2020 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP 2020)
Honolulu, Hawaii, USA, February 28 - March 2, 2020

Proposal of management cloud software supporting multi-thread processing for overlay network protocol

Shuhei Isomura[1], Yuki Yamada[1], Taiki Yoshikawa[2], Kohei Tanaka, Katsuhiro Naito[2]

[1] Graduate School of Business Administration and Computer Science
Aichi Institute of Technology
1-38-1 Higashiyamadori, Chikusa, Nagoya 464-0807, Japan
Phone: +81-52-789-1381
FAX: +81-52-782-5066
E-mail: syisomura, yamayu@pluslab.org

[2] Faculty of Information Science
Aichi Institute of Technology
1247 Yachigusa, Yakusa, Toyota 470-0392, Japan
Phone: +81-565-48-8121
FAX: +81-565-48-0277
E-mail :yoshikawataiki, naito@pluslab.org

Abstract

The internet has several issues such as accessibility due to the Network Address Port Translation (NAPT) technology, interoperability between IPv4 and IPv6, and disconnection due to mobility. As a technique for solving these issues, the authors have developed CYber PHysical Overlay Network over Internet Communication (CYPHONIC) that supports accessibility and seamless mobility over the Internet. Since IoT devices typically implement the full IP stack, both IPv4 and IPv6 can be supported. Therefore, CYPHONIC can provide inter-connectivity among IoT devices. CYPHONIC end node communicates with cloud service in advance to realize direct communication between nodes. Therefore, the traffic to the cloud service from each node will be expected to increase. However, the scalability of the current cloud service is limited because each service is designed for single processing. This paper develops an extensible cloud service implementation supporting multi-thread processing. According to the extension in this paper, the new implementation realizes the cloud service in a multi-thread environment.

1. Introduction

Internet Protocol (IP) exists as a basic communication technology for realizing IoT systems [1]. Since IPv4 is a mainstream protocol on the Internet, the lack of compatibility between IPv4 and IPv6 should be great issues to realize IoT services [2, 3]. Additionally, recent networks employ the Network Address Port Translation (NAPT) technology to protect a network and to reduce the number of required IPv4 global addresses. The introduction of NAPT causes an accessibility issue due to the blocking of incoming packets [4]. As a new issue, the mobility of IoT devices also becomes a concern about seamless communication because a change of IP address causes a disconnection of transport-layer sessions [5, 6].

As a technique for solving these issues, the authors have developed CYber PHysical Overlay Network over Internet Communication (CYPHONIC) that supports accessibility and seamless mobility over the Internet [7]. Since IoT devices typically implement the full IP stack, both IPv4 and IPv6 can be supported. Therefore, CYPHONIC can provide inter-connectivity among IoT devices. CYPHONIC also provides mobility transparency because it detects the movement of IoT devices on the network in advance.

CYPHONIC end nodes communicate with cloud service in advance to realize direct communication between devices. Therefore, the traffic to the cloud service from each device will be expected to increase. However, the scalability of the current cloud service is limited because each service is designed for single processing. When the cloud group is accessed frequently, it is expected that the delay will increase. This paper develops an extensible cloud service implementation supporting multi-thread processing. As the data-sharing scheme, we employ Redis supporting an in-memory fast database. According to the extension in this paper, the new implementation realizes the cloud service on a multi-thread environment.

2. CYPHONIC

Fig. 1 shows the system model of the overlay network technology for cyber-physical communication protocol called CYPHONIC. CYPHONIC can provide an end to end accessibility and IP mobility over IPv4 and IPv6 networks. CYPHONIC consists of the cloud service and end nodes called CYPHONIC nodes. The cloud service has three functions: authentification service (AS), node management service (NMS), and tunnel relay service (TRS). AS provides the authentification process for CYPHONIC nodes and distributes a shared encryption key for communication between NMS and CYPHONIC nodes. NMS manages CYPHONIC nodes and directs a signaling process to create a UDP tunnel between CYPHONIC nodes. TRS provides a relay service for communication between IPv4 and IPv6 networks and communication between private networks. CYPHONIC achieves IP mobility by assigning a virtual IP address and fully qualified domain name (FQDN) from the CYPHONIC Cloud group to CYPHONIC nodes.

