8): 1658 – 1670. DOI: 10.14778/3529337.3529349

[31] ZHANG H, YU J X, ZHANG Y K, et al. Parallel query processing: To separate communication from computation [C]//International Conference on Management of Data. ACM, 2022: 1447 – 1461. DOI: 10.1145/3514221.3526164

[32] POLYCHRONIOU O, SEN R, ROSS K A. Track join: distributed joins with minimal network traffic [C]//SIGMOD International Conference on Management of Data. ACM, 2014: 1483 – 1494

[33] STAMOS J W, YOUNG H C. A symmetric fragment and replicate algorithm for distributed joins [J]. IEEE transactions on parallel and distributed systems, 1993, 4(12): 1345 – 1354. DOI: 10.1109/71.250116

[34] YANG Y, YOUILL M, WOICIK M, et al. FlexPushdownDB: hybrid pushdown and caching in a cloud DBMS [J]. Proceedings of the VLDB Endowment, 2021, 14(11): 2101 – 2113

[35] ROY D, PANDA P, ROY K. Tree-CNN: a hierarchical deep convolutional neural network for incremental learning [EB/OL]. (2019-09-18) [2023-08-01]. http://arxiv.org/abs/1802.05800

[36] TAI K S, SOCHER R, MANNING C D. Improved semantic representations from tree-structured long short-term memory networks [EB/OL]. (2015-05-30) [2023-08-01]. http://arxiv.org/abs/1503.00075

[37] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]//The 31st International Conference on Neural Information Processing Systems. ACM, 2017: 6000 – 6010. DOI: 10.5555/3295222.3295349

[38] YANG Z K, YANG C H, HAN F S, et al. OceanBase: a 707 million tpmC distributed relational database system [J]. Proceedings of the VLDB endowment, 2022, 15(12): 3385 – 3397. DOI: 10.14778/3554821.3554830

[39] SIDDIQUI T, JINDAL A, QIAO S, et al. Cost models for big data query processing: Learning, retrofitting, and our findings [EB/OL]. (2020-02-07) [2023-08-01]. http://arxiv.org/abs/2002.12393

[40] MARCUS R, PAPAEMMANOUIL O. Plan-structured deep neural network models for query performance prediction [EB/OL]. (2019-01-31) [2023-08-01]. http://arxiv.org/abs/1902.00132

[41] WANG J C, DING D, WANG H, et al. Polyjuice: high-performance transactions via learned concurrency control [EB/OL]. (2021-06-15) [2023-08-01]. http://arxiv.org/abs/2105.10329

Biographies

GAO Jun (gaojun@pku.edu.cn) received his BE and ME degrees in computer science from Shandong University, China in 1997 and 2000, and his PhD degree in computer science from Peking University, China in 2003. Currently he is a professor with the School of Computer Science, Peking University. His major research interests include web data management, graph data management and AI+DB.

HAN Yinjun is a senior engineer with ZTE Corporation. He has published multiple papers, obtained more than ten authorized patents, won multiple provincial and ministerial awards, and is a senior member of CCF. His main research interests include database systems and storage systems.

LIN Yang is a research and development engineer of ZTE Corporation. She received her master degree from Nanjing University of Science and Technology, China in 2017. Her research interests include query optimization, AI4DB and DB4AI.

MIAO Hao s a postgraduate student in the School of Computer Science, Peking University, China. His major research interests include graph neural network and AI+DB.

XU Mo is a research and development engineer of ZTE Corporation. He received his master degree from Monash University, Australia. His research interests include query optimization, AI4DB and database kernel development.

Review on Service Curves of Typical Scheduling Algorithms

GAO Yuehong[1], NING Zhi[1], HE Jia[1], ZHOU Jinfei[1],
GAO Chenqiang[2,3], TANG Qingkun[2,3], YU Jinghai[2,3]

(1. Beijing University of Posts and Telecommunications, Beijing 100876, China;
2. ZTE Corporation, Shenzhen 518057, China;
3. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China)

DOI: 10.12142/ZTECOM.202402008

https://kns.cnki.net/kcms/detail/34.1294.TN.20240516.1410.002.html, published online May 16, 2024

Manuscript received: 2024-01-03

Abstract: In recent years, various internet architectures, such as Integrated Services (IntServ), Differentiated Services (DiffServ), Time Sensitive Networking (TSN) and Deterministic Networking (DetNet), have been proposed to meet the quality-of-service (QoS) requirements of different network services. Concurrently, network calculus has found widespread application in network modeling and QoS analysis. Network calculus abstracts the details of how nodes or networks process data packets using the concept of service curves. This paper summarizes the service curves for typical scheduling algorithms, including Strict Priority (SP), Round Robin (RR), Cycling Queuing and Forwarding (CQF), Time Aware Shaper (TAS), Credit Based Shaper (CBS), and Asynchronous Traffic Shaper (ATS). It introduces the theory of network calculus and then provides an overview of various scheduling algorithms and their associated service curves. The delay bound analysis for different scheduling algorithms in specific scenarios is also conducted for more insights.

Keywords: network calculus; service curve; scheduling algorithm; QoS

Citation (Format 1): GAO Y H, NING Z, HE J, et al. Review on service curves of typical scheduling algorithms [J]. ZTE Communications, 2024, 22(2): 55 – 70. DOI: 10.12142/ZTECOM.202402008
Citation (Format 2): Y. H. Gao, Z. Ning, J. He, et al., "Research on multi-core processor analysis for WCET estimation," *ZTE Communications*, vol. 22, no. 2, pp. 55 – 70, Jun. 2024. doi: 10.12142/ZTECOM.202402008.

1 Introduction

With the rapid advancement of internet technology and applications, the variety of services within networks has continuously expanded and network structure has grown increasingly complex. The traditional internet, which relies on the best effort (BE) service model and the First Come First Served (FCFS) scheduling algorithm, can no longer meet the diverse quality-of-service (QoS) requirements of different services. To ensure QoS, international standardization organizations have introduced several Internet architectures, including Integrated Services (IntServ), Differentiated Services (DiffServ), Time-Sensitive Networking (TSN), and Deterministic Networking (DetNet).

The IntServ model reserves network resources based on the QoS requirements of a given traffic flow before transmission[1]. This pre-allocation of network resources ensures end-to-end QoS guarantees for the traffic flow. However, due to its protocol implementation intricacies and inefficient bandwidth utilization, the IETF introduced the DiffServ model[2]. The DiffServ model distinguishes itself through its unique strategy of marking data at the network's edge nodes, which is crucial for defining the subsequent handling and processing of the transmitted data.

In 2005, the Institute of Electrical and Electronics Engineers (IEEE) established the Audio-Video Bridging (AVB) working group, aiming to develop Ethernet AVB technology[3]. AVB is a set of real-time audio and video transmission protocols based on a new Ethernet architecture. In 2012, the IEEE 802.1 task group officially renamed AVB as TSN. TSN encompasses a range of technical standards, primarily focused on clock synchronization, data stream scheduling strategies, and network and user configurations.

In 2015, the IETF established the DetNet working group, with a specific focus on achieving deterministic, worst-case bounds on delay, packet loss, and jitter by implementing deterministic transmission paths at the second-layer bridging and third-layer routing segments[4]. This allows for predictable latency in network communications.

In order to provide guidance for the development of network

This work was supported by ZTE Industry-University-–Institute Cooperation Funds.

technologies and the practical planning of networks, researchers are increasingly emphasizing the use of mathematical models to analyze network performance. In earlier years, researchers employed mathematical theories such as the probability theory and queueing theory to analyze network service performance. However, as network structures became more complex and services diversified, the limitations of these theories gradually became evident. Network calculus, a theoretical framework that incorporates min-plus algebra, has been instrumental in transforming intricate nonlinear queueing problems into mathematically tractable models. Therefore, in recent years, network calculus has been proven to be an effective and versatile tool for analyzing network component performance. These components can encompass links, schedulers, shapers, or even entire networks. As a result, network calculus has found widespread application in scenarios of internet QoS[5 – 12].

Network calculus uses service curves to describe the service capabilities of network elements such as routers, schedulers, and links. In the min-plus algebra theory, the service curves of interconnected components can be combined through convolution, resulting in an overall service curve. Consequently, individual systems along a network path can be easily connected by convolving their service curves, thus obtaining a specified end-to-end service curve for the network. By combining the service curve with the arrival curve, network performance bounds can be determined. Different network architectures provide QoS guarantees for services within the network by employing various traffic shaping and scheduling algorithms at network nodes. Consequently, the service curves of different scheduling algorithms constitute a crucial foundation for analyzing network performance. Therefore, this paper aims to summarize the service curves for different scheduling algorithms.

The main contribution of this paper is to compile the concepts and service curves of seven typical scheduling algorithms from existing studies. The remaining content of this paper is arranged as follows: Section 2 introduces the basic concept of network calculus; Section 3 introduces different scheduling algorithms and their service curves; Section 4 calculates the delay bounds using service curves from various scheduling algorithms and conducts simulation in burst flows situations scenarios; Section 5 concludes by summarizing the contributions of the study.

2 Network Calculus

Network calculus was initially developed to analyze the performance of networks with non-probabilistic distribution traffic. The origins of network calculus can be traced back to CURZ's research on traffic characteristics and network performance boundaries[13 – 14], as well as the studies by PAREKH and GALLAGHER on the service curves of Generalized Processor Sharing (GPS) schedulers[15 – 16]. Subsequently, researchers like CRUZ and SARIOWAN advance the study of network calculus[17 – 23], formalizing the concept of service curves and establishing the foundational framework for network calculus[24].

Ref. [25] systematically discusses the fundamental concepts of deterministic network calculus, along with the foundation and theory of min-plus algebra, and further integrates specific analyses involving internet traffic and scheduling mechanisms. This section provides a brief introduction to the basic framework of network calculus based on the content presented in Ref. [25].

Network calculus is built upon the foundation of min-plus algebra and is used for the analysis of network performance. It involves two fundamental non-decreasing operations: min-plus convolution $\otimes$ and min-plus deconvolution $\oslash$:

$$(a \otimes b)(x) = \inf_{0 \leqslant y \leqslant x} \left[a(y) + b(x-y) \right], \tag{1}$$

$$(a \oslash b)(x) = \sup_{y \geqslant 0} \left[a(x+y) - b(y) \right]. \tag{2}$$

Both arrival curves and service curves are determined through min-plus convolution. The arrival curve $\alpha(t)$ serves as a mathematical model for the constrained arrival process $A(t)$ of a traffic flow, where $A(t)$ represents the cumulative input function, quantifying the amount of data that has arrived at the network node up to time *t*.

$$A(t) \leqslant A \otimes \alpha(t) = \inf_{0 \leqslant s \leqslant t} \left\{ A(s) + \alpha(t-s) \right\}. \tag{3}$$

The affine arrival curve $\alpha_{r,b}(t) = rt + b$ is a common type of arrival curve. This model is frequently used to describe traffic patterns where data is transmitted at a constant rate of *r* after an initial burst of size *b*.

The service curve $\beta(t)$ describes the capacity of a node to provide services. Let the departure function be denoted as A^* (*t*), which is the cumulative output function representing the total amount of data output from the network node up to time *t*. $\beta(t)$ satisfies the following relationship:

$$A^*(t) \geqslant \inf_{0 \leqslant s \leqslant t} \left\{ A^*(s) + \beta(t-s) \right\} = A \otimes \beta(t). \tag{4}$$

A strict service curve $\beta(t)$ satisfies the following relationship throughout any backlog period:

$$A^*(t + \Delta t) - A^*(t) \geqslant \beta(\Delta t). \tag{5}$$

A commonly used service curve is the Latency-Rate (LR) service curve $\beta_{r,T}(t) = r[t-T]^+$, which provides a simple way to describe the worst-case behavior of various scheduling algorithms[26]. This service curve represents the service node's guarantee of providing a service rate of r to a traffic flow, with the additional constraint that delays do not exceed T.

Network calculus encompasses several fundamental theorems that serve as the theoretical underpinnings for analyzing network performance. Here are some of the basic theorems in network calculus[27]:

1) Delay bound: The delay $D(t)$ of traffic at a network node at time t is bounded by the maximum horizontal distance between the arrival curve $\alpha(t)$ and the service curve $\beta(t)$:

$$D(t) \leqslant h(\alpha,\beta) = \sup_{s \geqslant 0} \left\{ \inf\left\{ \tau \geqslant 0 : \alpha(s) \leqslant \beta(s+\tau) \right\} \right\}. \tag{6}$$

2) Backlog bound: The backlog of traffic at a network node is bounded by the maximum vertical deviation between the arrival curve $\alpha(t)$ and the service curve $\beta(t)$:

$$B(t) \leqslant v(\alpha,\beta) = \sup_{s \geqslant 0} \left\{ \alpha(s) - \beta(s) \right\}. \tag{7}$$

3) Output characterization: The deterministic arrival process of the departure process A^* can be represented by the deterministic arrival curve $\alpha^*(t)$:

$$\alpha^*(t) = \alpha \oslash \beta = \sup_{s \geqslant 0} \left\{ \alpha(t+s) - \beta(s) \right\}. \tag{8}$$

4) Concatenation property: The deterministic service curve provided to data flows after the concatenation of several nodes can be represented as the convolution of these nodes' service curves:

$$\beta(t) = \beta^1(t) \otimes \beta^2(t) \cdots \otimes \beta^n(t). \tag{9}$$

5) Superposition: The arrival curve of an aggregated flow can be represented as the pointwise sum of the arrival curves of individual flows:

$$\alpha(t) = \sum_{i=1}^{n} \alpha_i(t). \tag{10}$$

6) Leftover service: This theorem addresses the number of services that remain available in a network node after some data have been served to specific flows. It is often used to calculate the remaining service capacity or service curve of a network element. For a network node that includes two flows, A_1 and A_2, with the node's service curve being $\beta(t)$, $A^*{}_1(t)$ satisfies the following equation:

$$A^*{}_1(t) \geqslant A_1 \otimes \left(\beta - \alpha_2\right)^+(t). \tag{11}$$

3 Scheduling Algorithms and Service Curves

In the expansive landscape of scheduling algorithms for network communications, a multitude of strategies exist to address the diverse challenges posed by data transmission. Among this myriad of options, we focus on seven distinctive scheduling algorithms that encapsulate both classical methodologies and contemporary innovations. The selected algorithms include classical strategies such as Strict Priority (SP), Round Robin (RR), and Weighted Fair Queuing (WFQ). Furthermore, this paper delves into the advanced strategies introduced by TSN, namely Cycling Queuing and Forwarding (CQF), Time Aware Shaper (TAS), Asynchronous Traffic Shaper (ATS), and Credit Based Shaper (CBS). The specifics of these algorithms are discussed in this paper, as well as their relation to the creation of service curves.

3.1 Strict Priority

3.1.1 Scheduling Algorithm

SP is a classic and pivotal scheduling strategy known for its effectiveness in ensuring high-priority data stream transmission performance. It proves particularly suitable for applications requiring guaranteed low latency. SP strictly follows the order of queue priorities. Packets with the same priority are scheduled using the FCFS policy. Each queue's packets must wait until all packets in higher priority queues have been scheduled before they have the opportunity to be scheduled. It ensures that high-priority data receive preferential treatment within the constraints of limited resources.

3.1.2 Service Curve

In the context of preemptive SP scheduling, if data from a high-priority flow arrive at the server, they will receive an immediate service, even if data from a low-priority flow are currently being serviced. Consequently, lower-priority flows do not affect the service provided by the server to higher-priority flows. It is sufficient to analyze two types of flows: the flow with priority level i and the aggregate flow with priorities greater than i. According to the theorem of Leftover Service, the service curve for the flow with priority level i is as follows[25].

$$\beta_i(t) = \left[\beta(t) - \sum_{j=1}^{i-1} \alpha_j(t) \right]^+, \tag{12}$$

where $\beta(t)$ represents the overall service curve provided by the service node, $\alpha_i(t)$ represents the arrival curve for the data flow with priority level i, and $[x]^+$+ denotes $\max(0, x)$.

In practical networks, data flows are composed of data packets and non-preemptive strategies are often employed. In a non-preemptive SP scheduling, if a low-priority data packet has started being serviced, it continues to be serviced even if higher-priority data packets arrive. In such cases, the service curve for a non-preemptive SP scheduling for a flow with priority level i is as follows[25].

$$\beta_i^{SP}(t) = \beta(t) - \sum_{j=1}^{i-1} \alpha_j(t) - \max_{j>i} l_j^u, \tag{13}$$

where l_j^u represents the maximum packet length in the queue with priority level j.

3.2 Weighted Fair Queuing

3.2.1 Scheduling Algorithm

While the SP scheduling algorithm holds a crucial position in network communications, especially for latency-sensitive applications, it exhibits noticeable limitations. In instances where high-priority data streams are overly frequent, SP may lead to prolonged neglect of low-priority data streams, impacting overall fairness. In the contemporary field of network communications, ensuring both fairness and efficiency in scheduling various priority and data stream types is paramount for QoS. To address this, researchers have introduced WFQ, a strategy that employs weight assignments and a fair queuing approach. WFQ aims to provide equitable and efficient services to different data streams by ensuring that each stream receives a fair share of resources based on its assigned weight. This approach acknowledges the importance of maintaining fairness while effectively managing the diverse priorities and types of data streams in today's network communication landscape.

WFQ allocates the bandwidth to each flow based on their queue weights. WFQ, also known as Packet General Processor Sharing (PGPS), is a variant of the idealized scheduling strategy called GPS[15–16]. GPS is a scheduling strategy that allocates a certain proportion of service guarantees to each priority queue based on weight parameters. For n queues with weight parameters $\varphi_1,\cdots,\varphi_n$, queue i is guaranteed a service proportion of $\varphi_i/\sum\varphi_j$. However, GPS is an idealized strategy that assumes each data segment can be divided into infinitely small segments to ensure proportional sharing at every point in time and data size. In practical applications, bits cannot be divided, and data packets are usually not fragmented. Researchers have introduced PGPS scheduling to approximate the ideal GPS strategy under real-world constraints.

In a WFQ system, when a packet arrives, its departure time in the corresponding GPS system is calculated. The system selects the packet with the smallest departure time for transmission. This process introduces a monotonically increasing virtual time function, denoted as $V(t)$:

$$V\left(t_{j-1}+\tau\right)=V\left(t_{j-1}\right)+\frac{\tau\cdot C}{\sum_{i\in B_j}\varphi_i},\ \ \tau<t_j-t_{j-1}. \tag{14}$$

In Eq. (14), $V(0)=0$, and t_j represents the time at which the j-th event occurs in the system. These events can include either the departure or arrival of data packets. B_j represents the set of non-empty queues between the time t_{j-1} and t_j.

We can define the arrival time of the k-th data packet in queue i as a_i^k, and the length of this packet as L_i^k. Additionally, the allocation of transmission rates to each queue i in a WFQ system is represented by weight parameters φ_i. S_i^k and F_i^k represent the virtual start time and virtual finish time of the k-th data packet in queue i. If there is no special explanation, set S_i^0 to 0. When queue i is empty, $S_i^k=V(a_i^k)$, and when queue i is not empty, $S_i^k=F_i^{k-1}$.

$$S_i^k=\max\left\{F_i^{k-1},V\left(a_i^k\right)\right\},$$

$$F_i^k=S_i^k+\frac{L_i^k}{\varphi_i}. \tag{15}$$

The virtual finish time F_i^k for each queue is calculated according to Eq. (15). Then, the system selects the data packet for transmission by choosing the one with the smallest virtual finish time F_i^k among all the queues. This ensures that the packet from the queue with the earliest virtual finish time is sent next, maintaining fairness in resource allocation.

3.2.2 Service Curve

Refs. [15] and [16] introduce the concept of GPS scheduling strategy and conducted performance analysis of GPS schedulers in both single-node and multi-node scenarios. The derived expression for GPS node availability of a service is considered to be the first service curve formula in network calculus. This work laid the foundation for understanding and analyzing service curves in the context of network calculus. If data flow i experiences backlog within the time interval (s,t), the following condition holds for data flow j in all cases:

$$\phi_j\left(D_i(t)-D_i(s)\right)\geqslant\phi_i\left(D_j(t)-D_j(s)\right). \tag{16}$$

The service curve for data flow i is as follows.

$$\beta_i^{\mathrm{GPS}-1}(t)=\frac{\varphi_i}{\sum_{j=1}^{n}\varphi_j}\beta(t). \tag{17}$$

WFQ can be considered a packet-based form of GPS, which means that we can derive the service curve for data flow i within a WFQ node:

$$\beta_i^{\mathrm{WFQ}-1}(t)=\left[\beta_i^{\mathrm{GPS}-1}(t)-\max\nolimits_{1\leqslant j<n}l_j^u\right]^{+}. \tag{18}$$

The previous conclusion assumes that all data flows are continuously backlogged, fully utilizing the allocated bandwidth. In reality, if one or more flows do not make full use of their allocated resources, the remaining service shares will be redistributed to the backlogged flows based on their weights. Ref. [26] takes into account practical scenarios and used the departure process's arrival curve to provide a more general service curve for data flow i:

$$\beta_i^{GPS-2}(t) = \max_{M \subseteq \{1,\cdots,n\}} \left\{ \frac{\varphi_i}{\sum_{j \in M} \varphi_j} \left[\beta(t) - \sum_{j \notin M} \alpha_j^*(t) \right]^+ \right\}. \tag{19}$$

The departure process' s arrival curve α_j^* is determined based on the arrival curve α_j and the service curve specified in Eq. (18).

In addition, the research in Refs. [15] and [16] assumes that all flows have affine arrival curves of the form $\alpha_i(t) = r_i t + b_i$, with constant link rates, and the system is stable, ensuring that the overall average arrival rate does not exceed the link capacity. However, Ref. [28] conducted research without being constrained by these conditions and derived more general service curves:

$$\beta_i^{GPS-3}(t) = \mathop{max}_{M \subseteq \{1,\cdots,n\} \backslash \{i\}} \left\{ \frac{\varphi_i}{\sum_{j \notin M} \varphi_j} \left[\beta(t) - \sum_{j \in M} \alpha_j(t) \right]^+ \right\}. \tag{20}$$

3.3 Round Robin

3.3.1 Scheduling Algorithm

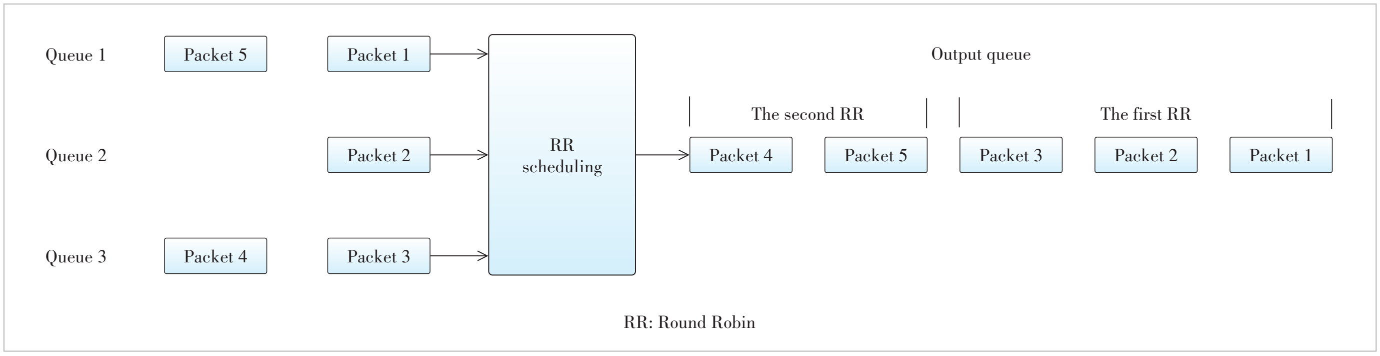
RR scheduling strategy[29] is proposed to address the limitations of SP. Similar to WFQ, RR utilizes weight assignments to enhance overall fairness within the system. RR also comes in various variants, each employing different methods of allocating transmission resources to improve overall system performance from distinct perspectives.

RR employs a polling mechanism to schedule multiple queues, and within each queue, a FCFS scheduling strategy is applied. During each polling round, the scheduler sequentially sends the first data packet from each queue, skipping over empty queues. Fig. 1 shows a scheduling scenario.

The RR algorithm provides uniform cyclic scheduling services to different queues. However, it may result in unfairness when dealing with queues with varying packet sizes. Additionally, it does not differentiate between services with different latency requirements, making it challenging to guarantee QoS for high-latency-sensitive services. As a result, various variants of RR are introduced, such as Weighted Round Robin (WRR)[30] and Deficit Round Robin (DRR)[31].

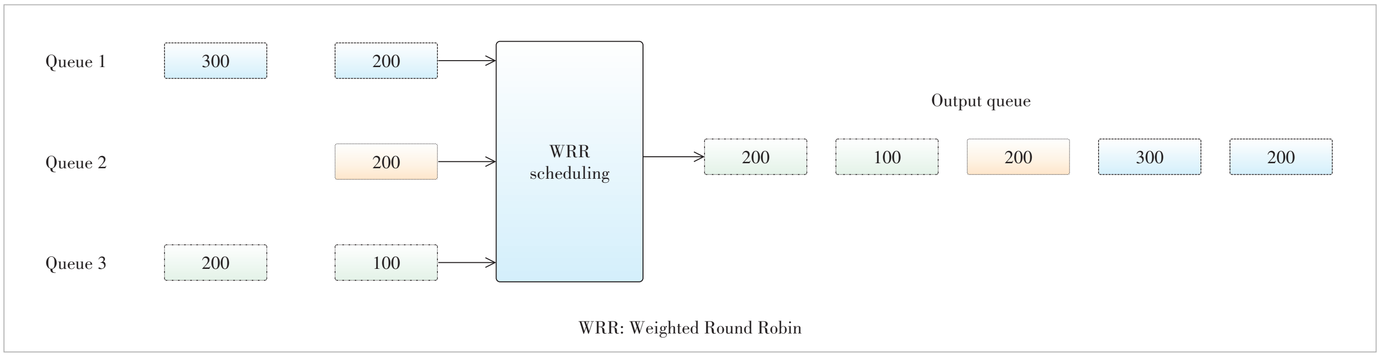
WRR allocates service proportionally to the weight of each queue during polling. In the classic WRR, during each polling round, each queue consecutively sends data packets w_i that is the weight of the queue. The standard RR scheduling can be seen as a special case of WRR, where the weight of each queue is 1. For example, in a WRR scenario with weights 2:2:1, the polling sequence would allocate services as shown in Fig. 2.

However, continuous sending of consecutive data packets w_i can lead to burstiness in data flows and potentially impact other queues. Interleaved Weighted Round-Robin (IWRR) addresses this issue by eliminating the impact through an alter-



RR: Round Robin

▲Figure 1. RR scheduling



WRR: Weighted Round Robin

▲Figure 2. WRR scheduling

nating mechanism. In this approach (Fig. 3), each queue is assigned a counter, which is initialized based on the queue's weight. When a queue with a non-zero counter is polled, it sends one data packet and reduces its counter by 1. Once the counter reaches 0, the queue is skipped, and the scheduler moves on. A new scheduling round begins when the counters of all queues are 0.

DRR is a scheduling algorithm that operates based on packet lengths. In DRR, each queue is assigned a counter and initialized to the maximum number of bytes that can be scheduled in one round, known as Quantum. During each round of polling, when a queue is reached and if the length of a packet in the queue is less than the counter's value, the packet is sent and the counter is reduced by the length of the sent packet. This process continues until the counter's value becomes smaller than the length of the first packet in the queue, at which point the next queue is scheduled. After each polling round, the counter is reset to its maximum value and a new scheduling round begins.

For example, when all queues have a Quantum of 200, the DRR scheduling mechanism functions as illustrated in Fig. 4.

3.3.2 Service Curve

Researchers have extensively studied the service curves of RR and its variants.

For WRR, it is evident that the packet length has a significant impact on the performance of the WRR algorithm. The researchers in Ref. [32] model the packet length sequence and provide three different service curve models with varying levels of precision and complexity:

$$\beta_i^{WRR-1}(t) = \frac{q_i}{q_i + Q_i}\left[\beta(t) - Q_i\right]^+, \tag{21}$$

$$\beta_i^{WRR-2}(t) = \lambda_1 \otimes \upsilon_{q_i, q_i + Q_i}\left(\left[\beta(t) - Q_i\right]^+\right), \tag{22}$$

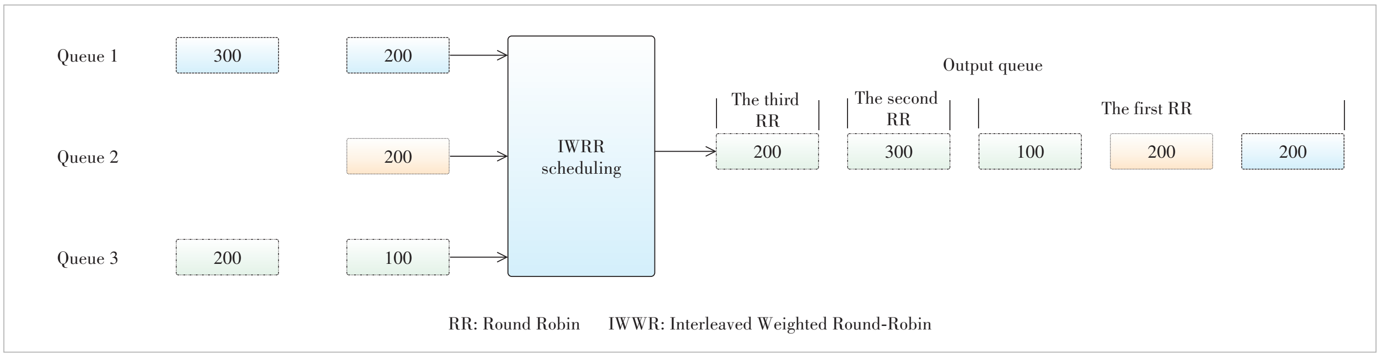
$$\beta_i^{WRR-3}(t) = f_i^{-1}\left(\beta(t)\right), \tag{23}$$

where $Q_i = \sum_{j \neq i} w_j l_j^u$ represents the maximum amount of data that all other queues can receive in one round of polling and $q_i = w_i l_i^l$ represents the minimum amount of data that queue i can receive in one round of polling (l_i^u and l_i^l represent the maximum and minimum packet lengths in queue i and w_i is the WRR weight of the queue).

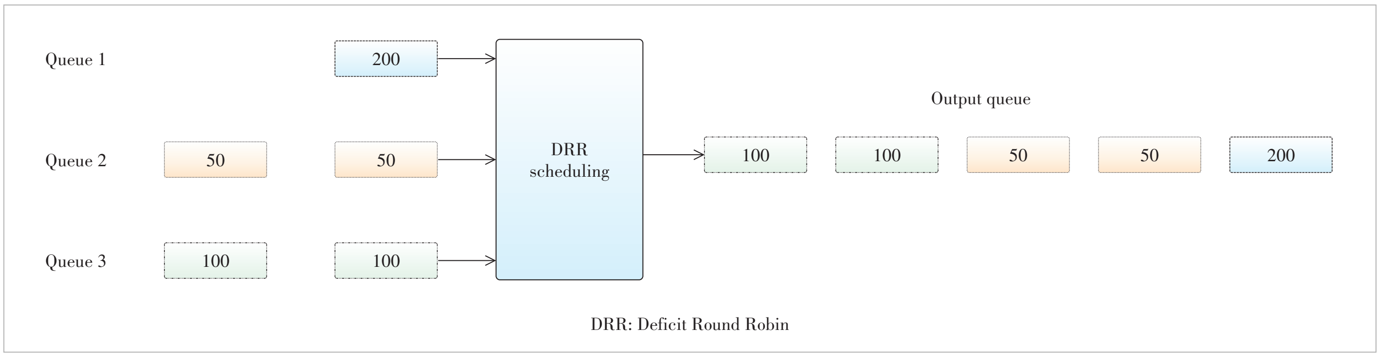
Functions $\upsilon_{a,b}(t)$, $\lambda_k(t)$ and $f_i(x)$ can be expressed as:

$$\upsilon_{a,b}(t) = a\left\lceil \frac{t}{b} \right\rceil,\ t > 0, \tag{24}$$

$$\lambda_k(t) = kt,\ t > 0, \tag{25}$$



▲Figure 3. IWWR scheduling



▲Figure 4. DRR scheduling

$$f_i(x) = x + \sum_{j \neq i} L_j^u\left(w_j\left\lfloor 1 + \frac{g(x)}{w_i}\right\rfloor\right), \tag{26}$$

$$g = \sup\left\{x | L_i^l(x) \leqslant y\right\}, \tag{27}$$

where L_i^u and L_i^l represent the upper and lower bounds of the cumulative packet length sequence of data stream i, respectively. The data packet length sequence L_i satisfies the following condition:

$$\forall j, n \in N, L_i^l(n) \leqslant \sum_{j}^{j+n-1} L_i(j) \leqslant L_i^u(n). \tag{28}$$

Among the three service curves mentioned above, Eq. (23) is the most accurate, as it utilizes the data packet curve. However, it is computationally more complex.

If the maximum packet length L_i^u and minimum packet length L_i^l are known, we can get $L_i^u(n) = nl_i^u(n)$ and $L_i^l(n) = nl_i^l$. In this case, Function $f_i(x)$ can be simplified as:

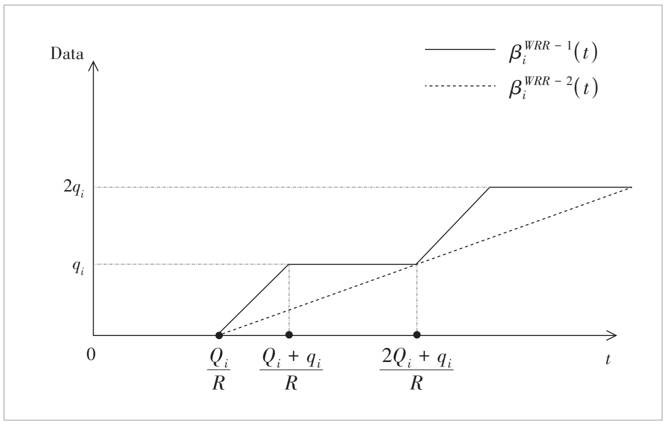
$$f_i(x) = x + Q_i\left\lfloor \frac{x}{w_i} \right\rfloor + Q_i. \tag{29}$$

In this case, the inverse function takes the form of $\lambda_1 \otimes \upsilon_{q_i, q_i + Q_i}$, which results in the service curve being transformed into the service curve in Eq. (22).

Eq. (22) obtains the details of the polling using a pseudo-inverse method, taking into account the total bandwidth used by the packets currently being serviced. On the other hand, Eq. (21) can be viewed as the linearized result of Eq. (22) and their service curve is depicted in Fig. 5.

