EAI Endorsed Transactions
on Mobile Communications and Applications

Research Article EAI.EU

Intelligent Distributed Data Storage for Wireless Communications in B5G Networks

Yajuan Tang[1], Shiwei Lai[2], Zichao Zhao[2], Yanyi Rao[2], Wen Zhou[3], Fusheng Zhu[4], Liming Chen[5], Dan Deng[6], Jing Wang[7], Tao Cui[7], Yuwei Zhang [7], Jun Liu[7,*], Di Wu[8], Zhusong Liu[9], Huang Huang[10], Xuan Zhou[11], Zhao Wang[12], Chao Li[13], Kai Chen[14], Wei Zhou[15], Yun Li[16], Kaimeno Dube[17], Abbarbas Muazu[18], Nakilavai Rono[19], Sunli Feng[20,21], Jiayin Qin[20,22], Haige Xiang[23], Zhigang Cao[20,7], Lieguang Zeng[20,7], Zhixing Yang[20,7]

[1]Shantou University, Shantou, China.
[2]Guangzhou University, Guangzhou, China.
[3]Nanjing Forestry University, Nanjing, China.
[4]Guangdong New Generation Communication and Network Innovative Institute (GDCNi), Guangzhou, China.
[5]Electric Power Research Institute of CSG, Guangzhou.
[6]University of Science and Technology of China, Hefei, China.
[7]Information Research Center, Tsinghua University, Beijing, China.
[8]Beijing University of Posts and Telecommunications, Beijing, China.
[9]Anhui University of Technology, Anhui, China.
[10]Huawei Technologies, D*ü*sseldorf, Germany.
[11]Huawei Technologies, Boulogne-Billancourt, France.
[12]Ericsson Sweden, Stockholm, Sweden.
[13]Advanced Research Center, Hamdard University, Pakistan.
[14]Huawei Technologies, Stockholm, Sweden.
[15]Huawei Technologies, Markham, Canada.
[16]University of Illinois Urbana-Champaign, Urbana, USA.
[17]Vaal University of Technology, Andries Potgieter Blvd, South Africa.
[18]Baze University, Airport Road, Abuja, Nigeria.
[19]Rongo University, Rongo.
[20]King Abdullah University of Science and Technology (KAUST), Kingdom of Saudi Arabia.
[21]School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China.
[22]School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China.
[23]Information Research Center of Peking University, China.

Abstract

With the continuous improvement of the fifth-generation (5G) communication networks and other network infrastructure and applications, and the continuous strengthening of the performance of edge computing terminals, billions of mobile and Internet of Things (IoT) devices are connected to the Internet, generating hundreds of millions of data bytes at the edge of the network. Taking beyond 5G (B5G) edge intelligent network as the research object, based on the deep integration of storage / computing and communication, this paper focuses on the theory and key technology of system data storage, so as to effectively support the related applications of B5G edge intelligent network in the future. Specifically, this paper analyzes the research status of data storage, studies the practical distributed storage computing system, and designs the corresponding flashback shift code and error correction scheme with low storage space overhead. The work in this paper can serve an importance guidance for the theoretical development of distributed data storage for B5G intelligent networks.

Received on 05 August 2022; accepted on 22 August 2022; published on 25 August 2022

Keywords: Data storage, wireless transmission, B5G.

doi:10.4108/eetmca.v7i2.2415

1. Introduction

With the continuous improvement of the fifth-generation (5G) communication networks and other network infrastructure and applications [1–4], and the continuous strengthening of the performance of edge computing terminals, billions of mobile and Internet of Things (IoT) devices are connected to the Internet, generating hundreds of millions of data bytes at the edge of the network. Gartner predicts that by 2022, more than 80% of enterprise IoT projects will include AI components [5–8].

In the process of digital transformation, all walks of life will deploy intelligent edges at the most appropriate balance point [9–12]. The industrial demand from intelligent manufacturing, transportation, retail, energy, smart city and other aspects will become a huge source to promote the outbreak of the edge intelligent market after being superimposed with 10 billion level networking devices [13–15].

