Scalable and Self-configurable Eduroam by using Distributed Hash Table

Hiep T. Nguyen Tri, Rajashree S. Sokasane, Kyungbaek Kim
Dept. Electronics and Computer Engineering
Chonnam National University
Gwangju, Republic of Korea
e-mails: {tuanhiep1232@gmail.com, sokasaners@gmail.com, kyungbaekkim@jnu.ac.kr}

Abstract—**In the recent years, the number of increased Wi-Fi networks and Wi-Fi-enabled devices shows how fast Wi-Fi technology is growing. Since a single network provider is usually not able to ensure Wi-Fi coverage for its own users across many geographic locations, we need Wi-Fi roaming. Eduroam is a Wi-Fi roaming system, which allows a user of a domain to access wireless resources in another domain with the unique credential of the user managed in the original domain. The authentication process in Eduroam is based on the hierarchical tree structured Remote Authentication Dial-In User Service (RADIUS) servers over wide area networks. Existing RADIUS-based tree structure of Eduroam is not self-configurable; joining/leaving of node is not automatically handled by the existing approach and it takes high communication delay as well. In order to improve the scalability of Eduroam with self-configurable feature and reduce communication delay as compared with tree structure-based Eduroam, we hereby proposed a Scalable & Self-configurable Eduroam by using Distributed Hash Table (DHT). Through a prototype implementation, we showed that the proposed system supports high scalability and high fault tolerance.**

Keywords-DHT; Eduroam; RADIUS server; Wi-Fi roaming.

I. Introduction

Wi-Fi technology has become increasingly popular due to its flexibility and mobility; as a result, the need of Wi-Fi roaming systems is increasing. The Eduroam [1] is a secure roaming system between educational institutions. The Eduroam allows users to access the Internet with their own credentials at visiting institution during roaming. The Eduroam principle is based on the fact that the user's authentication is done by the user's home institution, whereas the authorization decision allowing access to the network resources is done by the visited network. The authentication process of Eduroam is based on hierarchical tree structured RADIUS servers. However, hierarchical tree structure approach in Eduroam causes long communication delay, and also exposes a single point of failure because every authentication traffic flows through the tree hierarchy even though it is only of interest to a leaf RADIUS server. To overcome these issues of tree structured Eduroam, we developed a Flat Layer RADIUS server model with Eduroam in our previous work [9]. The Flat Layer RADIUS server model effectively reduces communication delay and avoids single point of failure.

In the Flat Layer RADIUS server model, we assumed that every node in the network must know about all other nodes in the network. In the Flat Layer RADIUS server model, each node (RADIUS server) directly communicates with each other, without using any intermediate RADIUS proxy servers. To evaluate the performance of the Flat Layer RADIUS server model and compare with tree structure model, we setup experiments by using open source based freeRADIUS (version 2.1.8) and ubuntu (version 10.04.4) as RADIUS server. Table I shows the authentication time comparison between the tree structure and the Flat Layer RADIUS models. Note that the authentication time includes request forwarding process, authentication process, network latency and response forwarding process. Table II shows request processing time of three stages in authentication process. From Table I, we can observe that Flat Layer RADIUS server model takes less authentication time than RADIUS-based tree structures.

TABLE I. AUTHENTICATION TIME (μS)

	Tree structure		**Flat Layer RADIUS model**
	3 hops away	***2 hops away***	
Request Forwarding Process	1155	711	273
Authentication Process	330	237	242
Response Forwarding Process	559	278	134
Network latency	620823	402997	201330

TABLE II. REQUEST PROCESSING TIME

Process/machine	**Time in μs**
Request Forwarding	357
Authentication	270
Response Forwarding	162

However, the Flat Layer RADIUS server model may face the scalability issue. If a node operation, such as joining or leaving the network, takes place all nodes in the network need to be updated to stay up-to-date with latest membership information of the network. If the number of nodes in the network goes up, the data transfer between all nodes lead to overhead and updating operation to all nodes takes much time, it may cause for bottleneck. Flat Layer RADIUS server model works well with small scale, but when the members in the network are going to be increased the maintenance cost is also increased with it.

