A Signaling Monitor Scheme of RRC Protocol in 5G Road Tester

Bingying Zhang(✉), Fang Cheng, and Bingguang Deng

School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
sl80131221@stu.cqupt.edu.cn

Abstract. Faced with the growth of massive data in the 5G network and the low query efficiency in the existing signaling synthesis algorithm, the traditional LTE/LTE-A signaling monitor scheme has been unable to satisfy with the new 5G network architecture. The Radio Resource Control (RRC) protocol is the core of the control plane, which manages and controls the wireless resources of the network. Based on this, a signaling monitor scheme of RRC protocol suitable for 5G road tester was proposed and the specific functions of its main submodule were introduced in detail. Additionally, this paper discussed an improved dynamic hash signaling synthesis algorithm based on AVL tree under a new network architecture. The tree structure is used to reduce the query time of traditional algorithm in the hash table, so as to quickly deal with hash collisions and improve the real time on the signaling synthesis. At present, the proposed scheme was applied to the real network testing of air interface data in the 5G road tester. The experimental results show that the improved algorithm can efficiently solve the issue of low efficiency of Call Detail Recording (CDR) synthesis and the average query time can be reduced by 49.3% and 33.1% compared with the two traditional algorithms. The signaling synthesis scheme described above achieves the expected effect and signaling messages in RRC protocol can be accurately decoded and monitored in real time.

Keywords: 5G road tester · Signaling synthesis · Signaling monitoring · Hash collisions · AVL tree

1 Introduction

In order to adapt to the growth of massive data and accelerate the development of various new service application scenarios, the fifth generation mobile communication network (5G) has arised [1]. More open and flexible 5G network will meet the requirements of mobile network development. 5G will support a diverse range of requirements such as the handling of low latency (in the order of 1 ms), massive Machine-Type Communication (mMTC) and extreme Mobile Broadband (xMBB) services. At present, international organizations such as 3GPP and ITU are actively promoting the standardization process of 5G technology. 5G is at a critical stage of the evolution from technology research to technology verification [2].

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2021
Published by Springer Nature Switzerland AG 2021. All Rights Reserved
H. Gao et al. (Eds.): ChinaCom 2020, LNICST 352, pp. 324–336, 2021.
https://doi.org/10.1007/978-3-030-67720-6_22

Faced with the higher services test requirements and more complex road test data than 4G network, the traditional Long Term Evolution-Advanced (LTE-A) signaling monitoring analyzer is difficult to adapt to the 5G network structure and handle massive mobile signaling data. 5G road tester with independent intellectual property rights will meet more high-performance test requirements of 5G network in the high rate and low delay scenario and promote the rapid development of 5G test industry. Signaling monitor is one of the core technologies of 5G road tester. It mainly completes the whole process from signaling processing to CDR synthesis. By monitoring the CDR signaling messages, it can provide an analysis basis for the various services of the 5G network.

In the recent years, many scholars have researched on signaling in different interfaces or protocols of mobile communication network. In [3], it used secondary re-hashing to complete CDR synthesis by studying the signaling monitor of layer 3 for Uu interface in LTE network. This method can avoid the clustering of hash collision elements in a certain area, but increased the search time. In [4], the authors proposed an effective monitoring scheme based on the study of signaling for S6a interface in LTE network, which used hash table instead of binary tree to store CDR key-value. In [5], NAS protocol monitoring in LTE-A network are researched; In [6], a multi-protocol association technology was proposed for Uu interface to monitor signaling association and synthesis in LTE-A network. Both of them completed the CDR synthesis by using separate chaining. This method can effectively reduce the clustering of hash elements when hash collisions occurred, but the disadvantage is that it will waste a lot of time to frequently visit memory because of using sequential search.

Therefore, this paper will propose a new scheme based on the traditional LTE-A signaling monitor, which is suitable for RRC protocol in 5G network to realize real-time monitoring and data analysis. Furthermore, in order to solve problems such as long search time and low index efficiency in the traditional signaling synthesis algorithm, an improved dynamic hash signaling synthesis algorithm based on AVL tree is proposed. It can efficiently and quickly process signaling messages under the massive user data in 5G network, so as to accurately and fully complete signaling monitor. Analysis results show that the improved signaling synthesis algorithm applied to the signaling monitor module in the 5G road tester can significantly improve the CDR synthesis efficiency and reduce memory consumption.

