A Proposal of Autonomous Control of Server Relocation for Fog Computing Systems

Kouki Kamada[†], Hiroshi Inamura[‡], Yoshitaka Nakamura[‡]

[†]Graduate School of Systems Information Science, Future University Hakodate, Hakodate, Japan
[‡]School of Systems Information Science, Future University Hakodate, Hakodate, Japan
{g2119012, inamura, y-nakamr}@fun.ac.jp

Abstract - Fog computing, which extends the paradigm of cloud computing to the edge of networking, has been proposed, and its research has been active. In the field of networking, research on Content Centric Networks (CCN) has been conducted. CCN have been shown to be able to handle cached content naturally within the network, reducing traffic and latency. However, in today's Internet, dynamic content with dynamic services is indispensable. A system that can handle dynamic services is desired by incorporating the way of handling computational resources in fog computing into CCN. In this paper, we propose an autonomous control of server relocation for fog computing systems for server relocation and allocating resources. In addition, Furthermore, we perform simulations on show the basic performance of the proposed system.

Keywords: Contents Centric Network, Fog Computing, In-Network Caching, Server Relocation

1 Introduction

The number of IoT devices, which is 27.4 billion as of 2017, is expected to increase to about 40 billion by 2020[1]. For these large volumes of data generated by IoT devices, processing-intensive architectures such as cloud computing do not take advantage of the processing power of the edge and the latency from the point of data generation to the remote data centers cannot be ignored. Therefore, fog computing, which extends the paradigm of cloud computing to the edge of the network, has been proposed and actively studied[2].

In the field of networking, research on CCN (Content Centric Networks) such as NDN (Named Data Networking) has been carried out instead of the conventional IP address-based architecture[3]. It has been shown that CCN can naturally handle cached content in the network by using location- independent content as an identifier, which can reduce traffic and latency.

We proposed an autonomous control of server relocation for fog computing systems[4]. In addition, we improved the autonomous control of server relocation to transfer services on the fog network where the processing capacity is heterogeneous, so that the service transfer is commensurate with the required processing capacity[5].

In this paper, to optimize end-user QoS, we control server transfers in a fog computing environment to achieve both shortening of the average response time and fairness between users. For this purpose, we set up a use cases to examine the fairness between users in uniform computer resources.

2 Related Works

2.1 Fog Computing

In fog computing, the delay time for execution is reduced by selecting and transporting the points necessary for the execution process. For example, in Wireless Sensor and Actuator Networking, simple processing can be performed at intermediate nodes, such as fog nodes, before the data collected by sensor nodes are moved to the cloud, and the intermediate nodes can reduce the delay time by giving commands to actuation instead of the cloud. There is another technique called code-offloading[6]–[9]. Code-offloading is the use of mobile on resource-constrained mobile devices. This technology aims to improve the energy efficiency and execution speed of applications. Specifically, in a mobile application, we can use the node on fog that has more computational resources to execute the code, rather than running on mobile devices. With the decision, we can save resources such as batteries in mobile devices. In fog computing, the optimal allocation of computational resources is a focus, but there has been no discussion on the optimal placement of content.

2.2 CCN

Jacobson et al.[3] proposed a CCN that does not use the traditional IP addressing architecture and Two types of CCN messages, Interest and Data, are used in the CCN communication. This is done by a protocol that is based on the Messages can be sent and received through the FIB(Forwarding Information Base), CS(Content Store), PIT (Pending Interest Table) to send the data back to the requester, three main data structures are used. Using these data structures, CCN exchange messages between Interest and Data. The result retains the simplicity and scalability of IP but offers much better security, delivery efficiency, and disruption tolerance. In this way, CCN put content closer to the user, which allows static contents to be disseminated. However, to treat the running system, we need to care the internal state to continue the process, it is not possible to handle in the same way to provide dynamic content and services.

There is research on cache efficiency in CCN and how to route Interest packets efficiently[10]. These studies have been discussing the treatment of static content and how efficiently distributed content can be considered as transparent, and there is no discussion on how dynamic services can be distributed and deployed on the network.

2.3 Necessity of Fog computing and CCN integration

In order to solve the problems we have seen from the research mentioned so far, it would be useful to consider an architecture that allows us to control the deployment of services running in the cloud and dynamically redeploy them as needed. For example, it may be possible to optimize the point of execution of services by running them closer to the user.

Since the CCN is based on the idea of replacing the current TCP / IP with the CCN, there are several discussions on static content caching schemes. However, in today's Internet, which is created by real-world TCP / IP, dynamic content with dynamic services is essential. For example, there is a web page that authenticates the user and displays information appropriate for the user. We wondered if a static content caching scheme is not enough to replace the current Internet with a CCN because of the large amount of these dynamic contents. In the study of fog computing, there is little discussion on the issue of how to place data on a fog network. Therefore, we believe that the problems in the research fields of fog computing and CCN can be mutually resolved by incorporating the way computational resources are handled in the CCN, as introduced in 2.1, into the CCN.

3 Challenge

We consider optimizing quality of service by allowing services running on the network to be dynamically relocated. In this paper, we focus on response time as seen by the client as quality of service. We assume that the response time is expressed as the sum of the network transmission delay and the processing time at the server. When considering the response time, we need to optimize the system in two ways: fairness of the response time among clients and minimization of the processing time. An example is shown and discussed below.

3.1 Use case that require fairness in delay times

There is a need for autonomous resource allocation that satisfies the fairness of delay times for participants. For example, in an Internet conferencing system, media quality for all participants may not be maintained if the server is in a single location for clients distributed in different locations on the network with different latency. Therefore, there is a need for autonomous resource allocation that satisfies the equity of delay time for participants.

3.2 Assumptions and Requirements for Autonomous Control of Server Relocation System

We need a system that aims at fairness in average response time and shortening of service execution time among users simultaneously. We define the service response time which is the sum of the network latency between the client and server and the processing time of the service.

In addition, the system should reduce the average response time on non-uniform computer resources. For machine learning applications, where the execution time varies greatly depending on the processing performance, the processing time of the service becomes a bottleneck due to the processing performance. Therefore, by monitoring the processing performance of each node and transferring services based on the predicted service response time, services can be transferred to nodes with appropriate processing capacity.

In this system, we assume that the client has a fixed position in the network because it communicates directly with the sensor and user. On the other hand, since servers providing services are arbitrarily located in the fog/cloud, we assume that it is possible to relocate the server by transferring the state of service execution to obtain the necessary resources to execute a process.

