Implementation and Evaluation of IP Mobility Functions in NTMobile for Android

*

Kaito Kuromiya
Graduate School of Science and Technology
Meijo University
Aichi 468-8502, Japan
kaito.kuromiya@wata-lab.meijo-u.ac.jp

Hisayoshi Tanaka
Graduate School of Science and Technology
Meijo University
Aichi 468-8502, Japan
hisayoshi.tanaka@ucl.meijo-u.ac.jp

Hidekazu Suzuki
Graduate School of Science and Technology
Meijo University
Aichi 468-8502, Japan
hsuzuki@meijo-u.ac.jp

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

Akira Watanabe
Graduate School of Science and Technology
Meijo University
Aichi 468-8502, Japan
wtnbakr@meijo-u.ac.jp

***Abstract*—NTMobile (Network Traversal with Mobility) is a technology which can realize IP Mobility that enables continuous communication even if nodes change their networks, along with solving the so-called NAT traversal ploblem and enabling communication between different IP versions. NTMobile was initially implemented on Linux PC. And now that applications by which this technology can be used for Android nodes have become available, using general applications. However, due to the reason that the function related to the detection of IP address change is not implemented yet, it has thus far not been possible to realize IP Mobility by using general applications for Android nodes. In this paper, we have implemented the full function of IP Mobility in Android nodes and verified the IP Mobility.**

***Index Terms*—IP Mobility, NAT Traversal, Smart Devices, VPN Service**

I. Introduction

With the popularization of smart phones, the mobile data traffic has been drastically increasing, and it is forecast that the volume of mobile data traffic of 3.7 EB/month recorded in 2015 is expected to increase to 30.6 EB/month by 2020 [1]. Therefore, it is required to offload data from mobile data communication network to IP networks. However, the present IP network has the following problems. Namely, while IPv4 and IPv6 addresses coexist in the IP network, direct communication between addresses of different IP versions is not possible. Moreover, in the case of IPv4, it is not possible to initiate communication from the Internet side to nodes existing behind the NAT (so-called "NAT traversal problem") and also, ongoing communication is interrupted by the change of IP addresses following the change of the network. In the IP network, an IP address have two roles of location identifier and communication identifier. For that reason, it is necessary to separate the roles of an IP address to realize "IP Mobility".

As a technology to solve these problems, we have been proposing "NTMobile" (Network Traversal with Mobility) [2]–[4].NTMobile is a communication technology to realize IP Mobility in the IPv4/IPv6 co-existing environment. For NTMobile, virtual IP addresses which are not influenced by the location of the nodes are firstly assigned to the nodes in which NTMobile is introduced (hereafter "NTM node"). Then, the application implemented in the NTM node initiates communication using the assigned virtual IP address. The actual communication is conducted by way of the encapsulation/ decapsulation processing of the real IP address of the node. In this way, the application can continue communication without being interrupted by the change of IP addresses. Also, the NTM node can receive communication route directions even if the node exists behind NAT, by sending periodical Keep Alive signals towards the communication route indicator set on the Internet. Moreover, even for the communication between IPv4 and IPv6, or for that between nodes behind different NATs where direct communication is normally not possible, the NTM node can conduct communication through encapsulation/ decapsulation processing by using original relay devices. In this way, NTMobile is a quite useful technology to solve the presently existing problems for the Internet, but there still exist some problems when implementing the system in smart phones.

It is necessary for NTMobile to obtain administrative right on the node in order to conduct communication based on NTMobile, as it performs encapsulation/decapsulation processing in the kernel space. However, it is difficult to promulgate the system for smart phones for which the rooting of the node is not recommended.

In order to solve the above-said problem, We have been proposing a method of realizing NTMobile communication by

using the encapsulation / decapsulation processing provided by the VpnService of Android (hereafter "VpnService-type NTMobile") [4], [5]. In VpnService-type NTMobile, Some of the functions such as the NAT traversal and intercommunication between different IP versions have been verified as effective. However, "IP Mobility" by this type of NTMobile has not been realized yet because the library conducting NTMobile communication cannot detect address changes by different operating systems, such as Linux, Android, iOS, etc. commonly.

Accordingly, in this study, we realize IP Mobility by separating the VpnService-type Mobile ' s function of detecting address changes from the library conducting NTMobile communication.

Hereafter, we describe the function of NTMobile in Chapter 2, explain our proposed method/system in Chapter 3, explain the implementation of our system in Chapter 4, describe our evaluation in Chapter 5, and finally summarize our conclusion in Chapter 6.