For IWRR, Ref. [33] employs a method similar to Eq. (22) to obtain the service curve for IWRR using a pseudo-inverse:

$$\beta_i^{IWRR}(t) = \lambda_1 \otimes \sum\nolimits_{k=1}^{w_i - 1} \upsilon_{l_i^l, L_{tot}}\left(\left[\beta(t) - \psi_i^{IWRR}\left(kl_i^l\right)\right]^+\right), \tag{30}$$



▲Figure 5. Service curve for $\beta_i^{WRR-1}(t)$ and $\beta_i^{WRR-2}(t)$

$$\psi_i^{IWRR}(x) = x + \sum_{j \neq i} \theta_{i,j}^{IWRR}\left(\left\lfloor \frac{x}{l_i^l} \right\rfloor\right) l_j^u, \tag{31}$$

$$\theta_{i,j}^{IWRR}(x) = \left\lfloor \frac{x}{w_i} \right\rfloor w_j + \left[w_j - w_i\right]^+ + \min\left(x\, mod\, w_i + 1, w_j\right). \tag{32}$$

For DRR, Ref. [31][34][35] study the worst-case performance under strict assumptions, mainly assuming that the server has a constant service rate. On the other hand, Ref. [36] uses network calculus to analyze the delay of DRR in a more general scenario, encompassing the results from Refs. [31], [34] and [35]. It derives the DRR service curve as:

$$\beta_i^{DRR}(t) = \left[\frac{Q_i}{F}\beta(t) - \frac{Q_i\left(L - l_i^u\right) + \left(F - Q_i\right)\left(Q_i + l_i^u\right)}{F}\right]^+, \tag{33}$$

where $F = \sum\nolimits_{i=1}^{n} Q_i$, representing the total maximum number of bytes processed in one round, and $L = \sum\nolimits_{i=1}^{n} l_i^u$, representing the sum of the upper bounds of packet sizes from all queues. Additionally, considering that packet sizes are discrete and multiples of a base unit ε (e.g., 1 byte), we define $l_i^{u-\varepsilon} = l_i^u - \varepsilon$ and $L = \sum\nolimits_{i=1}^{n} l_i^{u-\varepsilon}$. The service curve is defined as follows:

$$\beta_i^{DRR-\varepsilon-1}(t) = \left[\frac{Q_i}{F}\beta(t) - \frac{Q_i\left(L^\varepsilon - l_i^{u-\varepsilon}\right) + \left(F - Q_i\right)\left(Q_i + l_i^{u-\varepsilon}\right)}{F}\right]^+. \tag{34}$$

To some extent, the polling process can be considered as a simulation of GPS. Therefore, the analysis of RR can also leverage research on GPS. The researchers in Ref. [37] introduce a more comprehensive concept, the bandwidth sharing policy, to unify GPS and RR. They introduce a new method to derive the service curves for bandwidth-sharing scheduling policies and improve the performance boundaries by exploiting the characteristics of cross traffic. They prove that for a variable-capacity network node with service curve $\beta(t)$ and bandwidth-sharing parameters φ_j, $1 \leqslant j \leqslant n$, if $\beta(t)$ is a convex function and all data flow arrival curves $\alpha_j(t)$ are concave functions, there exists a non-negative integer set H_M, $M \subseteq \{1, \cdots, n\} / \{i\}$, and then the service curve for data flow i can be expressed as:

$$\beta_i^{BS}(t) = \sup_M \left\{\frac{\varphi_i}{\sum_{j \notin M} \varphi_j}\left[\beta(t) - \sum_{i \in M} \alpha_i(t) - H_M\right]^+\right\}. \tag{35}$$

Ref. [37] introduces an inductive process to compute H_M and applies the conclusion to GPS and DRR. For GPS, in the particular case where H_M=0, they obtain the same result as in Ref. [28]. For DRR, the service curve can be rewritten as:

$$\beta_i^{BS-DRR}(t) = \sup_{M} \left\{ \frac{Q_i}{\sum_{j \notin M} Q_j} \left[\beta(t) - \sum_{i \in M} \alpha_i(t) - H_M \right]^+ \right\}. \quad (36)$$

Furthermore, Ref. [38] makes further advancements using the pseudo-inverse and output arrival curves from the research in Refs. [36] and [37]. The pseudo-inverse is initially used to gain more insights into DRR, improving the service curve for DRR and resulting in the new service curve in the absence of arrival constraints.

$$\beta_i^{DRR-\varepsilon-2}(t) = Y_i\big(\beta(t)\big), \quad (37)$$

$$Y_i(x) = \lambda \otimes v_{Q_i, Q_{tot}}\left(\left[x - \psi_i\left(Q_i - l_i^{u-\varepsilon}\right)\right]^+\right) + \min\left(\left[x - \sum_{j \neq i}\left(Q_j + l_j^{u-\varepsilon}\right)\right]^+, \left(Q_i - l_i^{u-\varepsilon}\right)\right), \quad (38)$$

$$Q_{tot} = \sum_{j=1}^{n} Q_j, \quad (39)$$

$$\psi_i^{DRR}(x) = x + \sum_{j \neq i} \theta_{i,j}^{DRR}(x), \quad (40)$$

$$\theta_{i,j}^{DRR}(x) = \left\lfloor \frac{x + l_i^{u-\varepsilon}}{Q_i} \right\rfloor Q_j + \left(Q_i + l_j^{u-\varepsilon}\right). \quad (41)$$

Moreover, Ref. [38] introduces an iterative method to improve the service curve by taking into account arrival curve constraints of interfering flows (with concave arrival curves). This method is applicable to any available strict service curves for DRR.

$$\beta_i^{new}(t) = \max\Big(\beta_i^{old}(t), \max_{M \subseteq \{1,\cdots,n\} \setminus \{i\}} Y_i^M\left(\left[\beta(t) - \sum_{j \in \{1,\cdots,n\} \setminus \{i\} \setminus M} \alpha_j \oslash \beta_i^{old}(t)\right]^+\right), \quad (42)$$

$$Y_i^M(x) = \lambda \otimes v_{Q_i, Q_{tot}^{M,i}}\left(\left[x - \psi_i^M\left(Q_i - l_i^{u-\varepsilon}\right)\right]^+\right) + \min\left(\left[x - \sum_{j \in M}\left(Q_j + l_j^{u-\varepsilon}\right)\right]^+, \left(Q_i - l_i^{u-\varepsilon}\right)\right), \quad (43)$$

$$Q_{tot}^{M,i} = Q_i + \sum_{j \in M} Q_j, \quad (44)$$

$$\psi_i^M(x) = x + \sum_{j \in M} \theta_{i,j}(x). \quad (45)$$

On the other hand, inspired by Ref. [37], the researchers in Ref. [39] introduce a new service curve for WRR that demonstrates improved performance bounds when dealing with cross-traffic and arrival constraints. Assuming a set of flows $N = \{1, \cdots, n\}$ and each flow i is constrained by concave arrival curves $\alpha_i(t)$, the following relationship holds:

$$\beta_i^{BS-WRR}(t) = \sup_{i \in M \subset N} \left\{ \frac{q_i}{q_i + Q_i^M} \left[\beta(t) - \sum_{j \notin M} \alpha_j(t) - Q_i^M \right]^+ \right\}, \quad (46)$$

$$Q_i^M = \sum_{j \in M \setminus \{i\}} w_j l_j^u. \quad (47)$$

3.4 Cycling Queuing and Forwarding

3.4.1 Scheduling Algorithm

The emergence of TSN has brought about enhanced latency determinism and QoS in network communications. The CQF[40] scheduling algorithm, as a crucial component in the field of TSN, plays a vital role in efficiently forwarding data while ensuring fairness. CQF scheduling, through its cycling queuing and precise forwarding mechanism, effectively enhances the network's ability to control latency, reduce data transmission jitter, and simultaneously improve overall network throughput. This algorithm not only safeguards critical data but also contributes to the overall efficiency of the network.

CQF utilizes dual queues in conjunction with gating control principles and service scheduling strategies. It requires precise clock synchronization support. In the CQF mechanism, gate structures are applied exclusively at the output ports of the data buffer queues. When the gate is open, data from the queue is allowed to be forwarded to the next node. Conversely, when the gate is closed, incoming data is buffered within the queue, awaiting transmission.

3.4.2 Service Curve

Considering CQF alternates transmission using two identical queues, Ref. [41] divides the service model into odd and even queues and models the queues as greedy shapers. It constructs the shaping curves based on the service model of time division multiple access (TDMA) from Ref. [42]. Finally, it derives the service curves for the two queues in CQF.

$$\beta^{CQF-1}(t) = \sigma_{odd}(t) = C \cdot \min\left(\left\lceil \frac{t}{2T_Q} \right\rceil \cdot T_Q, t - \left\lfloor \frac{t}{2T_Q} \right\rfloor \cdot T_Q\right), \quad (48)$$

$$\beta^{CQF-0}(t) = \sigma_{even}(t) = C \cdot \max\left(\left\lfloor \frac{t}{2T_Q} \right\rfloor \cdot T_Q, t - \left\lceil \frac{t}{2T_Q} \right\rceil \cdot T_Q\right), \quad (49)$$

where $\beta^{CQF-1}(t)$ is the service curve for the odd queue, β^{CQF-0} is the service curve for the even queue, T_Q is the alternating queue period, and C is the port forwarding rate.

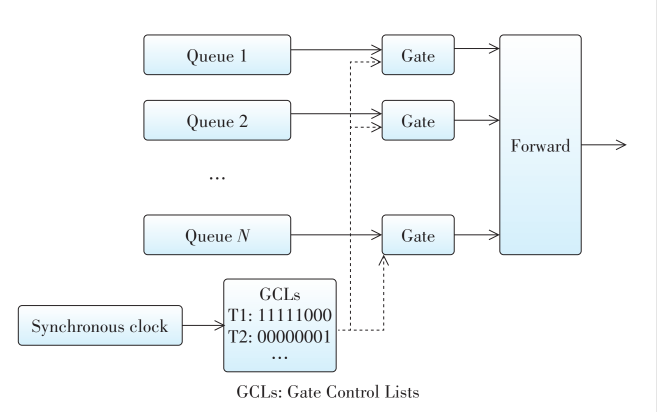
3.5 Time Aware Shaper

3.5.1 Scheduling Algorithm

In industrial IoT and similar contexts, there is a need for stringent requirements in terms of latency and jitter for certain types of data. Exceeding specified thresholds for either latency or jitter can potentially lead to severe consequences. Moreover, such data are often transmitted periodically. To meet the performance demands of these scenarios, TSN introduces scheduled traffic (ST)[43] to support low latency and low jitter applications. Additionally, the TSN framework includes TAS, which ensures that ST flows receive the necessary latency guarantees. TAS relies on highly precise clock synchronization to implement a gate-based scheduling mechanism. TAS periodically scans predefined Gate Control Lists (GCLs) to control the opening and closing of gates associated with different queues. When a gate is open, data frames in the corresponding queue can be transmitted, and when it is closed, they await their turn for transmission. CQF can be perceived as a solution built upon TAS. Fig. 6 shows the structure of TAS.

To ensure that low-priority data frames do not impact the transmission of high-priority data frames, TAS introduces the concept of the guard band (GB). Before the window for high-priority data frames opens, a segment of time equivalent to the GB duration is reserved. During this GB period, new data frames cannot begin transmission. The typical length of the GB is set to the maximum data frame transmission time within the communication network.

To address the issue of resource wastage caused by GB, TSN introduces the frame preemption (FP) mechanism[44]. This mechanism classifies frames based on their priority. The transmission of low-priority frames will be interrupted when high-priority frames arrive, and only resumes after the high-priority frames have been completely transmitted. When a high-priority frame arrives, the system checks if the remaining segment of the low-priority frame being transmitted satisfies the slicing conditions. If it does, the low-priority frame's transmission is paused, and a 4-byte Message Checksum Redundancy Check (MCRC) is added to the already transmitted portion, effectively functioning as a checksum. This enables the assembly of the transmitted frame segments into a complete data frame. The transmission of the high-priority frame begins when a frame interval is available. Once the high-priority frame's transmission is completed, the remaining part of the low-priority frame is supplemented with a preamble code containing assembly information associated with the previously transmitted portion. Subsequently, the low-priority frame's transmission continues.



▲**Figure 6. Structure of Time Aware Shaper (TAS)**

3.5.2 Service Curve

Through the careful construction of GCLs, TAS can periodically reserve transmission resources for different traffic classes, thereby providing reliable QoS guarantees. It can also offer a completely deterministic transmission mode for individual flows. In this fully deterministic mode, there is no need for network calculus analysis, but it has limited application scenarios. Therefore, as discussed in Ref. [45], the analysis of the delay bounds for ST flows focuses on the most general case without specifying the construction method of the GCL.

TAS shares some similarities with TDMA in the sense that they both allocate transmission resources to different services based on a time scale. However, TDMA allows different services to transmit in non-overlapping time slots, which eliminates conflicts between them. In contrast, in the TAS mechanism, when multiple gates are open simultaneously, lower-priority data frames must wait for the completion of the transmission of higher-priority data frames before they can start the transmission. Additionally, without utilizing the FP mechanism, they may also need to wait for even lower-priority data frames to complete their transmission.

Within a GCL period T_{GCL}, the data frames of different priority levels have varying periods and the regions of overlapping gate opening times for different priority queues differ. This results in varying lengths L of each transmission window during each GCL cycle. Let N represent the number of priority queues, S_i denote the interval between the reference position (the starting time of the GCLs period) and the first transmission window for priority queue i, and o_i^j represents the relative offset from the j-th transmission window of queue i to the first transmission window. N_i is the number of transmission windows within one GCL cycle for queue i. In the study of Ref. [44], these transmission window parameters are used to calculate length L_i^j of the j-th transmission window for queue i. Combining this with service model $\beta_{T,L}$ based on TDMA[42], the authors obtain the service curve provided by TAS nodes for data flows with priority level M in the non-preemptive mode.

$$\beta_i^{TAS}(t) = \sum_{j=0}^{N_i - 1} \beta_{T_{GCL}, L_i^j}\left(t + T_{GCL} - L_i^j - S_i - o_i^j\right), \tag{50}$$

$$\beta_{T,L} = C \cdot \max\left(\left\lfloor \frac{t}{T} \right\rfloor \cdot L, t - \left\lceil \frac{t}{T} \right\rceil \cdot (T - L)\right). \tag{51}$$

For TAS and other scheduling algorithms based on precise global synchronized clocks, the service curves discussed in this paper are related to the choice of reference time points. In Eq. (41), S_i represents the time waited from the reference time point to the time when the gate of queue i is opened, and it has a significant impact on the calculated delay results. In the case of CQF in Section 3.4, the different reference points form the distinction between odd and even queues, but in reality, both queues are equivalent.

3.6 Credit Based Shaper

3.6.1 Scheduling Algorithm

To efficiently transmit audio and video data in a local area network, the IEEE 802.1 formed the AVB task group in 2005, which introduced a series of standards. Among these, the 802.1QAV[46] presents the concept of CBS. It classifies data streams into stream reservation (SR) flows and BE flows. SR flows indeed have higher latency requirements and are granted higher priority compared to BE flows. Furthermore, different SR flows can have distinct priority levels among themselves.

For SR flows, CBS utilizes credits (*credit*) to indicate whether their data can be transmitted. When *credit* exceeds 0, the corresponding data category can commence transmission. If the data for the corresponding category is either awaiting transmission or its corresponding queue is empty while the *credit* is less than 0, the *credit* increases at a rate according to *idleSlope* and decreases during transmission at a rate defined by *sendSlope*. Typically, $sendSlope = idleSlope - R$, where R represents the node’s forwarding rate. The bounds for credit in the presence of two different SR flows are as follows[47].

$$sendSlope_A * \frac{l_A^u}{R} \leqslant credit_A \leqslant idleSlope_A * \frac{l_n^u}{R}, \tag{52}$$

$$sendSlope_B * \frac{l_B^u}{R} \leqslant credit_B \leqslant$$
$$idleSlope_B * \left(\frac{l_{BE}^u + l_A^u}{R} - l_n^u * \frac{idleSlope_A}{sendSlope_A * R}\right). \tag{53}$$

Fig. 7 shows the operation of CBS, considering two categories of SR flows, A and B.

In Fig. 7, at time t_1, BE frames arrive. After time t_2, Class-A and Class-B data frames arrive. Subsequently, Class-A and Class-B data frames continuously arrive. During the time interval from t_2 to t_3, the BE frame is transmitted, and Class A and Class B are waiting. Their credits increase at rates of $idleSlope_A$ and $idleSlope_B$, respectively. When BE frame transmission finishes, both Class-A and Class-B credits are greater than 0. Class A, having a higher priority, begins its transmission. Its credit decreases at the rate of $sendSlope_A$. Class-B credit is still increasing at the rate of $idleSlope_B$. After Class-A transmission is completed, Class B follows. At this point, no Class-A frames are waiting in the queue, but its credit is less than 0. The credit increases at the rate of $idleSlope_A$ until it reaches 0. Once Class-B transmission is finished, the credit becomes greater than 0, but there are no Class-B frames left in the queue, so the credit is reset to 0.

3.6.2 Service Curve

In the context of CBS service curves, extensive research has been conducted by AZUA[47], ZHAO[48–50] and MOHAMMADPOUR[10] among others.

The study in Ref. [47] employs network calculus to model AVB networks and derives the LR service curve for CBS nodes serving Class-A and Class-B data flows:

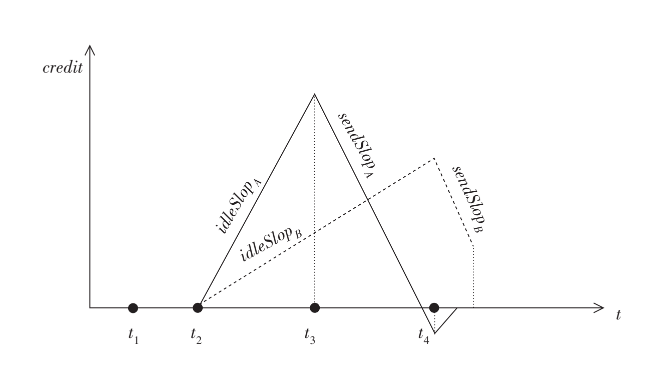
$$\beta_X^{CBS}(t) = R_X\left[t - T_X\right]^+, X = A \text{ or } B, \tag{54}$$

$$R_A = \frac{idSl_A * R}{idSl_A - sdSl_A}, T_A = \frac{l_n^u}{R} - l_A^u \frac{sdSl_A}{idSl_A * R}, \tag{55}$$

$$R_B = \frac{idSl_B * R}{idSl_B - sdSl_B}, T_B = \frac{l_B^u + l_n^u}{R} - \frac{l_n^u}{R}\frac{idSl_A}{sdSl_A} - \frac{l_B^u}{R}\frac{sdSl_B}{idSl_B}, \tag{56}$$

where $idSl_X$ and $sdSl_X$ respectively represent $idleSlope_X$ and $sendSlope_X$, and $X = A \text{ or } B$.

Recognizing that the two SR classes may not meet the diverse requirements of various traffic types in AVB networks, Ref. [48] establishes two types of worst-case delay models and proposes an approach to calculate the worst-case queuing de-



▲Figure 7. Changes in credit during Credit Based Shaper (CBS) operation lay for additional SR data streams. This results in the LR service curve provided by CBS nodes for the additional SR data streams. Rate R_N corresponds to $idSl_N$ and Delay T_N represents the queuing delay experienced by the first frame of additional SR data in the worst-case scenario, which is calculated based on the worst-case queuing delay in Ref. [47].

With the rapid development of communication networks, there is an increasing diversity in the types of services and traffic in the network. To provide service guarantees for a wide range of applications, related standards have continuously expanded, defining various traffic categories with different latency requirements. Researchers have also begun to apply network calculus to analyze SR flows in situations where multiple traffic categories are mixed.

In the study of Ref. [10], control data traffic (CDT) with higher priority than Class-A and Class-B flows is considered, which extends the conclusions from Ref. [46]. If the CDT flow has an affine arrival curve $\alpha_{CDT}(t) = r_{CDT}t + b_{CDT}$, a new service curve can be derived with the following parameters:

$$R_A^{CDT} = \frac{idSl_A\left(R - r_{CDT}\right)}{idSl_A - sdSl_A}, T_A^{CDT} = \frac{1}{R - r_{CDT}}\left(l_n^u + b_{CDT} + \frac{l^u * r_{CDT}}{R}\right), \tag{57}$$

$$R_B^{CDT} = \frac{idSl_A\left(R - r_{CDT}\right)}{idSl_A - sdSl_A}, T_B^{CDT} = \frac{1}{R - r_{CDT}}\left(l_E^u + l_A^u + b_{CDT} - \frac{l_n^u * idSl_A}{sdSl_A} + \frac{l^u * r_{CDT}}{R}\right). \tag{58}$$

On the other hand, Refs. [49] and [50] derive the latency bounds for SR flows under the presence of ST flows and provide service curves for SR flows in both non-preemptive and preemptive modes.

The researchers in Ref. [49] establish the arrival curve for aggregated ST flows based on the ST window, allowing the derivation of service curves for Class-A and Class-B data flows at network nodes. Similar to Ref. [45], an ST window is defined as the time interval when the gate of ST queues opens and closes within a GCLs period T_{GCL}. In each period, there are N windows, and the length of the i-th window is denoted as L_i. The relative offset between the i-th and j-th windows, which represents the time gap between the opening times, is denoted as o_i^j. Based on this, the aggregated arrival curve for ST flows is referenced to the i-th ST window and is given by the following equation:

$$\alpha_{ST,i}(t) = \sum_{j = i}^{i + N - 1} L_j R\left\lceil \frac{t - o_i^j}{p_{GCL}} \right\rceil. \tag{59}$$

If the impact of GB is considered, letting $L_{GB,i}$ represent the length of GB for the i-th window, the aggregated ST flow arrival curve can be expressed as:

$$\alpha_{GB + ST,i} = \sum_{j = i}^{i + N - 1} R \cdot \left(L_j + L_{GB,j}\right) \cdot \left\lceil \frac{t - o_i^j + L_{GB,j} - L_{GB,i}}{T_{GCL}} \right\rceil. \tag{60}$$

In the preemptive mode, when SR frame transmissions are interrupted and need to be supplemented with a preamble code to associate it with the transmitted portion, the arrival curve for the preamble code can be defined as follows if the time overhead of the preamble code is L_{OH}.

$$\alpha_{OH,i}(t) = \sum_{j = i}^{i + N - 1} R * L_{OH}\left\lceil \frac{t - o_i^j - L_j}{T_{GCL}} \right\rceil. \tag{61}$$

It is assumed in Ref. [49] that when an ST queue's gate is open, gates for other queues are closed. The service curve provided by the network node for SR flows is derived with reference to the i-th window as follows:

$$\beta_{X,i}^{CBS - ST - 1}(t) = \frac{idSl_X * R}{idSl_X - sdSl_X}\left[\sup_{0 \leqslant u \leqslant t}\left\{u - T_{X,i}^{\mathrm{mod}}(u)\right\}\right]^+, \tag{62}$$

where $mod \in \left\{np, p\right\}$ represents non-preemptive and preemptive modes, and $X \in \left\{A, B\right\}$. $T_{X,i}^{\mathrm{mod}}$can be expressed as:

$$T_{X,i}^{np}(u) = \frac{\alpha_{GB + ST,i}(u)}{R} + \frac{credit_X^{\max}}{idSl_X}, \tag{63}$$

$$T_{X,i}^{p}(u) = \frac{\alpha_{ST,i}(u)}{R} + \frac{\alpha_{OH,i}(u)}{R}\frac{\left(idSl_X - sdSl_X\right)}{idSl_X} + \frac{credit_X^{\max}}{idSl_X}. \tag{64}$$

However, in Ref. [49], it is assumed that credits remain frozen during the GB period, which contradicts the TSN standard. Additionally, the study in Ref. [50] supports only two SR classes. Ref. [50] extends the conclusions in Ref.[49] to multiple SR classes and the case where credits are not frozen during the GB period. The bound for the credits is provided and the service curves for SR flows are derived as:

$$credit_X^{\min} = sdSl_X * \frac{l_X^u}{R}, \tag{65}$$

$$redit_X^{NF - \max} = idSl_X \frac{\sum_{j = 1}^{X - 1} credit_j^{\min} - l_{> X}^u - \sigma_{GB}}{\rho_{GB} + \sum_{j = 1}^{X - 1} idSl_j - R}, \tag{66}$$

$$credit_X^{F - \max} = idSl_X \frac{\sum_{j = 1}^{X - 1} credit_j^{\min} - l_{> X}^u}{\sum_{j = 1}^{X - 1} idSl_j - R}, \tag{67}$$

$$\beta_X^{CBS-ST-2}(t) = \frac{idSl_X*R}{idSl_X - sdSl_X}\left[t - \frac{\alpha_X^I(t)}{R} - \frac{credit_X^{I-\max}}{idSl_X}\right]^+, \quad (68)$$

where $I \in \{F, NF\}$ represents whether credits freeze (F) or do not freeze (NF) during the GB period, $X \in [1, M]$ represents the maximum number of SR flow classes, which can be up to 6, $credit_X^{I-max}$ refers to the upper bound of the credit value for an SR flow of class X in the context of mode I, and the expression $\alpha_X^I(t)$ refers to the arrival curve for an SR flow of class X:

$$\alpha_X^{NF}(t) = \max_{1 \leqslant i \leqslant N}\left\{\alpha_{ST,i}(t)\right\}, \quad (69)$$

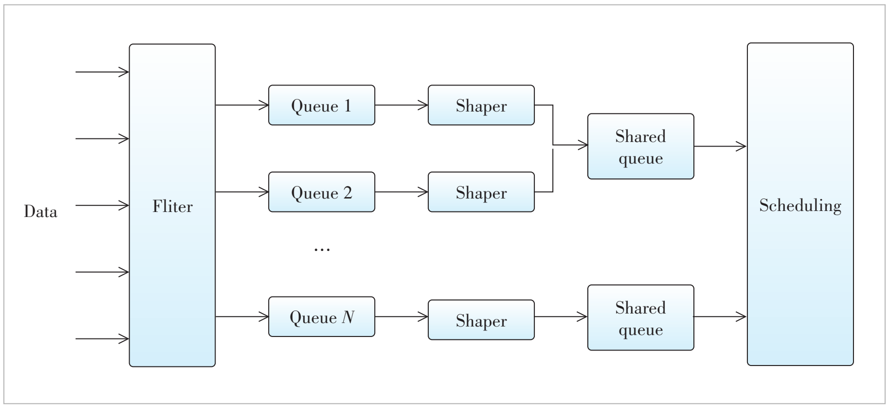
$$\alpha_X^{F}(t) = \max_{1 \leqslant i \leqslant N}\left\{\alpha_{GB+ST,i}(t)\right\}. \quad (70)$$

3.7 Asynchronous Traffic Shaper

3.7.1 Scheduling Algorithm

In addition to TAS and CBS, the TSN working group has also developed the ATS[51] mechanism based on the Urgency-Based Scheduler (UBS)[8]. ATS is designed to reshape asynchronous traffic flows in TSN networks, offering an alternative solution to predictable and real-time communications. ATS operates without the need for global clock synchronization, making it suitable for scenarios where asynchronous traffic needs to be regulated to ensure reliable communications. ATS reshapes traffic at each network hop, reducing burstiness and helping meet timing constraints for various applications. Fig. 8 shows the structure and component parts of ATS.

In ATS, the flow filter first filters out data frames that exceed the size limit. It then determines the internal priority of data streams based on the priority field provided by the virtual local area network (VLAN) tag and parameters associated with flow identification. Data streams with the same priority are grouped into a shared queue. This approach is implemented to effectively manage data streams with similar QoS requirements. To mitigate burstiness in traffic, data streams are required to pass through a shaper before entering shared queues. The shaper is a kind of minimal interleaved regulator. Finally, the ATS node schedules and forwards data from all shared queues.



▲**Figure 8. Structure of Asynchronous Traffic Shaper (ATS)**

3.7.2 Service Curve

Inspired by Ref. [10], the researchers in Ref. [7] analyze ATS using network calculus and divide it into two parts: shaping queues and shared queues. Shared queues are scheduled using SP scheduling. For a shared queue Q_i with priority i, its service curve is as follows:

$$\beta_{Q_i}^{ATS}(t) = R\left[t - \frac{\sum_{j=1}^{i-1}\alpha_{Q_i}(t) + \max_{j>i}\left\{l_{Q_j}^u, l_{BE}^u\right\}}{R}\right]^+, \quad (71)$$

where $\alpha_{Q_i}(t)$ represents the arrival curve for the shared queue, which is determined by the output arrival curve $\alpha_q^*(t)$ of shaping queue q. $\alpha_q^*(t)$ is the sum of the output arrival curves for the various flows f in the aggregated flow of queue q. ATS uses token bucket shaping, where the token bucket's burst size and committed transmission rate are denoted as b_f and r_f, respectively. This can be expressed as:

$$\alpha_q^*(t) = \sum_{f \in q} r_f \cdot t + b_f, \quad (72)$$

$$\alpha_{Q_i}(t) = \sum_{q \in Q_i} \alpha_q^*(t). \quad (73)$$

According to the theorem of delay bound, the upper bound of delay $D_{Q_i}^{ATS}$ can be derived from $\alpha_{Q_i}(t)$ and $\beta_{Q_i}^{ATS}(t)$.

For the shaping queue, the ATS shaper at the back of the FCFS system does not increase the overall system's upper bound on delay. This is a key consideration in understanding how the shaping queue impacts the network. Hence, the delay in the shaping queue, denoted as d_q^{ATS} and the delay in the previous node's shared queue, denoted as $d_{Q_i^-}^{ATS}$, satisfy the relationship $d_q^{ATS}+d_{Q_i^-}^{ATS} \leqslant D_{Q_i^-}^{ATS}$. Additionally, the minimum delay that queue q can experience in Q_i^- is l_q^{min}/C. Therefore, the delay bound for the shaping queue is given by $D_q^{ATS} = D_{Q_i^-}^{ATS} - l_q^{min}/C$. Then Ref. [7] provides the service curve for the shaping queue based on the discussion above:

$$\beta_q^{ATS}(t) = \begin{cases} 0, t \leqslant D_q^{ATS} \\ +\infty, t > D_q^{ATS}. \end{cases} \quad (74)$$

The arrival curve $\alpha_q^{ATS}(t)$ for queue q is determined by the output arrival curve from Q_i^-, and not all flows from Q_i^- will be forwarded to q. Therefore, for queue q, the output arrival curve from Q_i^- is given by:

$$\alpha_{Q_i^-}^*(t) = \frac{\sum\limits_{f \in [Q_i^-, q]} \left(r_f \cdot t + b_f\right)}{\delta_D^{Q_i^-}(t)}, \tag{75}$$

where $\delta_D^{Q_i^-}$ is 0 for $t \leqslant D_q^{ATS}$ and $+\infty$ for $t > D_{Q_i^-}^{ATS}$. Taking into account the physical link constraints that are governed by the link-shaping curve $\delta(t) = Ct$, we have:

$$\alpha_q^{ATS}(t) = \min\left\{\alpha_{Q_i^-}^*(t), \delta(t) + l_{Q_i^-}^{\max}\right\}. \tag{76}$$

By using Eqs. (70 – 75), the delay bounds for all nodes along each flow path can be obtained. Summing up the delays of all nodes can get the end-to-end delay bound.

4 Delay Bound

In this section, we will incorporate specific scenarios and leverage the service curves and the theorem of delay bound discussed earlier to provide theoretical latency bounds for various traffic flows at a single node under different scheduling algorithms. Additionally, the open-source simulation tool NS3 is utilized to construct a simulation platform that allows the configuration of various scheduling algorithms, and the maximum latency experienced by all flows through plenty of simulations is recorded for comparison. This section will conduct a comparative analysis between the simulation results and the theoretical outcomes.

The paper considers a node with a forwarding rate R of 10 Mbit/s, and the service curve for this node is as follows:

$$\beta(t) = Rt. \tag{77}$$

Furthermore, we assume several flows of different priorities or categories passing through the current network node. All flows are periodic burst traffic, represented as (τ_i, σ_i). Here, τ_i represents the period, and σ_i represents the burst size, which is the total size of burst packets within one period. All packets are 64 bytes. The arrival curve $\alpha_i(t)$ can be expressed using the following equation:

$$\alpha_i(t) = \frac{\sigma_i}{\tau_i} t + \sigma_i. \tag{78}$$

According to the theorem of delay bound, the upper bound of the latency for different business flows passing through this node can be expressed as:

$$d_i = h(\alpha_i, \beta_i) = \sup_{s \geqslant 0} \left\{\inf\left\{\tau \geqslant 0 : \alpha_i(s) \leqslant \alpha_i(s + \tau)\right\}\right\}. \tag{79}$$

For SP scheduling, considering three different priority periodic flows, the configurations and latency bounds for different flows are shown in Table 1.

For WFQ scheduling, we consider three flows with different weight configurations. The configurations for different traffic flows are presented in Table 2. The delay bounds for different data flows using different service curves are outlined in Table 3, where β_i^{WFQ-i}, $i = 1,2,3$, represents the WFQ service curves derived from Eqs. (18), (20) and (21).

Although the simulation results are bounded by all theoretical results, β_i^{WFQ-1} assumes that other flows fully utilize the allocated resources, which is not consistent with the scenario presented in this paper. Consequently, the obtained latency bounds are relatively loose. β_i^{WFQ-2} and β_i^{WFQ-3} consider the possibility that each flow may not fully utilize the allocated resources, rather than allocating bandwidth directly according to weights. This results in more stringent latency bounds. For RR scheduling, we consider three flows and allocate weights based on their data arrival rates. We set the base unit ε of DRR as 64 bytes. The configurations and delay bounds for different traffic flows are presented in Tables 4 and 5.

β_i^{WRR-1}, β_i^{WRR-2} and β_i^{WRR-3}, Similar to β_i^{WFQ-1}, allocate bandwidth directly according to weights and packet sizes.

▼Table 1. Latency bounds for Strict Priority (SP) scheduling

Flow ID	Priority	τ_i /ms	σ_i /B	d_i /ms	Simulation Results/ms
1	High	1	256	0.409 6	0.256 0
2	Middle	1	256	0.579 5	0.406 8
3	Low	1	256	1.040 1	0.614 4

▼ Table 2. Flow configurations for Weighted Fair Queuing (WFQ) scheduling

Flow ID	Weight	τ_i/ms	σ_i/B
1	4	1	256
2	3	1	256
3	2	1	256

▼Table 3. Delay bounds for Weighted Fair Queuing (WFQ) scheduling

Method	d_1/ms	d_2/ms	d_3/ms
β_i^{WFQ-1}	0.576 0	0.768 0	1.152 0
β_i^{WFQ-2}	0.576 0	0.631 5	0.665 6
β_i^{WFQ-3}	0.576 0	0.631 5	0.665 6
Simulation results	0.406 8	0.512 0	0.614 4

▼Table 4. Flow configurations for RR scheduling

Scheduling Algorithm	Flow ID	Weight	τ_i/ms	σ_i/B
WRR	1	4	1	256
	2	3	1	256
	3	2	1	256
DRR	1	256	1	256
	2	192	1	256
	3	128	1	256

DRR: Deficit Round Robin RR: Round Robin WRR: Weighted Round Robin

▼Table 5. Delay bounds for RR scheduling

Scheduling Algorithm	Method	d_1/ms	d_2/ms	d_3/ms
WRR	β_i^{WRR-1}	0.716 8	0.921 6	1.280 0
	β_i^{WRR-2}	0.460 8	0.819 2	0.921 6
	β_i^{WRR-3}	0.460 8	0.819 2	0.921 6
	β_i^{BS-WRR}	0.716 8	0.815 5	0.870 6
	Simulation	0.406 8	0.614 4	0.614 4
IWRR	β_i^{IWRR}	0.460 8	0.665 6	0.921 6
	simulation	0.460 8	0.563 2	0.614 4
DRR	$\beta_i^{DRR-\varepsilon-1}$	0.716 8	0.921 6	1.280 0
	β_i^{BS-DRR}	0.460 8	0.614 4	0.921 6
	$\beta_i^{DRR-\varepsilon-2}$	0.460 8	0.819 2	0.921 6
	Simulation	0.406 8	0.614 4	0.614 4

DRR: Deficit Round Robin RR: Round Robin
IWRR: Interleaved Weighted Round Robin WRR: Weighted Round Robin

However, β_i^{WRR-2} and β_i^{WRR-3} utilize pseudo-inverse to leverage the details of RR (i.e. details of resource allocation), with β_i^{WRR-3} additionally considering packet-level details to implement tighter bounds. As the flows in this scenario are similar to periodic flows, β_i^{WRR-2} and β_i^{WRR-3} yield identical results. Especially for Flow 1, its burst packets are sent within a complete RR cycle in this scenario. During this cycle, other flows fully utilize their own transmission resources, resulting in an accurate boundary. β_i^{IWRR} employs the same derivation method as β_i^{WRR-2}, hence yielding similar results. However, IWRR alleviates bursts caused by continuous transmission of packets from the same flow, leading to a reduction in both the theoretical latency bounds and the maximum simulated latency for flow 2. On the other hand, β_i^{BS-WRR} uses resource utilization situations of other flows to improve the latency bounds compared to β_i^{WRR-1}. In certain scenarios, the results obtained are better than those of β_i^{WRR-2} and β_i^{WRR-3}.