4 Proposal for Autonomous Control of Server Relocation System

In order to optimize QoS for end users, we propose the autonomous control of server relocation for fog computing to reduce both the average response time and the fairness between users, as described below. We describe each of the functions required by the system, and then we describe the basic functions of the system.

This system collects information about the computing environment of the surrounding nodes, searches for a candidate node that can minimize the service response time, and transfers the server to the selected node.

In searching for candidate nodes, it is not realistic to assume global knowledge across different computing environments such as fog and cloud. As a reasonable scope of search, we assume a routing topology of CCN interest messages when the server is regarded as a resource. It collects PCEL (Available Processing Capacity and Estimated Latency) management information for each node on the path where a message arrives and determines the server transfer based on this information.

The following sections describe the main components of the proposal, the estimation of the service processing time, the collection of PCEL information on the message arrival route, and the algorithm for selecting candidate nodes.

Service processing time

C which is the processing power of a node and L which is the amount of processing of the requested service executed by the node are represented as a two-dimensional vector to decompose the processing power of the node into CPUC and the purpose-specific unit T, respectively. The processing capacity of a fog node is represented by (C_C, C_T), and the amount of processing required for service execution is defined as (L_C, L_T). Based on these processing capacity and processing volume, the service processing time T_{est} is defined as follows.

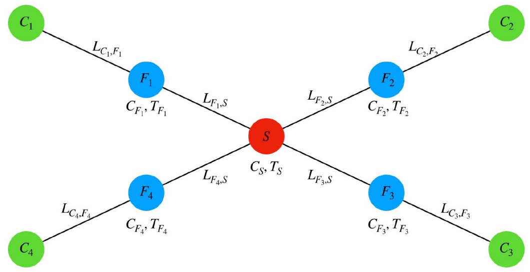

Figure 1: System Configuration Example(S : Server Node, F : Fog Node, C : Client Node, $L_{A,B}$: Communication delay from A to B, C_X : X's CPU Processing Power, T_X : X's processing of purpose-specific units)

$$T_{est}(C_C, C_T, L_C, L_T, \alpha) = \begin{cases} \frac{1}{C_C}(L_C + \frac{L_T}{\alpha}) & (C_T = 0) \\ \max(\frac{L_C}{C_C}, \frac{L_T}{C_T}) & (otherwise) \end{cases}$$

However, the CPU can perform the processing required for the purpose specific unit. The α that express ratio is set to 5 for use in later evaluation.

Collecting PCEL information on the message arrival path.

In order to treat all the nodes on the path from each client node to the server as candidates for transfer, it is necessary to collect information on the delays between the links traversed and the processing capacity of the nodes traversed. These are called the route PCEL information. In our system, PCEL information is added to the request message at the node that passes by the time the request message reaches the server node, and it is transmitted to the server.

For example, when our system is used as shown in Figure 1, the following parameters are added to the request message of C_1.

- L_{C_1,F_1}, which is is the communication delay of the link from C_1 to fog node F when a request message from client node C_1 goes through each fog node.
- $L_{F_1,S}$, which is communication delay of the link from server node S to F_1.
- C_{F_1}, which is the CPU processing power of F_1
- T_{F_1}, which is the processing power of the purpose-specific unit of F_1

This added PCEL information can be acquired by S from the request message. Similarly, S can obtain information on the route to and from all clients from the PCEL information attached to the request messages from C_1 to C_4, which are all participating client nodes.

Candidate node selection algorithm

The server selects candidate nodes for transfer from the PCEL information appended to the request message received from the client By using the algorithm shown in Algorithm 1. Algorithm 1 calculates the average and standard deviation of the service response time of the participating clients based on the information that the server shown in Figure 1 can obtain from the request message. The value is the sum of the service response time multiplied by $1 - R_{std}$ and the standard deviation multiplied by R_{std}, based on R_{std}, which specifies how much importance is placed on the fairness between clients and users. Then, we find the node whose evaluation value is at a minimum.

Algorithm 1 Find Candidate Node

```
Require: $L_{All}$:List of $L$ on the Path
Require: $L_{(F_i,C_j)}$:List of $L$ on the Path between $F_i$ to $C_j$.
Require: $F_{All}$:List of $F$ on the Path
Require: $C_{All}$:Clients connected to the service
Require: $L_C$:CPU processing capacity during service execution
Require: $L_T$:Purpose specific unit throughput during service execution
Require: $R_{std}$:Ratio of importance to the standard deviation
Require: $S_{F_i,C_{All}}$:Standard deviation of service response time between $F_i$ to $C_{All}$
Require: $\alpha$:Coefficient that represents the ratio when the CPU can handle the amount of processing of the target-specific unit
Ensure: $MinNode$ is candidate node.
  $MinCost \leftarrow \infty$
  for all $node$ in $F_{All}$ do
    for all $client$ in $C_{All}$ do
      $Latency_{node,client} \leftarrow \sum L_{node,client} * 2$
      $Latency_{sum} \leftarrow Latency_{sum} + Latency_{node,client}$
    end for
    $Latency_{ave} \leftarrow \frac{Latency_{sum}}{C_{All}.length}$
    $Latency_{var} \leftarrow \frac{1}{C_{All}.length} \sum_{i=1}^{C_{All}.length} (Latency_{Ave} - Latency_{node,C_i})^2$
    $ServiceRTT \leftarrow T_{est}(node.C_C, node.C_T, L_C, L_T, \alpha) + Latency_{ave}$
    $R_{RTT} \leftarrow 1 - R_{std}$
    $COST \leftarrow ServiceRTT * R_{RTT} + S_{node,C_{All}} * R_{std}$
    if $MinCost > Cost$ then
      $MinCost \leftarrow Cost$
      $MinNode \leftarrow node$
    end if
  end for
  return $MinNode$
```

4.1 Functions of the node

The movement of the proposed system is shown in Figure 2. This system is assumed to operate at the session layer of all participating fog nodes. It is preferable that the change in the point of service execution is done transparently to the client and server. In order to implement the collection of PCEL information and transparent addition to the request message, it is convenient to work at the session layer in the seven-layer model. Since management actions such as service transport are operated by resources below the transport layer, they must be located in the upper layer where they are visible, and

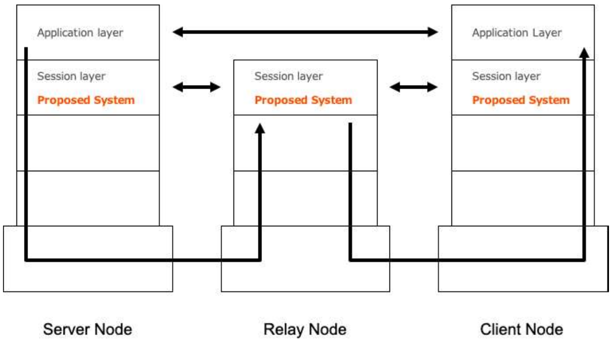

Figure 2: Autonomous Control of Server Relocation System

the session layer is lower than the application layer where clients and servers are running. By relaying communication between server nodes and client nodes at the application layer, the server relocation system at the session layer of server nodes, relay nodes and client nodes share information about resources available at each node. Based on the collected information, it realizes the selection of service execution points and resource allocation.