With the deployment of various intelligent edge devices, people began to pay attention to the landing of intelligent edge technology and how to solve the pain points of the industry [16–18]. In this process, how to make the edge more intelligent, and how to deploy intelligence to the “last mile" of the industry play an important role in the future network design.

At present, the traditional communication network still adopts the communication centric architecture, which is difficult to support the growing application demand of massive data computing [19–21]. There is an urgent need to deeply integrate communication and computing, and efficiently use the cross center data / computing power of the network to support the needs of massive applications in the future mobile communication network.

On the basis that 5G network focuses on enhancing the interconnection of everything such as mobile broadband, ultra reliable low delay communication and large-scale machine communication, mobile communication, IoT and other systems not represented by B5G/6G network need to be deeply integrated with artificial intelligence, have super strong communication, computing and intellectual capabilities, provide human machine thing intelligent connection of everything, and realize the comprehensive upgrade from information transmission to “information storage perception communication computing control integration", In order to support intelligent applications of edge networks such as smart cities, industrial automation and autonomous driving, the application scenario of B5G edge intelligent network is shown in Figure 1.

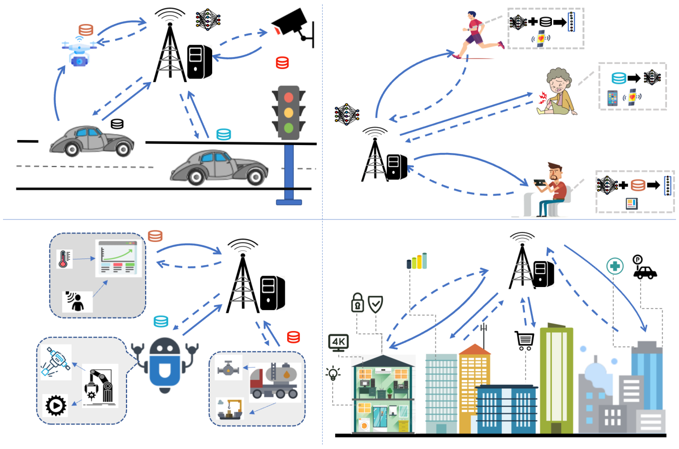

Figure 1. Application scenarios of B5G edge intelligent network

Taking beyond 5G (B5G) edge intelligent network as the research object, based on the deep integration of storage / computing and communication, this paper focuses on the theory and key technology of system data storage, so as to effectively support the related applications of B5G edge intelligent network in the future. Specifically, this paper analyzes the research status of data storage, studies the practical distributed storage computing system, and designs the corresponding flashback shift code and error correction scheme with low storage space overhead. The work in this paper can serve an importance guidance for the theoretical development of distributed data storage for B5G intelligent networks.

2. Research progress of data storage

Efficient data storage is the basis of realizing B5G edge intelligent network. However, the large-scale growth of data volume has caused great pressure on the storage system, which has been deeply studied at home and abroad. First of all, in order to improve the utilization of storage resources, distributed storage (DS) based on network coding with the combination property (CP) has been widely used. The most common CP code is the maximum distance separable (MDS), which is widely used in the design of disk fault storage codes. Reed Solomon (RS) code is the most popular MDS code. However, RS code needs to perform encoding and decoding operations on a very large finite field, and its encoding and decoding process includes cumbersome steps such as linear combination and corresponding Gauss elimination (multiplication and division), resulting in high computational complexity, high power consumption and equipment overheating. Therefore, RS code is not suitable for application scenarios where big data is frequently read and written, or devices with limited decoding computing power and resources. Therefore, researchers are committed to implementing low complexity coding and decoding schemes in DS systems. At present, the mainstream

*Corresponding author. Email: junliu.thu@ieee.org technology is to reduce the size of the finite field and avoid the Gauss elimination steps included in decoding, that is, the time-consuming line ladder transformation and simple reverse iteration in the inverse process. If we can avoid the time-consuming step of line ladder transformation, we only need to reverse the iteration, which obviously can reduce the decoding complexity. The typical representative is zigzag decoding (ZD), whose decoding steps look like sawtooth. By using shift and add (SA) operation in the encoding process and ZD operation in the decoding process, sawtooth code avoids the inverse process of coefficient matrix in the decoding process, which can solve the problems of high decoding complexity and poor numerical stability. The disadvantage of ZD is that it introduces additional storage space overhead.

