Enhancing Route Optimization for NAPT Traversal in CYPHONIC: Design and Implementation

Kazushige Matama
Aichi Institute of Technology,
Toyota, Aichi 470-0392,Japan
matama@pluslab.org

Hijiri Komura
hjr3ikmr@pluslab.org

Ren Goto
Aichi Institute of Technology,
Toyota, Aichi 470-0392,Japan
r0719en@pluslab.org

Hidekazu Suzuki
Meijo University,
Nagoya, Aichi 468-8502,Japan
hsuzuki@meijo-u.ac.jp

Katsuhiro Naito
Aichi Institute of Technology,
Toyota, Aichi 470-0392,Japan
naito@pluslab.org

***Abstract*—The modern Internet encounters challenges in establishing direct communication due to IP version differences and the presence of NAPT. The authors have proposed CYPHONIC to achieve communication connectivity and mobility transparency in IP communication. CYPHONIC facilitates NAPT traversal by leveraging a cloud service to relay device communications. However, this approach introduces challenges such as load concentration on the cloud service and redundant routes. This paper presents a route optimization technique that involves connecting devices through the cloud service and selectively transitioning to direct communication under specific conditions. The proposal focuses on the fact that NAPT involves several types of port translation techniques, a particular combination of which enables direct communication between devices. The introduction of the proposed technique extends the conventional CYPHONIC signaling and adds a function to try to communicate directly with both devices. The evaluation experiment shows whether the proposal is effective enough for the combination of NAPT types, thus clarifying the effectiveness of route optimization.**

***Index Terms*—NAPT, Overlay Network, CYPHONIC, IoT**

I. Introduction

In recent years, services based on the Internet of Things (IoT) have attracted significant attention. IoT services involve interconnected systems of IoT devices and find applications in various fields, including smart homes and smart factories [1], [2]. Two network models are commonly used for device coordination: the Client-Server type and the Peer-to-Peer (P2P) type [3], [4]. The Client-Server consists of a server that manages information and provides services and a client that uses the provided services. The Client-Server type consists of a server that manages information and provides services and clients that utilize these services. One drawback of the Client-Server type is the need for communication to continually pass through servers, resulting in redundant communication paths between clients and potential server overload. On the other hand, the P2P type involves clients with server functionalities, which reduces the concentration of processes and enables communication through the shortest path. However, there are situations where P2P communication becomes challenging due to the influence of communication protocols.

IoT devices use Internet Protocol (IP) for communication in the Internet [5]. IPv4 addresses are generally used as device identifiers in IP communication. The available IP addresses are limited to only 2^{32}. Thus, it concerns IP address exhaustion with the increase in the number of devices [6]. Network Address Port Translation (NAPT) and IPv6 address the IP address exhaustion problem [7], [8]. NAPT can convert private IP addresses used within internal networks into globally routable IP addresses for Internet usage. NAPT allows sharing of a single global IP address among multiple devices. However, NAPT conceals devices behind it from the external network, making it difficult for devices in the external network to initiate communication with devices behind NAPT. This challenge is known as the NAPT traversal problem [9]. On the other hand, IPv6 provides a much larger address space with 2^{128} available addresses, effectively mitigating the address exhaustion problem. However, due to the differences in packet structure and lack of compatibility between IPv4 and IPv6, mutual communication between the two protocols becomes challenging [10]. Furthermore, as the transition from IPv4 to IPv6 has not been completed in the current network environment, IPv4 and IPv6 coexist for a while. Furthermore, when considering recent devices, such as smartphones, they are equipped with multiple interfaces that allow them to switch interfaces and move within the network. However, the IP protocol does not consider device mobility, leading to the challenge of transport layer communication being disrupted when the network changes, known as the issue of mobility transparency.