Recently, to resolve scalability issues in many distributed systems, DHTs have been largely adopted as a useful substrate to the design and specification of scalable and self-configurable distributed systems. The basic operation in DHT-based systems is lookup (key), which returns the node controlling the region of the space corresponding to that key. In the lookup structure, DHT nodes form an overlay network where each node has a number of neighbors. One lookup (key) messages are then routed through the overlay network to the node responsible for that key.

In this paper, we propose Scalable and Self-configurable Eduroam by using DHT, in order to improve the scalability of Eduroam and make it self-configurable in case of joining/leaving node operation takes place frequently. In the proposed system, RADIUS servers run on DHT substrate and they form a DHT-based RADIUS network. A RADIUS server manages a single domain. It uses the domain name as a key and uses its IP address and port number as the concatenated string for a value in the DHT-based network. In the DHT-based RADIUS network, node joins and leaves are handled automatically by obeying the updating rules of DHT. When a client sends an authentication request, RADIUS server which receives the request will forward it to the corresponding RADIUS server through a DHT lookup operation with the target domain name of the request.

We implement the DHT-based RADIUS network for scalable and self-configurable Eduroam by modifying the freeRADIUS server with the bamboo DHT substrate. Through the extensive evaluation and the implementation, we observe that the proposed system is scalable and self-configurable. The paper is organized as follows. Section II describes Eduroam, DHT and our previous work on the Flat Layer Approach. Section III explains the detailed design of the proposed system. Next, we evaluate proposed system in Section IV. Finally, Section V concludes the paper.

II. BACKGROUND

This section provides an overview of the Eduroam, authentication process and related protocols, such as RADIUS and DHT.

A. Eduroam

Eduroam was originally proposed by TERENA (Trans-European Research and Education Networking Association) [1]. Eduroam allows students, researchers and staff from home institutions to obtain Internet connectivity when visiting other institutions. The Eduroam principle [1] is based on the fact that the user's authentication is done by the user's home institution, whereas the authorization decision allowing access to the network resources is done by the visited network. Eduroam is based on the most secure encryption and authentication standards in existence today [1]. It gives an access to authorized users only.

The Eduroam is based on hierarchical structured RADIUS proxy servers and IEEE 802.1X. Figure 1 shows an example of the RADIUS proxy tree in Eduroam. When a user accesses an Access Point (AP) in the network of visited institution, authentication information is transmitted from visited institution to user's home institution through RADIUS proxy tree [2]. If the authentication is successful, the user can access the network of visited institution.

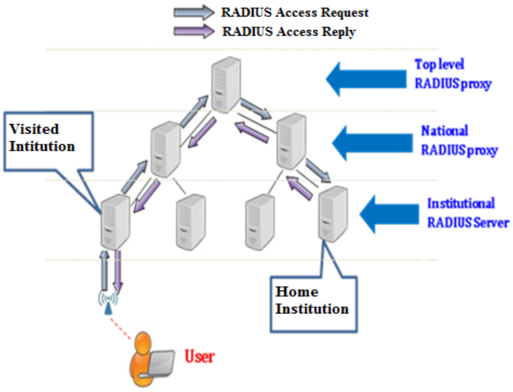


Figure 1. RADIUS-based tree structure in Eduroam.

When a user tries to log on to the wireless network of a visited Eduroam-enabled institution, the user's authentication request is sent to the user's home institution. This is done via a hierarchical system of RADIUS servers. The user's home institution verifies the user's credentials and sends to the visited institution (via the RADIUS servers) the result of such verification.