The proposed system consists of three types of nodes. The proposed system consists of multiple connections: a cloud node that has the contents necessary for service execution and has high processing power, a middle-class fog node that has medium processing power and can communicate with end devices with relatively low latency, and a client node that is an end device such as a smartphone that participates in the server. Based on the client-server communication model, cloud nodes and fog nodes play the role of servers, and leaf nodes play the role of clients. The server monitors the communication status of participating clients and decides whether it should autonomously play the role of the server or delegate the role of the server to other fog nodes based on the communication status. The delegated fog node takes over the role of the server. In this way, we try to optimize the server relocation of the server's role. This is an attempt to reduce the service response time.

There are Each fog node has an in-network resource monitoring function, a candidate selection function and a service transfer function. In this section, each function is explained. The autonomous control of server relocation system at each node operates at the session layer to superimpose management information on the communication messages between the server and the client to realize the resource monitoring function in the network. At the same time, it constantly monitors changes in the resources in the network, and if a change is observed, it executes the candidate selection function and, if it is judged to be necessary, it executes the service transfer function to the selected node to optimize the server relocation.

4.1.1 Overall Flow to Optimize Server Placement

We summarize the optimization process explained so far. The client sends a request to the server . The server stores information associated with the request by means of an in-network monitoring function. Using the candidate selection function, we select candidate nodes from the accumulated information.

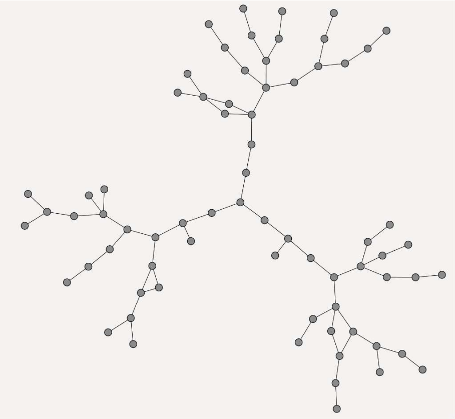

Figure 3: The network topology used in the experiment

After a candidate node is selected, it sends a message to the candidate node that it will transfer the service. A fog node that receives a message to transport a service confirms that no other service is established at its own node and starts collecting data for service execution, while at the same time sending a message to the node from which the service is being transported to inform it that the service is being prepared. When a fog node completes its data collection, it starts the service and sends a message to the source node telling it that it is ready. The server node that receives the ready message announces the new server to its current clients. The client receives information about the new server and changes the destination of the request to the new server. A client that joins from the middle of the process first sends a request to the original server node, receives information on the current server, and joins the service based on that information.

5 Evaluation

We defined three use cases to show three aspects: fairness between users on uniform computer resources, reduction of average response time on heterogeneous computer resources, and fairness and reduction of average response time between users on heterogeneous computer resources.

5.1 Simulation Environment

For the network topology, we used the topology generated by BRITE[11], which is a topology generator as shown in Figure 3. A county of three AS-equivalent nodes was prepared, with about 20 Fog nodes in each AS, and in each simulation, one AS was treated as a cloud environment and two AS were treated as a fog network with clients connected to it. Table 1 shows the parameters used in the simulation. The amount of content cache space owned by each node is also determined by the This was done assuming that the system has enough space to cache all the necessary data.

5.2 Internet Conference

this use case assumes a multi-point Internet conferencing system to show fairness among users with uniform computer resources. In the Internet conferencing system, the server mixes media data received from all connected terminals and

Table 1: Simulation Parameters

Parameters	Value
Cache Algorithm	LRU
Data Rate	10Mbps
Delay	1ms
Simulation time	100s
Server's C_C	100.0
Server's C_T	100.0
Dedicated Unit's C_C	20.0
Dedicated Unit's C_T	50.0
Regular Unit's C_C	20.0
Regular Unit's C_T	0

distributes them as a single stream to all terminals. The goal is to keep media quality fair in situations where geographically distributed participants connect to the system. Simulate the behavior of a server moving to the optimal location for clients distributed in different locations on the network with different latency times.

5.2.1 Simulation Scenario

Multiple meetings were defined and the participants of each meeting were placed in the same AS, and the servers of all the meetings were placed together in a different AS than the AS in which the clients were participating. The processing capacity for conducting the conference and the processing capacity of each node were assumed to be constant. The results were compared with the case in which no transfer was performed.

6 Results and discussion

In this use case experiment, we achieved fairness between users on a uniform computational resource. The experimental results are shown in Figure 6 and Figure 7. Figure 6 shows the change in service response time when the proposed system is not used, and Figure 7 plots the change in service response time against time when the proposed system is used. The users of the proposed system are gradually transferred to the one with less network latency.

At the timing of the start of the simulation, the two conference streams are shown in Figure 4. It is sent from different AS to a single AS, and the network traffic is aggregated. In the network after the transfer, the servers are transferred within each AS, as shown in Figure 5, and the two conference avoids the aggregation of meeting traffic.

In Figure 5, the fairness between users depends on which node on each communication path the server will be transferred to. It is possible to achieve the fairness required by each application by adjusting the R_{std} used in the algorithm 1 for destination determination. Figure 8 shows the trend of the mean and standard deviation of the service response time for a single conference among the results of the selecting candidate nodes for transfer, where the value of R_{std} is set to 0 and only the mean of the service response time is important. Figure 9 sets the value of R_{std} as 0.7 and the node selection with a weighted standard deviation of 70% of the response time, showed the average service response time for the same meeting as in figure 8. Comparing the two figures, we can confirm that the result of figure 9, weighted at 70%, is fairer (i.e. smaller deviation) than the result of figure 9, which shows the fairness of the transfer between users. We can confirm that the fairness of the transfer between users is maintained. In this way, we achieved fairness among users with uniform computer resources by performing transfers to shorten the service response time and adjusting the parameters to meet the requirements of the application.