In view of the fact that the shift matrix of Inc-Diff code (Increasing Difference code) is a vandermond matrix and the shift value is incremented by row, its overhead is $m(k-1)$. In order to further reduce the overhead, Cyc-Shift code is proposed on the basis of line cyclic shift. Cyc-Shift code determines the base vector according to the (n,k) relationship, and uses the base vector to perform cyclic shift to obtain the shift matrix. The cost of Cyc-Shift code can be divided into two cases according to the size relationship between m and k. When $m<k$, the overhead is $k(k-1)/2$; When $m>k$, the overhead is $m(m-1)/2$. The structure of this code is simple, but it has its limitations: Cyc-Shift code is only applicable to the scenario of $m \leq k$, when $m=k$, its overhead utilization reaches the maximum. When $m<k$, there is a certain overhead loss and low resource utilization. In order to make the cyclic shift code more widely used, researchers have constructed a full parameter (n,k) CP-ZD code named ID-CS based on the properties of Increasing Difference (ID) and Cyclic Shift (CS). The difference between this code and Cyc-Shift code is that: firstly, the shift value of ID-CS code increases in the vertical direction rather than the horizontal direction; secondly, the cyclic shift operation is carried out on the matrix rather than on the row vector. In terms of parameters, ID-CS can be applied to any (n,k) setting. However, in the case of $m<k$, the cost of this code is consistent with that of Cyc-Shift code, which can not reduce the cost loss. The existing CP-ZD code considers the problem of full parameters too much, and it is not applicable in the application scenario of $m<k$. As stated in the existing work, the size of m is often between 20% and 80% of k. The size of m is not suitable to be too large, which will waste resources. Therefore, in the scenario of $m<k$, especially for the edge nodes used by the edge intelligent network, it is necessary to design a coding framework with less overhead, which can not only solve the problem of high overhead of existing sawtooth codes in $m<k$, but also make the application of CP-ZD codes more universal. Specifically, the designed shift matrix is generally incremental. If we consider further increasing the decreasing mode of flashback, the overhead can be reduced by half, so as to significantly improve the system efficiency.

On the other hand, the separation of storage and computation will inevitably affect the efficiency of system intelligent training. General storage is to convert real numbers into finite fields and then store them in distributed coding. While the edge intelligent network needs frequent computation, it needs to frequently access data from the distributed storage system, and then needs frequent finite field and practical conversion. Therefore, the real data is directly distributed encoded and stored, so that direct calculation can be directly accessed and the calculation results can be decoded. This computing method is also known as coded distributed computing (CDC), and its computing results support resistance to straggler, so it can speed up computing in disguised form.

CDC is applied to the DNN training of intelligent network, and its coding redundancy is divided into two parts: on one hand, researchers use redundancy to resist the straggler problem, which is limited by the returned wrong calculation results, making the random gradient descent process deviate from the best target; On the other hand, researchers use redundancy to correct errors, which makes the scheme troubled by straggler. The work of these two aspects can not take into account both resistance to straggler and resistance to errors, which leads to the delay of training. How to give consideration to these two aspects at the same time for comprehensive design to obtain high-speed training should be studied. In addition, the above research is based on LC-CDC, while the research based on SAZD-CDC needs to be carried out. In terms of specific technology, the traditional error detection / correction decoding algorithm for LC coding in finite field is not suitable for real number calculation and the shift plus structure of CP-ZD code. The researchers designed an error detection / correction decoding algorithm in the early stage, but the algorithm is only applicable to one-dimensional CP-ZD codes in the binary domain without redundant packets, and its scope of application is relatively limited. For redundant packets and more complex two-dimensional CP-ZD codes in the real domain, it is urgent to carry out research on low complexity error detection / correction decoding in the real domain of the decoder.