B. Ditributed Hasht Table

In a peer-to-peer system, every node in the system plays the same role; each node has a piece of system data. In this case, looking up data has an important role in the distributed system. DHT provides a lookup service to help peer-to-peer system or other distributed applications to locate data more efficiently. DHT uses a key that is generated by hashing function to locate node which contains the value [4][5][6].

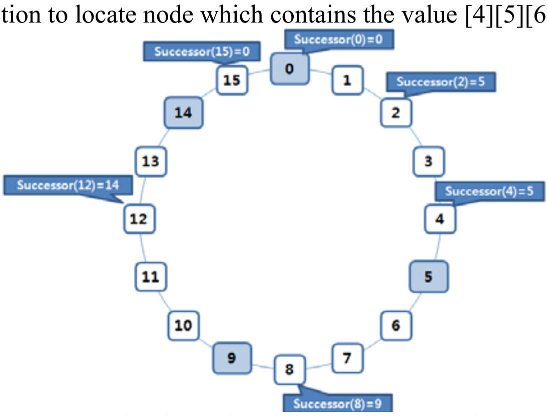


Figure 2. Identifier circle includes 4 nodes 0, 5, 9 and 14.

In a basic consistent hashing approach, nodes and value keys are hashed onto a circle ring, shown in Figure 2. The value keys are assigned to the nearest peer in the clockwise direction. Each node in the network maintains a routing table, which contains references of other nodes. In Pastry [6], the leaf set contains information about the closest node in identifier space. Routing table contains information of nodes whose identifier shares the present node’s identifier in the first n digits and whose $(n+1)^{th}$ digit has one of the 2b-1 possible value. The neighborhood set contains information of the closest (according the proximity metric) nodes.

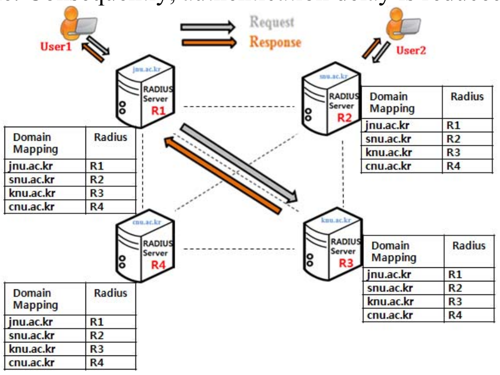
When there is a request to lookup data which is mapped to a key, firstly, Pastry lookups in leaf set. If the key identifier is within range of leaf set the message will be forwarded to the closet node in leaf set; otherwise, Pastry lookups routing table. The message is forwarded to a node that shares a common prefix with the key at least by one more digit. If there is no node in routing table satisfying that condition, the message will be forwarded to the node that shares a prefix with the key at least as long as the local node. The node is chosen in a set, is built from leaf set, routing table and neighborhood set.

DHT is conventional and popularly used in Peer-to-Peer system; BitTorrent [10][11][12][13][15] are examples of system or research related to DHT. DHT is also considered to support the Domain Name System (DNS) [12][13][15]. DNS which is used to translate domain name into IP address is a hierarchical distributed naming system [14]. Each node or leaf has resource records which contain domain name information. The tree is divided into zones. Each zone contains one or many domains. The tree begins at root zone. When a client wants to look up a domain name, the client sends request to root. Based on resource records, the root returns information of next node. If the node is responsible for the domain name, the node will return the IP address and the lookup process will end; otherwise, the node will return the information of next node. Cox implemented and evaluated DDNS, which is a system that is based on Chord and has the same function of DNS [15]. DDNS is self-configurable, so DDNS eliminates the pain of name server administration. DDNS also inherits good load balancing and fault tolerance. As we can see, DNS has the similar structure with Eduroam. Therefore, combination of DHT and Eduroam puts forward a promise to improve scalability and self-configuration to the Eduroam system.