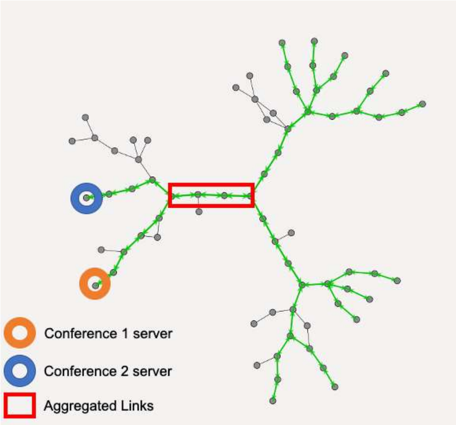

Figure 4: Network status before the transfer begins.

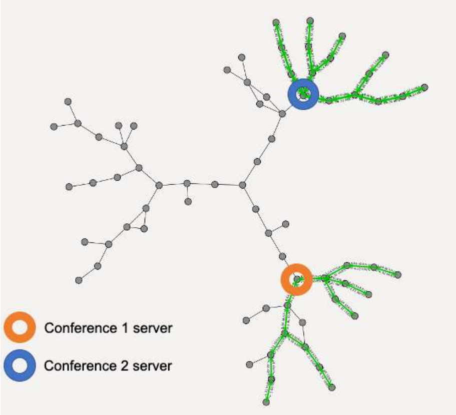

Figure 5: Network status after transfer

7 Conclusion

Fog computing, which extends the cloud computing paradigm to the edge of the network, has been proposed and is being actively researched. In the field of networking, there is research on CCN that use location-independent content as identifiers instead of the traditional IP address-based architecture. So far, we have proposed a system that aims at fairness in response time and shortening of service execution time between users, respectively. Therefore, in this study, we proposed the autonomous control of server relocation for fog computing systems to optimize QoS for end users, which achieves both shortening the average response time and fairness between users. To this end, the system achieves inter-user fairness on uniform computer resources, reduction of average response time on non-uniform computer resources, and We tested the fairness between users by setting up use cases .

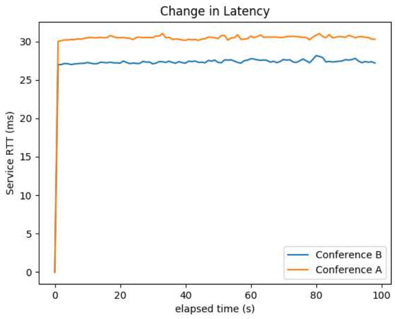

Figure 6: Changes in service response time if the proposed system was not used.

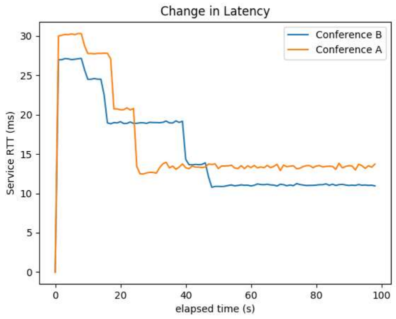

Figure 7: Changes in service response time when using the proposed system

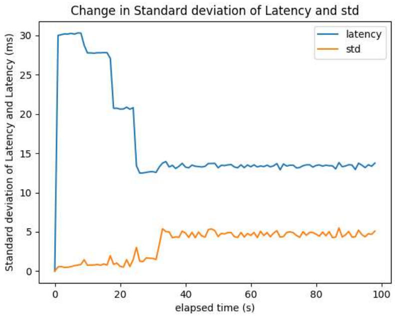

Figure 8: Mean and standard deviation of service response time for conference A when R_{std} is set to 0%.

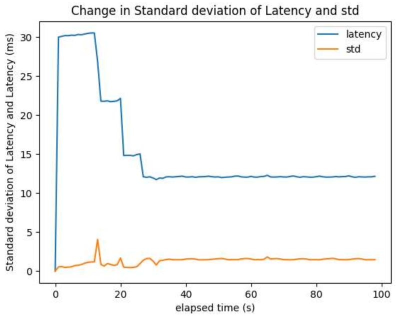

Figure 9: Mean and standard deviation of service response time for conference A when R_{std} is set to 70%.

REFERENCES

[1] Ministry of Internal Affairs and Communications, Japan: *The 2018 White Paper on Information and Communications in Japan*, Japanese Government (2018).

[2] Bonomi, F., Milito, R., Zhu, J. and Addepalli, S.: Fog Computing and Its Role in the Internet of Things, *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, pp. 13–16 (2012).

[3] Jacobson, V., Smetters, D. K., Thornton, J. D. et al.: Networking Named Content, *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, pp. 1–12 (2009).

[4] Kamada, K., Inamura, H. and Nakamura, Y.: A Proposal of Autonomous Control of Server Relocation for Fog Computing Systems(In Japanese), *IPSJ SIG Technical Report*, Vol. 2018-MBL-89, No. 1, pp. 1–5 (2018).

[5] Kamada, K., Inamura, H. and Nakamura, Y.: Autonomous Control Scheme of Server Relocation for Non-Uniform Computing Capacity in Fog Networks(In Japanese), *DICOMO2019*, Vol. 2019, pp. 1204–1211 (2019).

[6] Cuervo, E., Balasubramanian, A., Cho, D.-k. et al.: MAUI: making smartphones last longer with code offload, ACM Press, p. 49 (2010).

[7] Chun, B.-G., Ihm, S., Maniatis, P. et al.: CloneCloud: elastic execution between mobile device and cloud, ACM Press, p. 301 (2011).

[8] Kosta, S., Aucinas, A., Hui, P. et al.: ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading, *2012 Proceedings IEEE INFOCOM*, pp. 945–953 (2012).

[9] Berg, F., Dürr, F. and Rothermel, K.: Increasing the Efficiency of Code Offloading in N-tier Environments with Code Bubbling, *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MOBIQUITOUS 2016, ACM, pp. 170–179 (2019).

[10] Shanbhag, S., Schwan, N., Rimac, I. and Varvello, M.: SoCCeR: services over content-centric routing, *Proceedings of the ACM SIGCOMM workshop on Information-centric networking - ICN '11*, ACM Press, p. 62 (2011).