3. Challenges on the data storage

From the analysis of the above research status, it can be seen that the existing research has carried out an in-depth study on the reliability, availability and security of data storage, and conducted an in-depth analysis from multiple perspectives of exploring the improvement of storage efficiency and data update efficiency. From this point of view, it has been optimized in combination with storage coding and other technologies, which has significantly improved the performance of the storage system. These research works provide important reference and theoretical reference for the data storage of B5G edge intelligent network. However, B5G edge intelligence is applied in many application scenarios such as the IoT and industrial IoT, and the computing power and storage overhead of some network nodes are limited, which puts forward new requirements on the complexity and storage overhead of storage code encoding and decoding. How to design a storage coding scheme with low encoding and decoding complexity, low overhead and real number domain for B5G edge intelligent network, greatly improve the storage efficiency and enhance the reliability and security of storage is a difficult challenge.

4. Feasible solutions to distributed data storage

Firstly, in order to reduce the memory code overhead, aiming at the problem that the existing CP-ZD code has high overhead in the case of $(m < k)$, a low memory overhead codec framework is designed. Firstly, we should determine the base matrix according to the size relationship between n and k, then reverse it from the second row to form the reverse matrix, and finally intercept the first m rows of the base matrix and the reverse matrix to form the final shift matrix. Therefore, this code is called Reverse Shift Code (Rev-Shift code). The Rev-Shift code adopts SA encoding and ZD decoding, which can prove that the code meets the CP-ZD property, and the low overhead of Rev-Shift code is verified by experimental simulation. In addition, shift leads to additional overhead (storage space overhead and corresponding computing overhead), and the performance bound of this overhead has not been described, resulting in the lack of a measure of code performance. Therefore, based on the necessary conditions of CP-ZD code, we should characterize the lower bound of the possible overhead of CP-ZD code, provide a measurement standard, prove the possibility of its existence, and verify the numerical stability of the code.

Secondly, the design of error detection and error correction of practical storage code is studied. Let $h \leq N$ represent the number of returned results received by the master node. When all $h = K$ calculation results are collected, decoding can be attempted, and the redundancy of the shift can be used to quickly detect errors before decoding. If the detection passes, ZD decoding will be performed and other layers will be advanced. If the detection fails, we should wait for the return of other calculation results, and use redundant packets (i.e. $h > K$ calculation results) to try error correction decoding. If the error correction is successful, we should go to the next layer. If the error correction fails after collecting all N, the layer recalculates. This scheme can maximize the advantages of resisting the straggler problem and ensure that the random gradient descent process is not driven away from the optimal target by a small number of errors and slows down the training.

We should design for error detection $(h = K)$ scenario and error correction $(h > K)$ scenario respectively. As mentioned above, for the convenience of description and graphic display, we take the encoding and decoding of vectors as an example. In addition, since the decoding operation of the multiplication calculation result is essentially the same as the decoding of the data packet before multiplication (with only one multiplier coefficient difference), the decoding of the data packet before multiplication is also described as an example for convenience.

For $h = K$ (no redundant packets), we should design a low complexity fast error detection technology. The core idea is to use the redundant data introduced by the shift and the shift plus coding structure to obtain the constraints that should be satisfied between the data, and then detect whether there are errors according to the constraints. An example of one-dimensional CP-ZD system code is used to illustrate and assume that there is no truncation error. As shown in Fig. 2, data packets u_1 to u_3 with length $\Gamma = 7$ are encoded into v_1 to v_6 and decoded using v_4 to v_6 as an example. Arrange all 24 data from v_4 to v_6 in sequence, and then represent by y_1 to y_{24}, and make the column vector $y \triangleq (y_1, y_2, \cdots, y_{24})^T$. The coding relationship can be expressed as