C. *Flat Layer Approach*

In order to improve the performance of the authentication process in Eduroam, a Flat Layer approach was proposed by Sokasane and Kim [9]. In the Flat Layer RADIUS server model, the authentication delay is reduced because the visited institution server directly forwards the request to the home institution server. Flat Layer approach also helps system to avoid single point of failure problem.

Figure 3 depicts the structure of Flat Layer approach. When the visited institution RADIUS server receives an authentication request from user proxy by AP, the RADIUS server checks the user information in its database. If the domain name contained in the authentication request is a local domain, the authentication request will be handled locally; otherwise, the visited institution's RADIUS server will forward the authentication request directly to the home institution RADIUS server, based on the domain name information provided in the request. In order to direct forwarding authentication request to home institution server, each RADIUS server in system must know the information of all other RADIUS servers in the system. So, the total hop count message that passes through to reach the destination is just one. Consequently, authentication delay is reduced.



Domain Mapping	Radius
jnu.ac.kr	R1
snu.ac.kr	R2
knu.ac.kr	R3
cnu.ac.kr	R4

Domain Mapping	Radius
jnu.ac.kr	R1
snu.ac.kr	R2
knu.ac.kr	R3
cnu.ac.kr	R4

Domain Mapping	Radius
jnu.ac.kr	R1
snu.ac.kr	R2
knu.ac.kr	R3
cnu.ac.kr	R4

Domain Mapping	Radius
jnu.ac.kr	R1
snu.ac.kr	R2
knu.ac.kr	R3
cnu.ac.kr	R4

Figure 3. Flat Layer Approach.

Even though the Flat Layer approach reduces the authentication delay as well as avoids single point of failure, it has some disadvantages. The first disadvantage is scalability problem. As we said before, each RADIUS server must know the information of all other RADIUS servers present in the system. If a new RADIUS server joins or leaves, all RADIUS servers need to update their information. If there are a large number of RADIUS servers in system, the cost of updating information is very big.

III. PROPOSED SYSTEM

Although Eduroam is the secure roaming system between research and educational institutions, it has some disadvantages, especially with its RADIUS-based tree structure. Every authentication traffic flows through the whole hierarchy [3] even though it is only of interest to a leaf RADIUS server that causes high communication delay. Also, the existing RADIUS-based tree structure is not self-configurable and raises scalability issues.

To mitigate the disadvantages of Flat Layer RADIUS server model, we proposed a peer-to-peer based Eduroam approach. In this section, we will present proposed system in detail.

In DHT, data is distributed across nodes by using the hash function, and a routing scheme is implemented to efficiently look up the node on which data item is located. In DHT-based RADIUS server model, each node knows information about related nodes only. DHT provides a protocol for looking up the node in which data item is located [7]. DHT has some advantages, such as scalability, availability [5], self-configuration, and it only affects a set of nodes rather than every other node in system. When a new node joins the system, the system will redistribute data. The new node is automatically configured to build its routing table. This process will affect only a small set of nodes in the system, instead of affecting all nodes in the system. It helps to reduce the cost of data transferred.

In our proposed system, we consider DHT as lookup service. Figure 4 shows the architecture of proposed system. The system includes AAA servers (RADIUS) and DHT agents (or DHT nodes), which are parts of DHT system. In the proposed system, RADIUS server contains domain information of related domain(s) only rather than all domain(s) information present in the system. DHT agents play a very vital role. A DHT agent works as lookup service, which responds with the information of requested domain(s). In the proposed system, the DHT agent is responsible for finding appropriate domain information and sending it to visited institutions RADIUS server, then RADIUS server of visited institution sends user information to user's home institutions RADIUS server to get verification of user.

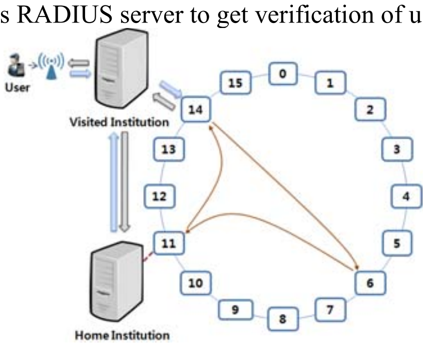


Figure 4. Architecture of Scalable and Self-configurable Eduroam.