These issues require comprehensive solutions in utilizing services based on mutual communication between devices. The authors have proposed and developed a technology called CYber PHysical Overlay Network over Internet Communication (CYPHONIC) to solve the above issues [11], [12]. CYPHONIC provides an overlay network that hides the actual network by assigning virtual IP addresses to the communicating devices. CYPHONIC provides a NAPT traversal mechanism by managing and coordinating the network information of communication devices. Also, when both communication

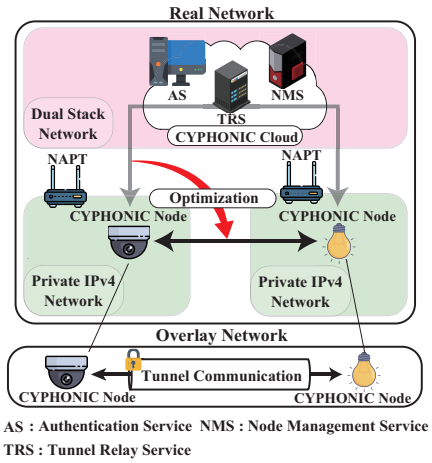

Fig. 1. Overview of CYPHONIC

devices are located behind separate NAPT, the cloud service acts as a relay for their communication. However, depending on the types and combinations of NAPT, there are cases where direct communication between the two devices is possible, resulting in issues of route redundancy.

This paper proposes a route optimization process that enables direct communication when a cloud service relays the communication. The proposed method focuses on the characteristics of each type of NAPT and switches the paths when NAPT traversal is possible through direct communication. NAPT has several patterns of rules for the translation of IP addresses and port numbers. Additionally, some translation rules allow communication from external networks under certain conditions. Therefore, we propose an approach that achieves NAPT traversal in direct communication by implementing specific processing for each type of NAPT and combinations of NAPT.

II. Overlay Network Protocol

A. CYPHONIC

CYPHONIC establishes a UDP tunnel on the network between the communicating end devices. This enables secure End-to-End (E2E) communication through the UDP tunnel. Fig. 1 provides an overview diagram of CYPHONIC. CYPHONIC comprises two components: the CYPHONIC Cloud, the cloud service, and the CYPHONIC Node, which represents the end devices. The CYPHONIC Cloud comprises three types of services.

- Authentication Service (AS)
  AS manages the account information of CYPHONIC Node and authenticates CYPHONIC Node. AS also assigns a Full Qualified Domain Name (FQDN), an identifier of the CYPHONIC Node, and a virtual IP address, in addition to distributing a common key necessary for encrypted communication with the NMS.
- Node Management Service (NMS)
  NMS manages network information such as IP addresses and port numbers of CYPHONIC Nodes and controls the tunnel establishment process between CYPHONIC Nodes.
- Tunnel Relay Service (TRS)
  TRS relays communication between CYPHONIC Nodes which are difficult to communicate directly, such as between nodes under NAPT routers and between nodes in IPv4/IPv6 networks.

CYPHONIC Node communicates with CYPHONIC Cloud and performs E2E communication with the other CYPHONIC Node. CYPHONIC Node has two processes: pre-tunnel communication and tunnel establishment.

In the pre-tunnel communication process, authentication and registration are performed. During the authentication process, the CYPHONIC Node proves its legitimacy as a user through TLS communication with AS. For this purpose, the CYPHONIC Node utilizes the address information and encryption keys provided by AS to communicate with NMS securely for registration. Through encrypted communication with the NMS, the CYPHONIC Node registers its network information in the registration process.

The tunnel establishment process involves route selection and tunnel construction. In the route selection process, NMS instructs the communication path based on the network information of both CYPHONIC Nodes involved in the communication. Additionally, when issuing route instructions, the NMS generates Temporary Key for encrypting communication between nodes and Tunnel Key for communication between nodes and TRS. If communications via the TRS are necessary, the CYPHONIC Nodes make a Hole Punching to TRS to allow CYPHONIC Nodes to communicate with TRS. In the tunnel establishment process, the End Key generated by one of the CYPHONIC Nodes is encrypted with the Temporary Key. Then, the CYPHONIC Node shares the End Key with the other CYPHONIC Node via the TRS using encrypted communication with the Tunnel Key.