2 Overview of RRC Protocol

The RRC protocol which is the most complex one in the air interface is responsible for the management and control of the radio resources of 5G network and it's the control center of the whole assess layer [7]. The functions of RRC protocol mainly include radio bears control and mobility management, system message broadcast, paging, RRC connection management [8]. In order to meet the requirements of mMTC service for low power consumption and large connection, a new RRC state (RRC connected inactive) has been proposed in 5G systems. The high-level characteristics of RRC Inactive are similar to LTE's Light Connected [9]. This new state enables the UE to maintain a connection to the core network at all times and save the UE context so that it

can quickly return to RRC Connected state. Figure 1 is the RRC state model for the 5G radio access network.

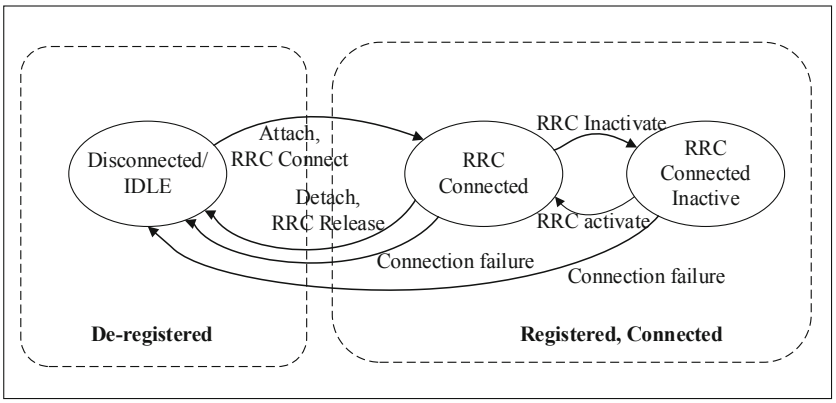


Fig. 1. RRC state model for the 5G radio access network.

3 System Module Design

The signaling monitoring module in 5G road tester combines the traditional signaling monitor technology in the LTE-A with data acquisition technology to complete the whole process from data acquisition to signaling analysis. Considering the characteristics of signaling in the RRC protocol and the application requirements, module is divided into four parts: data preprocessing, protocol decoding, CDR synthesis and statistics module. Adopting the thought of module design greatly improved independence of modules and subsequent extensibility. The design of signaling monitor module in 5G road tester is shown in Fig. 2.

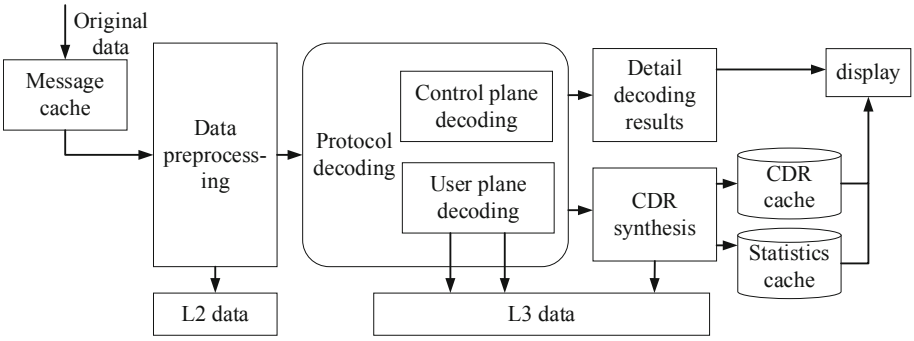


Fig. 2. Signaling monitor module framework in 5G road tester.

3.1 Data Preprocessing Module

The function of the module is mainly to receive the original signaling data from the data acquisition module and convert them into data structures such as binary code stream that can be recognized by the system, including three units of data elimination, protocol recognition and storage. Data elimination is to eliminate the underlying protocol data irrelevant to subsequent RRC protocol decoding and synthesis in the original signaling data. The control plane data mainly includes RRC and NAS protocol, and system information block (SIB), master information block (MIB), signaling radio bears such as SRB0, SRB1 and SRB2. User plane data mainly includes PDCP, SDAP and IP, HTTP protocols. The data type of messages can be judged by identifying the channel type contained in the data header. Then the message is respectively stored at the end of the queue for control plane data and user plane data. The channel types and data types in the data header are shown in Table 1.