The case is similar for DRR. $\beta_i^{DRR-\varepsilon-1}$ allocates resources directly, $\beta_i^{DRR-\varepsilon-2}$ utilizes pseudo-inverse to improve the bounds, and $\beta_i^{BS-DRR-1}$ considers the impact of cross-traffic to improve the bounds.

For CQF scheduling, the actual scheduling process involves the equivalence of two CQF queues. The odd or even queue depends on whether the corresponding queue's gate is open when data arrive. For an individual node, the upper bound on traffic flow delay is jointly determined by the delay bounds of data flows in both queues. However, for this case, the delay bounds are calculated separately for the two queues based on Eq. (49). This paper configures two flows with an alternating period of 4 ms. The different configurations and delay bounds for these business flows are presented in Table 6.

For TAS, this paper considers a simple scenario with three different flows. GLSs are configured based on the characteristics of the traffic flows. The queue gate opening period is the same as the flow period, and the GLS period is the least common multiple of all traffic flow periods. The gate control period is set to 1 ms. Based on the GCLs, the number of transmission windows for the three queues within one GCL period is 1, 2 and 2, respectively. Taking the start time of the GCL period as a reference, the waiting time S_i for the three queues is 0 ms, 2 ms and 2 ms, respectively. Based on Eq. (50), we can calculate the service provided to queue i within each transmission window and then derive the overall service curve. The flow configurations and their corresponding latency upper bounds are shown in Table 7.

In fact, the results in Table 7 do not represent the maximum latency of packets for each flow. This is because their service curves are referenced to a specific time point. In this paper, the starting time of the GCL period is taken as the reference point. The results reflect the maximum latency of all packets in the scenario where they arrive at the reference point. The differences in the results for different flows mainly arise from the time gap between the packet arrival time and the gate opening time.

On the other hand, ATS and CQF utilize gate structures to control transmission, ensuring precise allocation of transmission resources. With a comprehensive understanding of the flow characteristics, they can accurately calculate the latency bounds of packets, achieving deterministic transmission. Therefore, the configuration of the gate significantly influences the latency bounds of scheduling algorithms like CQF and ATS. In practical network scenarios, it is essential to set the configuration based on the characteristics of flows.

For the CBS scheduling, two SR flows are considered in this scenario: SR-A and SR-B. SR-A has a higher priority. Additionally, based on the settings in Refs. [10], [49] and [50], a CDT flow and a ST flow are introduced. In Refs. [49] and [50], the ST traffic arrival curve is determined by the GCLs without regard to the actual characteristics of the ST flows. In this paper, the ST queue's GCL period is set to 6 ms, with the gate opening for the first 1 ms of each period. The configuration and the corresponding delay bounds for each of these service flows are presented in Tables 8 and 9, respectively.

In the context of ATS, as discussed in Section 3.7, when multiple ATS nodes are connected in series, the shaper does not increase the overall system's delay bound. Therefore, this

▼ Table 6. Configurations and delay bounds for Round Robin (RR) scheduling

	τ_i/ms	σ_i/B	d_i/ms	Simulation Results/ms
β^{CQF-1}	1	256	0.204 8	0.204 8
β^{CQF-0}	1	256	4.204 8	4.204 8

▼Table 7. Configurations and delay bounds for TAS

Flow ID	GCL	τ_i/ms	σ_i/B	d_i/ms	Simulation Results/ms
1	100000	6	512	0.409 6	0.409 6
2	010010	3	521	1.409 6	1.409 6
3	001001	3	521	2.409 6	2.409 6

GCL: Gate Control List TAS: Time Aware Shaper

▼**Table 8. Configurations for CBS**

Flow ID	*idleSlope*/Mbit·s^{-1}	*sendSlope*/(Mbit·s^{-1})	τ_i/ms	σ_i/B
SR-A	4	−6	1	256
SR-B	3	−7	1	256
BE	-	-	1	256
CDT	-	-	1	64
ST	-	-	10	64

BE: best effort CDT: control data traffic ST: scheduled traffic
CBS: Credit Based Shaper SR: stream reservation

▼**Table 9. Delay bound for CBS**

		d_{SR-A}/ms	d_{SR-B}/ms	Simulation Results/ms	
				SR A	SR B
β_X^{CBS}		0.640 0	0.938 7	0.460 8	0.563 2
$\beta_X^{CBS-CDT}$		0.650 3	0.920 1	0.512 0	0.614 4
$\beta_i^{CBS-ST-1}$	P	1.588 2	1.852 5	1.470 8	1.573 2
	NP	1.614 4	1.870 4	1.512 0	1.614 4
$\beta_i^{BS-ST-2}$	F	1.614 4	1.870 4	1.512 0	1.614 4
	NF	1.615 3	1.907 7	1.460 8	1.563 2

CBS: Credit Based Shaper NF: non-frozen P: preemptive
F: frozen NP: non-preemptive SR: stream reservation

paper only considers delay $D_{Q_i}^{ATS}$ for the shared queue. The upper bound of the delay is determined by the output arrival curve of shaper $\alpha_q^*(t)$ and the service curve of shared queue $\beta_{Q_i}^{ATS}(t)$. This paper examines three different priority data flows, each entering a different shaper. The relevant parameters and the upper bounds of delay for these flows are presented in Table 10.

5 Conclusions

In this paper, an overview of seven common scheduling algorithms is provided. We summarize the service curves of these scheduling algorithms in different implementations (preemptive and non-preemptive) and under various traffic categories based on existing literature related to network calculus and QoS analysis. Finally, this paper applies different service curves to calculate their corresponding delay bounds in burst flow situations and conducts simulations in corresponding situations. Additionally, this paper conducts a comparative analysis between the simulation results and the theoretical outcomes. The study of this paper can serve as a reference for further research in the field of network modeling and QoS analysis using network calculus.

▼**Table 10. Configurations and delay bounds for ATS**

Flow ID	Committed Transmission Rate/(Mbit·s^{-1})	Burst Size/B	τ_i/ms	σ_i/B	d_i/ms
1	4	128	1	256	0.256 0
2	3	128	1	256	0.597 3
3	2	128	1	256	1.563 0

References

[1] IETF. Integrated services in the internet architecture: an overview: RFC 1633 [S]. 1994

[2] IETF. An architecture for differentiated services: RFC2475 [S]. 1998

[3] DON W. The history of the IEEE 802 standard [J]. IEEE communications standards magazine, 2018, 2(2): 4. DOI: 10.1109/MCOMSTD.2018.8412452

[4] IETF. Deterministic networking problem statement: RFC8557 [S]. 2019

[5] HE F, ZHAO L, LI E S. Impact analysis of flow shaping in ethernet-AVB/TSN and AFDX from network calculus and simulation perspective [J]. Sensors, 2017, 17(5): 1181. DOI: 10.3390/s17051181

[6] FINZI A, MIFDAOUI A, FRANCES F, et al. Incorporating TSN/BLS in AFDX for mixed-criticality applications: model and timing analysis [C]//Proc. 14th IEEE International Workshop on Factory Communication Systems (WFCS). IEEE, 2018: 1 – 10. DOI: 10.1109/WFCS.2018.8402346

[7] ZHAO L X, POP P, STEINHORST S. Quantitative performance comparison of various traffic shapers in time-sensitive networking [J]. IEEE transactions on network and service management, 2022, 19(3): 2899 – 2928. DOI: 10.1109/TNSM.2022.3180160

[8] SPECHT J, SAMII S. Urgency-based scheduler for time-sensitive switched Ethernet networks [C]//Proc. 28th Euromicro Conference on Real-Time Systems (ECRTS). IEEE, 2016: 75 – 85. DOI: 10.1109/ECRTS.2016.27

[9] LE BOUDEC J Y. A theory of traffic regulators for deterministic networks with application to interleaved regulators [J]. IEEE/ACM transactions on networking, 2018, 26(6): 2721 – 2733. DOI: 10.1109/TNET.2018.2875191

[10] MOHAMMADPOUR E, STAI E, MOHIUDDIN M, et al. Latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping [C]//Proc. 30th International Teletraffic Congress (ITC 30). IEEE, 2018: 1 – 6

[11] JIANG Y M. Some properties of length rate quotient shapers [EB/OL]. (2021-07-11)[2023-10-13]. http://arxiv.org/abs/2107.05021

[12] JIANG Y M. A basic result on the superposition of arrival processes in deterministic networks [C]//Proc. IEEE Global Communications Conference (GLOBECOM). IEEE, 2018: 1 – 6. DOI: 10.1109/GLOCOM.2018.8647202

[13] CRUZ R L. A calculus for network delay. part I: network elements in isolation [J]. IEEE transactions on information theory, 1991, 37(1): 114 – 131. DOI: 10.1109/18.61109

[14] CRUZ R L. A calculus for network delay, part II: network analysis [J]. IEEE transactions on information theory, 1991, 37(1): 132 – 141. DOI: 10.1109/18.61110

[15] PAREKH A K, GALLAGER R G. A generalized processor sharing approach to flow control in integrated services networks: the single-node case [J]. IEEE/ACM transactions on networking, 1993, 1(3): 344 – 357. DOI: 10.1109/90.234856

[16] PAREKH A K, GALLAGER R G. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case [J]. IEEE/ACM transactions on networking, 1994, 2(2):137 – 150. DOI: 10.1109/90.234856

[17] CRUZ R L. Quality of service guarantees in virtual circuit switched networks [J]. IEEE journal on selected areas in communications, 1995, 13(6): 1048 – 1056. DOI: 10.1109/49.400660

[18] SARIOWAN H, CRUZ R L, POLYZOS G C. Scheduling for quality of service guarantees via service curves [C]//Proc. Fourth International Conference on Computer Communications and Networks. IEEE, 1995: 512 – 520. DOI: 10.1109/ICCCN.1995.540168

[19] AGRAWAL R, RAJAN R. Performance bounds for guaranteed and adaptive services: IBM Technical Report RC 20649 [R]. 1996

[20] CRUZ R L. SCED: efficient management of quality of service guarantees [C]//Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE, 1998: 625 – 634. DOI: 10.1109/INFCOM.1998.665083

[21] LE BOUDEC J Y. Application of network calculus to guaranteed service networks [J]. IEEE transactions on information theory, 1998, 44(3): 1087 – 1096. DOI: 10.1109/18.669170

[22] CHANG C S. On deterministic traffic regulation and service guarantees: a systematic approach by filtering [J]. IEEE transactions on information theory, 1998, 44(3): 1097 – 1110. DOI: 10.1109/18.669173

[23] AGRAWAL R, CRUZ R L, OKINO C, et al. Performance bounds for flow

control protocols [J]. IEEE/ACM transactions on networking, 1999, 7(3): 310 – 323. DOI: 10.1109/90.779197

[24] FIDLER M. Survey of deterministic and stochastic service curve models in the network calculus [J]. IEEE communications surveys & tutorials, 2010, 12 (1): 59 – 86. DOI: 10.1109/SURV.2010.020110.00019

[25] LE BOUDEC J Y, THIRAN P. Network calculus: a theory of deterministic queuing systems for the Internet [M]. Berlin: Springer, 2001

[26] STILIADIS D, VARMA A. Latency-rate servers: a general model for analysis of traffic scheduling algorithms [J]. IEEE/ACM transactions on networking, 1998, 6(5): 611 – 624. DOI: 10.1109/90.731196

[27] JIANG Y M, LIU Y. Stochastic network calculus [M]. London: Springer, 2008

[28] BURCHARD A, LIEBEHERR J. A general per-flow service curve for GPS [C]//Proc. 30th International Teletraffic Congress (ITC 30. IEEE, 2018: 31 – 36

[29] NAGLE J. On packet switches with infinite storage [J]. IEEE transactions on communications, 1987, 35(4): 435 – 438. DOI: 10.1109/TCOM.1987.1096782

[30] KATEVENIS M, SIDIROPOULOS S, COURCOUBETIS C. Weighted round-robin cell multiplexing in a general-purpose ATM switch chip [J]. IEEE journal on selected areas in communications, 1991, 9(8): 1265 – 1279. DOI: 10.1109/49.105173

[31] SHREEDHAR M, VARGHESE G. Efficient fair queuing using deficit round-robin [J]. IEEE/ACM transactions on networking, 1996, 4(3): 375 – 385. DOI: 10.1109/90.502236

[32] BOUILLARD A, BOYER M, LE CORRONC E. Deterministic network calculus: from theory to practical implementation [M]. Hoboken: Wiley, 2018. DOI: 10.1002/9781119440284

[33] TABATABAEE S M, LE BOUDEC J Y, BOYER M. Interleaved weighted round-robin: a network calculus analysis [J]. IEICE Transactions on Communications, 2021, 104(12): 1479 – 1493. 10.1587/TRANSCOM.2021ITI0001

[34] KANHERE S S, SETHU H. On the latency bound of deficit round robin [C]//Proc. Eleventh International Conference on Computer Communications and Networks. IEEE, 2002: 548 – 553. DOI: 10.1109/ICCCN.2002.1043123.

[35] LENZINI L, MINGOZZI E, STEA G. Full exploitation of the deficit round Robin capabilities by efficient implementation and parameter tuning [R]. 2003

[36] BOYER M, STEA G, MANGOUA SOFACK W. Deficit round robin with network calculus [C]//Proc. 6th International Conference on Performance Evaluation Methodologies and Tools. IEEE, 2012: 138 – 147. DOI: 10.4108/valuetools.2012.250202

[37] BOUILLARD A. Individual service curves for bandwidth-sharing policies using network calculus [J]. IEEE networking letters, 2021, 3(2): 80 – 83. DOI: 10.1109/LNET.2021.3067766

[38] TABATABAEE S M, LE BOUDEC J Y. Deficit round-robin: a second network calculus analysis [J]. IEEE/ACM transactions on networking, 2022, 30 (5): 2216 – 2230. DOI: 10.1109/TNET.2022.3164772

[39] CONSTANTIN V C, NIKOLAUS P, SCHMITT J. Improving performance bounds for weighted round-robin schedulers under constrained cross-traffic [C]//Proc. IFIP Networking Conference (IFIP Networking). IEEE, 2022: 1 – 9

[40] IEEE. IEEE standard for local and metropolitan area networks: bridges and bridged networks: amendment 29: cyclic queuing and forwarding: 802.1Qch-2017 [S]. 2017

[41] YIN S W , WANG S , HUANG T . Analysis and optimization of queues based on network calculus in time-sensitive networking [J]. ZTE technology journal, 2022, 28(1): 21 – 28. DOI:10.12142/ZTETJ.202201007

[42] WANDELER E, THIELE L. Optimal TDMA time slot and cycle length allocation for hard real-time systems [C]//Proc. Asia and South Pacific Conference on Design Automation. IEEE, 2006: 479 – 484. DOI: 10.1109/ASPDAC.2006.1594731

[43] IEEE. IEEE standard for local and metropolitan area networks: bridges and bridged networks: amendment 25: enhancements for scheduled traffic: 802.1Qbv-2015 [S]. 2015

[44] IEEE. IEEE standard for local and metropolitan area networks: bridges and bridged networks: amendment 26: frame preemption: 802.1Qbu-2016 [S]. 2016

[45] ZHAO L X, POP P, CRACIUNAS S S. Worst-case latency analysis for IEEE 802.1Qbv time sensitive networks using network calculus [J]. IEEE access, 2018, 6: 41803 – 41815. DOI: 10.1109/ACCESS.2018.2858767

[46] IEEE. IEEE standard for local and metropolitan area networks: virtual bridged local area networks amendment 12: forwarding and queuing enhancements for time-sensitive streams: 802.1Qav [S]. 2009

[47] DE AZUA J A R, BOYER M. Complete modelling of AVB in Network Calculus Framework [C]//Proc. 22nd International Conference on Real-Time Networks and Systems. ACM, 2014: 55 – 64. DOI: 10.1145/2659787.2659810

[48] ZHAO L, HE F, LI E S, et al. Improving worst-case delay analysis for traffic of additional stream reservation class in ethernet-AVB network [J]. Sensors, 2018, 18(11): 3849. DOI: 10.3390/s18113849

[49] ZHAO L X, POP P, ZHENG Z, et al. Timing analysis of AVB traffic in TSN networks using network calculus [C]//Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS). IEEE, 2018: 25 – 36. DOI: 10.1109/RTAS.2018.00009

[50] ZHAO L X, POP P, ZHENG Z, et al. Latency analysis of multiple classes of AVB traffic in TSN with standard credit behavior using network calculus [J]. IEEE transactions on industrial electronics, 2020, 68(10): 10291 – 10302. DOI: 10.1109/TIE.2020.3021638

[51] IEEE. IEEE standard for local and metropolitan area networks-bridges and bridged networks amendment 34: asynchronous traffic shaping: 802.1 Qcr-2020 [S]. 2020

Biographies

GAO Yuehong (yhgao@bupt. edu. cn) received her PhD degree from Beijing University of Posts and Telecommunications (BUPT), China in 2010. She is an associate professor with the School of Information and Communication Engineering, BUPT. Her research interests include network calculus theory and application, quality of service guarantees in communication networks, simulation methodology, and digital twin networks.

NING Zhi received his BE degree in communication engineering from Beijing University of Posts and Telecommunications (BUPT), China in 2021. He is working towards his MS degree at BUPT. His research interests include wireless networks and deterministic networking.

HE Jia received his BE degree in communication engineering from Beijing University of Posts and Telecommunications (BUPT), China in 2022. He is working towards his MS degree in communication engineering at BUPT. His research interests include network calculus, 5G network architecture, and deterministic networking.

ZHOU Jinfei received his BE degree in communication engineering from Beijing University of Posts and Telecommunications (BUPT), China in 2023, where he is currently pursuing an MS degree. His research interests include wireless networks and deterministic networking.

GAO Chenqiang works at in the Data System Department of ZTE Corporation. His research interests include network calculus theory and application, DetNet, TSN, and SDN.

TANG Qingkun works at the Cable Software Platform Development Department of ZTE Corporation. His research interests include DetNet, TSN, and autonomous networks.

YU Jinghai works at the Data System Department of ZTE Corporation. He has more than 20 years of experience in the research and design of data network products including BIER, DetNet, TSN, Switch and Router, Data Center and SDN. He has won the 21st China Patent Silver Award and the first prize of the Science and Technology Award of the China Communications Society.

Deadlock Detection: Background, Techniques, and Future Improvements

LU Jiachen[1], NIU Zhi[2], CHEN Li[2], DONG Luming[2], SHEN Taoli[1]

(1. Zhejiang University, Hangzhou 310058, China;
2. ZTE Corporation, Xi'an 710114, China)

DOI: 10.12142/ZTECOM.202402009

https://kns.cnki.net/kcms/detail/34.1294.TN.20240604.1720.002.html, published online June 5, 2024

Manuscript received: 2023-12-19

Abstract: Deadlock detection is an essential aspect of concurrency control in parallel and distributed systems, as it ensures the efficient utilization of resources and prevents indefinite delays. This paper presents a comprehensive analysis of the various deadlock detection techniques, including static and dynamic approaches. We discuss the future improvements associated with deadlock detection and provide a comparative evaluation of these techniques in terms of their accuracy, complexity, and scalability. Furthermore, we outline potential future research directions to improve deadlock detection mechanisms and enhance system performance.

Keywords: deadlock detection; static analysis; dynamic analysis

Citation (Format 1): LU J C, NIU Z, CHEN L, et al. Deadlock detection: background, techniques, and future improvements [J]. ZTE Communications, 2024, 22(2): 71 – 79. DOI: 10.12142/ZTECOM.202402009

Citation (Format 2): J. C. Lu, Z. Niu, L. Chen, et al., "Deadlock detection: background, techniques, and future improvements," *ZTE Communications*, vol. 22, no. 2, pp. 71 – 79, Jun. 2024. doi: 10.12142/ZTECOM.202402009.

1 Introduction

Concurrency control is a critical aspect of parallel and distributed computing systems, as it ensures that multiple processes can access shared resources without conflicts or performance degradation. One of the major concerns in concurrency control is the occurrence of deadlocks, which can lead to indefinite delays and inefficient resource utilization. Deadlocks occur when a set of processes are blocked, each waiting for a resource held by another process in the set. This circular dependency prevents any of the processes from making progress, causing a significant impact on system performance.

A variety of deadlock detection techniques have been proposed in the literature, which can be broadly categorized into two categories. Static techniques analyze the system's source code, data structures, or control flow graphs to detect potential deadlocks without running the program. Dynamic techniques, on the other hand, monitor the system's execution at runtime to detect deadlock occurrences.

Moreover, each tool adopts a different set of strategies, with technical details not always fully documented or publicized. These factors have resulted in a knowledge gap that hinders users of these tools, particularly researchers conducting deadlock analysis. To narrow this gap, several questions must be addressed:

- Q1: How do current static deadlock detection techniques improve efficiency and reduce false positives?
- Q2: How do current dynamic deadlock detection techniques improve efficiency and reduce false positives?
- Q3: What is the future direction of development for deadlock detection techniques?

To answer these questions, we present a comprehensive analysis of existing deadlock detection techniques, through the study of five popular deadlock tools shown in Table 1. As the vast majority of deadlock detection tools are not open source, we can only discuss these techniques qualitatively and answer Q1 and Q2 accordingly. After the above discussion, we answer Q3 by providing an outlook and summary for the future development of deadlock detection.

By systematically dissecting and evaluating the tools, we can make new observations that amend or complement prior knowledge. Our major observations are as follows.

- Static deadlock detection typically improves efficiency

▼Table 1. Groups of deadlock detectors that our study covers

Type	Tools	Release Date
Static	D4[1]	Jun. 2018
	Peahen[2]	Nov. 2022
Dynamic	GoodLock[3]	Nov. 2005
	MagicLock[4–5]	Mar. 2014
	Sherlock[6]	Aug. 2014

through parallelization, pre-screening, and other methods.

- Dynamic deadlock detection typically improves efficiency by merging or deleting states in the lock graph.
- Despite dynamic and static methods generating lock traces using completely different approaches, there are commonalities in the subsequent deadlock detection process.

The remainder of this paper is structured as follows. Section 2 provides a brief background on deadlocks and deadlock detection. Sections 3 and 4 present a comprehensive analysis of current static and dynamic deadlock detection techniques, using a selection of tools as examples. Section 5 summarizes and compares the aforementioned tools. The future direction of development for deadlock detection techniques is discussed in Section 6. Section 7 concludes the paper.

2 Background: Deadlock and Defense

2.1 Deadlock

A deadlock is a state in which each member of a group is waiting for another member, including itself, to take action[7]. The occurrence of a deadlock requires the satisfaction of the following four necessary conditions[8].

- Mutual-exclusion: Each lock object can only be owned by one thread.
- Wait-for: A thread does not release the lock object it has acquired while waiting to acquire another lock object.
- No-preemption: A thread cannot seize the lock object of another thread.
- Circular-wait: Each thread holds one or more lock objects while simultaneously requesting lock objects that other threads have already acquired.

More specifically, for the abstract model of non-reentrant locks L_1 and L_2, and threads T_1 and T_2, the following two deadlock situations exist: When T_1 holds L_1 and is waiting for T_1 to release L_1 (e.g. T_1 locks L_1 twice). When T_1 is waiting for T_2 to release L_1, T_2 is also waiting for T_1 to release L_1.

In order to deal with deadlocks in concurrent systems, there are generally three preventive measures[9]:

- Deadlock prevention: breaking one of the four conditions mentioned earlier to prevent the occurrence of a deadlock;
- Deadlock avoidance: dynamically detecting the possibility of a deadlock and taking appropriate measures to avoid it;
- Deadlock detection: detecting the existence of a deadlock and taking appropriate measures to recover from it.

2.2 Deadlock Prevention

The objective of deadlock prevention is to ensure that deadlocks do not happen by breaking the necessary conditions for system deadlocks. As recommended by HAVENDER[10], the following approaches effectively negate each of the four remaining conditions in turn.

1) Mutual-exclusion condition denied. It allows multiple processes to access the same resource at the same time. This strategy can be applied to read-only data files[11], disks, and other software and hardware resources, but it is not always feasible because some resources may be inherently non-shareable.

2) Wait-for condition denied. All resources are allocated through a static approach. This means that a process must request all the necessary resources before execution and will not begin execution until all the required resources have been obtained. This approach is simple to implement but significantly reduces both resource utilization and language expressiveness. This is because the static allocation of resources cannot support runtime features such as recursion and polymorphism[12].

3) No-preemption condition denied. When a process requests a resource that is currently unavailable, it will be blocked until the resource is available. If the process cannot acquire the resource after a certain period of time, it will release all resources currently held and restart the request process. This approach is not always feasible as certain resources may inherently be non-preemptible. Currently, this strategy is only employed for the allocation of memory and processor resources.

4) Circular-wait condition denied. It assigns a unique number to each resource and requires processes to request resources in ascending order. This approach is more effective and widely used, such as Android[13], compared with the previous deadlock prevention methods.

2.3 Deadlock Avoidance

Deadlock avoidance takes a more proactive approach than deadlock prevention, striving to recognize and avoid potential deadlocks before they occur. This is typically achieved through careful resource allocation and monitoring of the system state.

The most well-known deadlock avoidance algorithm is the Banker' s Algorithm[14], which requires processes to declare their maximum resource needs upfront. The algorithm then allocates resources in a manner that guarantees a safe state, ensuring that no deadlock can occur. However, this approach requires accurate resource estimates and may become computationally complex for large-scale systems.

Deadlock avoidance is generally considered more favorable than prevention in database systems, as these systems already can abort transactions. Although avoidance may result in the unnecessary aborting of transactions, it is still preferred over prevention.

2.4 Deadlock Detection

Deadlock detection is the process of identifying deadlocks in a computing system, either before or during execution. Compared with deadlock prevention and avoidance, deadlock detection minimizes the need for human intervention in the program and avoids affecting the program' s performance and sharing capabilities. Due to its high practicality and versatility, deadlock detection technology has been widely researched in recent years, and significant progress has been made. Therefore, this article focuses on researching and summarizing deadlock detection technology.

Deadlock detection techniques can be classified into two categories based on whether the program needs to be executed: static deadlock detection discussed in Section 3 and dynamic deadlock detection discussed in Section 4.

3 Static Deadlock Detection

Static deadlock detection is a technique that can identify deadlocks without executing the program[15]. This technique involves analyzing the source code or program structure to identify potential deadlocks. In static analysis, detectors track the acquired lock objects and those being requested. When circular dependencies between locks result in a deadlock, a bug is detected. Techniques such as model checking[16], dataflow analysis[17], and control flow analysis[18] can be employed to detect deadlock-prone situations. Previous research[1 – 2, 19 – 20] used static analysis to detect deadlocks from source codes. While some promising results have been achieved as described in the following text, it is still a long way to achieve a complete solution to deadlock bugs. For instance, static analysis cannot account for dynamic program behavior and static detectors often produce many false positives[21].

In the remainder of this section, we discuss two representative static deadlock detectors to illustrate the recent development direction of static deadlock detection.

3.1 D4

D4 is a fast concurrency analysis framework based on concurrent and incremental pointer analysis. By redesigning the pointer analysis, correct conclusions can be obtained by only re-analyzing the incremental code, which avoids redundant computation in traditional whole-program analysis.

The pointer assignment graph (PAG) is a data structure used in pointer analysis algorithms to represent the assignments and relationships between pointers and objects in a program. Each program variable corresponds to a node within the PAG, and variable assignments are reflected through the creation of one or more edges. The PAG consists of two distinct node types: pointer nodes, representing pointer or reference variables, and object nodes, representing memory locations or objects. Each pointer node is associated with a points-to set denoted by pts, which contains the set of object nodes that the pointer may point to. Each edge represents a subset constraint between the points-to sets, i.e., $p \rightarrow q$ means $\text{pts}(p) \subseteq \text{pts}(q)$. We consider Fig. 1 as an illustrative example of a PAG, where p and q denote pointer nodes, and o_1 and o_2 represent object nodes. In this case, $\text{pts}(p)$ consists of $\{o_1\}$, while $\text{pts}(q)$ encompasses $\{o_1, o_2\}$.

The new parallel incremental pointer analysis is mainly based on the following properties of the acyclic PAG.

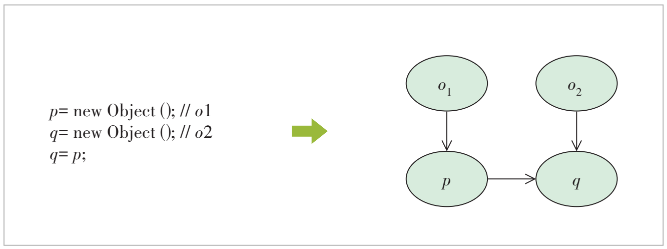
1) Deleting edges property. As shown in Fig. 2, for an object node o and two pointer nodes p and q in a PAG, if q has an incoming neighbor p (i.e., there exists an edge $p \rightarrow q$) and $o \in \text{pts}(p)$, o can reach p without going through q.

Based on the properties mentioned above, supposing an edge $q \rightarrow p$ is deleted from PAG and other edges remain unchanged, we only need to check the incoming neighbors of p (i.e., the deleted edge ' s destination), which is much faster than traversing the whole PAG for checking the path reachability.

2) Propagating changes property. To propagate a change to a node, it is sufficient to check the other incoming neighbors of the node. If the points-to set of any incoming neighbor contains the change, the node can be skipped. Otherwise, the change should be applied to the node and propagated further to all its outgoing neighbors.

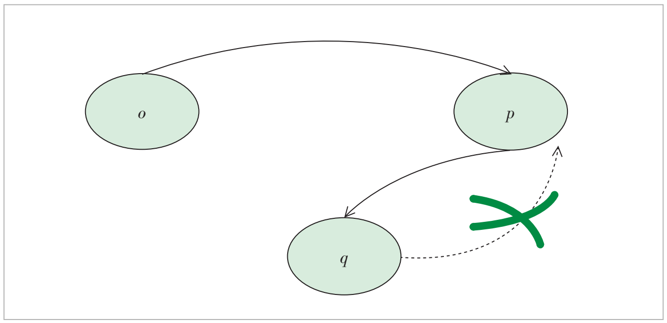
As shown in Fig. 3, for two object nodes o_1 and o_2, and four pointer nodes x, y, z, and w in a PAG, supposing that an edge $q \rightarrow p$ is deleted from PAG and other edges remain unchanged, we only need to check z and w, which are the outgoing neighbors of y.

The two theorems above ensure that when a statement is de-

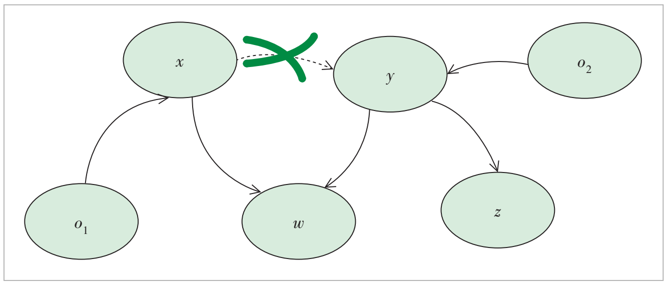


```
p= new Object (); // o1
q= new Object (); // o2
q= p;
```

▲**Figure 1. An example of pointer assignment graph (PAG)**



▲**Figure 2. Incoming neighbors property**



▲**Figure 3. Outgoing neighbors property** leted, it is only necessary to check the local neighbors of the affected nodes in the PAG to determine the changes in points-to sets and perform change propagation. This greatly reduces the amount of computation required for recomputing points-to sets or traversing the entire graph.

3.2 Peahen

Peahen[2] explores a context-reduction approach for fast and precise deadlock detection in real-world programs. Traditional static deadlock detection techniques first construct a context-sensitive lock graph based on lockset analysis, and then analyze the lock graph to discover precise deadlock cycles. However, it has been observed that large-scale context-sensitive lock calling contexts can cause state space explosion and make it difficult to eliminate false positive deadlock cycles. To address that problem, Peahen splits the static deadlock detection technique into two stages: the context-insensitive lock graph construction and three lazy deadlock cycles refinements.

1) Context-insensitive lock graph construction. Refs. [22 – 24] build context-sensitive lock graphs including a large scale of unnecessarily acquired edges. A context-insensitive lock graph with selected acquired edges cloning can simplify deadlock analysis. Peahen presents an inter-procedural algorithm that constructs a context-insensitive lock graph without requiring any context analysis. The algorithm then proceeds to clone selected multi-thread edges. For example, Fig. 4 shows a program using nested locks. Thread 1 and thread 2 both run functions foo() and bar(). If thread 1 is running at line 09 waiting for lock o_1 and thread 2 is running at line 16 waiting for lock o_2, a deadlock problem will occur. Fig. 5 is its lock graph. Peahen first adds edges in every function and then builds an intra-procedural lock graph using the bottom-up dataflow analysis. If an acquired edge represents different threads' lock dependencies, it must be cloned and distinguished as different thread IDs.

2) Deadlock cycle refinements. To identify deadlock cycles precisely and efficiently, Peahen performs the three following steps to refine lock graph cycles lazily.

```
T1:                          T2:
01: void thread1(){          11: void thread2(){
02:    fork(t2, thread2);    12:    foo();
03:    foo();                13: }
04:    join(t2);             14: void foo(){
05: }                        15:    lock(v1);
06: void bar(){              16:    lock(v2); // o2
07:    unlock(v1); // o1     17:    bar();
08:    x++;                  18:    unlock(v2);
09:    lock(v1);             19:    unlock(v1);
10: }                        20: }
```

▲Figure 4. Code using non-nested locks

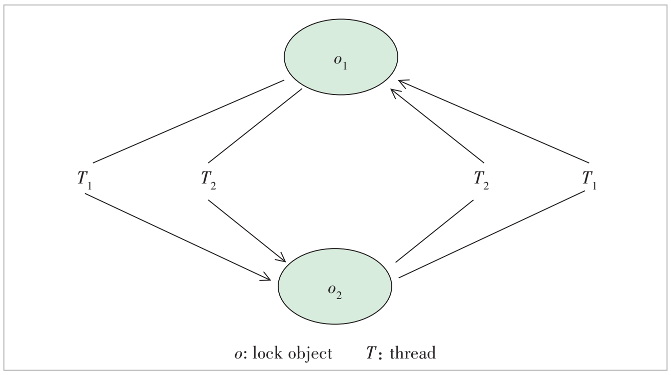


Figure 5. Context-insensitive lock graph

- Single- and multi-threaded cycle computation. Peahen divides lock cycles into single-threaded cycles and multi-threaded cycles. Single-threaded cycles indicate that a lock node owns an edge pointing towards itself. Multi-threaded cycles are defined in that every cycle edge is owned by different threads.
- Concurrent cycle computation. At this step, Peahen tries to refine multi-threaded cycles to concurrent cycles. Concurrent cycles must rule out two cases: A thread in the multi-threaded cycle has been destroyed, or the multi-threaded cycle has been guarded by a lock.
- Path-feasible cycle computation. Finally, Peahen performs path feasibility analysis on the concurrent cycles using the satisfiability modulo theory (SMT) solver. Peahen is the first one to introduce path feasibility analysis into deadlock cycle refinements.