[11] Medina, A., Lakhina, A., Matta, I. and Byers, J.: BRITE: An approach to universal topology generation, *MASCOTS 2001, Proceedings Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, IEEE, pp. 346–353 (2001).

On Improving Efficiency of CSMA/CA with RSSI-based Control-frame Detection

Yoshito Umezawa† and Takuya Yoshihiro‡

†Graduate School of Systems Engineering, Wakayama University, Japan
‡Faculty of Systems Engineering, Wakayama University, Japan
{s206036, tac}@wakayama-u.ac.jp

Abstract - CSMA/CA has been known as a representative media access control method since the dawn of wireless communication. Even now, it is widely used in the world. For example, it is well known to be adopted by IEEE 802.11, which is one of the most popular communication standards. CSMA/CA has long been known to have a problem that significantly degrades communication performance, called the hidden-terminal problem or the exposed-terminal problem. These problems have been tackled by many researchers over the years, and there is a huge amount of research. However, a fundamental solution to these problems has not yet been proposed. For this reason, even now, the communication performance is still significantly deteriorating when many terminals gather. In this paper, we propose a control frame multiplexing technique to detect CTS frame and ACK frame with high accuracy without demodulation by monitoring the received signal strength, i.e., RSSI (Received Signal Strength Indication), even when a node is receiving signals from neighboring nodes, which would normally be busy. The proposed technique enables simultaneous data communication, which solves the problem of exposed terminals and greatly improves the communication efficiency in CSMA/CA.

Keywords: CSMA/CA, RTS/CTS, RSSI, exposed-terminal problem.

1 INTRODUCTION

IEEE 802.11, which was standardized in 1997, is one of the most popular wireless communication standards even today. In this IEEE 802.11, a medium access control method called CSMA/CA is adopted. In CSMA/CA, a node detects if other nodes are transmitting before starting transmission. When no other node is transmitting, it starts communication after waiting a random backoff time. If a node detects that another node is transmitting, it waits for a while and after the node finishes the transmission, it starts its own communication. However, CSMA/CA has problems called the hidden-terminal problem and the exposed-terminal problem which significantly deteriorates communication performance. The hidden-terminal problem is a problem in which, when nodes that cannot detect the transmit radio waves of each other simultaneously, a collision occurs at the receiving node. The exposed-terminal problem is a problem in which transmission is actually possible, but transmission is unreasonably suppressed when there is transmission around there even if it does not prevent the transmission.

Many researchers have been tackling on these two problems for many years, and so there is a vast amount of studies. However, no essential solution has been proposed yet.

The purpose of this study is to eliminate the exposed terminal problem in wireless communication using CSMA/CA. We propose a control frame multiplexing technique that can detect the arrival of CTS and ACK frame with high accuracy by monitoring RSSI and transmit data even when a node is in BUSY state under the original CSMA.

This paper consists of five sections. In Section 2 describes related work. Section 3 describes the method proposed in this study. Section 4 describes the performance evaluation. Section 5 describes a summary of this study.

2 RELATED WORKS

Improvement in CSMA/CA has been undertaken by many researchers for many years, and so there is a vast amount of studies. Bharghavan et al. proposed a method called RTS/CTS to solve the hidden terminal problem that occurs in CSMA [1]. This method is also adopted in IEEE 802.11 standard. However, it is known that the performance degradation due to the exposed terminal problem is significant, and that frame loss due to radio interference from a distance occurs frequently especially during high-speed communication, so it does not work well as a countermeasure for the hidden terminal problem [2] [3]. As a result, RTS/CTS is rarely used in practice.

Recently, methods have been proposed to improve communication efficiency by using techniques in the physical layer. In wireless communication, a technique called SIC (Self Interference Cancellation) has been proposed in which a node has a two NICs (Network Interface Cards) for transmission and reception, respectively, and cancels the transmitted signal at the receiver to perform transmission and reception at the same time. This technique is known as full-duplex wireless communication, which has been actively studied [4] [5]. In addition, there is a technique called NOMA (Non-Orthogonal Multiple Access). When a node receives a strong signal and a weak signal at the same time, by demodulating the strong signal first, and by estimating its original signal to be removed from the received signal, the node can demodulate the weak signals [6]. Although these techniques are drawing attention as ones that significantly improve wireless communication capacity, only a few techniques for utilizing them in the MAC layer have been proposed. Therefore, it is doubtful whether or not it will contribute to solve the hidden terminal problem, which is an essential problem in CSMA/CA.

J.J.Garcia-Luna-Aceves proposed CRMA as a MAC protocol using SIC technology [7]. In addition, he proposed to use busy tone and pilot signals to realize a complete MAC protocol that does not cause both hidden and exposed terminal problems in wireless communication [8]-[10]. However, all of them are analyzed only theoretically, and so the performance in an actual wireless environment is unknown.

The purpose of this paper is to realize a MAC protocol that does not cause both hidden terminal problems and exposed terminal problems. The proposed method differs from the conventional method in that CTS and ACK frames can be received without demodulation. Since they are not demodulated, CTS and ACK frames can be received even if the signal from another node is being received only if the SN ratio is larger than about 3 dB. Therefore, it has the potential to greatly improve the flexibility of the MAC protocol compared to the conventional method.

3 PROPOSED METHOD

3.1 Overview

In the proposed method, nodes constantly monitor RSSI (Received Signal Strength Indication) during wireless communication using RTS/CTS. Even if the RSSI level reaches the threshold to transit to BUSY state or the NAV state, in the original CSMA if the certain condition is satisfied, it remains in idle state and transmitting RTS/CTS frames to start DATA transmission is allowed. In addition, after data frames are sent, CTS or ACK as the response is detected without demodulation by observing only the rise of RSSI at the timing when CTS or ACK is returned. As a result, we can achieve simultaneous communication of data frames and, solves the exposed terminal problem, and improves communication performance.

An example of the operation of the proposed method is described in Figure 1 and 2. Figure 1 shows the arrangement of nodes and the communication flow, and Fig. 2 shows a MAC operation of each node. First, RTS/CTS handshake is performed between node s_1 and node r_1. After that, s_1 starts transmitting DATA frame when CTS reception from r_1 is completed. s_2 that receives RTS and the data frame of s_1 does not transit to NAV or BUSY state because RSSI of RTS and DATA frame is below the predefined threshold. s_2 sends RTS to node r_2 after waiting the backoff time. r_2 returns CTS after receiving RTS from s_2. When CTS is returned from r_2, s_2 is detecting DATA frame transmitted by s_1. However, s_2 detects the rise in RSSI at the timing when CTS will be returned. As a result, it is confirmed that CTS has been returned from r_2 without demodulation, and s_2 starts DATA frame transmission.