The workflow of the proposed system is as follows (see Figure 6). While roaming, when a user wants to access the network of visiting institution through the credentials of home institution, the user needs to send an authentication request with user@domain.name format. At this point, the RADIUS server checks the domain name that user requested for; if the domain name of the authentication request is local domain name of the authentication server, authentication server will find it in database of local server. The database can be a database server or just a configuration file. If the username exists in the database, authentication server will respond Accept-Accept message otherwise authentication server respond Accept-Reject. If the domain name of the authentication request is not a local domain name of the authentication server; this is the case when a user wants to access the visited institution server. The visited institution's RADIUS server does not contain the user information; so, the visited institution RADIUS server needs to forward the message to the home institution's RADIUS server, where the user information is kept. After that, the visited institution's RADIUS server will send lookup request to the node called visited DHT node of DHT system. Each node in the DHT system contains a unique domain name regarding the domain name of the user. The visited DHT node employs the lookup function of the DHT system to send a lookup message to the home DHT node that hold home institution's information. The home DHT node will send a response message which contains the home institution server information to the visited DHT node. And then, the visited DHT node sends the response message to the visited institution's RADIUS server. The visited institution's RADIUS server forwards the authentication request to the home institution's RADIUS server based on the information that the DHT system returns. In the home institution's RADIUS server, the process is almost same with the first case except the response message is returned to the client through the visited institution's RADIUS server.

In the next sub-sections, we will discuss more detail about AAA server and DHT agent.

A. *Implemeting* AAA server

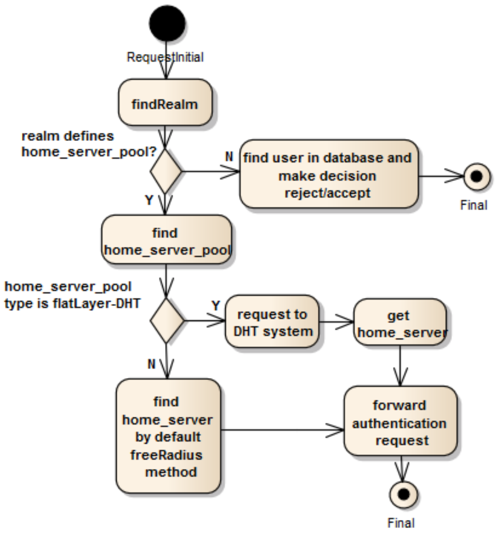


Figure 6. Work flow of DHT-based RADIUS server module.

In the Eduroam system, the freeRADIUS is used as an AAA server. The authentication process of the freeRADIUS is as follows; firstly, the freeRADIUS finds the correct realm based on the domain name given in the request.