- Authentication Service
  AS performs an authentication process to certify that

the CYPHONIC node is an authorized user. It also distributes necessary information for information management such as a common key for encrypting communication between NMS and the CYPHONIC node.

- Node Management Service
NMS allocates the virtual IPv6 address from the virtual IP address prefix region and Fully Qualified Domain Name (FQDN) according to NMS to CYPHONIC Node. Since each CYPHONIC Node registers its network information to its NMS, NMS can manage a signaling process to create a UDP tunnel between CYPHONIC Nodes. Additionally, CYPHONIC Nodes can realize secure communication because NMS distributes a common key to encrypt an end key for encrypting packets over the UDP tunnel.

- Tunnel Relay Service
TRS has the role of relaying communication between CYPHONIC nodes, and it is used only in the following cases: communication for CYPHONIC nodes behinds NAPT routers and communication for CYPHONIC nodes in IPv4 and IPv6 networks.

- CYPHONIC node
CYPHONIC node requests an authentification to AS and obtains an FQDN and a virtual IP address from MNS. This will allow the CYPHONIC node to join the CYPHONIC network When the CYPHONIC node starts communication with a correspondent node, it transmits a tunnel creation request including the FQDN of the correspondent node to its NMS. It also follows an instruction process from NMS to establish tunnel communication. After completing the tunnel construction, an application can perform data communication by using the virtual IP address. Since the virtual IP address does not change even when the real network changes, continuous communication is always possible.

3. Proposed System

The current implementation of the cloud service executes all processes in the parent thread. As a result, it cannot utilize a multi-thread environment to improve the performance even if a typical CPU implements multi-core processors. Therefore, it is necessary to change to a mechanism in which the cloud group operates in multi-thread. In addition, the cloud group keeps the state in several processes. In order to support multithreading, it is necessary to change to a mechanism that does not retain the state. Therefore, we add a State Database that holds the state temporarily. Thereby, the cloud group can be processed in a multi-thread without having a state. The details of the cloud group are described below.

3.1 Signaling

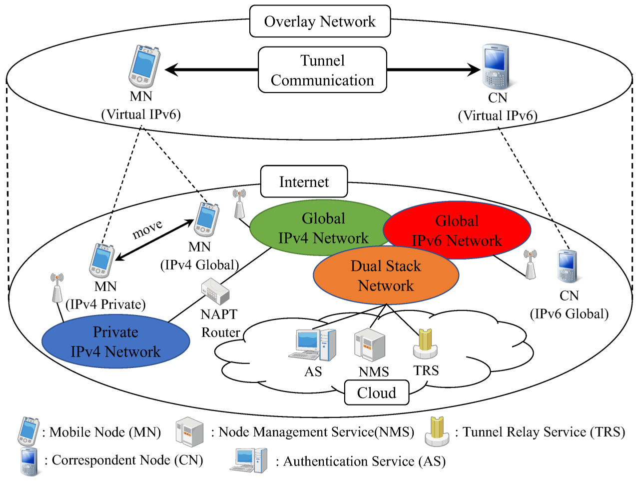

Figure 1: Overview of CYPHONIC.

CYPHONIC has several signaling patterns for multi-thread according to the network condition of end-nodes. The signaling is classified into three processes: the login process, the registration process, and the tunnel establishment process. The signaling process is shown in Fig. 3. We add a database called State Database to AS and NMS to support multi-threading. For simplicity of explanation, we assume the global to global connection, where both end-nodes have a pair of global IPv4 or IPv6 addresses. The following is a detailed signaling process for achieving intercommunication between CYPHONIC nodes.

- Authentification process
As the end-nodes must join the CYPHONIC network through the authentification process, the CYPHONIC node transmits the login request message to AS. If the authentication is successful, the common key is transferred to NMS. At this time, AS stores the common key and FQDN in State Database until waiting for an ACK from NMS. AS that has received ACK refers to the key information and FQDN from State Database and returns it to MN.

- Location registration process
Since NMS should manage network information of end-nodes, the CYPHONIC node transmits the registration request message to their NMS to register network information. When the location registration is completed, the NMS notifies the virtual IP address of MN.