When r_1 finished receiving DATA frame from s_1, it returns an ACK frame. When ACK is returned from r_1, s_1 is detecting DATA frame from s_2, but the rise in RSSI is observed at the timing when ACK will be returned. As a result, s_1 confirmed that ACK has been returned from r_1 without demodulation, and s_1 completes transmission. On the other hand, r_2 received DATA frame from s_2, returns ACK, and the transmission of s_2 is completed. If DATA frame of s_2 finishes its transmission before DATA frame of s_1, ACK from r_2 interferes with DATA frame of s_1. However, s_2 judges that ACK has arrived because of the rise of RSSI, and completes the communication. In this way, the communication of s_2 is not suppressed by RTS, and the simultaneous communication of DATA frames from s_1 and s_2 is possible, where the exposed-terminal problem is solved.

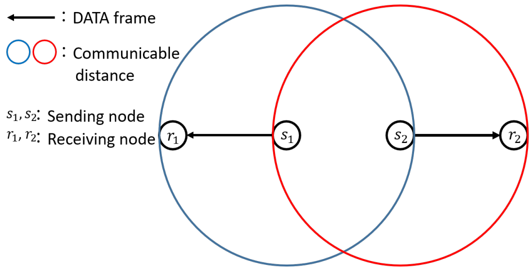

Figure 1: Layout of the operation example

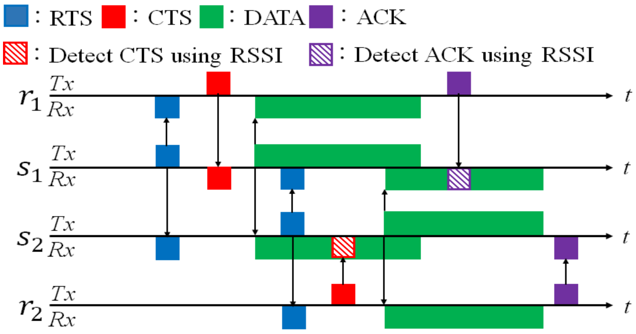

Figure 2: Operation example of each nodes

3.2 RSSI-based CTS and ACK detection

In this paper, even if frames are sensed, nodes do not transit to BUSY state when a certain condition is met, and the nodes start RTS/CTS handshake on their back-off expiration. This aims to eliminate the influence of the exposed-terminal problem and improve communication throughput. However, if a node sends a RTS or DATA frame when some other nodes are transmitting DATA frames, the returned CTS or ACK frames could collide with the DATA frame, resulting in retransmission of those frames.

Our approach to prevent this is to confirm the arrivals of CTS or ACK by observing only RSSI even if CTS or ACK is not high enough to demodulate it. At this time, the timing at which CTS or ACK frames will be returned depends on the fixed-length SIFS and the frame transmission rate, meaning that the arrival time can be easily expected. Therefore, if the rise in RSSI is observed at the timing when CTS or ACK is expected, it will be the reception of CTS or ACK. Even if demodulating CTS and ACK frames is impossible, the node can confirm that CTS and ACK frames have arrived.

This is explained in Fig. 3. Node B has data for node C, but at this moment B is receiving the data frame from neighboring node A. B holds the average radio signal strength S_A[dBm] of A. At this time, if the radio signal strength being observed is less than or equal to S_A+T[dBm], where T is a predefined threshold value, B starts data transmission with RTS. If node

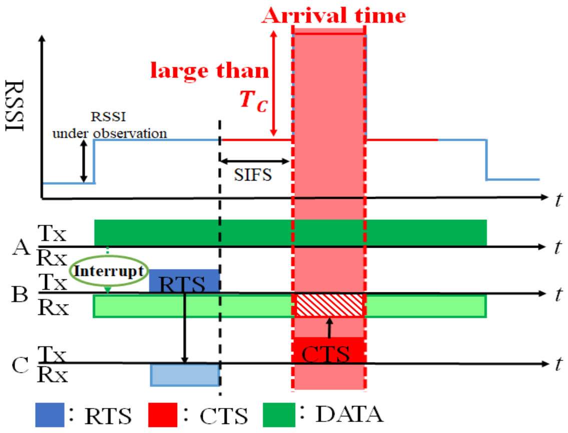

Figure 3: CTS detection using RSSI

C receives RTS of node B normally, node C returns CTS after SIFS interval. However, this time, B cannot decode CTS from C due to the interference from the data frame from node A. Therefore, node B compares the RSSI level of the estimated period for CTS arrival with that around the estimated arrival period, and if this difference is more than the threshold T_c, it judge that CTS has arrived and processes CTS. In addition, when the data transmitting nodes, i.e., A and B, are close to each other, RSSI level of DATA frames received by each other become high, and RSSI level of CTS and ACK becomes relatively small, consequently they would fail confirming CTS and ACK with high probability. Therefore, if the radio signal strength observed by B is S_A+T or more, B transits to BUSY state as in the conventional CSMA/CA, and waits for A finishing its transmission.

By detecting CTS or ACK using RSSI, communication is not suppressed by RTS or DATA frames transmitted from other nodes. As shown above, even under some interference, RTS/CTS handshake, data transmission and ACK transmission are possible, and consequently the exposed-terminal problem is resolved.

3.3 Proposed MAC protocol

Since the proposed method detects CTS and ACK through RSSI described in Section 3.2, a part of the conditions for state transition differs between the conventional CSMA/CA and the proposed method. Figure 4 shows the state transition diagram of the proposed method.

The difference between CSMA/CA and the proposed protocol is two folds :(1) the behavior when receiving RTS that does not destine itself, and (2) the behavior in face of carrier sensing. (1): In CSMA/CA, when RTS or CTS that does not destine itself is received in any of BACKOFF, DATA_WAIT, CTS_WAIT, and ACK_WAIT states, it transits to NAV state. In contrast, in the proposed method, the transmitting node in the BACKOFF state continues to stay BACKOFF state when it receives RTS to the node if the RSSI is below the threshold. If the received RSSI is above the threshold, it transits to BUSY state, which is the same as the normal CSMA/CA, because the returned CTS would fail to be detected with high probability. In carrier sensing, when the RSSI level of the received signal is above the threshold, it transits to the busy state.