III. Proposed System

This section discusses the route optimization technique for communication via NAPT in CYPHONIC. In CYPHONIC, when both communicating CYPHONIC Nodes exist behind NAPT, the TRS acts as an intermediary to facilitate NAPT traversal and enable communication between them. NAPT employs multiple port translation rules, and in specific scenarios, direct communication is feasible between devices situated behind the NAPT. However, the conventional CYPHONIC approach relies on the TRS for all communications when both devices are located behind NAPT, leading to increased traffic through the TRS and path redundancy concerns. A communication path switch between the two CYPHONIC Nodes is necessary to address these issues, enabling route optimization to bypass the TRS for direct communication.

The proposed method allows for E2E communication between CYPHONIC Nodes without going through the TRS if direct communication is possible through route optimization. During the route optimization, a tunnel is established via the TRS, and if E2E communication between CYPHONIC

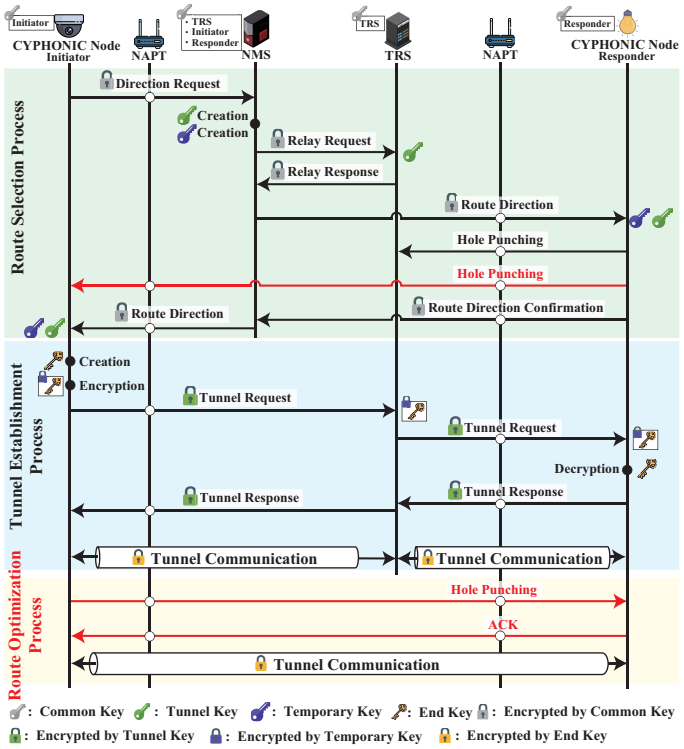

Fig. 2. Route Optimization Process

Nodes is available, the communication paths are switched accordingly. By obtaining NAPT translation information from each other and mapping it to the NAPT translation table, the route optimization process enables E2E communication between CYPHONIC Nodes behind the NAPT.

A. NAPT Type

Network Address and Port Translation (NAPT) offers two distinct types: Cone-type NAPT, which applies a uniform translation rule regardless of the external destination, and Symmetric NAPT, which employs separate translation rules for each external destination. The following section provides a detailed description of each NAPT type.

- Full Cone
  Full Cone is a type of NAPT that falls under the category of Cone-type NAPT. It can forward a received packet from an external address and port number to a corresponding internal address and port number when NAPT translation has occurred at least once, and a matching entry exists in the management table. Furthermore, Full Cone performs packet forwarding even if it receives a packet from a destination to which it has not previously sent any packets.
- Address-Restricted Cone
  An address-Restricted Cone is a Cone-type NAPT. When a corresponding entry exists, Address-Restricted Cone refrains from forwarding packets from a destination to which it has never received packets from the inside. However, it does forward packets from destinations to which it has received packets from the inside, directing them to the internal address and port number corresponding to the external IP address and port number.
- Port-Restricted Cone
  A Port-Restricted Cone is a type of Cone-type NAPT. In the presence of a corresponding entry, Port-Restricted Cone refrains from forwarding packets from a destination where it has never received packages from the inside. However, it does forward packets originating from the inside and destined for specific destination addresses and port numbers, directing them to the internal address and port number corresponding to the external IP address and port number.
- Symmetric
  Symmetric is a type of NAPT that performs address and port number mapping for each external destination. Consequently, Symmetric forwards packets exclusively from the destination address and port number sent from the inside to the corresponding internal address and port number.