II. NTMobile

A. Outline of NTMobile

Fig. 1 shows the configuration of NTMobile. Besides NTM nodes, NTMobile consists of "DC" (Direction Coordinator) which administers node information, instructs communication routes and assigns virtual IP addresses, and "RS" (Relay Server) which relays communication between IPv4 and IPv6 and also relays communication between NTM nodes existing behind different NATs. It is assumed that DC and RS are placed in the Dual Stack Network. Also, depending on the scale of the network, it is possible to set multiple units of DC and RS in order to disperse the load. The NTM node is assigned a virtual IP address by DC when it registers to DC through registration processing at the startup time. The application of the NTM node initiates communication by using the assigned virtual IP address. In NTMobile, the real IP address has the role of location identifier, while the virtual IP identifier has the role of communication identifier. When initiating communication, the packet which is created using the virtual IP address is encapsulated by the real IP address with UDP header. Through this method, IP Mobility can be realized, because even if the real IP address of the NTM node has changed, the virtual IP address having the role of communication identifier remains unchanged. The NTM node can always receive instructions from DC because it sends Keep Alive to DC periodically, even if the relevant NTM node is behind NAT. While communication between NTM nodes is basically designed to be conducted directly, RS relays the communication in the case of intercommunication between IPv4 and IPv6, or in the case where NTMobile nodes exist behind different NATs. In the latter case, however, depending on the type of the NAT, communication can be switched to direct communication without relaying by RS [6].

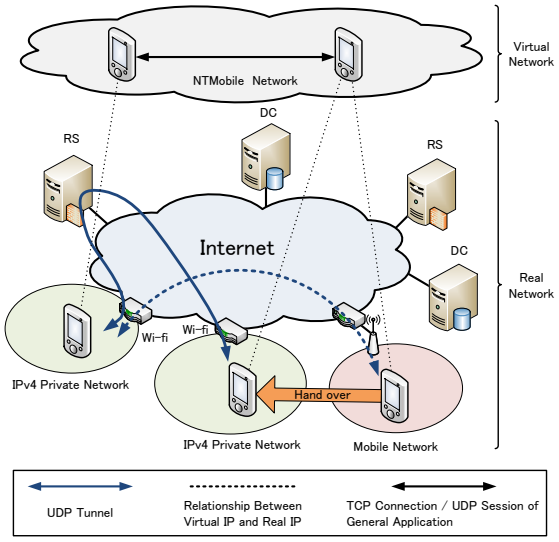

Fig. 1. Configuration of NTMobile.

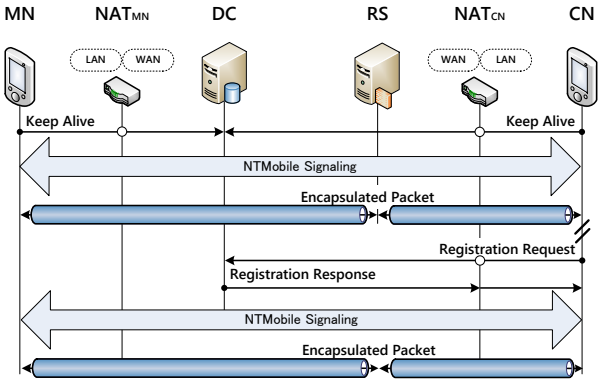

Fig. 2. Moving Sequence of NTMobile.

B. Moving Sequence of NTMobile

Fig. 2 shows the moving sequence of NTMobile relating to the change of a network. Fig. 2 is the sequence in which CN is assumed to have changed its address during communication. In NTMobile, when any address change was detected, a new tunnel route is created by the signaling processing performed anew, after the registration sequence to DC.

C. NTMobile Framework (NTMfw)

We have a communication library called "NTMobile framework" (NTMfw) which enables NTMobile communication. As the entire processing of NTMfw is designed to be conducted in the user space, it is not necessary for nodes to have root privileges. By incorporating NTMfw in the application of smart phones, NTMobile communication becomes feasible for general smart phones to which no rooting is applied. However, because each application has to be conscious of NTMfw, there

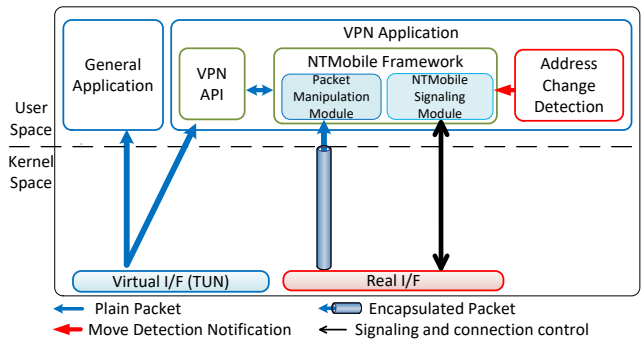

Fig. 3. Module construction of VpnService-type NTMobile.