(2): In CSMA/CA, DATA frame transmission starts only when CTS is received, and after the transmission ends, the node enters ACK_WAIT state. However, in the proposed protocol, when CTS is received in CTS_WAIT state, or when CTS is detected by RSSI as described in Section 3.2, DATA frame transmission starts and the node transits to ACK_WAIT state. If the received RSSI level is above the threshold, it transits to BUSY state. When CTS is not received, the node transits to the BACKOFF state. In the proposed protocol, transition to the BACKOFF state occurs either when ACK is received in ACK_WAIT state, or when ACK is detected by RSSI, or when a timeout occurs without receiving ACK.

Next, the common parts between the proposed protocol and CSMA/CA are described. First, the operation is the same in BUSY state and NAV state. When in BUSY state, if there is no radio in the communication channel, nodes transits to the backoff state. In NAV state, it transits to the backoff state when NAV period ends. In CSMA/CA and the proposed protocol, when a CTS for the node is received in the BACKOFF state, the state transits to NAV state. In the backoff state, the node waits before transmission until the random backoff time expires, and when the backoff expires, RTS is sent and transits CTS_WAIT state.

4 EVALUATION

4.1 Evaluation Method

We compare the performance of the proposed method and the existing method using the network simulator Scenargie ver. 2.1. The existing methods to compare are CSMA/CA (with RTS/CTS) and CSMA (without RTS/CTS). In the experiment, we evaluate whether the communication performance is improved by solving the exposed terminal problem by the proposed method compared with the conventional method. Therefore, we focus on and evaluate the frame delivery rate and average throughput. The frame delivery rate represents the rate of the received data frames out of the number of transmitted frames. The average throughput represents the amount of data sent/received per unit time. The communication speed of the entire network is measured by this measure.

Figure 5 shows the simulation scenario. We prepare two Access Points (AP), and one AP communicates with four Clients (C). The location of each AP and C is set so that both the hidden-terminal problem and the exposed-terminal problem occur. The distance between two APs, and the distance between C and the neighboring AP are both 250[m]. The simulation time is 120[s], and the communication flow generation time is 10 to 110[s]. The communication standard used IEEE802.11g, which is a commonly used wireless communication method, and the communication speed of all nodes is 6[Mbps]. The communication flow is CBR(Constant Bit Rate) and the frame size is 1000[Byte]. Then, the simulation was performed with variation of transmission rate per flow from 50 to 900[kbps] with 50[kbps] interval. Table 1 summarizes the conditions common to both the proposed method and

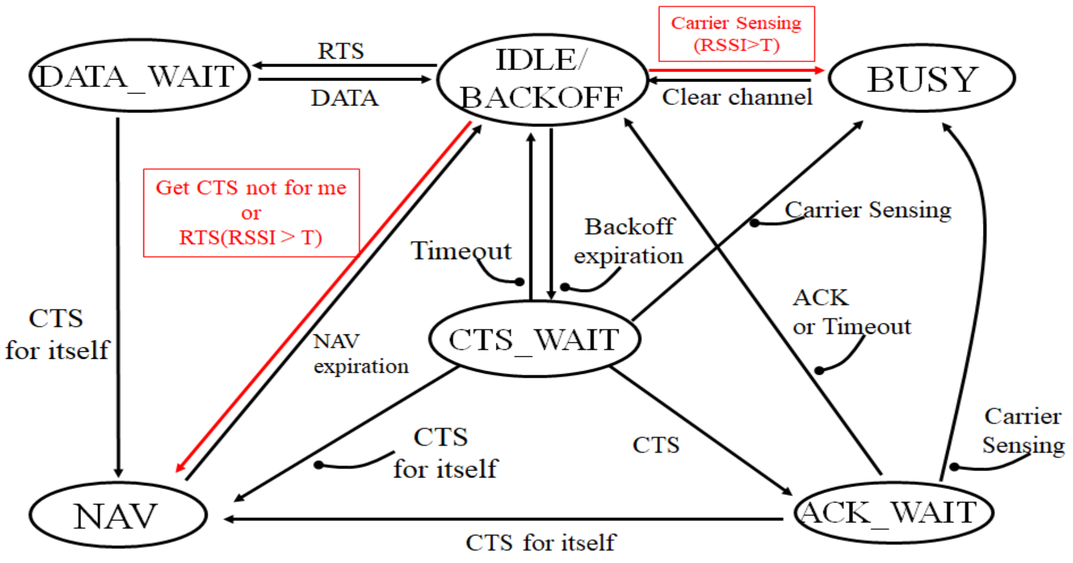

Figure 4: State transition diagram of proposed protocol

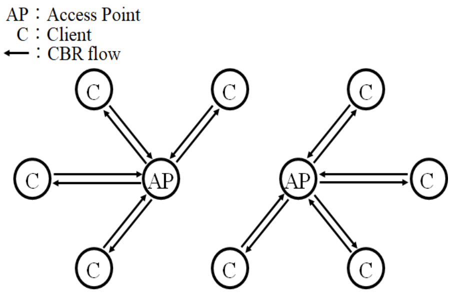

Figure 5: Node position

Table 1: Common condition

Parameter	Value
Threshold	3[dBm]
Simulation time	120[秒]
Number of nodes	10
Distance	250[m]
Flow type	CBR
Number of flows	16
Bit rate	50～900[kbps]
Occurrence time	100[s]
Frame size	1000[Byte]
Communication standard	IEEE802.11g
Communication speed	6[Mbps]
Transmission power	10[dBm]

the existing method. We set the detection threshold for CTS and ACK to 3 [dB] by t-test. This value is computed from a simple statistical calculation; we found we can identify CTS and ACK signal with 99[%] in probability if two randamly distributed signals have 3[dB] difference in the average signal strength.