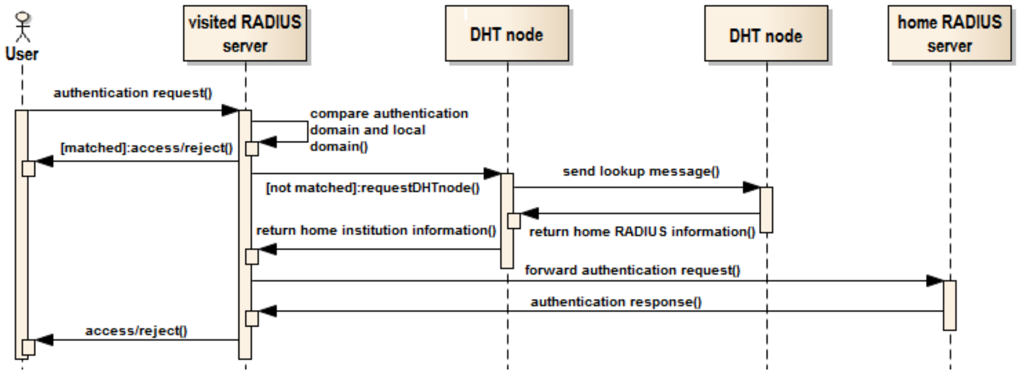

User
visited RADIUS
server
DHT node
DHT node
home RADIUS
server
authentication request()
compare authentication
domain and local
domain()
[matched]:access/reject()
[not matched]:requestDHTnode()
send lookup message()
return home institution information()
return home RADIUS information()
forward authentication request()
authentication response()
access/reject()

Figure 5. Proposed system flow.

Realm is an object that contains information to make the decision, either the request is forwarding to the other domain or to handle it locally. If the founded realm does not contain any server pool information, the requested domain name is the local domain and it should be handled by the local server; otherwise, the freeRADIUS finds the server pool for the requested domain name. The server pool contains information of the candidate home institution servers. In the home server pool, the candidate home servers and its type (property- that controls how home servers are chosen) are listed. By default, the type property can be assigned to one of the 6 values including fail-over, load-balance, client-balance, client-port-balance and keyed-balance. Based on the type value, server can choose the appropriate home institution server to forward the request.

With the backward compatible purpose, if in the case, we need to request to the DHT system to get information of the home institution server, we decided to assign a new value to the home server pool type. We added a module in the freeRADIUS, which is responsible for requesting and getting results from the DHT system. If the type property of home_server_pool object is “flatLayer-DHT”, then the newly added module will be executed. Figure 5 shows the work flow of the DHT-based RADIUS server module in the proposed system.

B. *Implementing DHT agent*

A DHT agent node is a part of DHT-based system, which works as lookup service of the proposed system. We decided to use bamboo-dht [8] as a routing layer to implement the DHT agent. Figure 7 shows the architecture of DHT agent. The freeRADIUS service module is responsible for receiving the requests from the freeRADIUS and returning the results. Bamboo routing is a routing layer module that provides routing API. After receiving a request from the freeRADIUS, the freeRADIUS service uses the bamboo routing layer to send a lookup message to the DHT node, which holds the information of the home institution’s server. When the destination DHT node receives the lookup message, the destination DHT node send a response message directly to the source DHT node, by using information attached in the lookup message. The source DHT node uses the message information to return to the freeRADIUS.

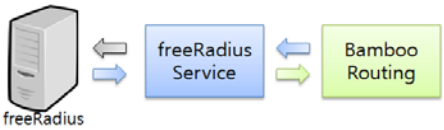


Figure 7. DHT agent architecture.

In this system, the domain name works as an identifier of each DHT node. Therefore, DHT system uses the domain name instead of the IP address and port to make an identifier. When a DHT node tries to join the system, it will send a joining request to the gateway node. The gateway node will send a message to the DHT node, that is identified by the domain name of the joining request and wait for the response. If the gateway node receives the response message within time out, the gateway node will reject the joining message. This will help to eliminate conflicting identifier in the DHT system.

IV. Evaluation

In order to evaluate our proposed system, we setup an environment in which we used 4 VMware machines. Each machine has CPU 3.4 GHz single core, 1 GB RAM and is running 64-bit ubuntu OS (version 10.04.4). We installed edited freeRADIUS (version 2.1.8) and bamboo-DHT which was modified based on the version released on March 3, 2006. On each machine we deployed the number of DHT nodes and the freeRADIUS with different configurations.

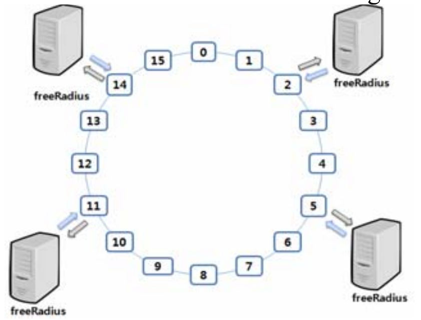


Figure 8. Evaluation architecture.