B. NAPT Combination

In the conventional CYPHONIC system, there are different patterns of NAPT traversal via TRS: scenarios where both devices exist behind other NAPT routers, cases involving multiple NAPT routers (known as multistage NAPT), and situations where both devices are behind the same NAPT. Consequently, achieving NAPT traversal for each of these scenarios is essential.

Firstly, when both devices exist behind different NAPT routers, they must exchange the translation information of each other's NAPT. Additionally, mapping to the NAPT is required if at least one of the NAPT routers is an Address-Restricted Cone or a Port-Restricted Cone. Furthermore, when at least one of the NAPT routers is Symmetric since the translation information differs for each destination, the external device must acquire the translation information directly.

In scenarios where at least one of the devices exists behind a multistage NAPT, the most muscular restriction among all the constituent NAPT routers governs the overall constraint for the entire multistage NAPT. NAPT routers are prioritized in terms of restrictions, with Full Cone being the least restrictive, followed by Address-Restricted Cone, Port-Restricted Cone, and Symmetric. Therefore, in a multistage NAPT configuration, if the external NAPT is Full Cone but the internal NAPT is Symmetric, the entire multistage NAPT is treated as Symmetric.

Finally, if both devices exist in the same NAPT, they should communicate within the NAPT network without traversing the NAPT. Consequently, checking whether each device is behind the same NAPT and facilitating communication using the internal IP address when required is necessary.

C. Routing Optimization Process

The NAPT traversal technique enables communication through the NAPT by employing NAPT mapping and obtain-

ing the external IP address and port number of the device behind the NAPT from the device outside the NAPT. In the route optimization process, NAPT mapping is achieved by sending a Hole Punching message from a device behind the NAPT to a device outside the NAPT. To obtain the NAPT's external IP address and port number, the information on the external network of the device stored in NMS is utilized.

Cone-type NAPTs, including Full Cone, Address-Restricted Cone, and Port-Restricted Cone, have a single translation rule for all external destinations. As a result, NAPT traversal is possible by using the network information on the external side of the NAPT stored in NMS. However, Symmetric NAPT maps addresses and port numbers differently for each external destination, resulting in a different set of translation information than what is registered in NMS. Therefore, using the external network information stored in NMS for NAPT traversal is not feasible in the case of Symmetric NAPT.

For scenarios where the other device exists behind a Symmetric NAPT, communication is addressed by interacting with the Hole Punching packet's source address and port number from the other device. This enables direct communication even when both devices are behind a multistage NAPT. In cases where both devices exist in the same NAPT, communication between internal networks is necessary. Hence, the proposal incorporates a process to switch to the internal IP address when the external IP addresses of both devices, obtained from NMS, match.

D. Signaling

Fig. 2 illustrates route optimization signaling during tunnel establishment via TRS. Upon receiving the Route Direction from NMS, the Responder sends a Hole Punching to the Initiator's external IP address and port number. The Responder's NAPT generates the translation information for the Initiator by processing the received Hole Punching packets. Additionally, the Initiator stores the source address and port number of the Hole Punching from the Responder, enabling direct communication if the Responder's NAPT is symmetric.

During tunnel communication via TRS, the Initiator performs the route optimization process for direct communication. After receiving the Tunnel Response from the Responder, the Initiator sends a Hole Punching to the Responder. Upon receiving the Hole Punching from the Initiator, the Responder stores the source address and port number, enabling direct communication if the Initiator's NAPT is symmetric.