4 Dynamic Deadlock Detection

Dynamic deadlock detection is a technique that detects deadlocks in a multi-threaded program based on the execution trace[3]. Most of the dynamic deadlock detection approaches map an execution trace to data structures such as lock-order graphs, and then the running detection algorithms to identify deadlocks. In recent years, dynamic deadlock detection has been widely used in the field of software testing (e.g., GoodLock[3, 25], DeadlockFuzzer[26], MagicLock[4 – 5], and Sherlock[6]). Compared with static deadlock detection, dynamic deadlock detection can better obtain happens-before relationships and other runtime information in a program. However, these approaches are limited in their ability to detect deadlocks, such as the failure to cover all possible program states and the possibility of false positives.

In the remainder of this section, we discuss three representative dynamic deadlock detectors to illustrate the recent development direction of dynamic deadlock detection.

4.1 GoodLock

GoodLock[3] is a dynamic deadlock detection algorithm analyzing a trace generated from the execution of the program. It consists of two main components: trace generation and detection. First, the program under test is instrumented to record synchronization events when executed. The detection algorithm analyzes the execution trace and constructs a lock graph that identifies potential deadlocks through the presence of cycles. Although GoodLock is not sound nor complete, it is an improvement on the basic lock graph algorithm[27], which reduces false positives in the presence of gate locks (a common lock taken first by involved threads). The main strategy for reducing false positives is described as follows.

- Extended lock graph. Traditional lock graphs can only represent partial information about the ordering of lock acquisitions by threads. For example, in an abstract model of thread T , locks L_1 and L_2, if there exists a state where thread T holds L_1 and acquires L_2 during execution, then a directed edge from L_1 to L_2 is added to the lock graph, written as $L_1 \rightarrow L_2$. Therefore, a cycle can be created in the lock graph through any cyclic acquisition of locks, even if the acquisition cannot happen parallelly, leading to false positives. To address this problem, GoodLock[3] introduced the concept of an extended lock graph. The extended lock graph is an extension of the traditional lock graph that includes more information about which thread causes the addition of the edge and which gate locks are held by that thread when the target lock is taken. Based on this extended information, false positives caused by single-threaded and guarded cycles can be eliminated during the detection phase.

- Segments. As the example in Fig. 6, the algorithm on the extended lock graph reports a cycle between threads T_1 (lines 05 – 06) and T_2 (lines 08 – 09) on locks L_1 and L_2. However, a deadlock is impossible since thread T_2 is joined on by the main thread in line 03. Therefore, the two code segments, lines 05 – 06 and lines 08 – 09, can never run in parallel. The algorithm to be presented will prevent such cycles from being reported by formally introducing such a notion of segments that cannot execute in parallel. A new directed segmentation graph will record which segments execute before others. The lock graph is then extended with extra-label information, which specifies what segment locks are acquired in, and the validity of a cycle now incorporates a check that the lock acquisitions occur in parallel executing segments. Based on this extended information, false positives caused by segmented cycles can be eliminated during the detection phase.

Apart from the strategies mentioned above, various optimization strategies have also been added to GoodLock's variants, as described in later sections.

4.2 MagicLock

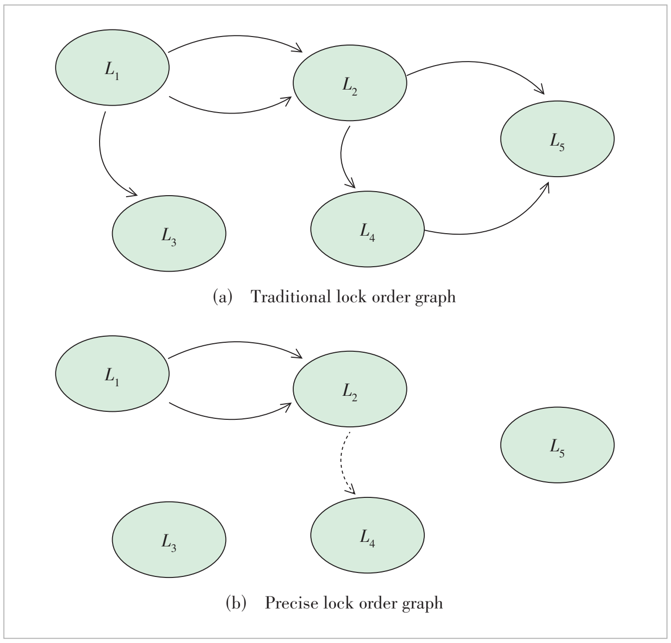
MagicLock[4 – 5] is a more efficient variant of GoodLock[3]. The two tools share a similar technological approach to deadlock detection, both consisting of two phases: trace generation and detection. In fact, directly checking on the lock order graph of GoodLock[3] for a large-scale program is impractical due to the huge cost of time. For example, in the ITCAM application, The authors in Ref. [28] reported a lock order graph with over 300 000 nodes and 600 000 edges. The nodes in the graph represent the procedures in the program, and the edges represent the call relationships between the procedures. GoodLock spent 48 h and 13.6 GByte memory to traverse it to find cycles if they exist[28]. So MagicLock[4] proposed some strategies to reduce the size of the lock order graph in the trace generation phase and the time spent on detecting deadlocks. The main strategies are described as follows.

```
Main:
01: t1 = new T1();
02: t1.start();
03: t1.join();
04: new T2().start();

T1:
05: synchronized(L1) {
06: synchronized(L2) {}
07: }

T2:
08: synchronized(L2) {
09: synchronized(L1) {}
10: }
```

Figur 6. Example program of false positives

- Graph pruning. Previous work has proposed several strategies for simplifying states, such as merging the state of locks[28]. Although merging locks can reduce the search space, all locks that cannot lead to deadlocks are still retained in the lock graph, leading to redundant traversal. MagicLock[4] iteratively removes the lockset and their edges, resulting in a more precise lock order graph. The strategy of iteratively removing edges is based on the following observation: for a node participating in a potential deadlock cycle, the node must have both incoming and outgoing edges. Therefore, during each iteration, the edges of nodes that possess solely incoming and outgoing edges will be eliminated. As the traditional lock order graph example in Fig. 7(a), the algorithm's first iteration will remove the edges pointing to L_3 and L_5, and the second iteration will remove the edge pointing to L_4, as indicated by the dotted line in Fig. 7(b). Based on this additional information, the states that would not cause potential deadlocks are deleted, which improves the efficiency of the algorithm.

- Thread-specific lock dependency. MagicLock uses a thread-specific lock dependency relation denoted by thread-specific triple $D_i = \langle t, m, Lt \rangle$ for each thread. Here, t represents the thread number, m denotes the lock that is being acquired by the current statement, and Lt represents the set of locks that are currently held by the thread t. This triple captures the dependencies between the locks that a thread holds and the locks that it attempts to acquire. Based on the afore-


L_1
L_2
L_3
L_4
L_5
(a) Traditional lock order graph
L_1
L_2
L_3
L_4
L_5
(b) Precise lock order graph

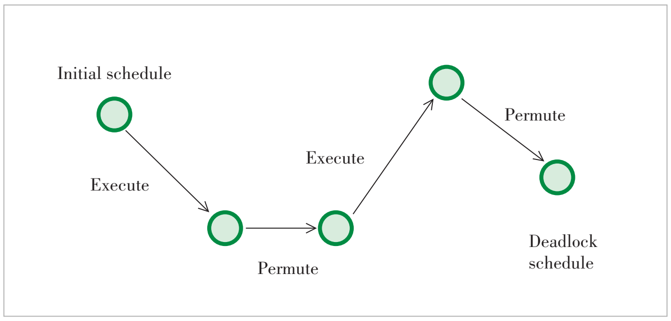
▲Figure 7. Lock order graph example

mentioned strategy, MagicLock employs a new depth-first-search algorithm to traverse D_i for each thread t_i. This approach differs from the iGoodLock algorithm in Deadlock-Fuzzer[26]. As iGoodLock employs transitive closure for iterative cycle detection, a noticeable limitation is that iGoodLock has to store all intermediate results, which consumes a lot of memory[26].

4.3 Sherlock

Sherlock[6] is a more effective variant of GoodLock[3]. Leveraging the power of concolic execution, Sherlock exhibits remarkable proficiency in identifying deadlocks after an extensive computation of one million steps, a feat beyond the reach of conventional technologies, which other existing technologies cannot discover. Sherlock also consists of two phases: producing deadlock candidates with GoodLock and concolic execution to drive an execution toward a deadlock candidate. As the algorithm for generating candidate deadlocks in GoodLock has already been introduced earlier, this section mainly focuses on the concolic execution in Sherlock[6].

Sherlock first produces a set of deadlock candidates using GoodLock. Then, it uses concolic execution to search for each of the deadlock candidates. The key idea is to turn each search for a deadlock into a search for an event sequence (schedule) that leads to the deadlock. As a deadlock search example in Fig. 8, for each schedule that leads to a deadlock, Sherlock alternates between the “execute” and “permute” steps. The “execute function” attempts to execute a given schedule and determine whether it leads to a deadlock. The “permute function” permutes a given schedule. The search begins with an initial schedule found simply by “InitialRun function”. The search fails if “execute” cannot execute a given schedule, “permute” cannot find a better permutation, or the search times out. In the following paragraphs, we will briefly discuss each of these phases: “InitialRun,” “Execute,” and “Permute”.



▲Figure 8. Sherlock deadlock search

- “InitialRun function”. The InitialRun function executes the program with some particular input and records the schedule. For each program, Sherlock uses the predetermined inputs to execute the program because those inputs are useful enough.
- “Execute function”. The arguments of “execute function” are a program, a schedule, and a deadlock candidate. The “execute function” will attempt to execute the given schedule, determine whether it leads to a deadlock, and return the input to the program that is used to execute the schedule. The implementation of the “execute function” uses concolic execution[29 - 32] collecting operational information (e.g. assignments and conditions), and generate inputs that are more likely to reach the specified state for the next running.
- “Permute function”. The arguments of “permute function” are a schedule and a deadlock candidate. The “permute function” will attempt to better the given schedule and primarily improves SAID et al.’s “permute function”[33]. The “permute function” in Sherlock encapsulates the encoding of lock-order graphs and alias information into constraints. These constraints are subsequently solved using an SMT solver, while satisfying conditions such as happens-before relationships. Finally, the feasible solutions are traversed to discover an optimal schedule.

5 Comparative Evaluation

Many existing deadlock detection methods lack open-source code, and a significant portion of open-source projects have low usability due to poor maintenance. Therefore, we provide a qualitative evaluation of these techniques. We evaluate these tools from two aspects: scalability and effectiveness. Scalability is the property of a system to handle a growing amount of work. One definition for software systems specifies that this may be done by adding resources to the system. The effectiveness is measured from three aspects: false positives, false negatives, and detection of new vulnerabilities, as shown in Table 2. Finally, we summarize our observations.

5.1 Static Deadlock Detection

- Scalability. Previous work[19] focused on whole-program analysis and it was difficult to efficiently detect deadlocks. D4 reduces redundant calculations during the analysis process through incremental analysis and further accelerates deadlock detection through parallelization. Peahen reduces the overhead of subsequent deadlock detection stages through a fast pre-processing phase.
- Effectiveness. D4[1] and Peahen[2] are both unsound and incomplete. Peahen is almost sound, aside from a few well-identified reasonable unsound choices for achieving higher precision. There are two sources of unsoundness in our implementations. First, the pointer analysis shares the same unsound sources as Peahen uses. For instance, it does not correctly handle pointer arithmetic, array accesses, containers, etc. Second, the lock graph construction ignores the locks that are inside blocks of the assembly code as the prior deadlock detectors. The unsoundness of D4 also mainly comes from two aspects. Firstly, the imprecision of pointer analysis that results in D4 cannot handle the situation where a lock variable may point to multiple objects. Secondly, D4 ignores reflection and library functions during the analysis process, leading to false negatives. Since Peahen has discovered new deadlock issues while D4 only demonstrates the efficiency of a new algorithm on the Dacapo benchmark[34] without finding new deadlock issues, we consider Peahen more effective than D4.

In summary, traditional static program analysis has become increasingly difficult to complete within an acceptable time frame for most existing programs. Therefore, much of the current research focuses on improving deadlock detection efficiency through parallelization, pre-screening, and other methods.

5.2 Dynamic Deadlock Detection

- Scalability. Traditional dynamic deadlock detection has only limited scalability. To relax the dynamic deadlock detection overhead, many seminal approaches have been proposed. The MulticoreSDK[27] firstly groups the locks being held by different threads at the same code location in the same group and then merges multiple groups into the same group whenever they have at least one shared lock to reduce the size of the lock order graph. The MagicLock[5] employs an iterative approach to eliminating locks that cannot cause deadlocks from the lock order graph. The latest one, AirLock[35], speeds up the online cycle discovery by first finding "simple cycles" without considering any execution information (e.g., threads) and then constructing deadlock cycles by taking full execution information into account.

▼Table 2. Evaluation results across all vulnerability discovery techniques

Type	Tools	Scalability	False Positives	False Negatives	New Bugs
Static	D4	High	True	True	False
	Peahen	High	True	Almost false	True
Dynamic	GoodLock	Low	True	True	False
	MagicLock	Medium	True	True	False
	Sherlock	Medium	True	True	True

- Effectiveness. Most dynamic deadlock detection techniques are also unsound and incomplete. Unlike typical fuzzing, dynamic deadlock detection usually only relies on lock-traces generated during runtime since it is difficult to produce inputs that can reach the deadlock state and verify the actual occurrence of deadlocks. An incomplete dynamic deadlock may induce false positives due to, for instance, ignoring happens-before relations. Thus, Sherlock[6] is integrated with other techniques via scheduling a real deadlock and identifying and solving execution constraints. DeadlockFuzzer[26] uses fuzzing to confirm whether the cycle of locks is a real deadlock. Similar to static deadlock detection, dynamic deadlock detection techniques like MagicLock[5] and AirLock[29] only demonstrate their speed and efficiency through evaluation. Only Sherlock discovers new deadlock problems through concolic execution. Therefore, we consider Sherlock more effective than MagicLock and AirLock.

In summary, traditional dynamic program analysis has also become increasingly difficult to complete within an acceptable time frame for most existing programs. Therefore, much of the current research focuses on improving deadlock detection efficiency by merging or eliminating states in the lock graph.

6 Future Works

Although significant progress has been made in both static and dynamic deadlock detection over the past years to address this security challenge, there are still many open and unsolved issues.

- Scalability. Due to the rapid expansion of software codes, existing deadlock detection tools still struggle to handle projects at the level of millions of lines of code, such as the Linux kernel[36] and Firefox[37], within an acceptable range. Therefore, future works should still focus on improving the scalability of deadlock detection tools.
- Recall. The false negatives of reports are the most serious problem of deadlock detection tools. The false negatives of reports can cause serious problems, because our ultimate goal is to eliminate deadlocks. Existing dynamic deadlock detection tools suffer from a significant number of false negatives due to their reliance on program execution traces, which makes it challenging to achieve high coverage. While static deadlock detection tools have fewer false negatives compared with their dynamic counterparts, they still have their own limitations in terms of false negatives. Therefore, enhancing the coverage of both static and dynamic deadlock detection tools remains an urgent and unresolved issue.
- Precision. The false positives of reports confuse developers and waste their time. Currently, besides DeadlockFuzzer[26] and other tools that report only the deadlocks confirmed by

fuzzing to avoid false positives, all other deadlock detection tools suffer from false positives. However, reporting only the deadlocks confirmed by fuzzing can lead to a large number of false negatives, which is not a sweet spot. Therefore, future work should focus on improving the accuracy of deadlock detection tools.

- Communication deadlocks. Communication deadlock[38] is another kind of deadlock. Traditional deadlock detection only models locks and focuses on whether there is a circular "wait-for" in acquiring the locks. However, communication deadlock occurs when one or more threads are waiting for certain messages/signals from other threads, which are suspended and unable to send the required messages/signals or have already sent the messages/signals before a waiting thread starts to wait for the messages/signals. Due to the diverse and complex characteristics of communication deadlocks, compared with resource deadlocks analysis, there are few achievements in static and dynamic detection of communication deadlocks. Program analysis for resource deadlocks is still in its early stages.

7 Conclusions

In this paper, we present a comprehensive study of deadlock detection techniques from two perspectives: static and dynamic. Our research summarizes the different strategies of existing deadlock detection works and qualitatively compares their differences. Throughout the study, we derive a group of new observations that can complement previous understandings and also inspire future directions of deadlock detection.

Acknowledgement:

We would like to express our heartfelt gratitude to Professor ZHAO Yongwang for his invaluable guidance, support, and encouragement throughout his research.

References

[1] LIU B Z, HUANG J. D4: fast concurrency debugging with parallel differential analysis [J]. ACM SIGPLAN notices, 2018, 53(4): 359 – 373. DOI: 10.1145/3296979.3192390

[2] CAI Y D, YE C F, SHI Q K, et al. Peahen: fast and precise static deadlock detection via context reduction [C]//The 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. ACM, 2022: 784 – 796. DOI: 10.1145/3540250.3549110

[3] BENSALEM S, HAVELUND K. Dynamic deadlock analysis of multi-threaded programs [EB/OL]. [2023-06-01]. https://www.havelund.com/Publications/padtad05.pdf

[4] CAI Y, CHAN W K. MagicFuzzer: scalable deadlock detection for large-scale applications [C]//Proceedings of 34th International Conference on Software Engineering (ICSE). IEEE, 2012: 606 – 616. DOI: 10.1109/ICSE.2012.6227156

[5] CAI Y, CHAN W K. Magiclock: scalable detection of potential deadlocks in large-scale multithreaded programs [J]. IEEE transactions on software engineering, 2014, 40(3): 266 – 281. DOI: 10.1109/TSE.2014.2301725

[6] ESLAMIMEHR M, PALSBERG J. Sherlock: scalable deadlock detection for concurrent programs [C]//The 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering. ACM, 2014: 353 – 365. DOI: 10.1145/2635868.2635918

[7] ARPACI-DUSSEAU R H, ARPACI-DUSSEAU A C. Operating systems: three easy pieces [M]. Raleigh, USA: Lulu Press, 2018

[8] COFFMAN E G, ELPHICK M, SHOSHANI A. System deadlocks [J]. ACM computing surveys, 1971, 3(2): 67 – 78. DOI: 10.1145/356586.356588

[9] ELMAGARMID A K. A survey of distributed deadlock detection algorithms [J]. ACM SIGMOD record, 1986, 15(3): 37 – 45. DOI: 10.1145/15833.15837

[10] HAVENDER J W. Avoiding deadlock in multitasking systems [J]. IBM systems journal, 1968, 7(2): 74 – 84. DOI: 10.1147/sj.72.0074

[11] Alvinashcraft. CreateFileW function (fileapi. h) [EB/OL]. [2023-05-17]. https://learn. microsoft. com/en-us/windows/win32/api/fileapi/nf-fileapi-createfilew#parameters

[12] BERLIZOV A N, ZHMUDSKY A A. The recursive adaptive quadrature in MS Fortran-77 [EB/OL]. [2023-05-17]. https://arxiv. org/abs/physics/9905035

[13] Google. mm/rmap. c-kernel/common-Git at Google-android. googlesource. com [EB/OL]. [2023-05-17]. https://android. googlesource. com/kernel/common/+/refs/heads/android13-5.15/mm/rmap.c

[14] IJKSTRA E W. Een algorithme ter voorkoming van de dodelijke omarming [EB/OL]. [2023-05-17]. http://www. cs. utexas. edu/users/EWD/ewd01xx/EWD108.PDF

[15] AYEWAH N, PUGH W, HOVEMEYER D, et al. Using static analysis to find bugs [J]. IEEE software, 2008, 25(5): 22 – 29. DOI: 10.1109/ms.2008.130

[16] CLARKE E M. Model checking [C]//The 17th International Conference on Foundations of Software Technology and Theoretical Computer Science. FSTTCS, 1997: 54 – 56

[17] KHEDKER U, SANYAL A, SATHE B. Data flow analysis: theory and practice [M]. Carrollton, UAS: CRC Press, 2017

[18] ALLEN F E. Control flow analysis [J]. ACM SIGPLAN notices, 1970, 5(7): 1 – 19. DOI: 10.1145/390013.808479

[19] NAIK M, PARK C S, SEN K, et al. Effective static deadlock detection [C]//The 31st International Conference on Software Engineering. IEEE, 2009: 386 – 396. DOI: 10.1109/ICSE.2009.5070538

[20] BROTHERSTON J, BRUNET P, GOROGIANNIS N, et al. A compositional deadlock detector for android java [C]//The 36th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2021: 955 – 966

[21] DELIGIANNIS P, DONALDSON A F, RAKAMARIC Z. Fast and precise symbolic analysis of concurrency bugs in device drivers (T) [C]//The 30th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2015: 166 – 177. DOI: 10.1109/ASE.2015.30

[22] WILLIAMS A, THIES B, ERNST M D. Static deadlock detection for java libraries [EB/OL]. [2023-05-17]. https://people.csail.mit.edu/amy/papers/deadlock-ecoop05.pdf

[23] KAHLON V, YANG Y, SANKARANARAYANAN S, et al. Fast and accurate static data-race detection for concurrent programs [C]//International Conference on Computer Aided Verification. CAV: 226 – 239.10.1007/978-3-540-73368-3_26

[24] KROENING D, POETZL D, SCHRAMMEL P, et al. Sound static deadlock analysis for C/Pthreads [C]//Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering. ACM, 2016: 379 – 390. DOI: 10.1145/2970276.2970309

[25] HAVELUND K. Using runtime analysis to guide model checking of java programs [C]//The 7th International SPIN Workshop on Model Checking of Software. SPIN, 2000: 245 – 264

[26] JOSHI P, PARK C S, SEN K, et al. A randomized dynamic program analysis technique for detecting real deadlocks [J]. ACM SIGPLAN notices, 2009, 44(6): 110 – 120. DOI: 10.1145/1543135.1542489

[27] HARROW J J. Runtime checking of multithreaded applications with visual threads [C]//The 7th International SPIN Workshop in Model Checking and Software Verification. SPIN, 2000: 331 – 342

[28] LUO Z D, DAS R, QI Y. Multicore SDK: a practical and efficient deadlock detector for real-world applications [C]//The 4th IEEE International Conference on Software Testing, Verification and Validation. IEEE, 2011: 309 – 318. DOI: 10.1109/ICST.2011.22

[29] MAJUMDAR R, XU R G. Directed test generation using symbolic grammars [C]//The 22nd IEEE/ACM international conference on Automated software engineering. IEEE, 2007: 134 – 143

[30] GODEFROID P, KLARLUND P, SEN K. Dart: directed automated random testing [C]//The ACM SIGPLAN conference on Programming language design and implementation. ACM, 2005: 213 – 223

[31] SEN K, AGHA G. CUTE and jCUTE: concolic unit testing and explicit path model-checking tools [C]//International Conference on Computer Aided Verification. CAV, 2006: 419 – 423. DOI: 10.1007/11817963_38

[32] SEN K. Concolic testing [C]//The 22nd IEEE/ACM international conference on automated software engineering. IEEE, 2007: 571 – 572

[33] SAID M, WANG C, YANG Z, et al. Generating data race witnesses by an SMT-based analysis [C]//The Third International Conference on NASA Formal Methods. NASA Formal Methods Symposium, 2011: 313 – 327

[34] BLACKBURN S M, GARNER R, HOFFMANN C, et al. The DaCapo benchmarks: java benchmarking development and analysis [C]//The 21st Annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications. ACM, 2006: 169 – 190. DOI: 10.1145/1167473.1167488

[35] CAI Y, MENG R, PALSBERG J. Low-overhead deadlock prediction [C]//The 42nd International Conference on Software Engineering. IEEE, 2020: 1298 – 1309

[36] Bugzilla. Bugzilla main page: bugzilla.kernel.org [EB/OL]. [2023-06-01]. https://bugzilla.kernel.org/

[37] Bugzilla. Bugzilla main page: bugzilla.mozilla.org [EB/OL]. [2023-06-01]. https://bugzilla.mozilla.org/home

[38] JOSHI P, NAIK M, SEN K, et al. An effective dynamic analysis for detecting generalized deadlocks [C]//The 18th ACM SIGSOFT international symposium on Foundations of software engineering. ACM, 2010: 327 – 336. DOI: 10.1145/1882291.1882339

Biographies

LU Jiachen (lujc@zju.edu.cn) is a master student in cybersecurity at Zhejiang University, China. His research interests include program analysis and formal methods.

NIU Zhi received his master degree in control engineering from Chongqing University, China. He is currently working at ZTE Corporation. His research interests include distributed system, formal verification and software reliability.

CHEN Li received his bachelor degree in computer science from Northeastern University, China. He is currently working at ZTE Corporation. His research interests include software reliability, open source software risk analysis and network security.

DONG Luming received his master degree in control theory and control engineering from Huazhong University of Science and Technology, China. He is currently working at ZTE Corporation. His research interests include distributed system, formal verification, software reliability and innovative security technology for wireless communication.

SHEN Taoli is a master student in cybersecurity at Zhejiang University, China. His research interests include push-button verification and formal methods.

A Distributed Acoustic Sensing System for Vibration Detection and Classification in Railways

ZHU Songlin[1], WANG Zhongyi[1], XIE Yunpeng[1], SUN Zhi[2]

(1. Wireline Product Planning Department, ZTE Corporation, Shanghai 201203, China;
2. Shenzhen Branch of China Telecom Corporation Limited, Shenzhen 518000, China)

DOI: 10.12142/ZTECOM.202402010

https://kns.cnki.net/kcms/detail/34.1294.TN.20240621.1412.002.html, published online June 21, 2024

Manuscript received: 2024-03-10

Abstract: A distributed acoustic sensing (DAS) system is proposed and a data processing method for vibration is designed in this paper. The proposed DAS system is based on the Rayleigh scattering signal and utilizes phase-sensitive optical time-domain reflectometry (ϕ-OTDR) to demodulate the environmental vibration. It can collect the vibration information in railways and implement vibration classification based on the feature of sensed vibration signals. This system has been deployed in Guangzhou Shenzhen High-Speed Railway, and the experimental results validate its effectiveness.

Keywords: DAS; φ-OTDR; vibration classification

Citation (Format 1): ZHU S L, WANG Z Y, XIE Y P, et al. A distributed acoustic sensing system for vibration detection and classification in railways [J]. ZTE Communications, 2024, 22(2): 80 – 84. DOI: 10.12142/ZTECOM.202402010

Citation (Format 2): S. L. Zhu, Z. Y. Wang, Y. P. Xie, et al., "A distributed acoustic sensing system for vibration detection and classification in railways," *ZTE Communications*, vol. 22, no. 2, pp. 80 – 84, Jun. 2024. doi: 10.12142/ZTECOM.202402010.

1 Introduction

Optical fibers are the carrier of optical signals for optical fiber communications, which have been research hotspots since its introduction. Recently, with the development of relevant research, optical fiber sensing technology has drawn more and more attention. Fiber can work as the sensing component and sense environmental variations, such as vibration, temperature, and strains[1]. Several researchers have applied optical fiber sensing to geophysics and obtained several achievements[2 – 4].

In addition to seismic wave monitoring in geophysics, distributed optical fiber sensing (DOFS) is a widely applied classical fiber sensing technology[5]. While an optical signal transmits through the fiber, the backscattering signals are generated and also propagated along the optical fiber. The scattering signals generated in different positions arrive at the start of fiber at different time points. Besides, the parameters of backscattering signals are related to environmental parameters in the scattering position. Hence, the sensing information can be obtained by analyzing the backscattering signals. These are the basic principles of DOFS.

Distributed acoustic sensing (DAS) is one Rayleigh scattering DOFS technology[6]. By applying a narrowband laser, coherent detectors can detect optical Rayleigh backscatter signals and obtain electrical signals. The electrical signal is sampled by an analog-to-digital converter (ADC) and further used to sense acoustic information by digital signal processing (DSP). Due to its high sensitivity, DAS is widely applied to provide unmanned real-time leak monitoring in the oil and gas industry.

Traffic monitoring is another important use case for DAS[7]. Traffic information can be obtained by analyzing sensing vibration information provided by DAS system. The characteristics of low cost and high coverage also match long-distance railways. However, the requirement of railway monitoring focuses on the identification of vehicle positions and intrusion detection, which involves positioning and vibration differentiation. To achieve this requirement, we propose a DAS system for railways in this paper, which can detect and locate vibration of vehicles and recognize the intrusion behavior.

This innovative solution can identify railway surrounding events through DAS systems utilizing single mode optical fibers within the communication network. This approach not only enhances the capabilities of communication infrastructure but also expands its potential commercial value. This system has been deployed in Guangzhou Shenzhen High-Speed Railway and the experimental results verified its effectiveness.

The remainder of this paper is organized as follows. Section 2 introduces the theory and principle of the DAS system. The proposed DAS system is designed and the data processing steps are given in Section 3. Experimental setup and results are given in Section 4 and we conclude this paper in Section 5.

2 Theory and Principle

The DAS system based on phase-sensitive optical time-domain reflectometry (ϕ-OTDR)[7] is introduced in this section, where the probe pulse signal is injected into the fiber and the sensing information is derived by analyzing the scatting signal. The probe pulse signal can be described as:

$$S_t = A_t \exp\left[i\left(\omega_t + \phi_t\right)\right], \tag{1}$$

where A_t, ω_t and ϕ_t are the amplitude, angular frequency, and phase of probe pulse signal i is the imaginary unit.

The probe pulse transmits through the fiber and generates the Rayleigh scattering signals. Rayleigh scattering does not change the frequency of light[8], and the scattering signal can be described as:

$$S_s = A_s \exp\left[i\left(\omega_t + \phi_s\right)\right], \tag{2}$$

where A_s, ω_t and ϕ_s are the amplitude, angular frequency, and phase of the Rayleigh scattering signal. The amplitude and phase are related to the strain received by the fiber. Therefore, the vibration information can be sensed.

To demodulate the parameters of the scattering signal, the coherent detection technology is applied. In a coherent detector, the local-oscillator (LO) signal and scattering signal work as inputs together and they should be at the near frequencies. Considering the heterodyne reception technology, the LO signal can be described as:

$$S_O = A_O \exp\left[i\left(\omega_O + \phi_O\right)\right], \tag{3}$$

where A_O, ω_O and ϕ_O are the amplitude, angular frequency, and phase of the LO signal. $\omega_{\mathrm{IF}} = \omega_O - \omega_t \neq 0$ is the frequency difference between the LO signal and scattering signal and represents the frequency of intermediate frequency signals.

The LO signal and scattering signal interfere and output two intermediate frequency signals after the hybrid. Two photodetectors convert the light signals into electrical signals, which can be described as:

$$\begin{cases} S_I = \dfrac{1}{2} R\left[A_s A_O \cos\left(\omega_{\mathrm{IF}} t + \phi_s - \phi_O\right) + A_s^2 + A_o^2\right] \\ S_Q = \dfrac{1}{2} R\left[A_s A_O \sin\left(\omega_{\mathrm{IF}} t + \phi_s - \phi_O\right) + A_s^2 + A_o^2\right]. \end{cases} \tag{4}$$

These medium-frequency electrical signals are down-converted into baseband signals by digital signal processing, and the DC components are filtered. The AC electrical signals are obtained as

$$\begin{cases} I_I = \dfrac{1}{2} R A_s A_O \cos\left(\phi_s - \phi_O\right) \\ I_Q = \dfrac{1}{2} R A_s A_O \sin\left(\phi_s - \phi_O\right), \end{cases} \tag{5}$$

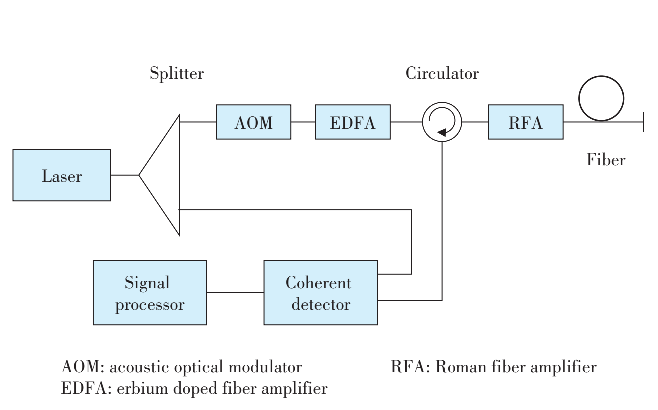
where R is the response factor of the photodetector.

There amplitude A_s and phase ϕ_s can be demodulated as follows:

$$\begin{cases} A_s \propto \sqrt{I_I^2 + I_Q^2} \\ \phi_s \approx \arctan \dfrac{I_Q}{I_I} + \phi_O . \end{cases} \tag{6}$$

3 System and Data Process

In this paper, we propose a DAS system as shown in Fig. 1. The laser generates a light signal, and one acoustic optical modulator (AOM) is utilized to modulate the probe pulse with a certain frequency shift. The AOM can achieve a high extinction ratio (ER) to support the following accurate date process. To achieve enough sensing length, the probe pulse is amplified by an erbium doped fiber amplifier (EDFA) before being injected into the fiber. Then the amplified probe pulse transmits through the fiber and generates backscatter signals. The Roman fiber amplifier (RFA) is utilized to keep probe pulse power sufficient during transmission. The scattering signals generated in different positions arrive at the


Splitter
Circulator
AOM
EDFA
RFA
Laser
Fiber
Signal processor
Coherent detector
AOM: acoustic optical modulator
EDFA: erbium doped fiber amplifier
RFA: Roman fiber amplifier

▲Figure 1. Proposed distributed acoustic sensing (DAS) system circulator at different times, and they can be distinguished in the time domain. For a specific scattering signal, it passes the circulator. The LO signal enters the coherent signal with the scattering signal, where the LO signal is one part light from the laser. As introduced in Section 2, the amplitude and phase of the scattering signal can be demodulated, which can be described as $A_{i,j}$ and $\phi_{i,j}$, where i represents this scattering signal is generated by the i-th probe pulse in the j-th position.