Table 1. The relationship of data types and channel types.

Data types	Channel types
Control plane data	BCCH
	PCCH
	CCCH
	DCCH
	MCCH
User plane data	DTCH

3.2 Protocol Decoding Module

RRC protocol messages are transmitted through Signaling Radio Bear (SRB) and it's divided into six processes: paging, RRC connection establishment, secure activation, RRC connection configuration, RRC connection re-establishment and release. The RRC decoding module uses the ASN.1 compiler to generate the decoding function of each signaling message in RRC protocol, and then completes the simple decoding and detailed decoding. The Protocol Data Unit (PDU) of the RRC protocol is expressed in an abstract language and multiple types of information elements are defined nested within the message. The decoding module puts the data structure of the decoded results on the stack, which can be called by the synthesis module, statistics module. When a valid RRC message is received, the corresponding data type of this message is obtained by judging the logical channel type (BCCH、PCCH、CCCH、DCCH、MCCH), transmission channel type (BCH、PCH、DL-SCH) and transmission direction (uplink、downlink) of RRC message, the decoding function of the data type is called to decode the message. The specific decoding process is shown in Fig. 3.

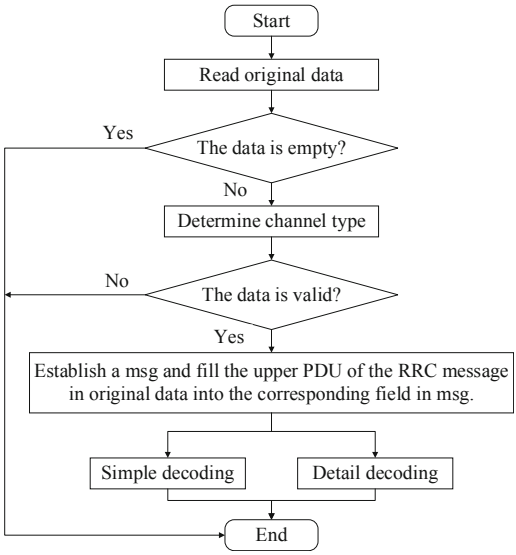


Fig. 3. Decoding process.

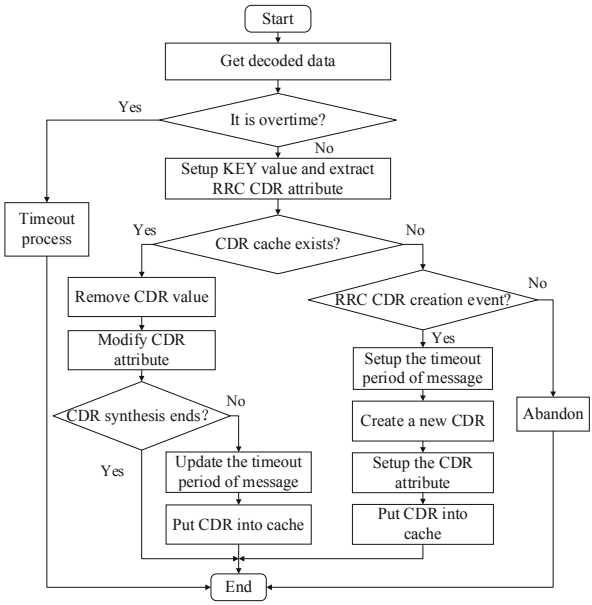


Fig. 4. RRC protocol CDR synthesis process.

3.3 CDR Synthesis Module

CDR synthesis module is the core module in the RRC protocol monitor scheme. It contains four steps: create, modify, store and delete. The key field information needs to be extracted from the protocol decoding module before signaling processes are analyzed. The information includes protocol type, CDR creation event, CDR completion event, associated key value. The specific implementation process is shown in Fig. 4. Each message from protocol decoding module needs to complete a timeout check. If it's overtime, the node associated with the key value in this message will be deleted and exit the synthesis process. Otherwise, extract the key value and CDR attribute. The message type is determined by the key value. Then, judge whether the CDR structure cache corresponding to the key value exists. If it exists, the message will be placed at the end of the corresponding CDR structure queue and modify its CDR attribute. The end of CDR synthesis is determined by judging whether the message type belongs to a CDR completion event in RRC signaling. If not, update the timeout period of message and put CDR of cache. Otherwise, the CDR synthesis process is ending. If the CDR cache isn't exist, a new CDR is created and its property value is placed in the CDR structure cache by identifying whether the message belongs to a CDR creation event in RRC signaling. Repeat the process until all the subsequent messages are associated and synthesized into the created CDR records.