- Tunnel establishment process
MN executes the tunnel establishment processing with the FQDN of the correspondent node to build a tunnel communication to CYPHONIC node. NMS acquires the FQDN of the correspondent node and manages the signaling process according to the network information of both CYPHONIC nodes. At this time, the NMS stores the key information and the FQDN in State Database until waiting for an ACK from CN. NMS that has received

the ACK refers to the information from State Database and passes the path information to the MN. MN generates an end key and performs route confirmation based on the tunnel establishment information. After that, both nodes can perform secure communication with the end key.

3.2 System model

Fig. 2 shows the system model of the cloud service corresponding to multi-thread processing. In the proposed implementation, the parent thread generates a child thread for each transaction. Since the child thread will be freed after the transmission of a packet, another thread should process the response packet. Therefore, the new implementation employs a database system to forward the state information of the previous thread to the new thread.

Firstly, the cloud group receives the packet from Client 1 in the parent thread. Then, the parent thread creates a child thread. The child thread executing the process saves state information to the state database before sending it to Client 2. The thread received from Client 2 similarly creates a child thread. Then, the child thread extracts state information from the state database and sends it to Client 1. As a result, the new thread can continue to process transactions and handle multi-threaded processing environments.

3.3 Prototyping

We have implemented the Authentication Service (AS) and Node Management Service (NMS) for a multi-thread processing environment. AS and NMS have implemented a function to execute event processing in a thread pool using an event-driven architecture. In addition, AS and NMS have a process of holding a state on the way. Therefore, we have implemented a function to store the state in a high-speed in-memory database called Redis. Also, if the child thread obtains state information from the state database, the state database must uniquely identify each state information. Therefore, the unique ID was added to the message format of CYPHONIC. Since a unique ID is assigned to one child thread in the unique ID, status information can be identified.

Table 1 shows the implementation environment. AS and NMS were implemented as virtual machines. As a result, it was confirmed that AS and NMS performed all signaling processes properly on Linux OS.

4. Conclusions

This paper has proposed a new architecture of CYPHONIC cloud service to support a multi-thread processing environment. The developed implementation manages state information of each transaction by using Redis that is an in-memory database system. The experimental results showed that the new implementation performs CYPHONIC cloud service on the multi-thread processing environment.

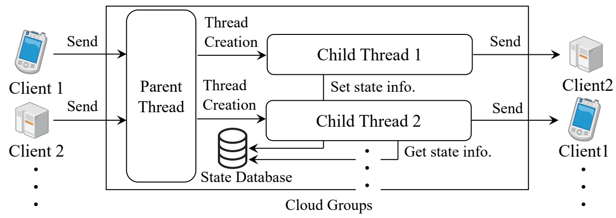

Figure 2: System model of cloud service.

Table 1: Implementation environment

Host Machine		Virtual Machine
OS	macOS Mojave 10.14.6	Ubuntu 19.04
CPU	1.4 GHz Intel Core i5	-
Memory	16GB	1GB
Software	VirtualBox 6.0	-
State Database	-	Redis 5.0.3

Acknowledgment

This work is supported in part by the collaborative research project with Mobiline Co., Ltd., Japan, Grant-in-Aid for Scientific Research (C)(17K00142), Japan Society for the Promotion of Science (JSPS), and the the Cooperative Research Project of the Research Institute of Electrical Communication, Tohoku University.

References

[1] W. Boulevard. Internet protocol. The Internet Engineering Task Force (IETF) RFC 791, September, 1981.

[2] C. Huitema. Teredo: Tunneling ipv6 over udp through network address translations (nats). The Internet Engineering Task Force (IETF) RFC 4380, February, 2006.

[3] M. Bagnulo, P. Matthews, and I. van Beijnum. Stateful nat64: Network address and protocol translation from ipv6 clients to ipv4 servers. The Internet Engineering Task Force (IETF) RFC 6146, April, 2011.

[4] T. Hain. Architectural implications of nat. The Internet Engineering Task Force (IETF) RFC 2993, November, 2000.

[5] C. Perkins. Ip mobility support for ipv4, revised. The Internet Engineering Task Force (IETF) RFC 5944, November, 2010.

[6] D. Johnson, C. Perkins, and J. Arkko. Mobility support in ipv6. The Internet Engineering Task Force (IETF) RFC 3775, June, 2004.