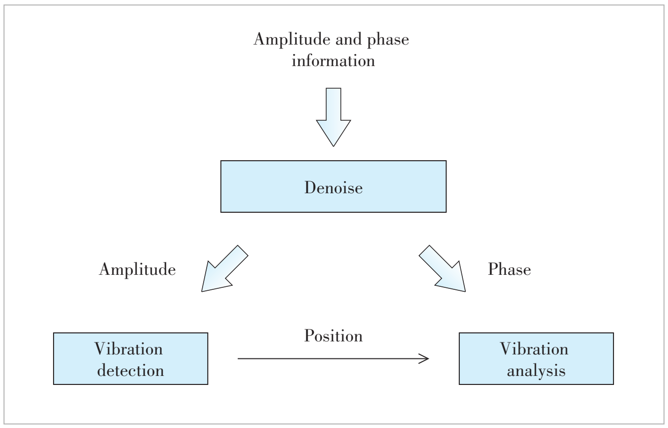
In the data processor, the vibration of vehicle detection and intrusion behavior recognition are realized based on the amplitude and phase information, and the detailed process is shown in Fig. 2. First, the denoising step is needed to achieve high accuracy. Then, the amplitude information is used to detect the vibration. The amplitudes generated by the i-th probe pulse are represented as the sequence $\left[A_{i,1}, A_{i,2}, \cdots, A_{i,m}\right]$, where m is the toal number of scattering signal sampled by ADC. The amplitude difference sequence $\left[\Delta A_{i,1}, \Delta A_{i,2}, \cdots, \Delta A_{i,m-1}\right]$ is calculated by $\Delta A_{i,k} = \left|A_{i,k+1} - A_{i,k}\right|$, where $k = 1, 2, \cdots, m - 1$. The peak value $\Delta A_{i,j}$ represents that the vibration occurs at the j-th position.

The short-term energy and short-term zero-crossing rate are used to detect the vibration position based on the amplitude sequence. Furthermore, the vibration may be caused by different reasons, and they can be classified into different vibration types. Supposing the vibration is detected at the j-th position in the i-th probe pulse, the phases at different times can be represented as sequence $[\phi_{i,j}, \phi_{i+1,j}, \cdots, \phi_{i+n,j}]$, where n is the total number of record phases. $[\Delta\phi_{i,j}, \Delta\phi_{i+1,j}, \cdots, \Delta\phi_{i+n-1,j}]$ is the phase difference sequence, which corresponds to the vibration waveform modulation on the scattering signal, where $\Delta\phi_{i,j} = \phi_{i+1,j} - \phi_{i,j}$.

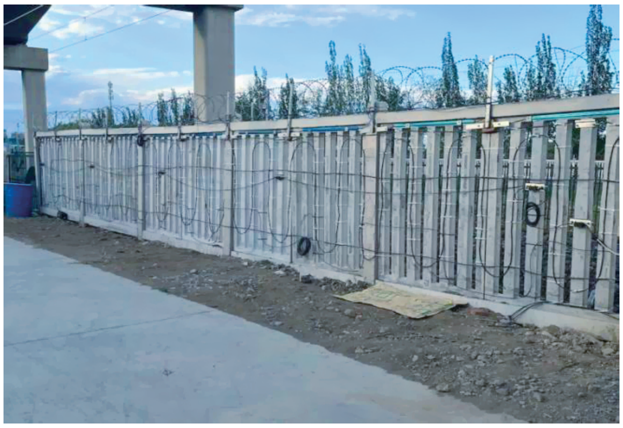
The time-frequency signal is obtained through a short-time Fourier transform of the phase difference sequence, and the features of the time-frequency signal are used in the following vibration classification. In detail, the key step of identifying vibration events is to select appropriate audio features to characterize the corresponding vibration waveform. In our system, we adopt the spectral image feature (SIF) of the phase difference sequence, which involves a short-time Fourier transform on the original sequence and the preprocessing of the two-dimensional time-frequency features.

4 Experimental Setup and Results

The proposed DAS system has been deployed in the Guangzhou Shenzhen High-Speed Railway. The sensing fiber is approximately 20 km long and is laid on railways through fixing with guardrails (Fig. 3). Through extensive testing and comparison of optical fiber deployment methods, it was found that the S-shaped optical fiber deployment has advantages in terms of the accuracy of vibration signal acquisition and accuracy of system identification for different events.



▲Figure 2. Proposed data processor

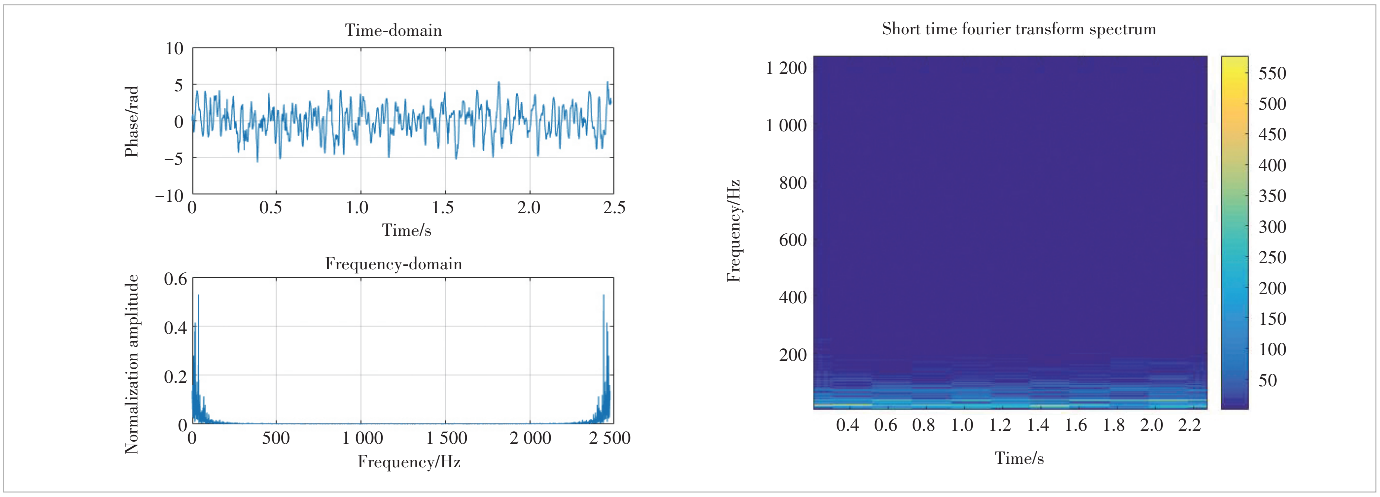


▲Figure 3. Fiber laying scenario

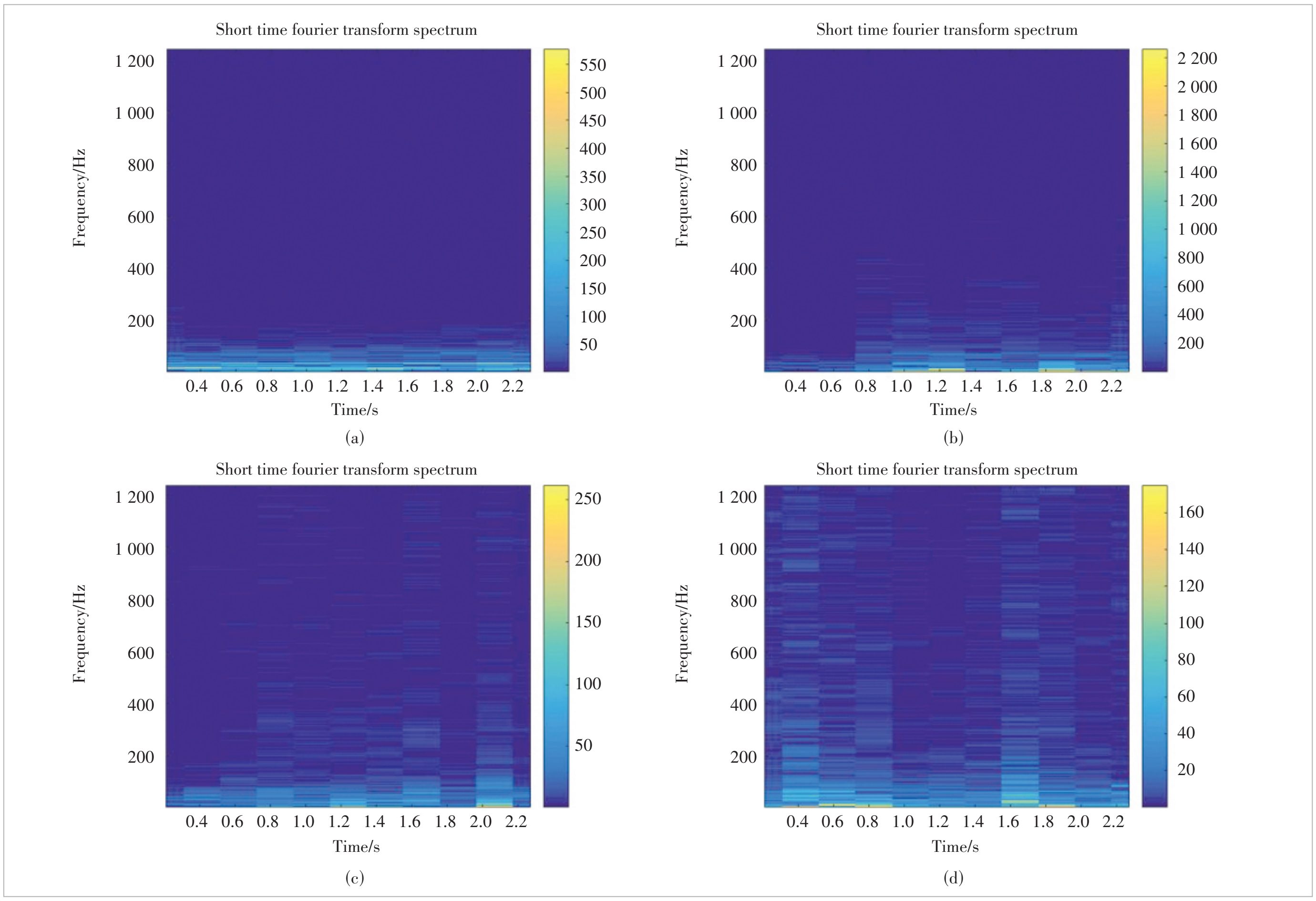
The laser generates the pulse light signal with a power of 23 dBm at 1 550.12 nm. The split ratio is set as 1:1. The repetition time of the probe pulse is set as 0.4 ms, which can cover the round-trip time of the probe pulse in a 40-km fiber. The width of the probe pulse is 80 ns, which corresponds to the spatial resolution of 8 m. The frequency shift introduced by AOM is 80 MHz. In RFA, the wavelength of pump light is 1 450 nm and the power is set as 21 dBm. The system can support the max sensing length of 40 km under these setups.

Fig. 4 shows a specific vibration signal. On the left, the time-domain signal and the frequency-domain signal are given, respectively. The time domain signal is the phase difference sequence obtained from the above step, which shows the vibration waveform. The time-domain signal has several features and the frequency-domain signal is obtained by the Fourier transform. It can be observed that the frequency-domain signal is between 0 to 2 500 Hz, and it mainly contains the low-frequency part (lower than 500 Hz) and high-frequency part (higher than 2 000 Hz). The time-frequency spectrum signal shown on the right is obtained by the short-time Fourier transform, and the colors correspond to different frequencies.

Fig. 5 gives several time-frequency spectrum signals of different situations, where Fig. 5(a) corresponds to the situation where there are no vehicles or intrusions, Fig. 5(b) the situation where there is a vehicle but no intrusion, Fig. 5(c) the situation where there are intrusions but no vehicle, and Fig. 5(d) the situation where there are both vehicles and intrusions. It can be seen that the proportion of high-frequency components varies. Based on this, we can design a classification algorithm to recognize different vibration situations. In our ex-



▲Figure 4. Vibration signal



▲Figure 5. Time-frequency spectrum signals in different situations periment, the classification accuracy can achieve 90%.

5 Conclusions

In this paper, we propose one DAS system to realize vibration detection and classification for railways. This system is based on φ-OTDR, and the amplitude and phase demodulated from scattering signals are analyzed to obtain the vibration waveform. Further, the vibration waveform is converted to the time-frequency spectrum signals, which show the frequency feature and are used for vibration situation classification. This system has been deployed in Guangzhou Shenzhen High-Speed Railway, and the classification accuracy can achieve 90%.

References

[1] LEE B. Review of the present status of optical fiber sensors [J]. Optical fiber technology, 2003, 9(2): 57 – 79. DOI: 10.1016/s1068-5200(02)00527-8

[2] MARRA G, CLIVATI C, LUCKETT R, et al. Ultrastable laser interferometry for earthquake detection with terrestrial and submarine cables [J]. Science, 2018, 361(6401): 486 – 490. DOI: 10.1126/science.aat4458

[3] MARRA G, FAIRWEATHER D M, KAMALOV V, et al. Optical interferometry-based array of seafloor environmental sensors using a transoceanic submarine cable [J]. Science, 2022, 376(6595): 874 – 879. DOI: 10.1126/science.abo1939

[4] ZHAN Z W, CANTONO M, KAMALOV V, et al. Optical polarization-based seismic and water wave sensing on transoceanic cables [J]. Science, 2021, 371(6532): 931 – 936. DOI: 10.1126/science.abe6648

[5] LU P, LALAM N, BADAR M, et al. Distributed optical fiber sensing: review and perspective [J]. Applied physics reviews, 2019, 6(4): 041302. DOI: 10.1063/1.5113955

[6] SHANG Y, SUN M C, WANG C, et al. Research progress in distributed acoustic sensing techniques [J]. Sensors, 2022, 22(16): 6060. DOI: 10.3390/s22166060

[7] LIU H Y, MA J H, XU T W, et al. Vehicle detection and classification using distributed fiber optic acoustic sensing [J]. IEEE transactions on vehicular technology, 2020, 69(2): 1363 – 1374. DOI: 10.1109/TVT.2019.2962334

[8] WANG Z N, ZHANG L, WANG S, et al. Coherent Φ-OTDR based on I/Q demodulation and homodyne detection [J]. Optics express, 2016, 24(2): 853 – 858. DOI: 10.1364/oe.24.000853

Biographies

ZHU Songlin (zhu.songlin@zte.com.cn) received his MS degree in theoretical physics from Hangzhou University, China in 1998 and PhD degree in electronic science and technology from Zhejiang University, China in 2001. He has been with the Wireline Product Planning Department, ZTE Corporation since 2001. His research interests include FTTx technology for optical communications and OTDR technology for optical sensing applications.

WANG Zhongyi received his bachelor's degree in communication engineering from Beijing University of Science and Technology, China in 2020 and his master's degree in information and communication engineering from Beijing University of Posts and Telecommunications, China in 2023. He joined the Wireline Product Planning Department, ZTE Corporation as a standard pre-research engineer in 2023. His research interests include the next generation PON technology and sensing technology in PON.

XIE Yunpeng obtained his bachelor's degree in computer science from the National University of Defense Technology, China in 1991 and his master's degree in communication engineering from Xidian University, China in 1999. He joined ZTE Corporation in 2000. His research interests include PON technology for optical access networks, FTTx optical link diagnosis, and fiber optic sensing technology.

SUN Zhi received his MS degree in microelectronics and semiconductor devices from Xidian University, China in 2000. He joined the Customer Service Dispatching and Resource Center, Shenzhen Branch of China Telecom in 2000. His research interests include fulfillment service, network resource management, operation support systems, and deterministic networks.

Adaptive Hybrid Forward Error Correction Coding Scheme for Video Transmission

XIONG Yuhui[1], LIU Zhilong[2], XU Lingmin[2], HUA Xinhai[2], WANG Zhaoyang[1], BI Ting[1], JIANG Tao[1]

(1. Huazhong University of Science and Technology, Wuhan 430074, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202402011

https://kns.cnki.net/kcms/detail/34.1294.TN.20240428.1442.002.html, published online April 28, 2024

Manuscript received: 2023-07-22

Abstract: This paper proposes an adaptive hybrid forward error correction (AH-FEC) coding scheme for coping with dynamic packet loss events in video and audio transmission. Specifically, the proposed scheme consists of a hybrid Reed-Solomon and low-density parity-check (RS-LDPC) coding system, combined with a Kalman filter-based adaptive algorithm. The hybrid RS-LDPC coding accommodates a wide range of code length requirements, employing RS coding for short codes and LDPC coding for medium-long codes. We delimit the short and medium-length codes by coding performance so that both codes remain in the optimal region. Additionally, a Kalman filter-based adaptive algorithm has been developed to handle dynamic alterations in a packet loss rate. The Kalman filter estimates packet loss rate utilizing observation data and system models, and then we establish the redundancy decision module through receiver feedback. As a result, the lost packets can be perfectly recovered by the receiver based on the redundant packets. Experimental results show that the proposed method enhances the decoding performance significantly under the same redundancy and channel packet loss.

Keywords: video transmission; packet loss; Reed−Solomon code; Kalman filter

Citation (Format 1): XIONG Y H, LIU Z L, XU L M, et al. Adaptive hybrid forward error correction coding scheme for video transmission [J]. ZTE Communications, 2024, 22(2): 85 – 93. DOI: 10.12142/ZTECOM.202402011

Citation (Format 2): Y. H. Xiong, Z. L. Liu, L. M. Xu, et al., "Adaptive hybrid forward error correction coding scheme for video transmission," *ZTE Communications*, vol. 22, no. 2, pp. 85 – 93, Jun. 2024. doi: 10.12142/ZTECOM.202402011.

1 Introduction

Mobile video services have proliferated with the growth of wireless communications and the Internet, and video traffic had been expected to account for 82% of total network traffic by 2022. However, networks with limited capacity are struggling to support the growing number of mobile video users. The complex uncertainty of wireless channels limits transmission rates, while wired transmissions suffer from packet loss due to buffer congestion at routing nodes. For the former, there are well-known channel coding methods[1 – 3] that promise transmission rates close to the Shannon bound. For the latter, existing solutions mainly involve retransmission techniques or forward error correction coding. The retransmission[4] does not require any redundant packets, but the extra round-trip time (RTT) increases the end-to-end delay of the entire video and therefore does not guarantee real-time video transmission. Forward error correction (FEC) coding[5], by way of contrast, is of interest due to its ability to recover lost source packets without adding any RTT.

The well-known WebRTC uses an exclusive OR (XOR)-based[6] FEC coding that generates new redundant packets by XORing the original packets. These redundant packets are sent to the receiver together with the original packets and the receiver recovers the lost packets according to the corresponding mapping relationships. There should be neither too many nor too few redundant packets, as this would result in waste or inadequate protection. Therefore, the FEC should adjust the number and size of redundant packets to the network environment to balance reliability and latency. To select the appropriate level of redundancy to cope with dynamic network environments, the WebRTC uses the current redundancy state to query the FEC redundancy for the next packet. The XOR encoding and single-step adaptive algorithms described above together form the FEC scheme for WebRTC.

Other FEC schemes also focus on improving packet loss recovery through advanced encoding methods and adaptive algorithms. Examples of such methods include fountain codes[7], Raptor codes[8 – 9], and Reed-Solomon (RS) codes[10 – 11]. Among them, RS codes are widely used by the telecommunication industry due to their superior protection capabilities. However, the drawback of RS codes is that decoding requires multiple matrix inversions, which results in extremely high computational complexity. In situations where the matrix dimension is too high, such as in high-definition video transmission with large packets, the decoding time may be too long to meet the needs of real-time transmission.

As for other adaptive algorithms, ATIYA et al. introduced a non-linear prediction method for automatic feature selection[12], and EMARA et al. combined an ingenious coding scheme with a network adaptive algorithm for parameter updating[13]. However, these approaches relied solely on historical network patterns to predict future patterns, overlooking the complex relationships that may exist between past and future patterns. To better exploit the correlation between current and previous network states in a weak network environment, CHENG et al. proposed a DeepRS[14] adaptive redundancy control algorithm in 2020. The algorithm uses a long short-term memory (LSTM) network[15–16] to predict the probability of packet loss and dynamically adjusts the redundancy rate of the RS encoder. However, integrating the LSTM network with the underlying user datagram protocol (UDP) protocol requires significant engineering effort.

In response to the shortcomings of existing coding and adaptive algorithms, we propose a new coding scheme covering low-density parity-check (LDPC) and RS to ensure smooth transmission of arbitrary definition video and design a multi-step Kalman filter-based adaptive algorithm for practical deployment. The contributions in this paper are summarized as follows.

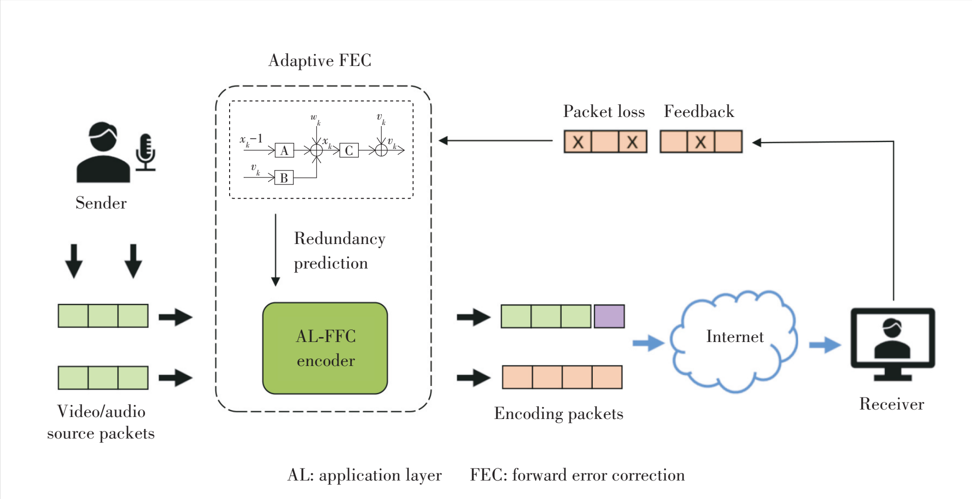
- We develop a hybrid FEC highly efficient encoding method to cope with continuous burst packet loss and different application scenarios. We design an encoding scheme of the LDPC code and the RS code in their respective optimal code length ranges. We optimize a progressive edge growth algorithm to get the LDPC coding matrix of the application layer. The coding scheme of the design system could cover various source code lengths and reduce the computational complexity of codes.
- We propose a Kalman filter-based multi-step adaptive method for video transmission. The system makes multi-step predictions based on packet loss feedback and then predicts the coding code rate based on the decoding terminal redundancy. It turns out that our method always maintains a high data recovery ratio with the interval change of the packet loss rate.

The rest of this paper is organized as follows. Section 2 describes the system architecture. Next, Section 3 proposes a frame-level partitioning method. Then, we present the hybrid coding method in Section 4. The code rate adaptation algorithm is evaluated in Section 5. Sections 6 and 7 give the evaluation and conclusions.

2 System Architecture

The prototype system verification framework for combating weak network conditions in video and audio transmission is shown in Fig. 1. The system uses the application-layer end-to-end FEC technology at the video frame level, and selects the optimal bit rate based on the network status information fed back by the terminal. Metrics such as historical packet loss rates and throughput are used to predict future network states and adaptive packet loss compensation is performed to optimize overall performance.

The sender serves as the video input source to the system, providing raw video streams that are processed into multimedia stream files with frame structures through generic protocol encoding. Before transmission, the sender side establishes a channel for transmission by selecting the optimal bitrate based on the network status information feedback from the terminal. To improve the reliability of end-to-end mobile video transmission, the sender-side encoder performs frame-level FEC (we give a frame-level partitioning method in this reign as shown in Section 3) encoding on the multimedia stream at the application layer and packages and sends the data according to the real-time transport protocol (RTP)/UDP principles. Specifically, on a small timescale, the application-layer FEC encoder processes each video frame in a fast serial manner based on the coding rate, that is, it divides each frame into source data packets, performs FEC encoding on these packets to generate repair data packets that facilitate the recovery of the original video data stream by the receiver. The mobile terminal serves as the video output and processes the received data packets by unpacking them. After obtaining the raw data packets with the RTP/UDP headers removed, the terminal-side decoder performs decoding and error correction accord-



▲Figure 1. Framework of FEC

ing to the agreed FEC encoding and decoding principles and finally obtains the raw video stream through the decoder.

Upon receiving the restored video stream, the mobile terminal sends real-time feedback on network status information (such as packet loss rate and throughput) through a control signaling port to the adaptive module located at the sender. The adaptive module monitoring and prediction unit in the module uses the feedback information to determine the encoding rate of the FEC encoder on the sender in the next slot. It is worth noting that during this process, the adaptive module needs to prevent excessive coding redundancy that may cause transmission congestion, while ensuring that the FEC encoder generates enough repair data packets to support data packet recovery, without compromising user experience in weak network environments. Additionally, the application-layer FEC encoder on the sender should always maintain the same data packet generation, validation, and recovery as the application-layer FEC decoder on the mobile terminal.

3 Frame-Level Partitioning

We divide the video into blocks according to the frames. Each block contains one or more frames, and we ensure the FEC encoding and decoding are synchronized with the video timestamp as much as possible. The sender obtains the video frame information by calling the FFmpeg tools and recombines the frames into blocks. The FEC encoding and decoding process at the application layer is based on the entire block. Successful decoding can obtain the entire video data of the block. In this paper, we set a limit of K_{frame} for the number of frames in a block, as excessively long blocks cause additional video delays. After the video is divided into blocks, we divide the blocks into coding data packets.

For short codes, the upper limit of the block size is 20×1 400 bytes, where 1 400 setups are based on the maximum transmission unit (MTU) and 20 based on the decoding performance of RS coding (we use RS coding because of its superior performance in this region as is shown in Section 4.2). Generally, the data sizes of B-frames and P-frames are small[17]. We take several consecutive B-frames or P-frames as a block, if the data size of the block does not exceed 28 000 bytes and the number of frames in the block does not exceed K_{frame}. The block data carries the encoding method and block size (in bytes). Optionally, we can place it in the block header as basic information.

The optimal interval selection $[N_{\min}, N_{\max}]$ for medium-long codes is more flexible. Due to the encoding characteristics of the LDPC code, longer code length results in better decoding performance. In addition, different code lengths and encoding code rates correspond to different LDPC generator matrices, where the storage complexity of LDPC generator matrices is $O(n^2)$. For instantaneous decoding refresh (IDR) frames that contain a large amount of information, they can be used directly as a block. Accordingly, we combine several P-frames with a large number of data into a single block, when the number of frames is not larger than K_{frame} and the block data size is not larger than $N_{\max}$×1 400 bytes. Within the allowable range of decoding delay, a longer code length means an enhanced ability to cope with continuous packet loss.

4 Hybrid Coding Method

In this section, we first present an improved LDPC that balances decoding latency and error correction performance in application layer packet loss scenarios. The optimal operating regions of RS and LDPC codes are then designed based on the performance analysis of the improved LDPC and RS codes. The two coding methods are combined to cover the requirements of various code lengths.

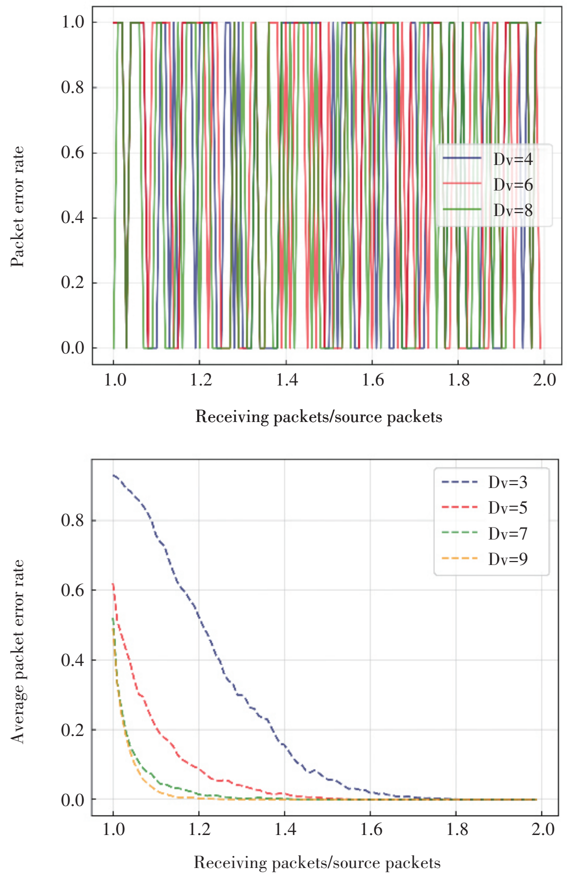
4.1 Improved Medium-Long LDPC Codes

As linear block codes, an LDPC or an XOR code is defined by its parity-check matrix $\boldsymbol{H}$ of dimensions $(n-k)\times n$, where n represents the number of all packets and k is the source packet number. The entries of the parity-check matrix $\boldsymbol{H}$ are exclusively 1 or 0, which means that it operates in the Galois Field GF(2). The parity-check matrix is so named because it provides $n-k$ parity-check equations that generate constraints between data bits and parity bits. Moreover, an LDPC code is defined as a linear block code for which the parity-check matrix $\boldsymbol{H}$ is very sparse, which means a low density (LD) of 1.

We construct an LDPC parity-check matrix $\boldsymbol{H}$ using the progressive edge growth (PEG) method, where the code length and the variable node (Dv) can be adjusted, like the physical layer LDPC channel coding. However, unlike before, when Dv is even, the decoding result fails irregularly, and the decoding result has a foreseeable change when Dv is odd, as shown in Fig. 2. Because in the binary erasure channel, the erased bits may appear to be unevenly distributed, and bits will interfere with each other when Dv is even. So only odd values of Dv can be selected. In addition, Dv represents the protection of the source packet in relation to the redundant packet. Therefore, LDPC decoding performance increases as Dv increases. At the same time, the decoding delay as a cost also increases. As shown in Fig. 3, with the number of Dv increases, there is a noticeable decrease within the range of 3 – 7, followed by a stabilizing trend. When Dv increases to a threshold value, the improvement in decoding performance no longer changes significantly as Dv increases further, but the latency still shows a linear increase (as shown in Fig. 4). Considering the delay and error correction performance, the threshold value of Dv is chosen to be 7 in the application layer of the LDPC scheme.

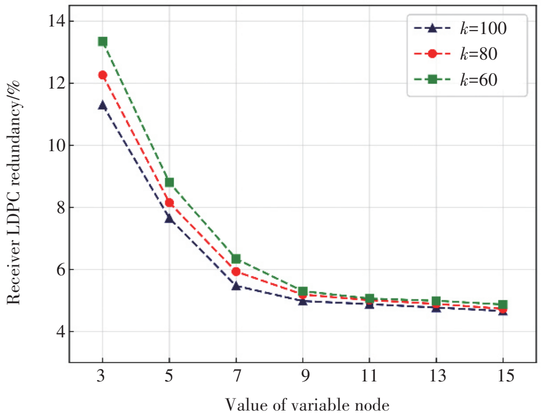
Then, to generate the code vector $\boldsymbol{c}$ from the data vector $\boldsymbol{s}$, we define the generator matrix $\boldsymbol{G}$, which holds:

$$\boldsymbol{c}=\boldsymbol{s}\boldsymbol{G}. \tag{1}$$



Dv: variable node

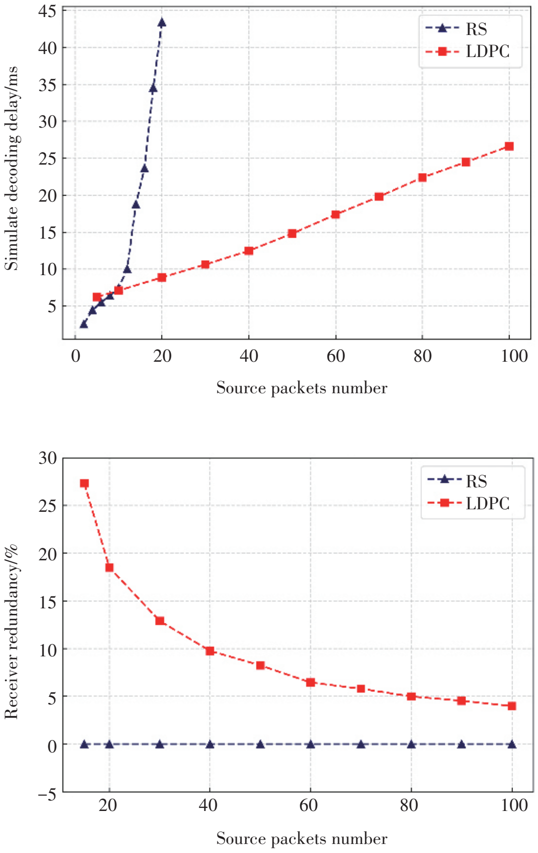
▲**Figure 2. Packet error rate performs irregularly when Dv is even or odd**



LDPC: low-density parity-check

▲**Figure 3. Average receiver redundancy in different values of variable nodes**

The main algorithms that create $\boldsymbol{G}$ from $\boldsymbol{H}$ consist in arranging $\boldsymbol{H}$ in an appropriate form that allows to develop $\boldsymbol{G}$ and construct it in a systematic form. Thus, $\boldsymbol{H}$ is randomly generated and then organized as:



LDPC: low-density parity-check RS: Reed-Solomon

▲**Figure 4. RS and LDPC performance with the number of source packets**

$$\boldsymbol{H} = [\boldsymbol{P}^T | \boldsymbol{I}_{n-k}], \tag{2}$$

where $\boldsymbol{I}_{n-k}$ is the identity matrix of dimensions $(n-k) \times (n-k)$ and $\boldsymbol{P}$ is a sparse matrix of dimensions $k \times (n-k)$. So, the corresponding $\boldsymbol{G}$ matrix is:

$$\boldsymbol{G} = [\boldsymbol{I}_k | \boldsymbol{P}]. \tag{3}$$

This approach is based on the use of the Gauss-Jordan elimination. However, if we transform the right part of $\boldsymbol{H}$ to an identity matrix $\boldsymbol{I}_{n-k}$, there is no way to ensure $\boldsymbol{P}$ is a sparse matrix. So we define $\boldsymbol{H}$ as the encoding matrix $\boldsymbol{G}$ directly. The encoding matrix $\boldsymbol{H}$ of dimensions $n \times k$ is different from the parity-check matrix $\boldsymbol{H}$ before. We define the encoding vector $\boldsymbol{sp}$ as:

$$\boldsymbol{sp} = \boldsymbol{Hs}. \tag{4}$$

We generate code vector $\boldsymbol{sp}$ from data vector $\boldsymbol{s}$. In the system code, $\boldsymbol{sp}$ includes $\boldsymbol{s}$, which means the source bit/byte and the redundancy bit/byte are separated. Whether it is a system or non-system code, we can rebuild the coefficient matrix $\boldsymbol{H}'$

to restore the original data by the approach of the Gauss-Jordan elimination as:

$$\boldsymbol{sp}' = \boldsymbol{H}'\boldsymbol{s}\,, \tag{5}$$

where a certain correspondence exists between $\boldsymbol{sp}'$ and $\boldsymbol{H}'$. $\boldsymbol{sp}'$ means the accepted sequence, and $\boldsymbol{H}'$ means the corresponding row in $\boldsymbol{H}$ with $\boldsymbol{sp}'$. Because in the packet erasure channel, a lost bit/byte can be located at a specific position.

The decoding end (LDPC uses soft decision at the physical layer, while at the application layer, the hard decision is used. Therefore, we employ the Gaussian elimination method at the decoding end) reassembles the received packets, where the rows in the coefficient matrix $\boldsymbol{H}'$ correspond to the $\boldsymbol{sp}'$ code bits. If the reassembled matrix is full rank, it satisfies the Gaussian elimination decoding requirements:

$$\boldsymbol{H}'_{m\times k} s_k = \boldsymbol{sp}'_m\,. \tag{5.1}$$

The receiver needs to provide feedback to the sender regarding the overall packet loss rate based on the total number of packets received. Once decoding is successful, the received information packets are arranged in sequence and the video data are extracted based on the block header information. With the improvement, LDPC can perform even better in the binary erasure channel in the application layer.