$$Gu = y, \tag{1}$$

where the column vector $u \triangleq (u_1, u_2, u_3)^T$ (assuming u_j is a row vector), and the 24×21 dimensional matrix G contains only real elements 1 and 0. We can make elementary row transformation for G and y synchronously until G is in row ladder form, and then the bottom three rows can be transformed into three constraint equations about y:

$$\begin{aligned} &y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 + y_9 + y_{10} \\ &+ y_{11} + y_{12} + y_{13} + y_{14} + y_{15} + y_{16} = 0 \end{aligned} \tag{2}$$

$$\begin{aligned} &y_8 + y_9 + y_{10} + y_{11} + y_{12} + y_{13} + y_{14} + y_{15} + y_{17} \\ &+ y_{18} + y_{19} + y_{20} + y_{21} + y_{22} + y_{23} = 0 \end{aligned} \tag{3}$$

$$\begin{aligned} &y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_9 + y_{10} \\ &+ y_{11} + y_{12} + y_{13} + y_{14} + y_{15} + y_{24} = 0 \end{aligned} \tag{4}$$

For the case of actual truncation error, the constraint detection in (2) needs to be replaced by $< \varepsilon$, where

$\gamma\varepsilon$ is a positive real number close to 0. In addition, in the process of elementary row transformation of G, the sparsity of G can be used to obtain the constraint equation in a low complexity way, so as to avoid high complexity Gaussian elimination.

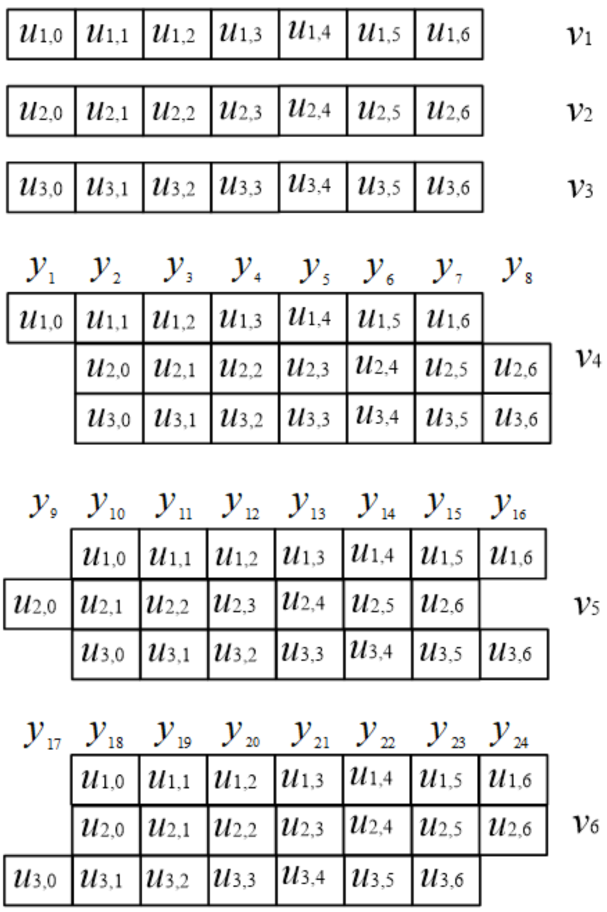
$u_{1,0}$ $u_{1,1}$ $u_{1,2}$ $u_{1,3}$ $u_{1,4}$ $u_{1,5}$ $u_{1,6}$
v_1
$u_{2,0}$ $u_{2,1}$ $u_{2,2}$ $u_{2,3}$ $u_{2,4}$ $u_{2,5}$ $u_{2,6}$
v_2
$u_{3,0}$ $u_{3,1}$ $u_{3,2}$ $u_{3,3}$ $u_{3,4}$ $u_{3,5}$ $u_{3,6}$
v_3
y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8
v_4
y_9 y_{10} y_{11} y_{12} y_{13} y_{14} y_{15} y_{16}
v_5
y_{17} y_{18} y_{19} y_{20} y_{21} y_{22} y_{23} y_{24}
v_6

Figure 2. Schematic diagram of Cyc-Shift code superposition network

For $h > K$ (with redundant packets), we should carry out low complexity fast error correction decoding, including the following operations:

First, SA coding is explained by convolutional code. Specifically, the data packet can be regarded as a data stream and the right shift can be regarded as a delay, so the SA operation is actually the addition of multiple data streams, which can then be equivalent to the encoding of convolutional codes. It differs from standard convolutional codes in that the addition is in the practical rather than the finite field, and the delay represents real numbers rather than bits.