To evaluate the scalability of system, we conducted some experiments with the different number of the DHT nodes and the freeRADIUS nodes to obtain the hop count number. Figure 8 shows an example of test-bed system. Since visited institution’s server forwards the request directly to the home institution’s server, there is a difference between the number of DHT nodes and freeRADIUS nodes. Concretely, the number of freeRADIUS nodes increases from 3 to 10 and the number of the DHT nodes increases from 5 to 800.

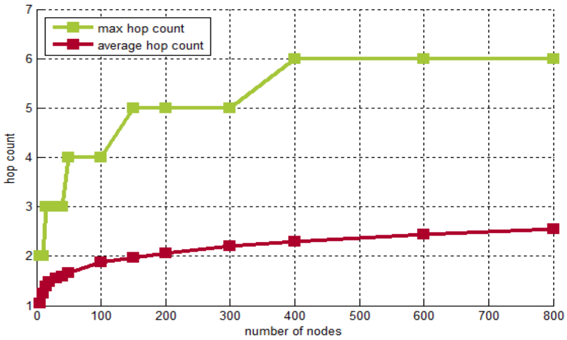


Figure 9. DHT look up hop count.

Figure 9 shows the number of hop count which is required to reach up to the destination DHT node. We can observe that the number of hop count is directly proportional to the number of nodes, but the increase speed of hop count is inversely proportional to the number of nodes. Although the number of nodes is up to 800 but the average hop count is 2.55.

Figure 10 shows the join time. The join time is a time period that a node needs to join completely the RADIUS network. During the join time, a node is unavailable. To measure the join time, we continuously start DHT nodes up to 800 and check the time when the node is available. From Figure 10, one can observe that the join time becomes more fluctuated with more number of nodes. The maximum join time is less than 1 second even though the number of DHT nodes increases up to 800.

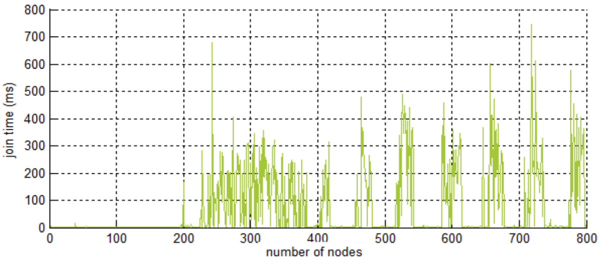


Figure 10. DHT node join time.

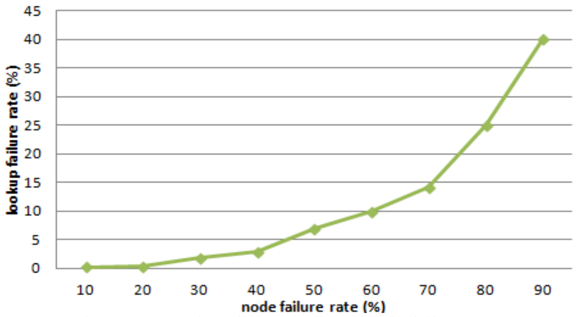


Figure 11. Lookup failure rate and node failure rate.

In order to test fault-tolerance of the system, we experimented in a network, which includes 800 DHT nodes. We simulated node's failure. We used a simple node failure model, which considers the crashing of node or the network failure. In the node failure model, all the failure nodes are simulated simultaneously. After simulated node's failure, we immediately sent successively lookup requests to the available DHT nodes, and we counted the number of failed lookup requests and total requests and calculated the lookup failure rate along with the node failure rate. The value of lookup timeout in this experiment is 10 seconds. Figure 11 shows the result of lookup failure rate and node failure rate. The lookup failure rate increases fast if the node failure rate is more than 40%. If the node failure rate is less than 40%, the lookup failure rate is small (<3%). The possibility of simultaneous failure of nodes is 40%, in reality is very small. Even though the failed node rate is 90%, the system is able to recover after just 1-2 minutes by itself.