3.4 Statistics Module

Statistics module is responsible for collecting related protocol processes and signaling in air interface. The messages are classified according to service types, procedure types and specific message types, which form a set to calculate the number of protocol messages and some statistical indicators. It is convenient for users to monitor and check abnormal data and print error information in real time. Because this module is used to synthesize the related information for the CDR synthesis, the trigger of the statistic function is merged into the interface function in the CDR synthesis module.

4 Algorithm Design

4.1 Hash Signaling Synthesis Algorithm

There is a huge amount of signaling data in 5G network at any moment. In order to increase efficiency of CDR synthesis, received signaling messages need to establish a hash index by certain rules and realize the one-to-one mapping between the Key value and the hash table. Firstly, the algorithm selects the Key from the RRC protocol association identifier as the independent variable, and calculate value which must be the same and unique by hash function. Then the Key value is mapped to the hash table to visit the CDR synthesis record when the query begins.

When the number of queried Key is much larger than the size of the hash table, hash collisions are inevitable. Traditional hash signaling synthesis algorithm (HSSA) used to resolve hash collisions include Open Addressing (Quadratic Probing, Linear Probing, and Rehashing/Double Hashing) and Separate Chaining [10, 11]. Open Addressing is simple to insert and search data. But the disadvantage is that the hash table will be full and memory will be accumulated if hash collision occurs. By contrast, Separate Chaining can avoid these problems. It's easier to change the linked list when inserting or deleting some elements. But it will waste some time to visit memory when the Key value is queried in the hash table because it uses Sequential storage structure.

4.2 Improved AVL-HSSA Algorithm

In order to solve problems such as long search time and low index efficiency in traditional hash signaling synthesis algorithm, this paper proposed an improved hash signaling synthesis algorithm based on AVL tree (AVL-HSSA). The tree structure is used to reduce the time of querying CDR cache in the hash table, so as to quickly deal with hash collisions and improve the timeliness of signaling synthesis.

The improved algorithm need to construct hash function and select the Key value before building hash table. Because there is a large amount of irregular signaling data from different users and protocols before signaling synthesis, this paper uses remainder of division method to construct hash function, which can quickly and uniformly distribute the data and realize the mutual mapping between Key and value [12]. Specifically speaking, the same user is identified by the source IP, the source port number, the destination IP address, and the destination port number. And we select IMSI, C-RNTI and CellID as Key value to realize CDR synthesis in the same signaling process from the same user. KEY_{i} is the Key of a signaling message. The Key value is divided by the size of hash table–N, then the result is $H_i(KEY)$, also known as Hash address. Its specific expression is as follows:

$$KEY = KEY_{\mathrm{i}}.IMSI + KEY_i.C{-}RNTI + KEY_i.CellID \tag{1}$$

$$H_{\mathrm{i}}(KEY) = KEY \bmod N \tag{2}$$

$$Hashed\_Addr = H_i(KEY) \tag{3}$$

The HSSA algorithm stores the root node of the tree in a hash table without a hash collision. Once collisions are occurred, it will use Hashed-AVL structure to store collision node. Each node in the tree contains a pointer for the left and right subtrees, a Key value for comparing node size, and a balance factor. The Key value of each node in the left subtree is smaller than the value of the root node, and the Key value of each node in the right subtree is larger than the value of the root node. The height difference between left and right subtrees isn't more than one [13]. The hash table structure when the AVL-HSSA algorithm handles collisions is shown in Fig. 5.

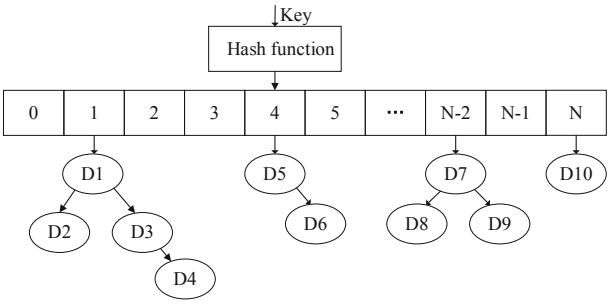


Fig. 5. Hash table structure of AVL-HSSA.