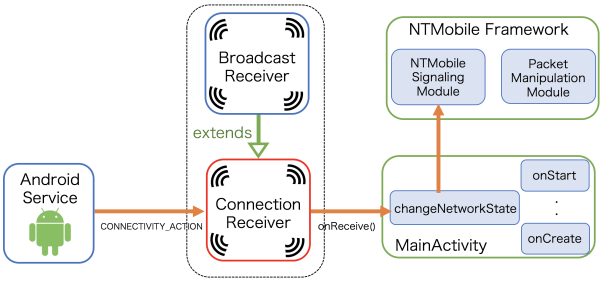

Fig. 4. Network configuration used for the measurements of performance.

exists a problem that it is not possible to apply NTMobile to general applications available at stores.

D. VpnService-type NTMobile

By applying this library, we have developed VpnService-type NTMobile which is a model for using as a smart phone application. The model is implemented by the NTMfw and the API of VpnService (hereafter VPN API).

VPN API is provided for implementing VPN communication smartphone application. As famous VPN API, Vpn Service is provided by Android Developer and NEPacketTunnelProvider is provided by Apple Developer. It is possible for VPN API to realize NTMobile communication with existing applications, by utilizing encapsulation/ decapsulation functions of VPN API. The NAT traversal by VpnService-type NTMobile as well as the end-to-end communication has already been verified. However, the NTMfw can not detect the changes of own IP addresses on other OS than Linux, so this mode has a problem that it is not possible to perform the mobility functions.

III. Realization of Mobility Function for VpnService-type NTMobile

In this chapter, we explain the method of implementing the function of detecting address changes in VpnService-type NTMobile.

In our proposed method, VpnService-type NTMobile perform the processing of the move, by notifying a change of addresses has been detected by the Android application to the NTMfw. Fig. 6 shows the module of VpnService-type NTMobile, the address detection processing of which is conducted separately.

VPN Application is one of application implemented on Android using VpnService-type NTMobile. In VPN Application, it creates TUN, which is a virtual interface, at the time of activation. Thereafter, General Application sends/receives packets via TUN interface. TCP/IP packets are generated in kernel space and the packets are sent to VPN Application via VPN API, instead of sending them to the network. NTMobile Signaling Module, which is contained in NTMobile Framework, carries out the address registration processing and the signaling processing by NTMobile at the time of activating VPN Application. In Packet Manipulation Module, the processing of sending and receiving packets encapsulated by NTMobile is performed. In our proposed method, an address-change detection module is added to VPN Application, independently of NTMfw. When addresses are changed, notification to convey the fact that addresses were changed is sent to NTMobile Signaling Module. Thereafter, IP Mobility of VpnService-type NTMobile is realized by carrying out the moving sequence indicated in Fig. 2, using the notification as the trigger.

IV. Implementation

In this Chapter we report the result of our experiment. We implemented our proposed system in order to verify the feasibility of detecting the change of addresses.

A. Outline of the Processing

Address change detection module detects the change of addresses, using VPN Application. To this end, Connectivity Manager offered by Android is used [7]. If and when the connection status of Android nodes changes, Connectivity Manager broadcasts CONNECTIVITY_ACTION ("android.net.conn.CONNECTIVITY_CHANGE"). Thus, in order to carry out the address change detection from VPN Application, it is necessary to have Connection Receiver to receive CONNECTIVITY_ACTION. For Address Change Detection, notification is sent to the Signaling Module of NTMfw, by using CONNECTIVITY_ACTION as the trigger. Therefore, I created a common library, by adding to NTMfw a function to call the processing which is required at the time of the change of the network for Android. Furthermore, at the time of receiving CONNECTIVITY_ACTION, we also called the processing to deal with the situation where the node moved, from the side of Java, by using the loadLibrary method. Fig. 4 shows the message flow of the implemented system. When the ConnectionReceiver receives CONNECTIVITY_ACTION, it notifies the network switching process of MainActivity.

B. Behavior of Connection Receiver

Connection Receiver is a type of receiver that inherits Broadcast Receiver to receive CONNECTIVITY_ACTION.

In implementing it, we created/installed a listener in the Connection Receiver to monitor the broadcast dispatched by Connectivity Manager at such times as when connection is switched to Wi-Fi, when connection is switched to 3G/LTE communication (mobile communication), and when nodes has become offline. When connection is switched to Wi-Fi or when connection is switched to mobile communication, the processing of the move described in NTMfw is to be executed.

V. Evalution

In this Chapter, we describe our verification of the behavior and the result of the measurements of performance of our system.