For convenience of understanding, take Cyc-Shift$(8, 4)$ system code as an example and assume that there is no truncation error. The original packets u_1 to u_4 are encoded into v_1 to v_8, where v_1 to v_4 are system packets, and the verification packets v_5 to v_8 are shown in Figure 3. See Fig. 3 for its equivalent convolutional code coding, where it represents the delay unit (a real number), $\oplus$ represents the sum of the input multiple data streams in the real number field, and the data packet u_i or v_i can be regarded as the data stream from the first data on the left to the right to the rightmost data (the blank part complements the real number 0).

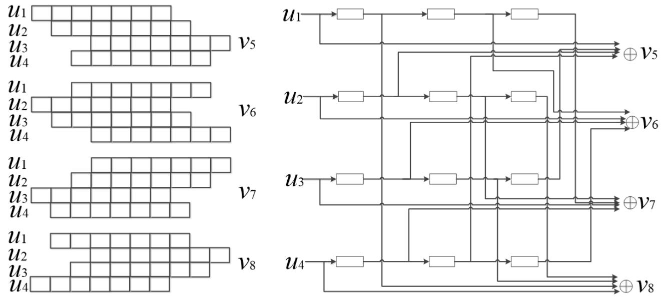

Figure 3. SA operation coding and convolutional coding diagram

In addition, we should design the error correction decoding algorithm of CP-ZD code within the framework of convolutional code. Specifically, the error correction decoding algorithms of convolutional codes mainly include Viterbi decoding, sequence decoding, and large number logic decoding. The first two methods are not applicable to the real number field in practice. The decoding complexity of large number logic decoding is low and it can be applied to real number field after modification. Let the error $e_{i,j} \triangleq r_{i,j} - v_{i,j}$ represent the difference between the actually received $r_{i,j}$ and the correct codewords $v_{i,j}$, and then solve $v_{i,j}$ is equivalent to solve $e_{i,j}$. Taking Cyc-Shift $(8, 4)$ system code as an example, where the correction subsequence can be obtained by simple derivations.

From the above formula, we get all the corrector equations as follows,

$$\begin{aligned} s_{5,0} &= -e_{1,0} \quad + e_{5,0} \\ s_{6,2} &= -e_{1,0} - e_{2,2} - e_{3,1} \quad + e_{6.2} \\ s_{7,3} &= -e_{1,0} - e_{2,1} - e_{3,3} - e_{4,2} \quad + e_{7,3} \\ s_{8,1} &= -e_{1,0} - e_{4,1} + e_{8,1}. \end{aligned} \tag{5}$$

The premise of applying the decision rule of large number logic decoding is that all error data to be decoded ($e_{1,0}$ in this example) have self orthogonality (for example, in the above four equations in this example, except $e_{1,0}$, any $e_{i,j}$ will not appear in more than one equation). Observation shows that this case naturally has self orthogonality.

Different from the decision of convolutional codes in finite fields (such as binary), $e_{i,j}$ at this time is real numbers rather than binary, and the influence of the corresponding truncation error also needs to be considered. The decision rule for $e_{1,0}$ needs to be transformed from checking that all equation values in (4) are 1 and 0 to judging whether they fall into the interval $(e-\varepsilon, e+\varepsilon)$ and $(-\varepsilon, \varepsilon)$, where e represents the true value of $e_{1,0}$, and a positive real number εclose to 0 represents the disturbance tolerance caused by the truncation error. Because the values of each equation may be different, it is necessary to eliminate the useless equation (which is greatly different from the real value e) first, and then make a combined judgment on the remaining multiple equations. In the process of combining decision, the truncation error will cause the error between the combined result and the real value, which is called the combining error, and it will accumulate with the decoding process, which may cause great interference to the accuracy of the subsequent decoded data. Therefore, when truncating floating-point numbers, we should choose to round to zero or round to the nearest, so that the truncation error is expected to be 0, which leads to the expectation that the combined error and cumulative error are both 0.