The specific search steps of the algorithm are as follows:

Step 1: extract the signaling association identification KEY_i from the protocol decoder before CDR synthesis, and find the corresponding hash address by hash function.
Step 2: judge whether hash collision occurs. If not, execute step3; otherwise, execute step4.
Step 3: insert the Key as root node into the hash table if the hash address hasn't any Key cache.
Step 4: insert the Key with the same hash address into the corresponding AVL tree and compare it with the Key value in the root node. If the queried Key is larger than the root node, it is compared with the nodes of the right subtree until the queried Key find its CDR cache and modify the CDR property [14, 15].

Algorithm: Searching of key in AVL-HSSA

```
1: procedure SEARCH(key)
2: hash = hashValue(key)
3: key_node = searchForHash
4: if key_node is NULL then
5:     create a new root node in AVL and set data
6: else
7:     Node = searchfordata
8:     while the key value is large than Node
9:         Node = Node.rchild
10:    while the key value is less than Node
11:        Node = Node.lchild
12:    return Node
```

4.3 Analysis of Algorithms

Usually, we use average search length (ASL) to evaluate the performance of the hash signaling synthesis algorithm in the process of searching signaling. Both Chain-HSSA and AVL-HSSA are analyzed in Table 2 when the Key value is successfully queried.

Table 2. Comparison of signaling synthesis algorithm

Algorithm	KEY search method	Average search length	Average time complexity
Chain-HSSA	sequential search	$(n+1)/2$	$O(n)$
AVL-HSSA	AVL tree search	$\log_2(n+1)-1$	$O(\log n)$

Chain-HSSA algorithm uses a sequential search method to compare the queried Key with each Key in the single linked list in turn until the same Key value is found. Its average search length is (n + 1)/2 and its average time complexity is O(n). It will waste a lot of time to compare CDR cache if there are a lot of number in linked list, so the efficiency of query is low. AVL-HSSA proposed in this paper breaks through the original storage structure, so that the AVL search tree can be used in the hash table. Using AVL tree to query the Key value may improve the efficiency of signaling synthesis and reduce the query time. Its average search length is $\log_2(n+1)-1$ and its average time complexity is O(logn).

5 Analysis of Experimental Results

In order to evaluate the performance of the proposed AVL-HSSA algorithm in signaling synthesis, we respectively compared Open-HSSA and Chain-HSSA in the traditional hash signaling synthesis algorithm with AVL-HSSA to finish the test. The experimental environment of this paper is: Windows 10 operating system, Visual Studio 2017 compiler running environment and the hardware configuration is 64-bit operating system, the processor is Inter(R) Core(TM) i5-8500 CPU 3.00 GHZ. The C-RNTI allocated by the gNB to the UE is used as the Key value in the signaling synthesis. The test data source of this experiments is obtained by extracting the dynamic identifier C-RNTI from the decoding module.

5.1 Search Time Analysis

The load factor α is a marker to measure how full a hash table is. When α is 1, we respectively select 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000 data from the test data with LP-HSSA, Chain-HSSA and AVL-HSSA to test. The test results are shown in Fig. 6.

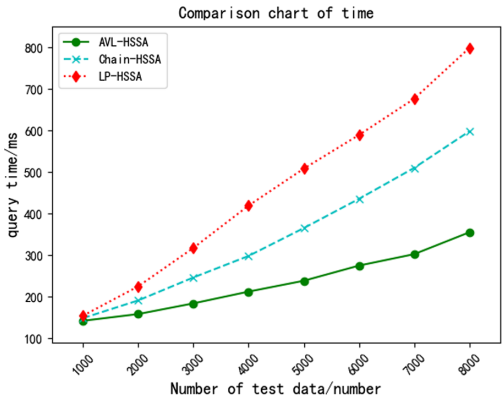


Fig. 6. Comparison of search time ($\alpha = 1$).