4.2 Length Bounds for LDPC and RS

For encoders, the coding method directly affects the delay and effectiveness of packet loss recovery. We use a hybrid coding method of LDPC and RS, and restrict the code lengths of LDPC and RS in their own optimal interval to cover all code length requirements.

The RS code, as an ideal code, can be successfully decoded when receiving several packets equal to the number of information source packets. So, while the sender pays for extra n–k redundant packets, the receiver only needs to receive k arbitrary packets to recover the source packet. However, as the code length increases, the computational complexity of the RS code also increases sharply. In contrast, LDPC is a non-ideal code, so it is necessary to receive extra redundancy to ensure successful decoding (when the decoding sparse matrix is not full rank, the Gaussian elimination method cannot be used). However, the computational complexity of LDPC increases linearly with the code length. LPDC is therefore expected to perform better in the long code region.

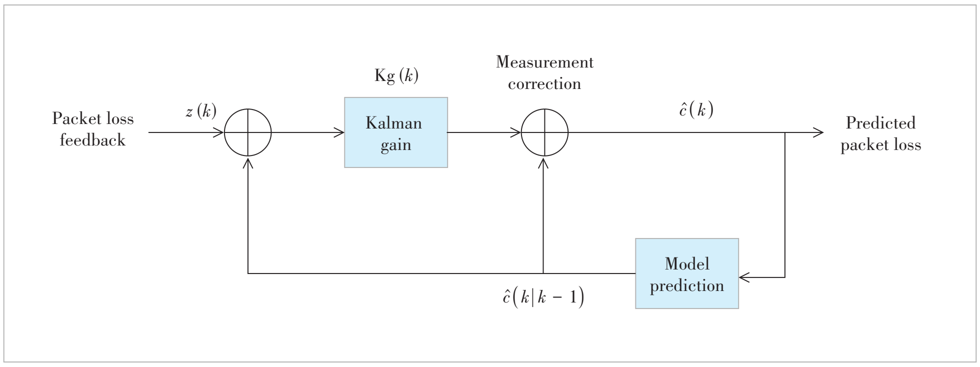
Fig. 4 shows the performance of RS and LDPC in the decoding end under the simulation environment. It is shown that RS has better performance in short codes, but as the code length increases, the delay of RS cannot meet requirements. Therefore, we choose to use LDPC instead of RS. LDPC operates in the binary field GF(2) different from RS in GF(2^8). It causes LDPC as an upper-layer coding method to have a lower decoding delay but also requires considering the additional redundancy at the receiving end of LDPC. Fig. 4 shows as the code length increases, LDPC has a high receiver redundancy in 20 source packets and it gradually becomes lower and closer to the ideal code performance, so we decide to use RS coding for code lengths not greater than 20 and LDPC coding for other situations. Besides, we set the upper limit of $N_{\max}$ comprehensive weight decoding delay and receiver redundancy to meet video transmission needs (the upper limit is relatively flexible due to LDPC characteristics, but we give an upper limit according to practical needs as shown in Section 4.1).

In Section 3, we limit the selection of RS and LDPC code lengths to certain intervals. However, these intervals are chosen to optimize the performance of RS and LDPC encoding and decoding within those specific code length ranges. It does not mean that we are restricted to only those code lengths.

5 Code Rate Adaptation Algorithm

FEC encoding redundancy allocation is mainly based on the estimated packet loss rate of the channel. Therefore, we establish a packet loss rate prediction module based on a multi-step Kalman filter system which is shown in Fig. 5. By calculating the linear minimum mean square error of the data and iteratively predicting results, we can obtain the next predicted value. The prediction process of this algorithm involves using the optimal result predicted at time $k-1$ and the measurement value at time k to calculate and update the optimal result of the state prediction at time k, and then continue iterating to obtain the predicted value of the next time.

First, we determine the state space equation used to estimate the packet loss rate of the channel. The state prediction equation is $c_k = \boldsymbol{A}c_{k-1} + \boldsymbol{B}u_k + w_k$, where $\boldsymbol{A}$ is the state transition matrix, c_{k-1} is the previous prediction value, $\boldsymbol{B}$ is the input gain matrix u_k system input vector, w_k has a mean of 0, covariance matrix $\boldsymbol{Q} = E\left[w_{k^2} \right]$, and follows a normal distribution



▲**Figure 5. Packet loss ratio predictor by using a Kalman filter** process noise. The state measurement equation is $z_k = c_k + v_k$, where v_k has a mean of 0, covariance matrix $\boldsymbol{R} = E[v_{k^2}]$, and follows a normal distribution measurement noise.

The single-step adaptive algorithm consists of two processes, namely prediction and correction. In the prediction phase, the filter uses the estimated packet loss rate of the previous state to predict the current state. In the correction phase, the filter uses the measurement value of the current state to correct the predicted value obtained in the prediction phase, i.e., the feedback value of the packet loss rate, to obtain a new estimate value that is closer to the true value. This new estimate value is the Kalman estimate value, which is used as the estimate of the previous state for the next Kalman estimate. Since the uncertainty of process noise and measurement noise cannot be modeled, the Kalman filter is used to continuously correct the estimation model to minimize the mean square error between the true value and the estimated value. The main steps are as follows:

$$\text{Step1: } Kg(k) = \frac{P(k-1) + W}{P(k-1) + W + Q}, \tag{6}$$

$$\text{Step2: } \hat{c}(k) = \hat{c}(k|k-1) + Kg(k)\left[z(k) - \hat{c}(k|k-1)\right], \tag{7}$$

$$\text{Step3: } P(k) = \left(1 - Kg(k)\right)\left(P(k-1) + W\right), \tag{8}$$

where $P(k) = E[(\hat{c}(k) - c(k))^2]$ is the error variance of the model. Through the Kalman filter, we obtain a packet loss rate c_k at a certain time scale.

To obtain the multi-step packet loss rate prediction value, the average packet loss rate of five blocks in the future is predicted at each step. Since the average packet loss rate measurement $z(k+1), z(k+2), z(k+3), z(k+4), z(k+5)$ is still unknown when encoding block $z(k)$ of the next blocks meets $z(k+1) = z(k+2) = z(k+3) = z(k+4) = z(k+5) = z(k)$.

As an important parameter for adaptive redundancy calculation, the packet loss is accompanied by another important parameter, which is the redundancy at the receiving end. Given that the RS code is ideal, decoding can be successfully achieved by receiving any k packets. However, the long-code LDPC method is not an ideal one, and the scheme needs to establish the cost of receiving redundancy at the receiver, which describes the mapping relationship between the number of additional packets required by the receiver and the retransmission rate due to decoding failures.

We establish a receiver redundancy cost function, and the selected code rate is influenced by the receiver redundancy cost. The goal is to minimize the overall transmission cost as much as possible. The receiver redundancy cost of the target video can be characterized by the receiver redundancy cost function:

$$C = [Q(k, r-m) - \delta \cdot P_{\text{rtt}}(k, r-m)]^2, \tag{9}$$

where C represents the receiver redundancy cost of the t-th block, i.e., the total cost of transmitting block t at the current code rate. $Q(k, k+r-m)$ represents the code rate with the source information bit length of k and the redundancy bit length of r (since the medium-long code is not in the form of a system code, r can be expressed as $n-k$, where n is the total information bit length after encoding), and m represents the number of lost packets. The code rate can be expressed as $k/(k+r)$ and the packet loss rate can be expressed as $m/(k+r)$. $k+r-m$ represents the number of packets received, and $Q(k, r-m)$ represents the function related to k and $k+r-m$. Optionally, $Q(k, r-m) = (r-m)/k$ can be directly expressed as the redundancy ratio of the receiver, where $P_{\text{rtt}}(k, r-m)$ represents the decoding failure rate under the current receiver redundancy ratio which is the ratio of the overall decoding errors of the block in the simulation process. Decoding failure requires retransmission of the block, and δ represents the weight of the decoding failure item. An increasing value of δ indicates that we consider the current decoding failure rate to be unacceptable.

By reversely solving the redundancy cost function problem, we can obtain the proportion of additional packets that the receiver needs to receive, r_{rec}, when the expected retransmission probability is not greater than a probability P using LDPC encoding. Then, by estimating the packet loss rate below, we can obtain the encoding redundancy:

$$r = \left[\left(k + \frac{k}{r_{\text{rec}}}\right)/c\right] - k, \tag{10}$$

where c is the current packet loss estimate and k is the length of the information source code.

6 Evaluation

6.1 Setup

1) Datasets: This paper considers a transmission channel based on the Gilbert-Elliott model[18], which is acknowledged as a common simulation environment of network packet delivery. The Gilbert-Elliott model is a Markov process, where B and G indicate the bad and good network state. The probability of transitioning from state B to G is denoted as P_{BG}, and the probability of transitioning from state G to B is denoted as P_{GB}. The transition matrix of the Markov chain is as follows:

$$\boldsymbol{A} = \begin{pmatrix} 1 - P_{\text{GB}} & P_{\text{GB}} \\ P_{\text{GB}} & 1 - P_{\text{GB}} \end{pmatrix}. \tag{11}$$

In a stable state, π_{G} is the probability of being in a good state and π_{B} a bad state.

$$\pi_{\text{G}} = \frac{P_{\text{GB}}}{P_{\text{GB}} + P_{\text{GB}}},$$

$$\pi_B = \frac{P_{GB}}{P_{GB} + P_{GB}}\ . \tag{12}$$

The formula for calculating the probability of packet loss is:

$$P_E = \pi_G(1 - k) + \pi_B(1 - h), \tag{13}$$

where k represents the probability of successful reception in a good state, and h represents the probability of packet loss in a bad state. Therefore, P_E is decided by setting the value of four parameters. Table 1 shows four parameter values for network packets loss ratio range of 5% – 20%, 20% – 40%, and 40% – 80%.

2) The high-bitrate 4K/30fps video is performed frame-level cutting by FFMPEG. We sample the generalized GE channel with different packet loss rates based on the total number of blocks. Then, the proposed scheme is compared with the WebRTC FEC algorithm in the GE channel:

- WebRTC-FEC based on the XOR algorithm employs a redundancy protection scheme. When the original packet size is less than or equal to 12, the redundancy level is directly obtained from a lookup table and the packet is encoded. When the original packet size is greater than 12, an interval-based grouping redundancy encoding method is used. The adaptive solution of WebRTC predicts the network status based on the video bitrate and the packet loss rate of video transmission. The redundancy level is obtained from a lookup table based on the video bitrate and packet loss rate, and is combined with the network status prediction that increases the RTT of transmission.
- The proposed AH-FEC is a long and short code adaptive redundancy coding algorithm based on RS and LDPC codes. It selects the coding redundancy degree based on the decoding redundancy at the receiver and the packet loss rate of the channel, balancing system latency and redundancy. At the encoding end, the future packet loss rates of multiple video blocks are predicted based on the packet loss rate of past video blocks for a certain period. Then, the encoding redundancy is dynamically adjusted accordingly.

3) Metric: In performance comparison, we consider the following measurement metrics.

- Data recovery ratio. The percentage of data blocks that are successfully recovered is the proportion of all data blocks.
- Redundancy ratio. The redundancy ratio is the proportion of encoded extra packets relative to packets, and it is expressed as $\frac{n - k}{k}$.

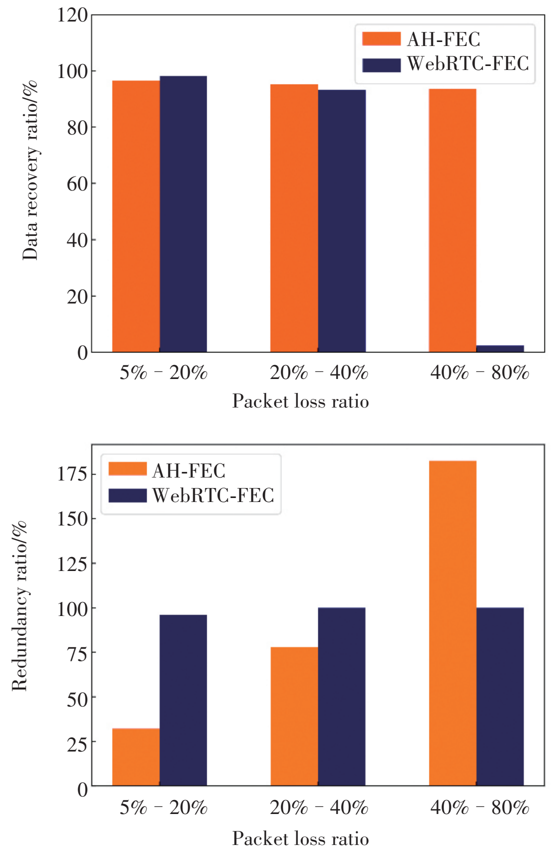
▼Table 1. Parameter values for different packets loss ratio ranges

Parameter	$P_{GB}(p)$	$P_{BG}(r)$	k	h	P_E	Range
Value	0.130	0.910	0.970	0.030	0.114	(0.050, 0.200)
	0.360	0.840	0.980	0.050	0.299	(0.200, 0.400)
	0.900	0.600	0.980	0.020	0.596	(0.400, 0.800)

6.2 Experiments on Simulation

Simulation traces contain 300 sample frames, and the redundancy strategy in WebRTC treats each frame as a block directly. We reorganize it according to the frame partition method and evaluate the results of the two FEC schemes.

Fig. 6 presents the recovery ratio and redundancy ratio of the two algorithms under network packet loss rates of 5% – 20%, 20% – 40%, and 40% – 80%. As evidenced by Fig. 6, the recovery success rates of both the proposed scheme and the WebRTC scheme approach 100% at packet loss rates of 5% to 20%, while the proposed scheme only requires about 32% redundancy. This lower redundancy requirement is attributed to the proposed scheme using RS-LDPC hybrid coding, which proves more efficient than the XOR coding employed by WebRTC, thereby requiring less redundancy to offer comparable protection capability. In addition, the adaptive capability of Kalman filtering is also superior to the fixed redundancy table of WebRTC, so the redundancy rate is greatly compressed. When the redundancy rate ranges between 20% and 40%, the WebRTC redundancy reaches its peak in the redundancy table at 100%. At this stage, a substantial reduction in the data recovery rate is observed due to WebRTC's lim-



AH-FEC: adaptive hybrid forward error correction

▲Figure 6. Data recovery and redundancy ratio in different packet loss ratios ited capability. In contrast, the proposed solution outperforms WebRTC in terms of both recovery and redundancy rates. Finally, the proposed solution works stably even at packet loss rates of 40% to 80%, when the redundancy rate reaches over 180% and the recovery rate reaches over 93%. In contrast, the redundancy of WebRTC is limited by an offline table so the FEC cannot work.

To illustrate the specific results in Fig. 6, we present the specific results of data recovery ratio and redundancy ratio in Table 2.

In summary, the proposed scheme presents the following advantages over the current FEC techniques (RFC5109) implemented in WebRTC:

- For network packet loss rates of 5% – 20%, the retransmission rate is kept at a relatively low level, achieving a 65.65% reduction in the redundancy ratio of the sent data.
- For network packet loss rates of 20% – 40%, in comparison to WebRTC, this approach improves the data recovery ratio by 2.21% and decreases redundant data by 22.06%.
- For network packet loss rates of 40% – 80%, the redundancy ratio increases by 82.56%, achieving a tremendous reduction in the retransmission rate.

These results suggest that the redundancy strategy employed by WebRTC lacks adaptability and depends heavily on retransmission, making it unsuitable for high-bitrate videos. Conversely, the proposed AH-FEC succeeds in reducing both redundancy and retransmission rates under identical conditions, demonstrating adaptability to complex channel loss scenarios, especially in weak network environments with high packet loss rates.

7 Conclusions

The FEC scheme employed in WebRTC is unsuitable for high-bitrate video due to limitations in coding efficiency and adaptive capacity. Therefore, we develop a hybrid coding method based on RS/LDPC codes, which determines the coding redundancy according to the receiver redundancy and packet loss rate. When contrasted with the group XOR method in WebRTC, the proposed scheme significantly reduces sending redundancy while ensuring low delay and high recovery rate. We also implement a redundancy decision algorithm based on multi-step packet loss rate prediction, which generates forward-looking redundancy decisions based on feedback from the packet loss rate of the receiver. In comparison to the static table lookup method in WebRTC, this approach can adapt to complex and dynamic packet loss environments. The proposed AH-FEC consistently maintains a high data recovery ratio with the interval change of packet loss rates.

▼Table 2. Performance comparison of two adaptive FEC methods in different packet loss rates

(a) Packet loss ratio: 5% – 20%

	WebRTC	AH-FEC
Data recovery ratio	98.20%	96.49%
Redundancy ratio	95.89%	32.18%

(b) Packet loss ratio: 20% – 40%

	WebRTC	AH-FEC
Data recovery ratio	2.29%	93.54%
Redundancy ratio	100.00%	182.56%

(c) Packet loss ratio: 40% – 80%

	WebRTC	AH-FEC
Data recovery ratio	93.23%	95.29%
Redundancy ratio	100.00%	77.94%

AH-FEC: adaptive hybrid forward error correction

References

[1] CAI S H, ZHAO S C, MA X. Free ride on LDPC coded transmission [J]. IEEE transactions on information theory, 2022, 68(1): 80 – 92. DOI: 10.1109/TIT.2021.3122342

[2] JUN M. Binary polar codes based on bit error probability [C]//Proceedings of IEEE International Symposium on Information Theory (ISIT). IEEE, 2022: 2148 – 2153. DOI: 10.1109/ISIT50566.2022.9834407

[3] XU J L, CHEN W, AI B. Deep Joint source-channel coding based CSI feedback [J]. ZTE Technology journal, 2022, 27(2): 29 – 33. DOI: 10.12142/ZTETJ.202302007

[4] KOTABA R, MANCHÓN C N, BALERCIA T, et al. How URLLC can benefit from NOMA-based retransmissions [J]. IEEE transactions on wireless communications, 2021, 20(3): 1684 – 1699. DOI: 10.1109/TWC.2020.3035517

[5] WANG Y, ZHU Q F. Error control and concealment for video communication: a review [J]. Proceedings of the IEEE, 1998, 86(5): 974 – 997. DOI: 10.1109/5.664283

[6] LI A. RTP Payload Format for generic forward error correction [EB/OL]. [2023-07-10]. https://www.rfc-editor.org/info/rfc5109

[7] MACKAY D J C. Fountain codes [J]. IEE proceedings-communications, 2005, 152(6): 1062 – 1068. DOI: 10.1049/ip-com: 20050237

[8] DEMIR U, AKTAS O. Raptor versus Reed-Solomon forward error correction codes [C]//International Symposium on Computer Networks. IEEE, 2006: 264 – 269. DOI: 10.1109/ISCN.2006.1662545

[9] BOURAS C, KANAKIS N, KOKKINOS V, et al. Evaluating RaptorQ FEC over 3GPP multicast services [C]//The 8th International Wireless Communications and Mobile Computing Conference (IWCMC). IEEE, 2012: 257 – 262. DOI: 10.1109/IWCMC.2012.6314213

[10] WICKER S B, BHARGAVA V K. Reed-Solomon codes and their applications [M]. New York: John Wiley & Sons, 1999

[11] SUDAN M. Decoding of reed Solomon codes beyond the error-correction bound [J]. Journal of complexity, 1997, 13(1): 180 – 193. DOI: 10.1006/jcom.1997.0439

[12] ATIYA A E, YOO S G, CHONG K T, et al. Packet loss rate prediction using the sparse basis prediction model [J]. IEEE transactions on neural networks, 2007, 18(3): 950 – 954. DOI: 10.1109/TNN.2007.891681

[13] EMARA S, FONG S L, LI B C, et al. Low-latency network-adaptive error control for interactive streaming [C]//Proceedings of IEEE Transactions on Multimedia. IEEE, 2021: 1691 – 1706. DOI: 10.1109/TMM.2021.3070134

[14] CHENG S, HU H, ZHANG X G, et al. DeepRS: Deep-learning based network-adaptive FEC for real-time video communications [C]//Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2020: 1 – 5. DOI: 10.1109/ISCAS45731.2020.9180974

[15] HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural computation, 1997, 9(8): 1735 – 1780. DOI: 10.1162/

neco.1997.9.8.1735

[16] HU H, CHENG S, ZHANG X G, et al. LightFEC: network adaptive FEC with a lightweight deep-learning approach [C]//The 29th ACM International Conference on Multimedia. ACM, 2021: 3592 – 3600. DOI: 10.1145/3474085.3475528

[17] WANG R, SI L, HE B F. Sliding-window forward error correction based on reference order for real-time video streaming [J]. IEEE access, 2022, 10: 34288 – 34295. DOI: 10.1109/ACCESS.2022.3162217

[18] HASSLINGER G, HOHLFELD O. The gilbert-elliott model for packet loss in real time services on the Internet [C]//The 14th GI/ITG Conference: Measurement, Modelling and Evaluations of Computer and Communication Systems. IEEE, 2008: 1 – 15

Biographies

XIONG Yuhui is pursuing his master degree at the Research Center of 6G Mobile Communications, School of Cyber Science and Engineering and School of Electronic Information and Communications, Huazhong University of Science and Technology, China. His research interests include multimedia transmission technology and cell-free massive MIMO.

LIU Zhilong is currently the cloud video system chief planning engineer of ZTE Corporation. His main research directions are multimedia transmission technology, SRTN products, cloud desktop products and remote secure office solutions.

XU Lingmin is currently the cloud video product chief planning engineer of ZTE Corporation. His main research directions are cloud computing, IP-based multimedia transmission technology, IPTV/OTT products, cloud desktop products and remote secure office solutions.

HUA Xinhai is currently the Vice President of ZTE Corporation and general manager of the cloud video product project. His main research directions are cloud computing, IP-based video product technology and solutions, security solutions of video service, technology and product solutions of content distribution network, etc.

WANG Zhaoyang is pursuing his PhD degree at the Research Center of 6G Mobile Communications, School of Cyber Science and Engineering and School of Electronic Information and Communications, Huazhong University of Science and Technology, China. His research interests include multimedia transmission technology and cell-free massive MIMO.

BI Ting (ting.bi@ieee.org) is currently an associate professor with the Research Center of 6G Mobile Communications and School of Cyber Science and Engineering, Huazhong University of Science and Technology, China. He received the BE degree in software engineering from Wuhan University, China in 2010, and the ME and PhD degrees in telecommunications from Dublin City University, Ireland in 2011 and 2017, respectively. His research interests include mobile and wireless communications, multimedia and multi-sensory media streaming.

JIANG Tao is currently a Distinguished Professor with the Research Center of 6G Mobile Communications and School of Cyber Science and Engineering, Huazhong University of Science and Technology, China. He received a PhD degree in information and communication engineering from Huazhong University of Science and Technology in 2004. He is/was a symposium technical program committee membership of some major IEEE conferences, including INFOCOM, GLOBECOM, and ICC. He was invited to serve as a TPC Chair for IEEE GLOBECOM 2013, IEEE WCNC 2013, and ICCC 2013. He is/was an associate editor of some technical journals in communications, including the *IEEE Network*, *IEEE Transactions on Signal Processing*, *IEEE Communications Surveys and Tutorials*, *IEEE Transactions on Vehicular Technology*, and he is the area editor of *IEEE Internet of Things Journal* and associate editor-in-chief of *China Communications*.

Waveguide Bragg Grating for Fault Localization in PON

HU Jin[1], LIU Xu[1], ZHU Songlin[2], ZHUANG Yudi[1],
WU Yuejun[1], XIA Xiang[3], HE Zuyuan[1]

(1. School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China;
2. Wireline Product Planning Department, ZTE Corporation, Shanghai 201203, China;
3. Hangzhou Electric Connector Factory, Hangzhou 310052, China)

DOI: 10.12142/ZTECOM.202402012

https://kns.cnki.net/kcms/detail/34.1294.TN.20240605.1755.002.html, published online June 6, 2024

Manuscript received: 2024-03-23

Abstract: Femtosecond laser direct inscription is a technique especially useful for prototyping purposes due to its distinctive advantages such as high fabrication accuracy, true 3D processing flexibility, and no need for mold or photomask. In this paper, we demonstrate the design and fabrication of a planar lightwave circuit (PLC) power splitter encoded with waveguide Bragg gratings (WBG) using a femtosecond laser inscription technique for passive optical network (PON) fault localization application. Both the reflected wavelengths and intervals of WBGs can be conveniently tuned. In the experiment, we succeeded in directly inscribing WBGs in 1×4 PLC splitter chips with a wavelength interval of about 4 nm and an adjustable reflectivity of up to 70% in the C-band. The proposed method is suitable for the prototyping of a PLC splitter encoded with WBG for PON fault localization applications.

Keywords: planar light circuit; power splitter; waveguide Bragg gratings; femtosecond laser; optical network fault localization

Citation (Format 1): HU J, LIU X, ZHU S L, et al. Waveguide Bragg grating for fault localization in PON [J]. ZTE Communications, 2024, 22 (2): 94 – 98. DOI: 10.12142/ZTECOM.202402012

Citation (Format 2): J. Hu, X. Liu, S. L. Zhu, et al., "Waveguide Bragg grating for fault localization in PON," *ZTE Communications*, vol. 22, no. 2, pp. 94 – 98, Jun. 2024. doi: 10.12142/ZTECOM.202402012.

1 Introduction

Fiber-to-the-x (FTTX) technology has made rapid progress in recent decades thanks to the adoption of passive optical network (PON) technology. This technology can effectively reduce the number of fiber channels and eliminate the need for power supply to transmission devices, resulting in low-cost and high-performance solutions.

In a PON system, the network structure is complex, with a large number of users scattered across various locations. More than one-third of network failures are caused by fiber damage[1], which is difficult to locate and repair. As a result, real-time fault localization in the PON system is an important issue that directly impacts the quality of network service and the cost of network maintenance.

Several technologies have been developed for failure detection and localization[2 – 3]. Among them, optical time-domain reflectometry (OTDR) is the most widely used one due to its versatility and convenience. OTDR characterizes fibers using power traces of fiber-backscattered signals, which can be used to extract information and localization related to network failures[4]. However, it is difficult to directly detect failures using OTDR in a point-to-multiple-point topological network due to the superposition of the backward signals. Researchers have improved the conventional OTDR method to distinguish different branches in PON by installing film-type filters as reflectors on optical network units (ONU) and comparing the measured signals with a standard signal[5]. However, this method will inevitably increase the complexity of both the operation and maintenance of ONUs located at the end-user side, which are inconvenient to access and are the most devices in the whole system.

As a result, realizing fault localization before reaching the ONUs is desirable. Fiber Bragg grating (FBG) encoded planar lightwave circuit (PLC) splitters have been proposed to overcome this problem[6 – 7]. Periodic coding schemes have been proposed[8 – 9], which use a pair of FBGs with different reflectance connected by a piece of fiber to generate the periodic codes. A centralized PON fault localization scheme based on optical coding has also been proposed[10], deploying an optical encoder containing a series of FBGs of different wavelengths in front of the user. This scheme achieved a small correlation

This work was supported by the ZTE Industry-University-Institute Fund Project under Grant No. IA20221202011.

distance and a low multiple-customer interference probability. Another FBG-based OTDR scheme involves probing the signal of a tunable OTDR reflected by the FBGs placed in front of the customers. However, the disadvantages of using FBG-based OTDR solutions for PON fault localization are obvious because the fabrication process for FBG arrays or ribboned FBGs is complicated and not suitable for mass production, which implies difficulty in cost reduction. Moreover, the poor correlation characteristic of FBG-based OTDR solutions increases the difficulty of the recognition process[11].

Recently, a remote coding scheme for PON fault localization using waveguide Bragg gratings (WBGs) in power splitters fabricated by PLC technology has been proposed[11]. Multiple cascaded gratings written on the branches of different stages of a PLC-based splitter can be used to generate the corresponding optical codes. Fig. 1 shows how the PLC chips integrated with WBGs realize the remote coding for the PON fault localization system. The first stage splitters reflect four different wavelengths of λ_1, λ_2, λ_3, and λ_4, and the second stage splitters reflect four wavelengths of λ_5, λ_6, λ_7, and λ_8, resulting in 16 different combinations such as $\lambda_1\lambda_7$ and $\lambda_3\lambda_8$ to determine the optical path where the ONU is located. The key advantage of using WBGs encoded power splitters for PON fault localization is their potential for mass production at the wafer level. This is the most effective way to cut cost, which is the most critical point for PON applications. There are other advantages such as easy installation and maintenance, compact size, and no need for additional devices or components. However, the fabrication of complex Bragg gratings on chip-level or even wafer-level PLC devices using conventional UV exposure assisted with hydrogen loading technique is difficult, high-cost, and time-consuming for prototype device fabrication and system-level validation tests.

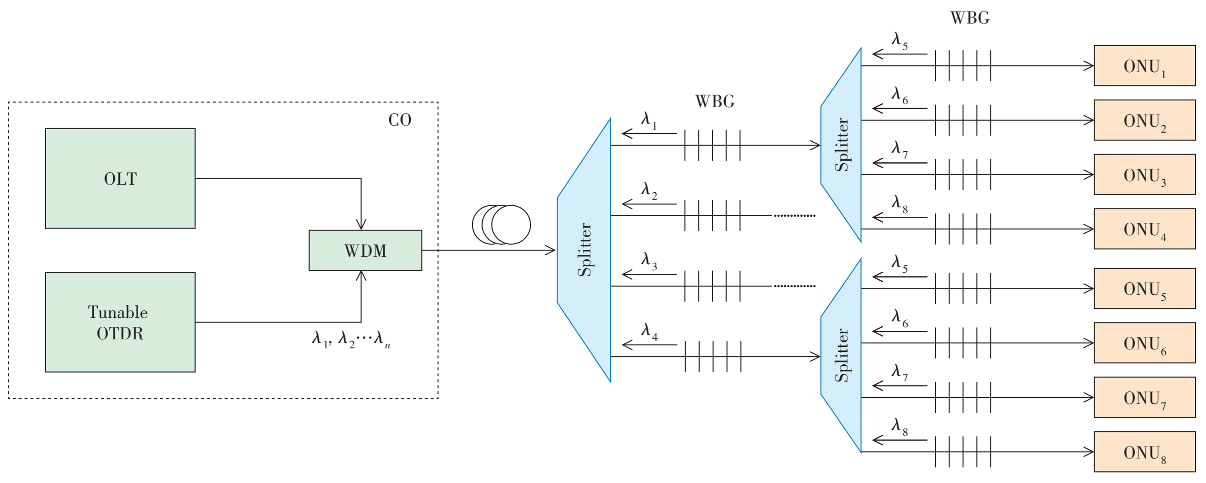
In this paper, we propose a femtosecond laser direct inscription technique for the fabrication of prototype PLC splitters encoded with WBGs for passive optical network fault localization applications. Reflected wavelengths, their intervals of WBGs, and reflectance can be conveniently tuned by adjusting parameters such as period, length, and refractive index modulation of the WBGs. In the experiment, we succeeded in directly inscribing WBGs in the 1×4 PLC splitter chips with a wavelength interval of about 4 nm and an adjustable reflectance of up to 70% in the C-band. The proposed method is suitable for prototyping PLC splitters encoded with WBGs for PON fault localization applications.

2 Fabrication and Evaluation

We utilized customized single-channel PLC waveguide chips and 4-channel PLC chips with splitter structures for our experiments. The dimensions of four-channel PLC chips are about 30 mm in length, 3 mm in width, and 2.5 mm in height. There is a comparatively long straight PLC waveguide after the splitter structure and a partly removed cover glass, for waveguide Bragg grating inscription purposes.

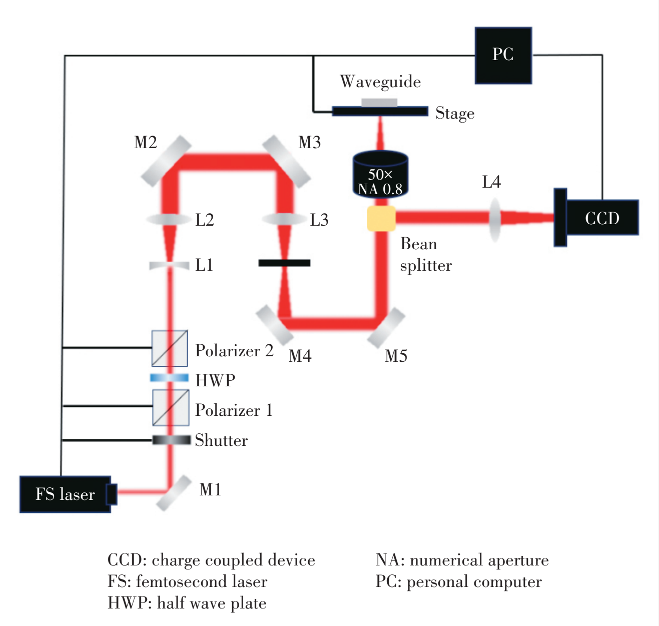
The schematic of the femtosecond laser process system and its picture for WBG inscription in PLC chips are shown in Figs. 2 and 3, respectively. The femtosecond laser operates at a wavelength of 515 nm with a pulse duration of 350 fs and a repeating rate of 25 kHz. During inscription, the laser moves at a speed of 50 μm/s. The incident femtosecond laser light is reflected by a total reflection mirror and focused into the center of the waveguide structure of the PLC chip through a 50× objective lens. By optimizing the moving speed, pattern and distance of the displacement stage, laser light can scan over the waveguide and inscribe desirable Bragg gratings on the waveguide in PLC chips.

The pitch Λ of WBG can be represented by $\lambda_B = 2n_{eff}\Lambda$, where λ_B is the Bragg wavelength and n_{eff} is the effective re-



CO: central office
OLT: optical line terminal
ONU: optical network unit
OTDR: optical time-domain reflectometry
PLC: planar lightwave circuit
WBG: waveguide Bragg grating
WDM: wavelength division multiplexer

▲Figure 1. Passive optical network (PON) fault localization using PLC splitter encoded with WBG



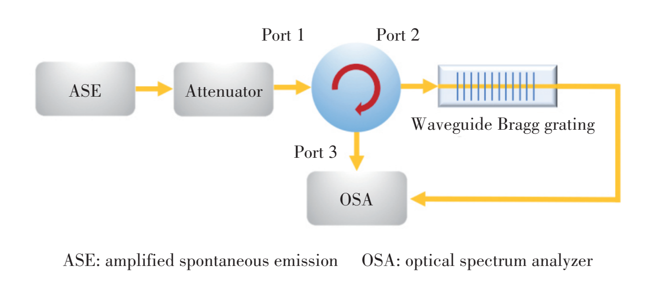
▲ **Figure 2. Femtosecond laser process system for waveguide Bragg grating (WBG) inscription in planar lightwave circuit (PLC) chips**

▲ **Figure 3. Picture of femtosecond laser process system**

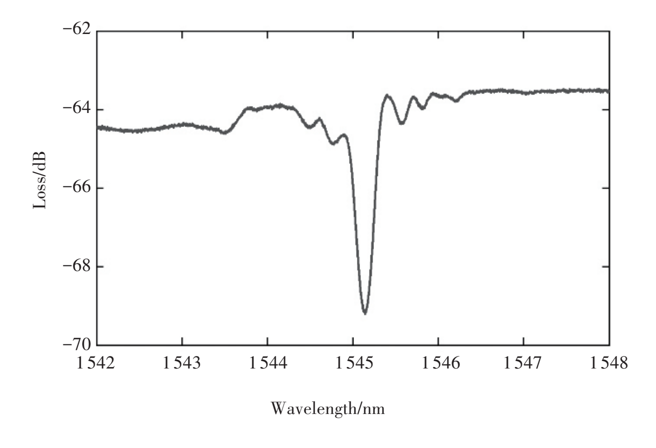
fractive index of the waveguide in the PLC splitter. In our case, the periods of the waveguide Bragg gratings inscribed on channels 1, 2, 3, and 4 are 1.609 μm, 1.605 μm, 1.601 μm and 1.597 μm, respectively. The length of the inscribed grating is 3 000 periods, which is approximately 4 800 μm. During the inscription process, the transmitted spectra of the waveguide could be observed in real time to optimize the inscription parameters.