In addition to the error problem, there are also missing data streams in practical applications (corresponding to some calculation results not returned, that is, $h < N$). Further, for the new returned calculation results (i.e. increased to $h + 1$), the incremental error correction decoding algorithm should be studied to reduce the amount of calculation, (the goal is to make the best use of the existing error correction results).

5. Conclusions

With the deployment and commercialization of 5G mobile communication network, the access nodes and data volume of wireless network has shown a massive and blowout growth trend. Taking B5G edge intelligent network as the research object, based on the deep integration of storage with computing and communication, this paper focused on the theory and key technology of system intelligent transmission, so as to effectively support the related applications of B5G edge intelligent network in the future. In particular, This paper analyzed the research status of data storage, studied the practical distributed storage computing system, and designed the corresponding flashback shift code and error correction scheme with low storage space overhead.

5.1. Data Availability Statement

The part of Introduction is completed by Yajuan Tang (yjtang@ieee.org), Shiwei Lai (swlai@ieee.org), Zichao Zhao (zichaozhao@hotmail.com), Yanyi Rao (yanyirao@ieee.org), Wen Zhou (wenzhou.nfu@gmail.com), and Fusheng Zhu (fushengzhu.gdcni@hotmail.com). The research progress of data storage is completed by Liming Chen (lmchen_CSPG@hotmail.com), Dan Deng (dengdan.ustc@hotmail.com), Jing Wang (jingwang.thu@ieee.org), Tao Cui (taocui@ieee.org), Yuwei Zhang (yzhang.thu@ieee.org), and Jun Liu (junliu.thu@ieee.org). The challenges to data storage are completed by Di Wu (dwu@ieee.org), Zhusong Liu (zhusongliu@ieee.org), Huang Huang (hhuang.huawei@ieee.org), Xuan Zhou (xzhou.huawei@ieee.org), Zhao Wang (zhaowang.ericsson@ieee.org), and Chao Li (chaoli.eecs@ieee.org). The challenges to feasible solutions to distributed data storage are completed by Kai Chen (kchen.huawei@ieee.org), Wei Zhou (wzhou.huawei@ieee.org), Yun Li (yunli.ericsson@ieee.org), Kaimeno Dube (Kaimeno.Dube@ieee.org), Abbarbas Muazu (Abbarbas.Muazu@ieee.org), Nakilavai Rono (Nakilavai.Rono@ieee.org), Jiayin Qin (jyqin@ieee.org), Haige Xiang (haigexiang@ieee.org), Zhigang Cao (zhigangcao.kaust@ieee.org), Lieguang Zeng (lieguangzeng@ieee.org), and Zhixing Yang (zhixingyang@ieee.org).

5.2. Acknowledgements

The work in this paper was supported by the NSFC with grant number 61871235.

5.3. Copyright

References

[1] H. Wang and Z. Huang, "Guest editorial: WWWJ special issue of the 21th international conference on web information systems engineering (WISE 2020)," *World Wide Web*, vol. 25, no. 1, pp. 305–308, 2022.

[2] Y. Guo and S. Lai, "Distributed machine learning for multiuser mobile edge computing systems," *IEEE J. Sel. Top. Signal Process.*, vol. 16, no. 3, pp. 460–473, 2022.

[3] L. He, K. He, L. Fan, X. Lei, A. Nallanathan, and G. K. Karagiannidis, "Toward optimally efficient search with deep learning for large-scale MIMO systems," *IEEE Trans. Commun.*, vol. 70, no. 5, pp. 3157–3168, 2022.