When there are the same number of data, Open-HSSA consumes the most search time and owns the least efficiency. By contrast, Chain-HSSA consumes less search time than Open-HSSA, but it is still higher than AVL-HSSA algorithm. As the number of data increases, the comparison of search time required by the three algorithms becomes more obvious. In the above test results, the average search time of the improved AVL-HSSA is reduced by 49.3% and 33.1% compared with Open-HSSA and Chain-HSSA.

We select 8000 test data from the data source and use three algorithms under different loading factors to finish the test. As shown in Fig. 7, when α is 1/10, the probability of hash collision is small because the length of hash table is longer than the number of queried Key. For Open-HSSA, the method can directly locate into the hash table by hash function when the queried Key exists in the hash table, so its search time of 269.7 ms is shortest. Using AVL-HSSA to locate into hash table is the same as Open-HSSA, the difference of them is that AVL-HSSA use the AVL tree to search data, so the search speed of the algorithm is slightly slower than that of Open-HSSA when α is 1/10, and the search time is 276.4 ms. By contrast, Chain-HSSA uses sequential search to visit linked list. Therefore, it needs to consume more time to visit memory.

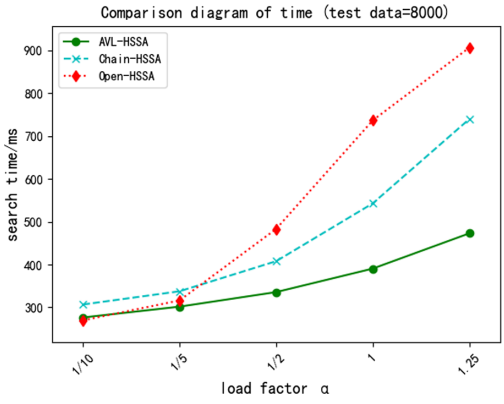


Fig. 7. Comparison of search time (test data = 8000).

When α is 1/2, the search time of Open-HSSA is obviously higher than the other two algorithms and its efficiency is lowest. By contrast, the AVL-HSSA algorithm has the shortest search time. And the greater the loading factor becomes, the more obvious the advantages of this method is. Its stability and adaptability are much higher than the other two hash signaling synthesis algorithms.

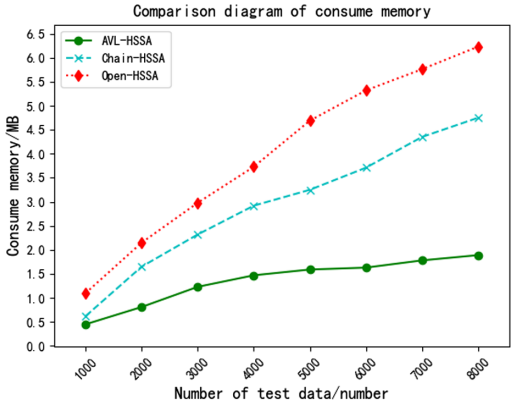


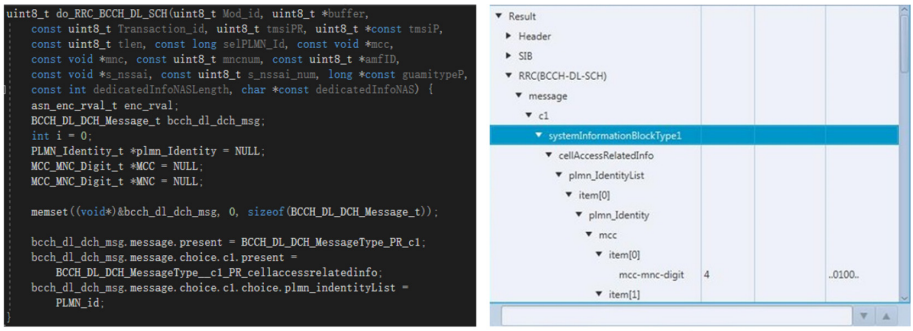
Fig. 8. Comparison of consume memory.

5.2 Memory Consumption Analysis

The comparison diagram of consume memory for AVL-HSSA algorithm and LP-HSSA, Chain-HSSA algorithm in the process of signaling synthesis are shown in Fig. 8. It can be intuitively seen from the figure that AVL-HSSA algorithm has an advantage in memory consumption, which occupies the lowest memory. The LP-HSSA algorithm allocates memory for all table items when it starts to create a hash table, and it will consume the most memory because of memory accumulated when the number of data source is more in the signaling synthesis. Chain-HSSA algorithm dynamically applies for memory when inserting into a hash table, and the memory occupied will increase with the more number of test data.