4.2 Results

Figure 6 and 7 show the evaluation results of the simulation. Figure 6 shows the average throughput of the proposed method and the existing method. Figure 7 shows the average delivery rate. The horizontal axis represents the transmission rate of one communication flow from 50 to 900[kbps], and the vertical axis represents the average throughput [kbps] in Fig. 6 and the average delivery rate [%] in Fig. 7. In Fig. 6, we see that the performance of the conventional method and the proposed method are the same in throughput up to 400 kbps in the transmission rate. However, over 400 kbps, the proposed method exceeds the existing method and the difference reaches about 1000 kbps at the transmission rate of 700 kbps. In Fig. 6, the performance in delivery rate is the same between the conventional method and the proposed method up to 400 kbps in the transmission rate. Over 400kbps, the proposed method exceeded the delivery rate at all transmission rates. This is because the exposed-terminal problem was solved by detecting CTS/ACK using RSSI, and the communication opportunity was not lost by solving the exposed-terminal problem, and consequently that simultaneous data communication was performed. Specifically, the DATA frames transmitted from the AP on the left side reach the AP on the right side, but the AP on the right side transmits RTS or DATA frame without suppressing the transmission, so the exposed terminal problem is solved. Also, when CTS/ACK arrives at the right AP while the data frame of the left AP is arriving, the arrival of CTS or ACK is detected by RSSI, so the hidden terminal problem can be solved. From the results of the evaluation, it is clear that the proposed method solves both the hidden-terminal problem and the exposed-terminal problem to some extent. However, even with low transmission rates, the delivery rate is about 63[%], so it can be considered that it has not been completely resolved. From the above, it is clear

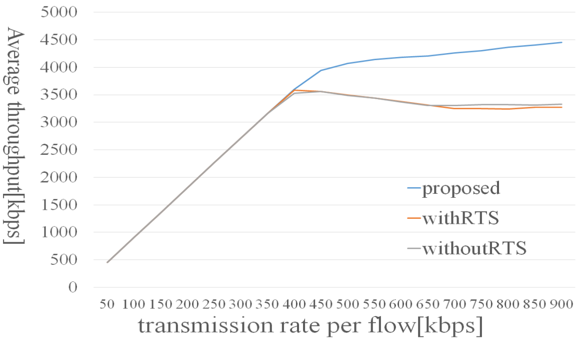

Figure 6: Average throughput results

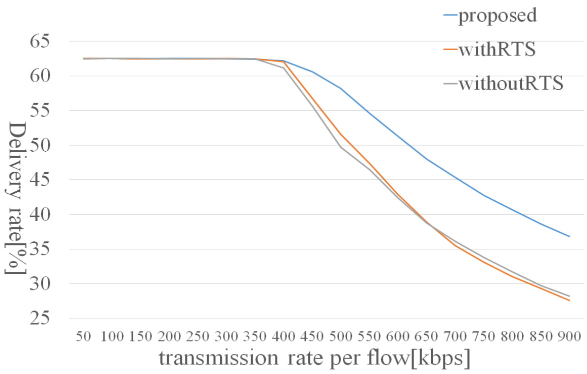

Figure 7: Average delivery rate result

that the proposed method improves the communication performance by eliminating the hidden-terminal problem and the exposed-terminal problem to some extent, as compared with the existing methods. In the proposed method, the capacity of the entire network is increased, and both the throughput and the delivery rate are improved by collision avoidance and simultaneous data communication.

5 CONCLUSION

In this paper, we proposed a method to solve the exposed terminal problem in wireless communication using CSMA/CA. As an evaluation, we compared the proposed method with the existing method using the network simulator Scenargie ver. 2.1. The evaluation results reveal that the proposed method improves the performance by simultaneous transmission of data, eliminating the hidden-terminal problem and the exposed-terminal problem to some extent. However, since it has not been completely resolved, we will analyze in detail the hidden-terminal problem and the exposed-terminal problem that could not be solved in the future. In addition, it is considered that the proposed method is open to discussion on the effects of false detection of CTS/ACK and of the fading effect in a certain environment. Therefore, we think that it is necessary to analyze the physical layer behavior by using MATLAB or real machine implementation. In the future, we will investigate the reality in a real environment by actual experiments using software defined radio such as USRP.

Recently, techniques for improving wireless communication efficiency by a physical layer has been proposed, and a full-duplex wireless communication technology for simultaneously performing transmission and reception has been proposed. we would also try to apply this study to not only the current half-duplex wireless communication but also the full-duplex wireless communication.

Acknowledgment

A part of this work is supported by KAKENHI(19K22845)

REFERENCES

[1] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: A media access protocol for wireless LAN’s,” in Proc. ACM SIGCOMM ’94, pp. 212–225, 1994.

[2] J.L. Sobrinho, R. de Haan, J.M. Brazio, “Why RTS-CTS Is Not Your Ideal Wireless LAN Multiple Access Protocol,” In Proc WCNS ’05, 2005.

[3] K. Xu, M. Gerla, and S. Bae, “Effectiveness of RTS/CTS Handshake in IEEE 802.11 Based Ad Hoc Networks,” Ad Hoc Networks, Vol.1 Issue.1, pp.107-123, 2003.

[4] M. Jainy et al., “Practical, Real-Time, Full Duplex Wireless,” In Proc. ACM MobiCom ‘ 11, 2011.

[5] D. Kim, H. Lee, and D. Hong A Survey of In-Band Full-Duplex Transmission: From the Perspective of PHY and MAC Layers Perspective of PHY and MAC Layers, IEEE Communications Surveys & Tutorials 17(4), 2017–2046, 2015.

[6] Z. Ding, X. Lei, G.K. Karagiannidis, R. Schober, J. Yuan, V. Bhargava, A Survey on Non-Orthogonal Multiple Access for 5G Networks: Research Challenges and Future Trends, IEEE Journal on Selected Areas in Communications, 35(10), pp.2181-2195, 2017.

[7] J.J. Garcia-Luna-Aceves, Carrier Resolution Multiple Access, In Proc. PE-WASUN ’ 17, 2017.

[8] J.J. Garcia-Luna-Aceves, “CTMA: A More Efficient Channel Access Method for Networks with Hidden Terminals,” PE-WASUN ’ 17 , pp 9-16, 2017.

[9] J.J. Garcia-Luna-Aceves, Busy-Tone Multiple Access with Collision Avoidance and Detection for Ad-Hoc Networks, In Proc. of IEEE ICNC2019, 2019.

[10] J.J. Garcia-Luna-Aceves, “Design and Analysis of Carrier-Sense Multiple Access with Collision Avoidance and Detection,” Proc. ACM MSWIM ‘ 17, 2017.

[11] bladeRF 2,0 micro, Nuand LLC, https://www.nuand.com/bladerf-2-0-micro/ (referred in June 2020)

[12] USRP B200, Ettus Research, http://www.ettus.com/all-products/UB200-kit/ (referred in June 2020)