By optimizing both displacement and aberration correction, we successfully fabricated single-channel WBG and 4-channel WBG in PLC splitters, respectively, with different reflected wavelengths and differently designed reflectance.

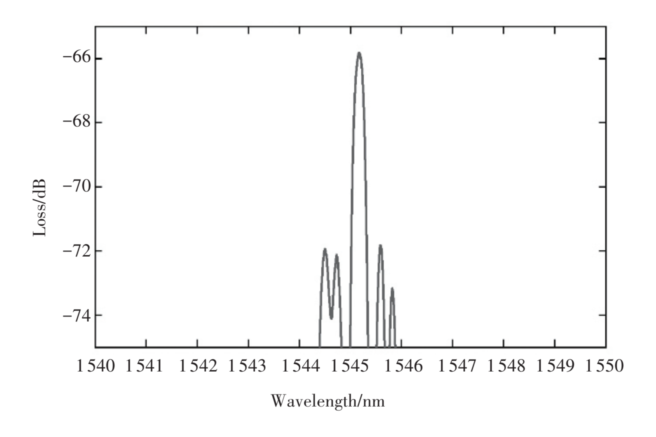
The measurement setup for transmitted and reflected spectra is shown in Fig. 4. The transmitted and reflected spectra can be obtained from port 2 and port 3, respectively. The measured transmitted and reflected spectrum of the single-channel WBG is shown in Figs. 5 and 6. The reflected wavelength of WBG is 1 545.1 nm with a reflectance of about 70%, and its 3 dB bandwidth is about 0.3 nm. The main parameters and their transmitted spectra of the four-channel PLC splitter are shown in Table 1 and Fig. 7. The reflected wavelengths of the four channels are intended to be 1 552 nm, 1 548 nm, 1 544 nm, and 1 540 nm respectively, with a negligible wavelength shift up to 0.5 nm. The reflectivity of Channels 1 and 2 is about 30%, while that of Channels 3 and 4 is about 40%. Their 3 dB bandwidths are varying from 0.5 nm to 0.9 nm. The reflected wavelength of the


Port 1
Port 2
ASE
Attenuator
Waveguide Bragg grating
Port 3
OSA
ASE: amplified spontaneous emission
OSA: optical spectrum analyzer

▲ **Figure 4. Measurement setup for transmitted and reflected optical spectra**



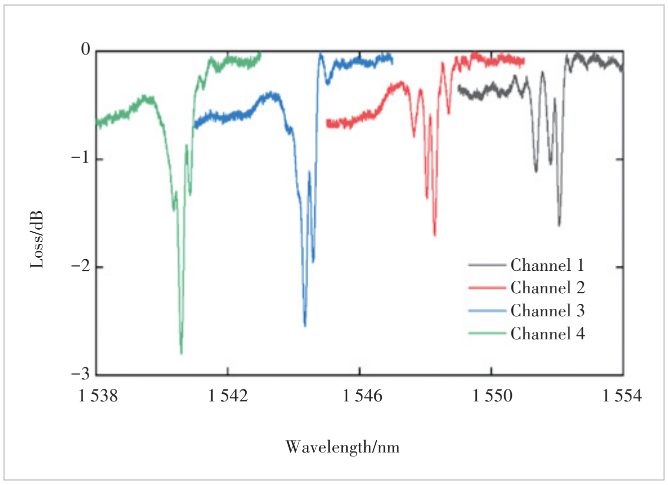
▲ **Figure 5. Measured transmitted spectrum of single-channel waveguide Bragg grating (WBG)**



▲ **Figure 6. Measured reflected spectrum of single-channel waveguide Bragg grating (WBG)**

▼Table 1. Measured results of four-channel PLC splitter

Channel No.	Reflectance/%	Reflected Wavelength/nm	3 dB Bandwidth/nm
1	28	1 552.2	0.8
2	29	1 548.3	0.5
3	40	1 544.4	0.9
4	45	1 540.7	0.5

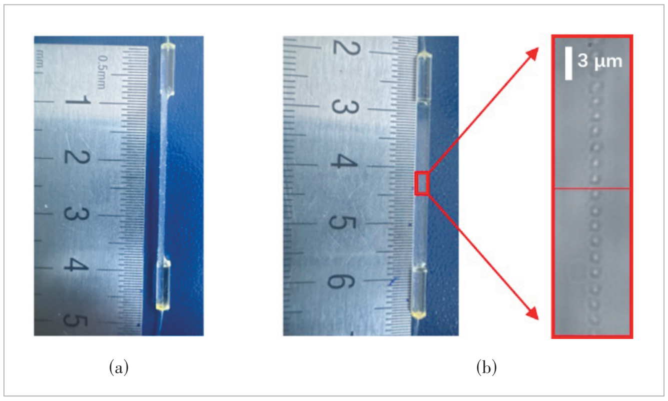


▲ **Figure 7. Measured transmitted spectra of 4-channel waveguide Bragg grating (WBG) splitter**

single-channel PLC waveguide remains the same at 1 545.1 nm and the measured fluctuation in reflectance is within 0.2 dB under different polarization states, which implies that WBG is not sensitive to polarization. Pictures of the fabricated prototype PLC splitter encoded with WBG are shown in Fig. 8. A microscope graph of WBG structure is also shown in the inset of Fig. 8(b).

3 Conclusions

We implement a femtosecond laser direct inscription technique to fabricate prototype PLC splitters integrated with WBGs for PON fault localization application. By manipulating parameters such as the period, length, and refractive index modulation of the WBGs, we can effectively control the reflected wavelengths and their intervals. Our experimental results demonstrate the capability to directly inscribe WBGs into PLC splitter chips, achieving a wavelength interval of approximately 4 nm and a reflectance of up to 70% in the C-band. This method is suitable for prototyping PLC splitters encoded with WBGs for PON fault localization.



(a)

(b)

▲ **Figure 8. Fabricated prototype waveguide Bragg grating (WBG) in planar lightwave circuit (PLC) chip: (a) side-view; (b) top-view**

References

[1] RAD M M, FOULI K, FATHALLAH H A, et al. Passive optical network monitoring: challenges and requirements [J]. IEEE communications magazine, 2011, 49(2): s45 – s52. DOI: 10.1109/MCOM.2011.5706313

[2] URBAN P J, DAHLFORT S. Cost-efficient remote PON monitoring based on OTDR measurement and OTM functionality [C]//Proc. 13th International Conference on Transparent Optical Networks. IEEE, 2011: 1 – 4. DOI: 10.1109/ICTON.2011.5970975

[3] ZHANG S C, JI W, LI X, et al. Efficient and reliable protection mechanism in long-reach PON [J]. Journal of optical communications and networking, 2016, 8(1): 23 – 32. DOI: 10.1364/jocn.8.000023

[4] YUKSEL K, MOEYAERT V, WUILPART M, et al. Optical layer monitoring in passive optical networks (PONs): a review [C]//Proc. 10th Anniversary International Conference on Transparent Optical Networks. IEEE, 2008. DOI: 10.1109/icton.2008.4598379

[5] ITU-T. Optical fibre maintenance depending on topologies of access networks: ITU-T L.310 [S]. 2016

[6] RAD M M, PENON J, FATHALLAH H A, et al. Probing the limits of PON monitoring using periodic coding technology [J]. Journal of lightwave technology, 2011, 29(9): 1375 – 1382. DOI: 10.1109/JLT.2011.2125946

[7] GE Z Q, LV T, YE X K, et al. Adaptive design for 2D optical coding PON link health detection system in complex environment [J]. Journal of lightwave technology, 2020, 38(23): 6458 – 6464. DOI: 10.1109/JLT.2020.3013276

[8] RAD M M, FATHALLAH H A, MAIER M, et al. A novel pulse-positioned coding scheme for fiber fault monitoring of a PON [J]. IEEE communications letters, 2011, 15(9): 1007 – 1009. DOI: 10.1109/LCOMM.2011.070711.111006

[9] FERNÁNDEZ M P, COSTANZO CASO P A, BULUS ROSSINI L A. False detections in an optical coding-based PON monitoring scheme [J]. IEEE photonics technology letters, 2017, 29(10): 802 – 805. DOI: 10.1109/LPT.2017.2686000

[10] ZHOU X, ZHANG F D, SUN X H. Centralized PON monitoring scheme based on optical coding [J]. IEEE photonics technology letters, 2013, 25 (9): 795 – 797. DOI: 10.1109/LPT.2013.2251622

[11] ZHANG X, LU F J, CHEN S, et al. Remote coding scheme based on waveguide Bragg grating in PLC splitter chip for PON monitoring [J]. Optics express, 2016, 24(5): 4351 – 4364. DOI: 10.1364/oe.24.004351

Biographies

HU Jin received his BS degree in electronic information science and technology from Nanjing University, China in 2020, and MS degree in electronic communication engineering from Shanghai Jiao Tong University, China in 2024, focusing on the research of novel planar lightwave circuit-based devices for next generation optical communications system.

LIU Xu received his BS degree in Internet of Things engineering from University of Electronic Science and Technology of China in 2020. He is currently working toward his PhD degree at Shanghai Jiao Tong University, China. His research interests include optical interconnects and polymer optical waveguide devices.

ZHU Songlin (zhu.songlin@zte.com.cn) received his MS degree in theoretical physics from Hangzhou University, China in 1998 and PhD degree in electronic science and technology from Zhejiang University, China in 2001. He joined the Wireline Product Planning Department, ZTE Corporation in 2001. His research interests include FTTx technology for optical communications and OTDR technology for optical sensing applications.

ZHUANG Yudi received her MS degree in electrical engineering from Illinois Institute of Technology, USA in 2014. She is currently an engineer of the Department of Electronic Engineering, Shanghai Jiao Tong University, China. Her research interests include specialty optical fibers and waveguides for optical communications and sensing application.

WU Yuejun received his MS degree in measurement technology and instruments from Shanghai University, China in 2002. In the subsequent years, he joined Philips (China) Semiconductor, Atmel (Shanghai) Semiconductor and Intel (Asia) R&D center, where he worked as a senior staff engineer/application manager until 2020. Currently he is the Laboratory Director of School of Sensing Science and Engineering, Shanghai Jiao Tong University, China. His research interests include information detection and control, intelligent instruments and sensing application.

XIA Xiang received his PhD degree in optical engineering from Zhejiang University, China in 2016. He joined Hangzhou Electric Connector Factory in 2017, primarily responsible for the research and development of optical coating and planar lightwave circuit products. He has a wealth of experience in the end-to-end process of passive optical chips, encompassing design, development, fabrication, and packaging. His main research interests include planar lightwave circuits and thin film filter for optical communication systems.

HE Zuyuan received his BS and MS degrees in electronic engineering from Shanghai Jiao Tong University, China in 1984 and 1987, respectively, and PhD degree in photonics from the University of Tokyo, Japan in 1999. He joined CIENA Corporation, Linthicum, USA in 2001, as a lead engineer heading the optical testing and optical process development group. He returned to the University of Tokyo as a lecturer in 2003, and then became an associate professor in 2005 and a full professor in 2010. He is now a Chair Professor and the head of Department of Electronic Engineering, Shanghai Jiao Tong University. His current research interests include optical fiber sensors, specialty optical fibers, and optical interconnects.

Cooperative Distributed Beamforming Design for Multi-RIS Aided Cell-Free Systems

ZHU Yuting[1], XU Zhiyu[2], ZHANG Hongtao[1]

(1. Key Lab of Universal Wireless Communications, Ministry of Education of China, Beijing University of Posts and Telecommunications, Beijing 100876, China；
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202402013

https://kns.cnki.net/kcms/detail/34.1294.TN.20240509.1550.002.html, published online May 10, 2024

Manuscript received: 2023-12-19

Abstract: Cell-free systems significantly improve network capacity by enabling joint user service without cell boundaries, eliminating inter-cell interference. However, to satisfy further capacity demands, it leads to high-cost problems of both hardware and power consumption. In this paper, we investigate multiple reconfigurable intelligent surfaces (RISs) aided cell-free systems where RISs are introduced to improve spectrum efficiency in an energy-efficient way. To overcome the centralized high complexity and avoid frequent information exchanges, a cooperative distributed beamforming design is proposed to maximize the weighted sum-rate performance. In particular, the alternating optimization method is utilized with the distributed closed-form solution of active beamforming being derived locally at access points, and phase shifts are obtained centrally based on the Riemannian conjugate gradient (RCG) manifold method. Simulation results verify the effectiveness of the proposed design whose performance is comparable to the centralized scheme and show great superiority of the RISs-aided system over the conventional cellular and cell-free system.

Keywords: cell-free systems; reconfigurable intelligent surface; cooperative distributed beamforming; Riemannian conjugate gradient

Citation (Format 1): ZHU Y T, XU Z Y, ZHANG H T. Cooperative distributed beamforming design for multi-RIS aided cell-free systems [J]. ZTE Communications, 2024, 22(2): 99 – 106. DOI: 10.12142/ZTECOM.202402013

Citation (Format 2): Y. T. Zhu, Z. Y. Xu, and H. T. Zhang, "Cooperative distributed beamforming design for multi-RIS aided cell-free systems," *ZTE Communications*, vol. 22, no. 2, pp. 99 – 106, Jun. 2024. doi: 10.12142/ZTECOM.202402013.

1 Introduction

To satisfy the ever-increasing demands for massively-connected, high-throughput, and energy-efficient communications, several promising technologies have emerged and been discussed in 5G communication standards, including massive multiple-input multiple-output (MIMO)[1], millimeter-wave communications[2], and ultra-dense networks (UDNs)[3]. Among them, massive MIMO and UDN both aim at increasing the number of antennas or the deployment of small base stations (BSs) in a cell-centric way to achieve high capacity. However, the performance of multi-cell MIMO architecture suffers from inter-cell interference with the concept of cell boundaries[4]. To address this problem, a new architecture named the cell-free network has been proposed in a user-centric paradigm, where all access points (APs) in the network coordinate with each other to serve all users in the network simultaneously[5 – 6]. By deploying a mass of low-cost APs across the network and through effective cooperation among APs, cell-free networks achieve high-capacity coverage and diversity enhancement. However, when it comes to further capacity improvement, both hardware and power consumption require high costs, which cannot be ignored for next-generation communications.

Fortunately, reconfigurable intelligent surface (RIS) emerges as a key candidate technology for future 6G wireless systems[7 – 8]. Different from the traditional ways of antennas or BS densification, RIS, which comprises numerous low-cost passive reflecting elements, provides an energy-efficient alternative to improve the system spectrum efficiency by adjusting the phase shifts of its elements smartly, while being free from radio frequency chains and amplifiers. With the ability to manipulate the incident electromagnetic signals, RIS can be used to improve the channel rank[9], extend the coverage area[10], enhance the desired signals at the users, and constructively mitigate the undesired signals at unintended users. By introducing RIS into the cell-free network, higher spectrum and energy efficiency can be achieved with less power consumption[11 – 12].

Several research works have been devoted to jointly optimizing the transmit beamforming at the AP and the phase shifts at the RIS to guarantee the performance gain, including single-cell[13 – 14] and multi-cell[10] scenes. Additionally, in Ref. [15], the authors first added an ariel RIS to a cell-free system and proposed an iterative optimization algorithm to maximize the achievable rate of the user by the power allocation and the beamforming vector. In Ref. [16], the channel estimate scheme was investigated for RIS-assisted cell-free systems under spatially correlated channels. While the above works only considered the system with a single RIS, authors in Ref. [11] studied the sum-rate optimizing problem with multiple RISs in a centralized beamforming scheme. However, with the increase of the network scale, it is very intractable to collect all the instantaneous channel state information (CSI) and compute the high-dimensional information. Later, in Ref. [17], a fully decentralized design framework was proposed to incrementally and locally update the beamformers. However, in the presence of RISs, multiple iterations are required to reach a consensus for the phase shift design due to the coupling effect of the active beamformer and phase shift design. Despite reducing the complexity, it increases signaling exchange among APs and the processor cost for APs, introducing more potential delay and errors for CSI.

Inspired by Ref. [18], we propose a distributed framework for cooperative beamforming and phase shift design in RISs-aided cell-free systems. In order to avoid the drawbacks of centralized high-dimensional CSI exchange and high CPU processing complexity, as well as the issues of frequent CSI exchange and latency time among fully distributed APs, we leverage the centralized processing capability of the CPU to optimize the high-dimensional phase shifts brought by multiple RISs to improve system capacity, while each AP only locally optimizes the small-scale active beamforming. The main contributions of this work are summarized as follows. 1) A cooperative distributed beamforming design framework is proposed for the multi-RIS aided cell-free system, showing lower complexity and comparable spectrum efficiency to the centralized framework in Ref. [11]. 2) A weighted sum-rate (WSR) maximization problem for the cooperative distributed scheme is formulated, subject to transmit power constraints at APs and unit-modulus constraints of RIS. By employing the alternating optimization framework to decompose the nonconvex problem, we innovatively derive a closed-form distributed solution to active beamforming, while the effective Riemannian conjugate gradient (RCG) algorithm is adopted to deal with the phase shifts of multiple RISs under unit-modulus constraints, and the discrete phase-shift case is additionally discussed. 3) Simulation results demonstrate the superior performance of the multi-RIS aided cell-free system compared with the traditional cellular network and the conventional cell-free system, and verify the effectiveness of the proposed low-complexity design.

The rest of this paper is organized as follows. Section 2 presents the system model and the formulation of the discussed problem. Section 3 introduces the proposed cooperative distributed beamforming design. Section 4 provides the numerical results to discuss the performance of the proposed design. Finally, we conclude this paper in Section 5.

2 System Model and Problem Formulation

In this paper, we consider a downlink RISs-aided cell-free system, where multiple distributed APs (each equipped with M_t transmit antennas) cooperatively serve K users (each equipped with M_r receive antennas) with the aid of multiple RISs. All RISs are controlled by the CPU through wired or wireless control, while all APs are connected to the CPU by the backhaul link. The CPU is deployed for joint planning and control, which coordinates APs and RISs. We denote the index sets of APs, RISs, users and RIS reflecting elements as $\mathcal{B} = \{1,\cdots,B\}$, $\mathcal{L} = \{1,\cdots,L\}$, $\mathcal{K} = \{1,\cdots,K\}$ and $\mathcal{N} = \{1,\cdots,N\}$, respectively.

2.1 Transmitters

In the proposed cell-free network, all APs cooperate to serve all users by coherent transmission. Let $s_k, \forall k \in \mathcal{K}$ denote the transmitted symbol for the k-th user, satisfying $\mathbb{E}\left\{\left|s_k\right|^2\right\} = 1$. Then, the transmitted signal at the b-th AP is given as

$$\boldsymbol{x}_b = \sum_{k=1}^{K} \boldsymbol{w}_{b,k} s_k, \forall b \in \mathcal{B}, \tag{1}$$

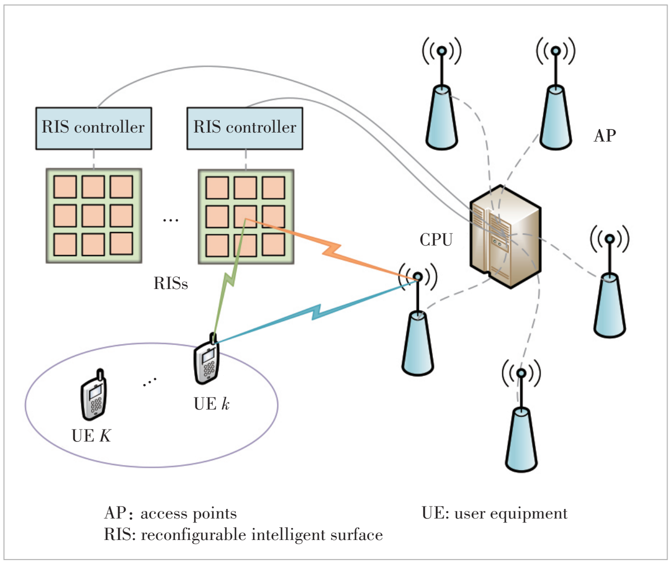
where $\boldsymbol{w}_{b,k} \in \mathbb{C}^{M_t \times 1}$ denotes the corresponding active beamforming vector designed for the k-th user at the b-th AP. The beamforming vectors satisfy the transmit power constraint $\sum_{k=1}^{K} \left\| \boldsymbol{w}_{b,k} \right\|^2 \leqslant P_{b,\max}$, where $P_{b,\max}$ denotes the power budget of the AP b.

2.2 Channel Model

Let $\boldsymbol{H}_{b,k}^H \in \mathbb{C}^{M_r \times M_t}$, $\boldsymbol{H}_{r,l,k}^H \in \mathbb{C}^{M_r \times N}$ and $\boldsymbol{G}_{b,l} \in \mathbb{C}^{N \times M_t}$ denote the complex equivalent baseband channel matrix between the b-th AP and the k-th user, between the l-th RIS and the k-th user, and between the b-th AP and the l-th RIS, respectively, $\forall b \in \mathcal{B}, \forall k \in \mathcal{K}$, and $\forall l \in \mathcal{L}$. We assume that the CSI of all the links can be perfectly known at the AP via the channel acquisition method[19]. Denote the phase shift of the n-th reflection element of the l-th RIS by $\theta_n^l \in [0, 2\pi]$. Then, by defining $\boldsymbol{\Phi}_l \triangleq \mathrm{diag}\left\{\phi_{l,1},\cdots,\phi_{l,N}\right\}, \forall l \in \mathcal{L}$, where $\phi_{l,n} = e^{j\theta_n^l}$, the received signal at the k-th UE can be expressed and simplified as:

$$\begin{aligned} \boldsymbol{y}_k = \sum_{b=1}^{B}\sum_{m=1}^{K}\left(\boldsymbol{H}_{b,k}^H + \sum_{l=1}^{L}\boldsymbol{H}_{\mathrm{r},l,k}^H \boldsymbol{\Phi}_l \boldsymbol{G}_{b,l}\right)\boldsymbol{w}_{b,m} s_m + \boldsymbol{n}_k \overset{(a)}{=} \\ \sum_{b=1}^{B}\sum_{m=1}^{K}\left(\boldsymbol{H}_{b,k}^H + \boldsymbol{H}_{\mathrm{r},k}^H \boldsymbol{\Phi}\, \boldsymbol{G}_b\right)\boldsymbol{w}_{b,m} s_m + \boldsymbol{n}_k \overset{(b)}{=} \\ \sum_{b=1}^{B}\sum_{m=1}^{K}\bar{\boldsymbol{H}}_{b,k}^H \boldsymbol{w}_{b,m} s_m + \boldsymbol{n}_k, \end{aligned} \tag{2}$$

where (a) holds by defining $\boldsymbol{\Phi} = \mathrm{diag}\left(\boldsymbol{\Phi}_1,\cdots,\boldsymbol{\Phi}_L\right)$, $\boldsymbol{H}_{\mathrm{r},k} = \left[\boldsymbol{H}_{\mathrm{r},1,k}^T,\cdots,\boldsymbol{H}_{\mathrm{r},L,k}^T\right]^T$, and $\boldsymbol{G}_b = \left[\boldsymbol{G}_{b,l}^T,\cdots,\boldsymbol{G}_{b,L}^T\right]^T$, and (b) holds by defining



▲**Figure 1. Downlink transmission in the multi-RIS aided cell-free system**

$$\bar{\boldsymbol{H}}_{b,k}^{H} = \boldsymbol{H}_{b,k}^{H} + \sum_{l=1}^{L} \boldsymbol{H}_{\mathrm{r},l,k}^{H} \boldsymbol{\Phi}_{l} \boldsymbol{G}_{b,l}, \tag{3}$$

and $\boldsymbol{n}_k \sim \mathcal{CN}\left(\boldsymbol{0},\sigma^2 \boldsymbol{I}_{M_r}\right)$ denotes the noise at the k-th user following the Gaussian distribution. Then, the achievable data rate of user k can be given by:

$$R_k = \log\det\left(\boldsymbol{I} + \left(\sum_{b=1}^{B} \bar{\boldsymbol{H}}_{b,k}^{H} \boldsymbol{w}_{b,k}\right)\left(\sum_{b=1}^{B} \boldsymbol{w}_{b,k}^{H} \bar{\boldsymbol{H}}_{b,k}\right) \boldsymbol{Q}_k^{-1}\right), \tag{4}$$

where $\boldsymbol{Q}_k = \sum_{m=1,m\neq k}^{K} \left(\sum_{b=1}^{B} \bar{\boldsymbol{H}}_{b,k}^{H} \boldsymbol{w}_{b,m}\right)\left(\sum_{b=1}^{B} \boldsymbol{w}_{b,m}^{H} \bar{\boldsymbol{H}}_{b,k}\right) + \sigma^2 \boldsymbol{I}_{M_r}$.

2.3 Problem Formulation

In this paper, we aim at maximizing the WSR of the RISs-aided cell-free system by jointly optimizing the AP transmit beamforming $\boldsymbol{W}$ and phase shift matrix $\boldsymbol{\Phi}$, with the WSR written as

$$R_{\mathrm{sum}} = \sum_{k=1}^{K} \omega_k R_k, \tag{5}$$

where $\omega_k \in \mathbb{R}^{+}$ is a weighting factor representing the priority for user k.

Then, subject to the AP transmit power constraint and the unit-modulus constraints of RIS elements, the optimization problem can be expressed as

$$\begin{aligned} &\max_{\boldsymbol{W},\boldsymbol{\Phi}} \; R_{\mathrm{sum}} \\ &\text{s.t.} \; \sum_{k=1}^{K} \left\| \boldsymbol{w}_{b,k} \right\|^2 \leqslant P_{b,\max}, \forall b \in \mathcal{B}, \\ &\quad \theta_n^l \in \mathcal{F}, \forall l \in \mathcal{L}, \forall n \in \mathcal{N}, \end{aligned} \tag{6}$$

where $\boldsymbol{W} \triangleq \left[\boldsymbol{w}_1,\cdots,\boldsymbol{w}_K\right] \in \mathbb{C}^{BM_t \times K}$, and $\boldsymbol{w}_k = \left[\boldsymbol{w}_{1,k}^{T},\cdots,\boldsymbol{w}_{B,k}^{T}\right]^{T}$. Here, we assume $\mathcal{F} \triangleq \left\{\theta_n^l \middle| \left|e^{j\theta_n^l}\right| = 1\right\}, \forall l \in \mathcal{L}, \forall n \in \mathcal{N}$, and we will also discuss the design under the discrete phase shift constraints as follows. Apparently, due to the non-convex complex objective function and the unit-modulus constraint in Problem (6), the optimization of the phase shift matrix $\boldsymbol{\Phi}$ and the active beamforming matrix $\boldsymbol{W}$ is very challenging.

3 Proposed Cooperative Distributed Beamforming Design

To avoid the centralized overwhelming computation, we propose a cooperative distributed beamforming design for solving Problem (6), since the constraints are distributed. Meanwhile, due to the large dimension of variables and the coupling effect of active beamformer and phase shift design, the full distributed framework will lead to extensive CSI exchange among APs and even more to reach a consensus on the phase shift design. Therefore, in the proposed design, we take full advantage of the centralized processing of the CPU to optimize the high-dimensional $\boldsymbol{\Phi}$, while the active beamformers are computed locally by each AP, in a cooperative distributed way.

In the following, the alternating optimization approach is adopted to address the joint optimization problem, which is decomposed into the active beamforming and the phase optimization subproblems.

3.1 Reformulation of the Original Problem

By exploiting the equivalence of the sum-rate maximization problem and the weighted mean-square error (MSE) minimization problem[20], the original non-convex problem can be reformulated into a more tractable form. First, considering a linear receiver filter $\boldsymbol{u}_k \in \mathbb{C}^{M_r \times 1}$, the estimated signal vector of each user is given by $\hat{\boldsymbol{s}}_k = \boldsymbol{u}_k^H \boldsymbol{y}_k, \forall k \in \mathcal{K}$. Then, under the independence assumption of the signal and the noise, the MSE matrix can be written as

$$\begin{aligned} &\mathrm{mse}_k \triangleq \mathbb{E}_{\mathbf{s},\mathbf{n}}\left[\left(\hat{\boldsymbol{s}}_k - \boldsymbol{s}_k\right)\left(\hat{\boldsymbol{s}}_k - \boldsymbol{s}_k\right)^H\right] = \\ &\left(\boldsymbol{u}_k^H \sum_{b=1}^{B} \bar{\boldsymbol{H}}_{b,k}^{H} \boldsymbol{w}_{b,k} - 1\right)\left(\boldsymbol{u}_k^H \sum_{b=1}^{B} \bar{\boldsymbol{H}}_{b,k}^{H} \boldsymbol{w}_{b,k} - 1\right)^H + \\ &\boldsymbol{u}_k^H\left(\sum_{m=1,m\neq k}^{K} \left(\sum_{b=1}^{B} \bar{\boldsymbol{H}}_{b,k}^{H} \boldsymbol{w}_{b,m}\right)\left(\sum_{b=1}^{B} \boldsymbol{w}_{b,m}^{H} \bar{\boldsymbol{H}}_{b,k}\right) + \sigma_k^2 \boldsymbol{I}_{M_r}\right)\boldsymbol{u}_k, \forall k \in \mathcal{K} \; . \end{aligned} \tag{7}$$

By introducing a set of auxiliary matrices $\boldsymbol{f} = \left\{f_k, \forall k\right\}$, Problem (6) can be reformulated as follows[20]:

$$\begin{aligned} &\max_{\boldsymbol{W},\boldsymbol{u},\boldsymbol{f},\boldsymbol{\Phi}} \; \sum_{k=1}^{K} \omega_k\left(\log\left(f_k\right) - f_k \mathrm{mse}_k + 1\right), \\ &\text{s.t.} \; \sum_{k=1}^{K} \left\| \boldsymbol{w}_{b,k} \right\|^2 \leqslant P_{b,\max}, \forall b \in \mathcal{B}, \\ &\quad \theta_n^l \in \mathcal{F}, \forall l \in \mathcal{L}, \forall n \in \mathcal{N}. \end{aligned} \tag{8}$$

3.2 Optimizing Active Beamforming

Fixing all of the auxiliary matrices $\boldsymbol{u},\boldsymbol{f}$ and the phase shift matrix of RISs $\boldsymbol{\Phi}$, we can rewrite the active beamforming optimization problem as:

$$\begin{aligned}&\min_{\boldsymbol{W}} \ \sum_{k=1}^{K} \omega_k f_k \mathrm{mse}_k \\ &\text{s.t.} \ \sum_{k=1}^{K} \left\| \boldsymbol{w}_{b,k} \right\|^2 \leqslant P_{b,\max}, \forall b \in \mathcal{B}.\end{aligned} \tag{9}$$

By substituting mse_k in Eq. (7) into the objective function in Problem (9) and ignoring the unrelated constant terms, we simplify the above optimization problem as

$$\begin{aligned}&\min_{\boldsymbol{W}} \ \mathrm{Tr}\left(\boldsymbol{W}^H \boldsymbol{V} \boldsymbol{W}\right) - 2\Re\left\{\mathrm{Tr}\left(\boldsymbol{Q}^H \boldsymbol{W}\right)\right\} \\ &\text{s.t.} \ \sum_{k=1}^{K} \left\| \boldsymbol{w}_{b,k} \right\|^2 \leqslant P_{b,\max}, \forall b \in \mathcal{B},\end{aligned} \tag{10}$$

where

$$\boldsymbol{V} \triangleq \begin{pmatrix} \boldsymbol{V}_{1,1} & \cdots & \boldsymbol{V}_{1,B} \\ \vdots & \ddots & \vdots \\ \boldsymbol{V}_{B,1} & \cdots & \boldsymbol{V}_{B,B} \end{pmatrix}, \tag{11}$$

$$\boldsymbol{V}_{bb'} \triangleq \sum\nolimits_{k=1}^{K} \omega_k f_k \bar{\boldsymbol{H}}_{b,k} \boldsymbol{u}_k \boldsymbol{u}_k^H \bar{\boldsymbol{H}}_{b',k}^H, \tag{12}$$

$$\boldsymbol{Q} \triangleq \left[\boldsymbol{q}_1, \cdots, \boldsymbol{q}_K\right], \tag{13}$$

$$\boldsymbol{q}_k \triangleq \left[\boldsymbol{q}_{1,k}^T, \boldsymbol{q}_{2,k}^T, \cdots, \boldsymbol{q}_{B,k}^T\right]^T, \tag{14}$$

$$\boldsymbol{q}_{b,k} \triangleq \omega_k f_k \bar{\boldsymbol{H}}_{b,k} \boldsymbol{u}_k, \tag{15}$$

and $\Re\{\cdot\}$ denotes the real part of its argument. We can observe that Problem (10) is a standard quadratically constrained quadratic program (QCQP) problem, which can be optimally solved by many existing methods such as the alternating direction method of multipliers (ADMM) and the standard convex tools[11]. However, these centralized methods contribute to high computational complexity. Here, with the power budget constraint, we provide a closed-form distributed solution by introducing the Lagrange multipliers method. According to the first-order optimal condition for each AP b and each user k, we can obtain

$$\boldsymbol{w}_{b,k}^{\mathrm{opt}} = \left(\boldsymbol{V}_{bb} + \lambda_b \boldsymbol{I}_{M_t}\right)^{-1}\left(\boldsymbol{q}_{b,k} - \boldsymbol{\xi}_{b,k}\right), \tag{16}$$

where λ_b is the introduced Lagrange multiplier updated via the bisection method. $\boldsymbol{\xi}_{b,k} \triangleq \sum\nolimits_{b' \in \mathcal{B} \backslash \{b\}} \boldsymbol{V}_{bb'} \boldsymbol{w}_{b',k}$, which implies the information about the channel between AP b and the other APs, and about the beamforming designs adopted by the other APs for user k. Moreover, in the distributed design, each AP locally computes its beamformer $\boldsymbol{w}_{b,k}$ in parallel with the other APs. So based on the fixed $\boldsymbol{\xi}_{b,k}$, each AP updates its beamformer vector at iteration t as

$$\boldsymbol{w}_{b,k}^{(t)} = (1-\alpha)\boldsymbol{w}_{b,k}^{(t-1)} + \alpha \boldsymbol{w}_{b,k}^{\mathrm{opt}}, \tag{17}$$

where $\alpha \in (0,1]$. The update in Eq. (17) is to limit the variation of the precoding vectors between consecutive iterations, where the step size α needs to be chosen properly to strike a balance between convergence speed and accuracy.