V. CONCLUSION

In this paper, we presented experimental results of the Eduroam system that adopts DHT as a lookup service. The proposed system is scalable and self-configurable. We also evaluated fault-tolerance of the system. In the proposed system, the RADIUS servers run on DHT substrate and they form a DHT-based RADIUS network. A RADIUS server manages a single domain.

The proposed system should have a stable node which plays as a role of a gateway which will be used while new nodes join the system. When a new node wants to join, it requires connecting to one available node in the system. The proposed system has to face the problem of handling routing function when there are a huge number of nodes leaving the system.

AKNOWLEDEGEMENT

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (NIPA-2014-H0301-14-1014) supervised by the NIPA(National IT Industry Promotion Agency).

REFERENCES

[1] Eduroam. https://www.eduroam.org [accessed: 2014-04-29]

[2] Y. Miyamoto, Y. Yamasaki, H. Goto, and H. Sone, "Optimization System of IP Address Using Terminal ID in eduroam" in Proceedings of 2011 IEEE/IPSJ International Symposium on Applications and the Internet, July. 2011, pp. 342 – 346, ISBN: 978-1-4577-0531-1.

[3] K. Wierenga and L. Florio, "eduroam: Past, Present and Future", Computational Method in Science and Technology, vol. 11 (2005), pp. 169-173, 2005.

[4] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network", In Proc. ACM SIGCOMM'01, San Diego, CA, Aug. 2001, pp. 161-172.

[5] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. "Chord: A scalable peer-to-peer lookup service for Internet applications", In Proc. ACM SIGCOMM'01, San Diego, CA, Aug. 2001, pp. 149-160.

[6] A. Rowstron and P. Druschel. "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", Proceeding Middleware '01 Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, 2001, pp. 329-350, ISBN:3-540-42800-3.

[7] Distributed Hash Table. http://en.wikipedia.org/wiki/Distributed_hash_table [accessed: 2014-04-29]

[8] The Bamboo Distributed Hash Table. http://bamboo-dht.org/ [accessed: 2014-04-29]

[9] R. Sokasane and K. Kim. "Flat Layer RADIUS Model: Reducing Authentication Delay in eduroam", In Proceedings of the 2nd International Conference on Smart Media and Applications, Kota Kinabalu, Malaysia, Oct. 2013, pp. 161-168.

[10] K. Kim and D. Park. "Efficient and Scalable Client Clustering for Web Proxy Cache" IEICE Transactions on Information and Systems, vol. E86-D, no. 9, Sept. 2003, pp. 1577-1585.

[11] K. Kim. "Lifetime-aware Replication for Data Durability in P2P Storage Network" IEICE Transactions on communications, vol. E91-B, no. 12, Dec. 2008, pp. 4020-4023.

[12] V. Pappas, D. Massey, A. Terzis, and L. Zhang "A Comparative Study of the DNS Design with DHT-Based Alternatives" In Proceedings of IEEE INFOCOM'06, Barcelona, Catalunya, Spain Apr. 2006, pp. 1-13, ISBN: 1-4244-0221-2.

[13] Y. Doi, "DNS meets DHT: Treating Massive ID Resolution Using DNS Over DHT", International Symposium on Applications and the Internet, Trento, Italy, Feb. 2005, pp. 9-15, ISBN: 0-7695-2262-9.

[14] Domain Name System. http://en.wikipedia.org/wiki/Domain_Name_System [accessed: 2014-04-29]

[15] R. Cox, A. Muthitacharoen, and R. Morris. "Serving dns using a peer-to-peer lookup service", In Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS), Cambridge, MA, Mar. 2002, pp. 155-165, ISBN:3-540-44179-4.