5.3 Analysis of Signaling Monitoring Scheme Test Results

The signaling decoding results show the protocol data information with the form of tree structure in Fig. 9. According to the definition of RRC BCCH-DL-DCH message in the 3GPP standard, each field in the decoding tree matches the definition of the message. The field value of mcc-mnc-digit is 4, and the bitmask is 0100.



(a) (b)

Fig. 9. RRC BCCH-DL-DCH message decoding. (a) The function declarations. (b) Detail decoding result

6 Conclusion

By studying the RRC protocol in a new 5G network architecture, this paper proposed a signaling monitor scheme of RRC protocol based on Hashed-AVL and discussed the specific function of its main submodule. The implementation process of the protocol decoding module and the CDR synthesis module was introduced in detail. Focused on the low query efficiency and high average traversal time complexity in the traditional signaling synthesis algorithm, an improved dynamic hash signaling synthesis algorithm under a new network architecture is used to reduce the query time of traditional algorithm in the hash table, so as to quickly deal with hash collisions and improve the real time on signaling synthesis. It shows that the system is effective and feasible. The signaling synthesis scheme described above achieves the expected effect and signaling messages in RRC protocol can be accurately decoded and monitored in real time.

Acknowledgment. It is supported by the Science and Technology Major Project in Chongqing (R&D and application of 5G road test instruments: No.cstc2019jscx-zdztzxX0002).

References

1. Hucheng, W., Hui, X., Zhimi, C.: Current research and development trend of 5G network technologies. Telecommun. Sci. **9**, 149–155 (2015)
2. Navarro-Ortiz, J., Romero-Diaz, P., Sendra, S., et al.: A survey on 5G usage scenarios and traffic models. IEEE Commun. Surv. Tutorials, **22**(2), 905–929 (2020). Secondquarter
3. Jiang, Y.: Design and implementation of LTE Uu Interface Layer 3 Protocol Monitor System. Beijing University of Posts and Telecommunications (2014)
4. Yingying, H., Bing, X., Zhizhong, Z.: Research and implementation on synthetic scheme of S6a interface in the LTE network monitoring system. Appl. Mech. Mater. **3670**, 237–240 (2015)
5. Pei, P., Longhan, C., Zhizhong, Z.: Research of NAS protocol monitor scheme of air Interface in LTE-A network monitoring instrument. Video Eng. **39**(21), 44–48 + 60 (2015)

6. Lei, L., Zhizhong, Z., Bing, X.: Research and implementation of multi-protocol association scheme on Uu interface in LTE-Advanced network. Telecommun. Sci. **32**(6), 167–176 (2016)
7. Ryoo, S., Jung, J., Ahn, R.: Energy efficiency enhancement with RRC connection control for 5G new RAT. In: 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, pp. 1–6 (2018)
8. 3GPP, NR., radio resource control (RRC) protocol specification (release 15).: 3GPP 38.331 (2019)
9. Hailu, S., Saily, M., Tirkkonen, O.: RRC state handling for 5G. IEEE Commun. Magazine **57**(1), 106–113 (2019)
10. Gou, X., et al.: Single hash: use one hash function to build faster hash based data structures. In: 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, pp. 278–285 (2018)
11. Atighehchi, K., Rolland, R.: Optimization of tree modes for parallel hash functions: a case study. IEEE Trans. Comput. **66**(9), 1585–1598 (2017)
12. Agrawal, A., Bhyravarapu, S., Venkata Krishna Chaitanya, N.: Matrix hashing with two level of collision resolution. In: 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, pp. 14–15 (2018)
13. Chinnaiyan, R., Kumar, A.: Construction of estimated level based balanced binary search tree. In: 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, pp. 344–348 (2017)
14. Dhar, S., Pandey, K., Premalatha, M., Suganya, G.: A tree based approach to improve traditional collision avoidance mechanisms of hashing. In: 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, pp. 339–342 (2017)
15. Cao, Y., et al.: Binary hashing for approximate nearest neighbor search on big data: a survey. IEEE Access **6**, 2039–2054 (2018)