[7] Katsuhiro Naito, Kohei Tanaka, and Kensuke Tanaka. Cyphonic: Overlay network technology for cyber physical communication. In The 10th International Multi-Conference on Complexity, Informatics and Cybernetics: IMCIC 2019, pp. 1–6, March 2019.

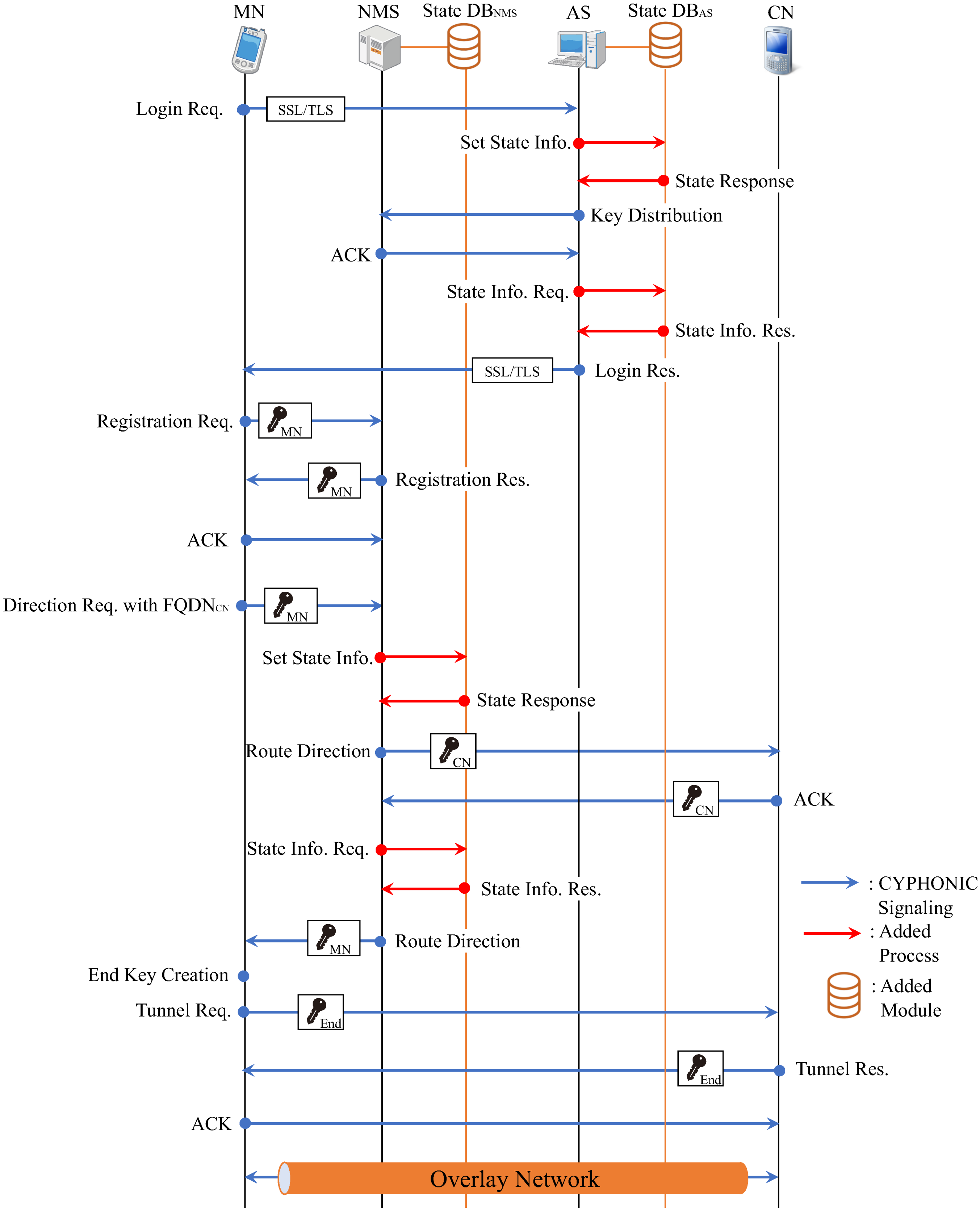

Figure 3: CYPHONIC Signaling