A. Verification of the Behavior

In order to verify that VPN Application can execute the move processing correctly at the time when nodes are in move, we performed the following experiment by using a Virtual Private Server. When conducting the experiment, we used Network Analyzer [8] offered by Google Play Store for Android applications. Under the condition that both MN and CN were set behind the NAT, Pings were periodically sent, using the Network Analyzer. In this condition, we changed access points of MN or CN to see the difference. As a result of our behavior verification, it was confirmed that mobility was realized with VPN Application.

B. Measurement of Performance

We measured the time required for the move processing, based on the method. We show the specifications of the devices used for measurements in Tab. I. Also, Fig. 5 shows the network configuration used for the measurements of performance. Here, a router manufactured by NEC Corp. is shown as "Wi-Fi" and that manufactured by Buffalo is shown as "Wi-Fi2".

Fig. 6 shows the result of measurements. From the result, we can see that the time required to switch from Wi-Fi to Mobile was 525[msec] and that from Wi-Fi to Wi-Fi2 was 889 [msec]. Meanwhile, the time required for the signaling of NTMobile was 122.85 [msec].

TABLE I
Specification of each equipment.

	MN, CN	DC, RS (VPS)
OS	Android 7.1.1	CentOS 6.9
CPU	NVIDIA Tegra K1@2.50GHz	Intel(R) Xeon(R) E5-2667v3
Memory	2GB	1536MB

The reason for the short time required for the switching from Wi-Fi to Mobile is inferred to be the result that the switching of IP addresses in a short time is possible because the IP address to make Mobile communication is maintained even during the communication which smart phone is communicating through Wi-Fi. Furthermore, it is considered that it took quite a long time for the communication switching from Wi-Fi to Wi-Fi2, because for that switching, it is necessary to firstly release IP address to Wi-Fi and then to receive assignment of a new IP address from Wi-Fi2.

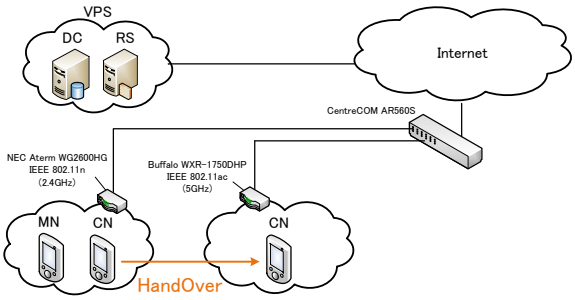

Fig. 5. Network configuration used for the measurements of performance.

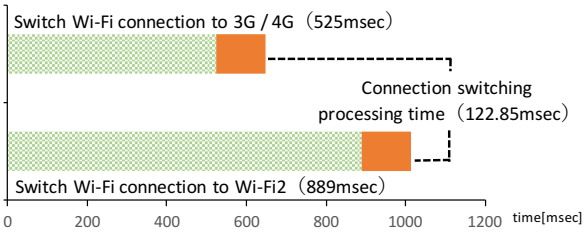

Fig. 6. Results of measurements.

VI. Summary

In this paper, we have proposed that the detection of the change of IP Address is to be separated from the mobility function because the method is different for each OS. We implemented and evaluated the proposed method to realize IP Mobility in VPNService type NTMobile and confirmed the effectiveness. This method can be applied to other OSes, such as iOS, LINUX, etc.

References

[1] Cisco visual networking index. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html.

[2] K. Naito, K. Kamienoo, T. Nishio, H. Suzuki, A. Watanabe, K. Mori, and H. Kobayashi. Proposal of seamless ip mobility schemes: Network traversal with mobility (ntmobile). In *Proc. of the IEEE Global Communications Conference (GLOBECOM 2012)*, Dec. 2012.

[3] H. Suzuki, K. Naito, K. Kamienoo, T.Hirose, and A. Watanabe. Ntmobile: New end-to-end communication architecture in ipv4 and ipv6. In *Proc. of the Proc. of ACM MobiCom 2013*, pp. 171–174, Oct. 2013.

[4] T. Yamada, H. Suzuki, K. Naito, and A. Watanabe. Ip mobility protocol implementation method using vpnservice for android devices. In *Proc. of The 9th International Conference on Mobile Computing and Ubiquitous Networking (ICMU2016)*, pp. 67–68, Oct. 2016.

[5] Vpnservice — android developers. https://developer.android.com/reference/android/net/VpnService.

[6] H. Nodo, H. Suzuki, K. Naito, and A. Watanabe. A proposal of autonomous route optimization in ntmobile. In *IPSJ Journal*, Vol. 54, pp. 1–10, Jan. 2012.

[7] Connectivitymanager — android developers. https://developer.android.com/reference/android/net/ConnectivityManager.html.

[8] Network analyzer - google play. https://play.google.com/store/apps/details?id=net.techet.netanalyzerlite.an&hl=ja.