The Responder then switches the path from communication via TRS to direct communication with the Initiator and sends an ACK. The Initiator switches from communication via TRS to direct communication with the Responder upon receiving the ACK. The outlined process completes the route optimization procedure.

IV. PERFORMANCE EVALUATION

This section conducts the operational verification of the proposed route optimization process. In the conventional CYPHONIC setup, when both CYPHONIC nodes exist behind NAPT, TRS is used as an intermediary for relaying, irrespective of the type and combination of the NAPTs. However, the issue of route redundancy arises since there are specific patterns where direct communication is feasible regardless of the NAPT type and combination. To address this concern, we verify the route optimization process for each NAPT pattern in the actual network environment to ensure that route optimization is effectively executed.

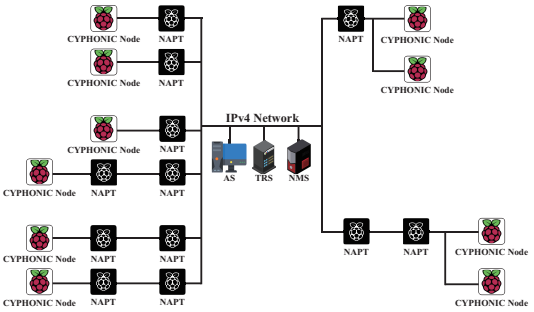

Fig. 3. Evaluation environment

TABLE I
SPECIFICATIONS OF THE MEASURING DEVICES

Virtual Machine (AS,NMS,TRS)	
OS	Ubuntu 21.10
CPU	3.50GHz 2cores Intel(R) Core i9-11900K
Memory	2GB RAM
Raspberry Pi 4 Model B (NAPT Router)	
OS	Raspbian GNU/Linux 10.0 (Buster)
CPU	Quad Core 1.5GHz Broadcom BCM2711 64bit
Memory	4GB RAM
Raspberry Pi 3 Model B (CYPHONIC Node)	
OS	Raspbian GNU/Linux 10.0 (Buster)
CPU	Quad Core 1.2GHz Broadcom BCM2837 64bit
Memory	1GB RAM

A. Verification Experiments

This paper introduces the route optimization process and verifies its functionality in a real network environment where NAPTs are present. The verification environment is depicted in Fig. 3. It consists of the CYPHONIC Cloud (comprising AS, NMS, TRS, and NAPT devices) integrated into a network that simulates a global network. Each CYPHONIC Node is deployed in the network behind the NAPT.

The specifications of the measuring devices are outlined in Table I. The evaluation includes a total of 292 combinations, representing different NAPT type patterns when up to two NAPTs are added for each CYPHONIC Node.

B. Evaluation

1) Both devices exist behind the different single NAPT: Table II presents the verification results for the route optimization process when passing through one NAPT for each CYPHONIC node. Firstly, it was observed that route optimization is feasible for communication between Cone-type NAPTs when both devices exist under the same NAPT. For cases

TABLE II
INITIATOR : SINGLE NAPT, RESPONDER : SINGLE NAPT

Initiator \ Responder	Full Cone	Address-Restricted Cone	Port-Restricted Cone	Symmetric
Full Cone	possible	possible	possible	possible
Address-Restricted Cone	possible	possible	possible	conditionally possible
Port-Restricted Cone	possible	possible	possible	impossible
Symmetric	possible	possible	impossible	impossible

where the Responder's NAPT is Symmetric, and the Initiator's NAPT is Full Cone, it was confirmed that Hole Punching from the Responder could be received, thus enabling route optimization even when the Responder's NAPT is Symmetric.