3.3 Optimizing Auxiliary Variables

For given active beamforming matrices $\left\{\boldsymbol{w}_{b,k}, \forall b, \forall k\right\}$ and $\boldsymbol{\Phi}$, the optimization problem can be expressed as

$$\max_{\boldsymbol{u},\boldsymbol{f}} \ \sum_{k=1}^{K} \omega_k \left(\log\left(f_k\right) - f_k \mathrm{mse}_k\right). \tag{18}$$

By substituting Eq. (7) into the objective function in Problem (18), it can be easily seen that the form is concave with respect to $\boldsymbol{u}_k$ and to f_k. Thus, the optimal solution of them can be easily obtained by checking the first order optimality condition as follows:

$$\boldsymbol{u}_k^{\mathrm{opt}} = \left(\sum_{m=1}^{K}\left(\sum_{b=1}^{B} \bar{\boldsymbol{H}}_{b,k}^H \boldsymbol{w}_{b,m}\right)\left(\sum_{b=1}^{B} \boldsymbol{w}_{b,m}^H \bar{\boldsymbol{H}}_{b,k}\right) + \sigma_k^2 \boldsymbol{I}_{M_r}\right)^{-1} \sum_{b=1}^{B} \bar{\boldsymbol{H}}_{b,k}^H \boldsymbol{w}_{b,k}, \tag{19}$$

$$f_k^{\mathrm{opt}} = \mathrm{mse}_k^{-1}, \tag{20}$$

where

$$\mathrm{mse}_k = 1 - \sum_{b}^{B} \boldsymbol{w}_{b,k}^H \bar{\boldsymbol{H}}_{b,k} \boldsymbol{u}_k. \tag{21}$$

3.4 Optimizing Phase Shifts

Next, we focus our attention on optimizing the phase shifts $\boldsymbol{\Phi}$, based on the optimized $\boldsymbol{u},\boldsymbol{f}$ and $\left\{\boldsymbol{w}_{b,k}, \forall b, \forall k\right\}$. By ignoring the unrelated terms, the phase shifts optimization problem is presented as

$$\begin{aligned}&\min_{\boldsymbol{\Phi}} \ \sum_{k=1}^{K} \omega_k f_k \mathrm{mse}_k \\ &\text{s.t.} \ \theta_n^l \in \mathcal{F}, \forall l \in \mathcal{L}, \forall n \in \mathcal{N}.\end{aligned} \tag{22}$$

This problem is non-convex due to the unit-modulus constraint. Substituting Eq. (3) into Eq. (7) and following some further manipulations, the objective function is represented as

$$\sum_{k=1}^{K}\omega_k f_k\left[\sum_{i=1}^{L}\sum_{j=1}^{L}\mathrm{Tr}\left(\boldsymbol{\Phi}_i^H\boldsymbol{A}_{i,j,k}\boldsymbol{\Phi}_j\boldsymbol{B}_{i,j}\right)+\right.$$
$$\left.\sum_{l=1}^{L}\mathrm{Tr}\left(\boldsymbol{\Phi}_l^H\left(\boldsymbol{C}_{l,k}-\boldsymbol{D}_{l,k}\right)\right)+\sum_{l=1}^{L}\mathrm{Tr}\left(\boldsymbol{\Phi}_l\left(\boldsymbol{C}_{l,k}-\boldsymbol{D}_{l,k}\right)^H\right)\right], \tag{23}$$

with the notations as follows:

$$\boldsymbol{A}_{i,j,k}=\boldsymbol{H}_{\mathrm{r},i,k}\boldsymbol{u}_k\boldsymbol{u}_k^H\boldsymbol{H}_{\mathrm{r},j,k}^H, \tag{24}$$

$$\boldsymbol{B}_{i,j}=\sum_{m=1}^{K}\left(\sum_{b=1}^{B}\boldsymbol{G}_{b,j}\boldsymbol{w}_{b,m}\right)\left(\sum_{b=1}^{B}\boldsymbol{w}_{b,m}^H\boldsymbol{G}_{b,i}^H\right), \tag{25}$$

$$\boldsymbol{C}_{l,k}=\boldsymbol{H}_{\mathrm{r},l,k}\boldsymbol{u}_k\boldsymbol{u}_k^H\sum_{m=1}^{K}\left(\sum_{b=1}^{B}\boldsymbol{H}_{b,k}^H\boldsymbol{w}_{b,m}\right)\left(\sum_{b=1}^{B}\boldsymbol{w}_{b,m}^H\boldsymbol{G}_{b,l}^H\right), \tag{26}$$

$$\boldsymbol{D}_{l,k}=\boldsymbol{H}_{\mathrm{r},l,k}\boldsymbol{u}_k\sum_{b=1}^{B}\boldsymbol{w}_{b,k}^H\boldsymbol{G}_{b,l}^H\ . \tag{27}$$

By defining vector $\boldsymbol{\phi}_l=\left[\phi_{l,1},\cdots,\phi_{l,n},\cdots,\phi_{l,N}\right]^T$, and $\boldsymbol{\phi}=\left[\boldsymbol{\phi}_1^T,\cdots,\boldsymbol{\phi}_L^T\right]^T$, we arrive at $\mathrm{Tr}\left(\boldsymbol{\Phi}_i^H\boldsymbol{A}_{i,j,k}\boldsymbol{\Phi}_j\boldsymbol{B}_{i,j}\right)=\boldsymbol{\phi}^H\left(\boldsymbol{A}_{i,j,k}\odot\boldsymbol{B}_{i,j}^T\right)\boldsymbol{\phi}$, where $\odot$ is a Hadamard product operator. For ease of representation, we let $\boldsymbol{Z}_{i,j}=\left(\sum_{k=1}^{K}\omega_k f_k\boldsymbol{A}_{i,j,k}\right)\odot\boldsymbol{B}_{i,j}^T$, $\hat{\boldsymbol{Z}}=\begin{pmatrix}\boldsymbol{Z}_{1,1}&\cdots&\boldsymbol{Z}_{1,L}\\ \vdots&\ddots&\vdots\\ \boldsymbol{Z}_{L,1}&\cdots&\boldsymbol{Z}_{L,L}\end{pmatrix}$, $\boldsymbol{p}_l=\left[\sum_{k=1}^{K}\omega_k f_k\left[\boldsymbol{C}_{l,k}-\boldsymbol{D}_{l,k}\right]_{1,1},\cdots,\sum_{k=1}^{K}\omega_k f_k\left[\boldsymbol{C}_{l,k}-\boldsymbol{D}_{l,k}\right]_{N,N}\right]^T$, and $\boldsymbol{p}=\left[\boldsymbol{p}_1^T,\cdots,\boldsymbol{p}_L^T\right]^T$. Hence, the optimization of Problem (22) for phase shifts $\boldsymbol{\phi}$ can be reformulated as:

$$\begin{aligned}&\min_{\boldsymbol{\phi}}\ \boldsymbol{\phi}^H\hat{\boldsymbol{Z}}\boldsymbol{\phi}+2\Re\mathrm{e}\left\{\boldsymbol{p}^H\boldsymbol{\phi}\right\}\\&\text{s.t. }\theta_n^l\in\mathcal{F},\forall l\in\mathcal{L},\forall n\in\mathcal{N}.\end{aligned} \tag{28}$$

We have $\mathcal{F}\triangleq\left\{\theta_n^l\,\middle|\,\left|e^{j\theta_n^l}\right|=1\right\}$ in the non-convex problem, and we notice that the unit modulus constraints form a complex circle manifold in fact, as

$$\mathcal{M}^{NL}=\left\{\phi\in\mathbb{C}^{NL}:\left|\phi_{1,1}\right|=\ldots=\left|\phi_{L,N}\right|=1\right\}. \tag{29}$$

The formed search space is the product of NL circles in the complex plane, which is a Riemanifold of $\mathbb{C}^{NL}$ with the product geometry. Thereby, we propose the Riemannian conjugate gradient method for the phase shifts optimization. Specifically, Problem (28) can be alternately solved by carrying out the following steps at each iteration r: 1) Firstly, compute the gradient in Euclidean space $\nabla f\left(\boldsymbol{\phi}_r\right)=2\hat{\boldsymbol{Z}}\boldsymbol{\phi}_r+2\boldsymbol{p}^*$; 2) Compute the Riemannian gradient $\mathrm{grad}\ f\left(\boldsymbol{\phi}_r\right)=\mathrm{Proj}_{\boldsymbol{\phi}_r}\nabla f\left(\boldsymbol{\phi}_r\right)=\nabla f\left(\boldsymbol{\phi}_r\right)-\Re\left\{\nabla f\left(\boldsymbol{\phi}_r\right)\odot\boldsymbol{\phi}_r^*\right\}\odot\boldsymbol{\phi}_r$; 3) Then, update the search direction for the RCG method on manifold $\boldsymbol{\eta}_{r+1}=-\mathrm{grad}f\left(\boldsymbol{\phi}_{r+1}\right)+\beta_r\mathcal{T}_{\phi_r\to\phi_{r+1}}\left(\boldsymbol{\eta}_r\right)$, where $\mathcal{T}_{\boldsymbol{\phi}_r\to\boldsymbol{\phi}_{r+1}}\left(\boldsymbol{\eta}_t\right)\triangleq T_{\boldsymbol{\phi}_r}\mathcal{M}^{NL}\mapsto T_{\boldsymbol{\phi}_{r+1}}\mathcal{M}^{NL}$: $\boldsymbol{\eta}_r\mapsto\boldsymbol{\eta}_r-\Re\mathrm{e}\left\{\boldsymbol{\eta}_r\odot\boldsymbol{\phi}_{r+1}^*\right\}\odot\boldsymbol{\phi}_{r+1}$ and β_r is chosen as the Polak-Ribiere parameter; 4) Finally, map the solution into the manifold $\mathcal{M}^{NL}$ as $\boldsymbol{\phi}_{r+1}=\mathcal{R}_{\boldsymbol{\phi}_r}\left(\alpha_r\boldsymbol{\eta}_r\right)$ with step size α_r by retraction operator $\mathcal{R}_{\boldsymbol{\phi}_r}\left(\alpha_r\boldsymbol{\eta}_r\right)\triangleq T_{\boldsymbol{\phi}_r}\mathcal{M}^{NL}\mapsto\mathcal{M}^{NL}$: $\alpha_r\boldsymbol{\eta}_r\mapsto\mathrm{vec}\left[\frac{\boldsymbol{\phi}_r+\alpha_r\boldsymbol{\eta}_r}{\left|\boldsymbol{\phi}_r+\alpha_r\boldsymbol{\eta}_r\right|}\right]$.

3.5 Complexity Analysis and Algorithm Supplements

Based on the solutions to the above sub-problems, we implement the proposed cooperative distributed beamforming design by iteratively updating the variable set $\left\{\boldsymbol{w},\boldsymbol{u},\boldsymbol{f},\boldsymbol{\xi},\boldsymbol{\phi}\right\}$. At each iteration, $\boldsymbol{w}$ is optimized locally at each AP, while CPU optimizes $\boldsymbol{u},\boldsymbol{f},\boldsymbol{\phi}$ and computes $\boldsymbol{\xi}$ in a centralized mode, which is guaranteed to converge at least a locally optimal solution[18]. Note that APs need to share the estimated CSI $\left\{\boldsymbol{H}_{b,k},\forall k\right\}$, $\left\{\boldsymbol{H}_{\mathrm{r},l,k},\forall k,l\right\}$, and $\left\{\boldsymbol{G}_{b,l},\forall l\right\}$ with CPU, so the required backhaul signaling for CSI exchange is $BM_t\left(KM_r+NL\right)+NLKM_r$. Moreover, according to Section 3.2, each AP needs to receive $\left\{\boldsymbol{u},\boldsymbol{f},\boldsymbol{\xi},\boldsymbol{\phi}\right\}$ from CPU (or initialize them) at each iteration, which requires $KM_r+K+NL+BM_tK$ backhaul signaling, and then APs need to feed back $\left\{\boldsymbol{w}_{b,k},\forall k\right\}$ to CPU, which requires BM_tK backhaul signaling. Therefore, the total required signaling overhead of the proposed design is $BM_t\left(KM_r+NL\right)+NLKM_r+I\left(KM_r+K+NL+2BM_tK\right)$, where I denotes the number of iterations. It reduces the signaling overhead compared with the fully distributed framework, which leads to $B^2\left(M_t\left(KM_r+NL\right)+NLKM_r+I\left(KM_r+K+NL+M_tK\right)\right)$ signaling overhead, in the case of large B in the cell-free network.

In the meantime, the main complexity is dominated by the matrix inverse, which involves complexity $\mathcal{O}\left(BKM_t^3\right)$, and by the gradient computation in the RCG method, which involves $\mathcal{O}\left(K^2N^2L^2\right)$. Compared with the design using semidefinite relaxation (SDR) or convex optimization toolbox, which leads to the complexity $\mathcal{O}\left(N^{3.5}L^{3.5}\right)$, the proposed approach realized great computational complexity reduction.

As a supplement, when the discrete phase shifts are considered, we adopt the common solution and approximation projection[21], to address the non-convex constraint. The core idea of this method is first to obtain a continuous solution $\theta_n^{l\ \mathrm{opt}},\forall l,\forall n$ that satisfies the unit modulus constraint, and then simply project the solution to the nearest discrete value in the set

$\hat{\mathcal{F}} \triangleq \left\{ \theta_n^l \middle| \ \theta_n^l = e^{j\frac{2\pi(x-1)}{\Delta}}, x = 1,\cdots,\Delta \right\}$, where $\Delta = 2^{\hat{b}}$ and $\hat{b}$ is the number of discrete bits. It can be written as follows:

$$\theta_n^{l\star} = \arg\min_{\varphi \in \hat{\mathcal{F}}} \left| \theta_n^{l\ \mathrm{opt}} - \varphi \right|, \forall l \in \mathcal{L}, \forall n \in \mathcal{N}. \tag{30}$$

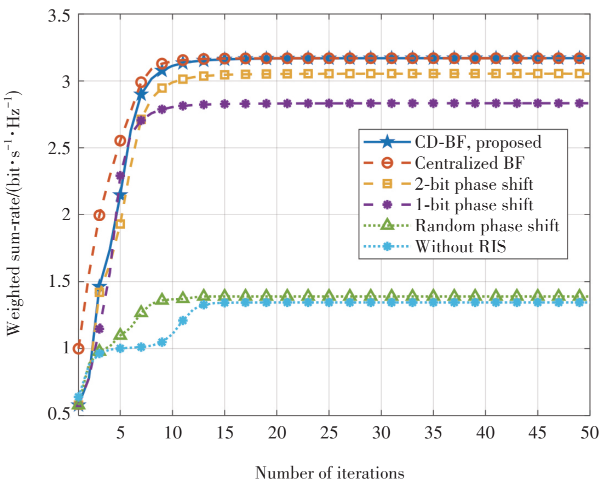
4 Simulation Results

In this section, simulation results are presented to demonstrate the performance of the proposed cooperative distributed beamforming design in the multi-RIS aided cell-free system. The numbers of APs, users and RISs are $B = 3$, $K = 3$, and $L = 3$, and each AP and user is equipped with $M_t = M_r = 2$ antennas. Considering a 3D scenario, three APs are located at $(0, 100\ \mathrm{m}, 3\ \mathrm{m})$, $\left(-50\sqrt{3}\ \mathrm{m}, -50\ \mathrm{m}, 3\ \mathrm{m}\right)$, and $\left(50\sqrt{3}\ \mathrm{m}, -50\ \mathrm{m}, 3\ \mathrm{m}\right)$, respectively, and users are randomly distributed in a circle centered at $(0, 0)$ with a radius of 5 m. The height of the users is set as 1.5 m. In particular, three RISs are deployed near users right above the points $(0, 20\ \mathrm{m})$, $(-15\ \mathrm{m}, -15\ \mathrm{m})$, and $(15\ \mathrm{m}, -15\ \mathrm{m})$, respectively, facing the ground with an altitude of 6 m, so that all of them can cooperate with all APs to serve the users.

For the large-scale fading, we use the urban macro (UMa) path loss model in 3GPP specification TR 38.901[22] as the distance-dependent channel path loss model, with a carrier frequency set as 5.8 GHz. Specifically, the path loss model for $\boldsymbol{G}_{b,l}$ and $\boldsymbol{H}_{r,l,k}$ can be given by $\mathcal{L}_{\mathrm{LoS}} = 43.27 + 22.0\log(d)$, where d represents the distance between the transmitter and the receiver. Meanwhile, due to the randomness of users and the long distance between the AP and users, LoS propagation may not necessarily be guaranteed for the AP-user channels, so the path loss for $\boldsymbol{H}_{b,k}$ is assumed as $\mathcal{L}_{\mathrm{NLoS}} = \max\left(\mathcal{L}_{\mathrm{LoS}}, \mathcal{L}'_{\mathrm{NLoS}}\right)$, where $\mathcal{L}'_{\mathrm{NLoS}} = 28.81 + 39.08\log(d)$. For the small-scale fading, we consider the Rician fading channel model. Let $\kappa_{\mathrm{AU}} = 0$, $\kappa_{\mathrm{RU}} = 3$, and $\kappa_{\mathrm{AR}} \rightarrow \infty$ denote the Rician factors of the AP-user, RIS-user, and AP-RIS channels, respectively. The transmit power budget is set as $P_{b,\max} = P_{\max} = 20\,\mathrm{dBm}, \forall b$, the noise power is set as $\sigma^2 = -80$ dBm, and the weight ω_k for each user is set as 1 equally.

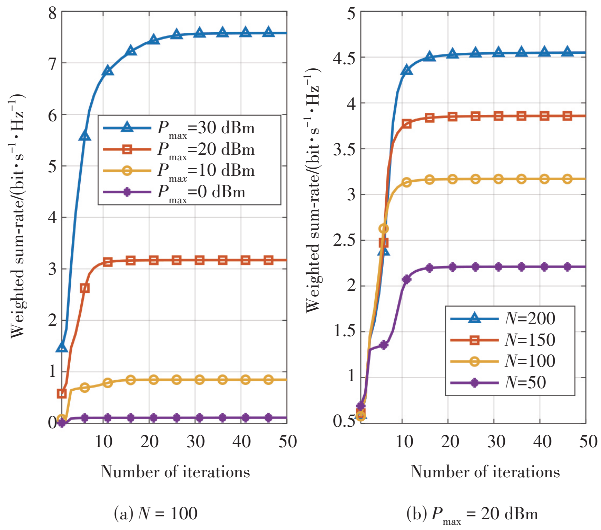
Fig. 2 illustrates the convergence behavior of all the proposed algorithms. It can be seen that, when $N = 100$ and the convergence error is not greater than 0.1%, the proposed cooperative distributed beamforming (CD-BF) algorithm converges within 15 iterations. Despite the distributed design of active beamforming in the proposed method, the convergence performance is almost the same as that of the centralized beamforming (BF)[11], without causing any performance loss. Besides, the cases of discrete bits, random phases, and without RIS converge within 15 iterations as well.

Fig. 3 presents the convergence behavior of the proposed design under different AP transmit power budgets $P_{\max}$ and varying numbers of elements N. Fig. 3(a) illustrates that as $P_{\max}$ increases, the convergence speed noticeably slows down, while the weighted sum-rate performance significantly improves. Similarly, in Fig. 3(b), it can be observed that as N increases, the algorithm converges slightly slower but within 20 iterations. This depicts the good convergence performance of the proposed design.



CD-BF: cooperative distributed beamforming algorithm
RIS: reconfigurable intelligent surface

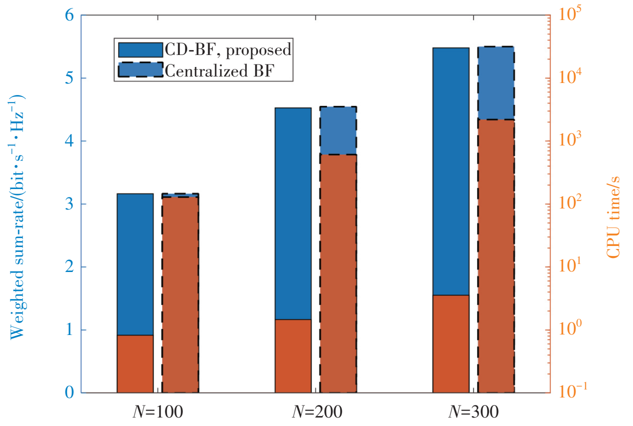
▲**Figure 2. Convergence behavior when *N*=100**



▲ **Figure 3. Convergence behavior under different power budgets and numbers of elements *N***

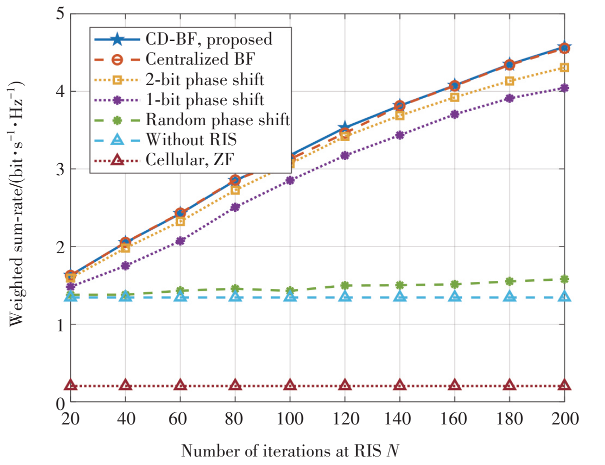
Fig. 4 shows the performance comparison between the proposed and the centralized algorithm in terms of weighted sum-rate and CPU running time, with different numbers of reflecting elements N=100, 200, and 300. First, it is observed that both algorithms exhibit almost identical sum-rate performance under different N. However, due to the proposed cooperative distributed design, which avoids high-dimensional matrix calculations, and the fast optimization speed of the manifold method, the proposed one achieves better speed performance than the centralized algorithm. In addition, as N increases, the runtime of the proposed algorithm remains in the same order of 10^0, while the centralized algorithm's runtime increases from 10^2 to 10^3, highlighting the low complexity advantage of the proposed algorithm.

Fig. 5 compares the sum-rate performance with the size N of RIS. The results indicate that the proposed design achieves higher performance gain as N increases, comparable to centralized design. Additionally, with the rise of N, the approximation loss of low-bit discrete phases becomes larger, which implies the significance of the precise phase design when it comes to a large size of RIS, while balancing the overhead and complexity of channel estimation. Furthermore, we discuss a traditional cellular network baseline with multiple small cells serving the nearest user in the setting scene without RIS assistance, where classical zero-forcing (ZF) precoding is used for transmission. It can be observed that compared with this baseline and the traditional cell-free network without RIS, the RIS-assisted cell-free system architecture achieves significant spectral efficiency improvement.



CD-BF: cooperative distributed beamforming algorithm

▲ **Figure 4. Weighted sum rate and CPU running time comparison between the proposed and the centralized BF**



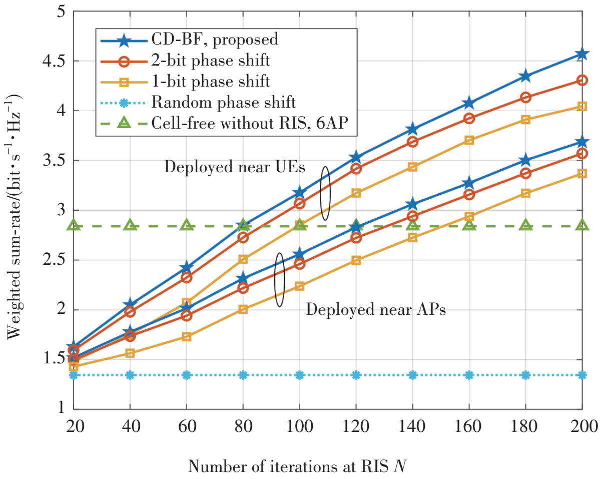
CD-BF: cooperative distributed beamforming algorithm
RIS: reconfigurable intelligent surfaces
ZF: zero-forcing

▲ **Figure 5. Weighted sum rate versus the number of reflecting elements N**

In Fig. 6, we compare the WSR performance under two different deployment strategies, namely, the near-AP side and the near-user side. In the setting scenario, the results reveal that, considering edge users that are far from APs, the near-user deployment outperforms the near-AP one, regardless of continuous or discrete phase shifts. Moreover, it can be seen that, even compared with the traditional cell-free scenario with $B' = B + L$ APs, the proposed RIS-aided architecture still provides significant gains while being cost-effective and energy-efficient.

5 Conclusions

In this paper, we investigate joint active beamforming and phase shift design for the multi-RIS aided cell-free system. The weighted sum-rate maximization problem under the proposed cooperative distributed beamforming design framework has been considered, which is firstly converted to a tractable form by exploiting the relationship between the sum rate and the sum MSE. Further, we derive the distributed closed-form



AP: access point
CD-BF: cooperative distributed beamforming algorithm
RIS: reconfigurable intelligent surfaces
UE: user equipment

▲ **Figure 6. Weighted sum-rate versus N with different RIS deployment strategies** solution from the active beamforming and update the phase shifts using the RCG method. By iteratively optimizing the two objectives across APs and CPU, the proposed design converges to a stationary point, outperforming the centralized framework in terms of lower complexity and equivalent spectrum efficiency. In addition, our numerical results demonstrate the remarkable potential of RIS in improving the network capacity compared with conventional cellular and cell-free systems.

References

[1] LARSSON E G, EDFORS O, TUFVESSON F, et al. Massive MIMO for next generation wireless systems [J]. IEEE communications magazine, 2014, 52(2): 186 – 195. DOI: 10.1109/MCOM.2014.6736761

[2] XIAO M, MUMTAZ S, HUANG Y M, et al. Millimeter wave communications for future mobile networks [J]. IEEE journal on selected areas in communications, 2017, 35(9): 1909 – 1935. DOI: 10.1109/JSAC.2017.2719924

[3] AN J P, YANG K, WU J S, et al. Achieving sustainable ultra-dense heterogeneous networks for 5G [J]. IEEE communications magazine, 2017, 55 (12): 84 – 90. DOI: 10.1109/MCOM.2017.1700410

[4] GESBERT D, HANLY S, HUANG H, et al. Multi-cell MIMO cooperative networks: a new look at interference [J]. IEEE journal on selected areas in communications, 2010, 28(9): 1380 – 1408. DOI: 10.1109/JSAC.2010.101202

[5] NGO H Q, ASHIKHMIN A, YANG H, et al. Cell-free massive MIMO versus small cells [J]. IEEE transactions on wireless communications, 2017, 16(3): 1834 – 1850. DOI: 10.1109/TWC.2017.2655515

[6] NAYEBI E, ASHIKHMIN A, MARZETTA T L, et al. Precoding and power optimization in cell-free massive MIMO systems [J]. IEEE transactions on wireless communications, 2017, 16(7): 4445 – 4459. DOI: 10.1109/TWC.2017.2698449

[7] WU Q Q, ZHANG R. Towards smart and reconfigurable environment: intelligent reflecting surface aided wireless network [J]. IEEE communications magazine, 2020, 58(1): 106 – 112. DOI: 10.1109/MCOM.001.1900107

[8] WU Q Q, ZHANG S W, ZHENG B X, et al. Intelligent reflecting surface-aided wireless communications: a tutorial [J]. IEEE transactions on communications, 2021, 69(5): 3313 – 3351. DOI: 10.1109/TCOMM.2021.3051897

[9] YUE D W, NGUYEN H H, SUN Y. MmWave doubly-massive-MIMO communications enhanced with an intelligent reflecting surface: asymptotic analysis [J]. IEEE access, 2020, 8: 183774 – 183786. DOI: 10.1109/ACCESS.2020.3029244

[10] PAN C H, REN H, WANG K Z, et al. Multicell MIMO communications relying on intelligent reflecting surfaces [J]. IEEE transactions on wireless communications, 2020, 19(8): 5218 – 5233. DOI: 10.1109/TWC.2020.2990766

[11] ZHANG Z J, DAI L L. A joint precoding framework for wideband reconfigurable intelligent surface-aided cell-free network [J]. IEEE transactions on signal processing, 2021, 69: 4085 – 4101. DOI: 10.1109/TSP.2021.3088755

[12] ZHANG Y T, DI B Y, ZHANG H L, et al. Beyond cell-free MIMO: Energy efficient reconfigurable intelligent surface aided cell-free MIMO communications [J]. IEEE transactions on cognitive communications and networking, 2021, 7(2): 412 – 426. DOI: 10.1109/TCCN.2021.3058683

[13] GUO H Y, LIANG Y C, CHEN J, et al. Weighted sum-rate maximization for reconfigurable intelligent surface aided wireless networks [J]. IEEE transactions on wireless communications, 2020, 19(5): 3064 – 3076. DOI: 10.1109/TWC.2020.2970061

[14] LI Z F, HUA M, WANG Q X, et al. Weighted sum-rate maximization for multi-IRS aided cooperative transmission [J]. IEEE wireless communications letters, 2020, 9(10): 1620 – 1624. DOI: 10.1109/LWC.2020.2999356

[15] ZHOU T, XU K, XIA X C, et al. Achievable rate optimization for aerial intelligent reflecting surface-aided cell-free massive MIMO system [J]. IEEE access, 2021, 9: 3828 – 3837. DOI: 10.1109/ACCESS.2020.3047450

[16] VAN CHIEN T, NGO H Q, CHATZINOTAS S, et al. Reconfigurable intelligent surface-assisted cell-free massive MIMO systems over spatially-correlated channels [J]. IEEE transactions on wireless communications, 2022, 21(7): 5106 – 5128. DOI: 10.1109/TWC.2021.3136925

[17] HUANG S C, YE Y, XIAO M, et al. Decentralized beamforming design for intelligent reflecting surface-enhanced cell-free networks [J]. IEEE wireless communications letters, 2021, 10(3): 673 – 677. DOI: 10.1109/LWC.2020.3045884

[18] ATZENI I, GOUDA B, TÖLLI A. Distributed precoding design via over-the-air signaling for cell-free massive MIMO [J]. IEEE transactions on wireless communications, 2021, 20(2): 1201 – 1216. DOI: 10.1109/TWC.2020.3031807

[19] WANG Z R, LIU L, CUI S G. Channel estimation for intelligent reflecting surface assisted multiuser communications: Framework, algorithms, and analysis [J]. IEEE transactions on wireless communications, 2020, 19 (10): 6607 – 6620. DOI: 10.1109/TWC.2020.3004330

[20] SHI Q J, RAZAVIYAYN M, LUO Z Q, et al. An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel [J]. IEEE transactions on signal processing, 2011, 59(9): 4331 – 4340. DOI: 10.1109/TSP.2011.2147784

[21] WU Q Q, ZHANG R. Beamforming optimization for wireless network aided by intelligent reflecting surface with discrete phase shifts [J]. IEEE transactions on communications, 2020, 68(3): 1838 – 1851. DOI: 10.1109/TCOMM.2019.2958916

[22] 3GPP. Study on channel model for frequencies from 0.5 to 100 GHz: TR 38.901 [S]. 2019

Biographies

ZHU Yuting received her bachelor's degree in communication engineering from the Beijing University of Posts and Telecommunications (BUPT), China in 2022. She is currently working toward her master's degree in communication and information engineering at the School of Artificial Intelligence, BUPT. Her research interests include the emerging technologies of 5G wireless communication network.

XU Zhiyu received his PhD degree in electrical and electronic engineering from the Hong Kong University of Science and Technology (HKUST), China in 2002. He is currently a senior architect with ZTE Corporation, China. He has authored or coauthored more than 10 articles on international journals and conferences, and has filed more than 10 patents. He is the author of two technical books. His research interests include 6G wireless communication and signal processing.

ZHANG Hongtao (htzhang@bupt.edu.cn) received his PhD degree in communication and information systems from the Beijing University of Posts and Telecommunications, China in 2008. He is currently a full professor with the Beijing University of Posts and Telecommunications, China. He has authored or coauthored more than 100 articles on international journals and conferences, and has filed more than 50 patents. He is the author of ten technical books. His research interests include 5G wireless communication and signal processing. He is a senior member of IEEE.

The 1st Youth Expert Committee
for Promoting Industry-University-Institute Cooperation

Director	**CHEN Wei,** Beijing Jiaotong University
Deputy Director	**QIN Xiaoqi,** Beijing University of Posts and Telecommunications
	LU Dan, ZTE Corporation

Members (Surname in Alphabetical Order)

CAO Jin	Xidian University
CHEN Li	University of Science and Technology of China
CHEN Qimei	Wuhan University
CHEN Shuyi	Harbin Institute of Technology
CHEN Siheng	Shanghai Jiao Tong University
CHEN Wei	Beijing Jiaotong University
GUAN Ke	Beijing Jiaotong University
HAN Kaifeng	China Academy of Information and Communications Technology
HE Zi	Nanjing University of Science and Technology
HOU Tianwei	Beijing Jiaotong University
HU Jie	University of Electronic Science and Technology of China
HUANG Chen	Purple Mountain Laboratories
LI Ang	Xi'an Jiaotong University
LIU Chunsen	Fudan University
LIU Fan	Southern University of Science and Technology
LIU Junyu	Xidian University
LU Dan	ZTE Corporation
LU Youyou	Tsinghua University
NING Zhaolong	Chongqing University of Posts and Telecommunications
QI Liang	Shanghai Jiao Tong University
QIN Xiaoqi	Beijing University of Posts and Telecommunications
QIN Zhijin	Tsinghua University
SHI Yinghuan	Nanjing University
TANG Wankai	Southeast Univeristy
WANG Jingjing	Beihang University
WANG Xinggang	Huazhong University of Science and Technology
WANG Yongqiang	Tianjin University
WEN Miaowen	South China University of Technology
WU Qingqing	Shanghai Jiao Tong University
WU Yongpeng	Shanghai Jiao Tong University
XIA Wenchao	Nanjing University of Posts and Telecommunications
XU Mengwei	Beijing University of Posts and Telecommunications
XU Tianheng	Shanghai Advanced Research Institute, Chinese Academy of Sciences
YANG Chuanchuan	Peking University
YIN Haifan	Huazhong University of Science and Technology
YU Jihong	Beijing Institute of Technology
ZHANG Jiao	Beijing University of Posts and Telecommunications
ZHANG Yuchao	Beijing University of Posts and Telecommunications
ZHANG Jiayi	Beijing Jiaotong University
ZHAO Yuda	Zhejiang University
ZHAO Zhongyuan	Beijing University of Posts and Telecommunications
ZHOU Yi	Southwest Jiaotong University
ZHU Bingcheng	Southeast University

ZTE COMMUNICATIONS

中兴通讯技术（英文版）

ZTE Communications has been indexed in the following databases:

- Abstract Journal
- Cambridge Scientific Abstracts (CSA)
- China Science and Technology Journal Database
- Chinese Journal Fulltext Databases
- Index Copernicus
- Ulrich's Periodicals Directory
- Wanfang Data
- WJCI 2021−2023

ZTE COMMUNICATIONS
Vol. 22 No. 2 (Issue 87)
Quarterly
First Issue Published in 2003

Supervised by:
Anhui Publishing Group

Sponsored by:
Time Publishing and Media Co., Ltd.
Shenzhen Guangyu Aerospace Industry Co., Ltd.

Published by:
Anhui Science & Technology Publishing House

Edited and Circulated (Home and Abroad) by:
Magazine House of ZTE Communications

Staff Members:
General Editor: WANG Xiyu
Editor-in-Chief: WANG Li
Executive Editor-in-Chief: HUANG Xinming
Editorial Director: LU Dan
Editor-in-Charge: ZHU Li
Editors: REN Xixi, XU Ye, YANG Guangxi
Producer: XU Ying
Circulation Executive: WANG Pingping
Assistant: WANG Kun

Editorial Correspondence:
Add: 12F Kaixuan Building, 329 Jinzhai Road,
Hefei 230061, P. R. China
Tel: +86-551-65533356
Email: magazine@zte.com.cn
Website: http://zte.magtechjournal.com

Annual Subscription: RMB 120
Printed by:
Hefei Tiancai Color Printing Company
Publication Date: June 25, 2024

China Standard Serial Number: $\frac{\text{ISSN 1673−5188}}{\text{CN 34−1294/TN}}$