[4] X. Lai, "Outdated access point selection for mobile edge computing with cochannel interference," *IEEE Trans. Vehic. Tech.*, vol. 71, no. 7, pp. 7445–7455, 2022.

[5] H. Wang, J. Cao, and Y. Zhang, *Access Control Management in Cloud Environments*. Springer, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-31729-4

[6] J. Chen, Y. Wang, J. Ou, C. Fan, X. Lu, C. Liao, X. Huang, and H. Zhang, "Albrl: Automatic load-balancing architecture based on reinforcement learning in software-defined networking," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[7] C. Ge, Y. Rao, J. Ou, C. Fan, J. Ou, and D. Fan, "Joint offloading design and bandwidth allocation for ris-aided multiuser mec networks," *Physical Communication*, p. 101752, 2022.

[8] C. Yang, B. Song, Y. Ding, J. Ou, and C. Fan, "Efficient data integrity auditing supporting provable data update for secure cloud storage," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[9] H. Wang, Y. Wang, T. Taleb, and X. Jiang, "Editorial: Special issue on security and privacy in network computing," *World Wide Web*, vol. 23, no. 2, pp. 951–957, 2020.

[10] J. Lu, L. Chen, J. Xia, F. Zhu, M. Tang, C. Fan, and J. Ou, "Analytical offloading design for mobile edge computing-based smart internet of vehicle," *EURASIP journal on advances in signal processing*, vol. 2022, no. 1, pp. 1–19, 2022.

[11] L. Zhang, W. Zhou, J. Xia, C. Gao, F. Zhu, C. Fan, and J. Ou, "Dqn based mobile edge computing for smart internet of vehicle," *EURASIP journal on advances in signal processing*, vol. 2022, no. 1, pp. 1–19, 2022.

[12] J. Liu, Y. Zhang, J. Wang, T. Cui, L. Zhang, C. Li, K. Chen, S. Li, S. Feng, D. Xie *et al.*, "Outage probability analysis for uav-aided mobile edge computing networks," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 9, no. 31, pp. e4–e4, 2022.

[13] J. Sun, X. Wang, Y. Fang, X. Tian, M. Zhu, J. Ou, and C. Fan, "Security performance analysis of relay networks based on-shadowed channels with rhis and cees," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[14] X. Deng, S. Zeng, L. Chang, Y. Wang, X. Wu, J. Liang, J. Ou, and C. Fan, "An ant colony optimization-based routing algorithm for load balancing in leo satellite networks," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[15] C. Wang, W. Yu, F. Zhu, J. Ou, C. Fan, J. Ou, and D. Fan, "Uav-aided multiuser mobile edge computing networks with energy harvesting," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[16] J. Liu, Y. Zhang, J. Wang, T. Cui, L. Zhang, C. Li, K. Chen, H. Huang, X. Zhou, W. Zhou *et al.*, "The intelligent bi-directional relaying communication for edge intelligence based industrial iot networks: Intelligent bi-directional relaying communication," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 9, no. 32, pp. e4–e4, 2022.

[17] J. Liu, T. Cui, L. Zhang, and Y. Zhang, "Deep federated learning based convergence analysis in relaying-aided MEC-IoT networks," *Journal of Engineering*, vol. 2022, no. 8, pp. 101–108, 2022.

[18] Y. Tang and S. Lai, "Energy-efficient and high-spectrum-efficiency wireless transmission," *EAI Endorsed Transactions on Mobile Communications and Applications*, vol. 2022, no. 8, pp. 129–135, 2022.

[19] ——, "An overview on active transmission techniques for wireless scalable networks," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 2022, no. 2, pp. 19–28, 2022.

[20] J. Liu and W. Zhou, "Deep model training and deployment on scalable iot networks: A survery," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 2022, no. 2, pp. 29–35, 2022.

[21] S. Tang, "Dilated convolution based CSI feedback compression for massive MIMO systems," *IEEE Trans. Vehic. Tech.*, vol. 71, no. 5, pp. 211–216, 2022.