When the Initiator's NAPT is Address-Restricted Cone, route optimization can be achieved by executing the route optimization process again within a short interval. During the first route optimization process, Hole Punching from the Responder cannot be received due to the restriction of the Initiator's NAPT. However, the subsequent Hole Punching from the Initiator to the Responder facilitates mapping the Responder's IP address to the Initiator's NAPT. The mapping to the Initiator's NAPT enables the Responder to receive Hole Punching from the Initiator during the second route optimization process, thus enabling successful route optimization. As a result, route optimization was confirmed to be feasible for most cases, except for a few NAPT patterns.

2) Devices exist behind the NAPT of different multistage NAPT: We verified the route optimization process in scenarios where the Initiator device exists behind a single NAPT, and the Responder device exists behind two different NAPTs, and where the Initiator device exists behind two different NAPTs, and the Responder device exists behind a single NAPT. Furthermore, we tested the route optimization process when passing through two NAPTs for each case.

The results indicate a certain rule for the combinations of NAPTs that can be optimized. Specifically, we confirmed that the type of NAPTs with more substantial restrictions significantly impacts the optimization. The NAPTs are prioritized in terms of restrictions in the order of Full Cone, Address-Restricted Cone, Port-Restricted Cone, and Symmetric.

For example, the inner NAPT is a Port-Restricted Cone, the Responder's outer NAPT is Symmetric, and the inner NAPT is an Address-Restricted Cone when the Initiator's outer NAPT is a Full Cone, In this case, the Port-Restricted Cone restriction on the Initiator's side and the Symmetric restriction on the Responder's side significantly influence the success of optimization. As a result, we confirmed that route optimization is feasible in most cases, surpassing a significant portion of the total combinations.

3) Both devices exist behind the same NAPT: We verified the route optimization process in scenarios where both devices exist behind the same NAPT. We confirmed that route optimization is feasible in all cases, regardless of the number of NAPTs involved.

V. CONCLUSIONS

In this paper, we have proposed extension mechanisms for CYPHONIC. This protocol achieves mobility transparency and communication connectivity by adding a route optimization method for direct communication environments without relying on cloud service relays. The proposed approach considers specific types and combinations of NAPTs and switches the communication paths when NAPT traversal allows for direct communication. Through comprehensive verification, we have confirmed that route optimization is feasible in most cases on the Internet.

ACKNOWLEDGMENT

This work is supported in part by Grant-in-Aid for Scientific Research (C)(21K11877), the Japan Society for the Promotion of Science (JSPS), and the Cooperative Research Project of the Research Institute of Electrical Communication, Tohoku University.

REFERENCES

[1] M. Rokonuzzaman, M. I. Akash, M. K. Mishu, W. Tan, M. A. Hannan and N. Amin, "IoT-based Distribution and Control System for Smart Home Applications," May 2022.
[2] P. A. Okeme, A. D. Skakun, and A. R. Muzalevskii, "Transformation of Factory to Smart Factory," January 2021.
[3] Y. Zhu, W. Wu, and D. Li, "Efficient Client Assignment for Client-Server Systems," August 2016.
[4] L. Meftah, R. Rouvoy, and Chrisment, I., "Testing Nearby Peer-to-Peer Mobile Apps at Large," May 2019.
[5] J. Postel, "Internet Protocol," September 1981.
[6] J. Beeharry, and B. Nowbutsing, "Forecasting IPv4 exhaustion and IPv6 migration," August 2016.
[7] P. Srisuresh and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations, " August 1999.
[8] S. E. Deering and B. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," July 2017.
[9] H.-C. Wang, C. Chen, and S.-H. Lu, "An SDN-based NAT Traversal Mechanism for End-to-end IoT Networking, " September 2019.
[10] Z. Qin, "Seamless converging system for IPv4/IPv6 transition," November 2017.
[11] T. Yoshikawa, H. Komura, C. Nishiwaki, R. Goto, K. Matama, and K. Naito, "Evaluation of new CYPHONIC: Overlay network protocol based on Go language," January 2022.
[12] S. Horisaki, K. Matama, K. Naito, and H. Suzuki, "A Proposal of QUIC-based CYPHONIC for Encrypted End-to-End Communications," November 2022.