IP Mobility Protocol Implementation Method Using VpnService for Android Devices

Takayuki Yamada†, Hidekazu Suzuki†, Katsuhiro Naito‡, Akira Watanabe†
†Graduate School of Science and Technology, Meijo University, Aichi 468-8502, Japan
‡Faculty of Information Science, Aichi Institute of Technology, Aichi 470-0392, Japan
Email: 153430022@ccalumni.meijo-u.ac.jp, hsuzuki@meijo-u.ac.jp, naito@pluslab.org, wtnbakr@meijo-u.ac.jp

***Abstract*—In this paper, we propose Network Traversal with Mobility (NTMobile), a mobility protocol in the IP layer that realizes secure connectivity and mobility by enabling network switching during IPv4 and IPv6 network communication. In conventional IP mobility protocol implementation methods for Linux, these modules must be implemented in a kernel space for packet handling. In the proposed NTMobile implementation method, on the other hand, we realize functions in only the user space by employing the Android VpnService API. Using NTMobile, we implemented its modules as an Android service for Android devices. The results of our performance evaluation show that NTMobile can realize an IP mobility protocol without requiring root privileges while significantly reducing throughput.**

I. Introduction

The popularization of smartphones has fostered communication with innumerable devices at any time and place over the Internet. However, in the current Internet environment, IPv4 and IPv6 networks coexist, and these networks are not compatible. Therefore, connectivity cannot be established between devices located in the respective networks. Moreover, devices located behind a network address translation (NAT) router cannot be directly accessed from outside the IPv4 network. This familiar restriction is known as the NAT traversal problem.

To address the above problems, we propose the Network Traversal with Mobility (NTMobile) protocol, which provides encrypted end-to-end connectivity and flexible mobility in IPv4 and IPv6 networks [1]. In the conventional implementation method, it is necessary to install a kernel module in the Linux kernel space of Android devices [2], [3]; e.g., mobility protocols, such as Mobile IP. Therefore, it is difficult to widely spread NTMobile for general smartphones. The proposed NTMobile protocol is an implementation method of an IP mobility protocol for Android devices. All NTMobile modules function as services by using the Android VpnService API. This API is a standard virtual private network (VPN) solution provided by the Android SDK. Using the proposed method, general users can easily install the NTMobile service application through the Google Play store. In addition, we developed an NTMobile tunnel service based on the proposed method and evaluated the end-to-end throughput performance in actual IPv4 and IPv6 networks.

II. NTMobile

The NTMobile architecture introduces a virtual IP address as a fixed IP address that does not depend on a connecting IP network. NTMobile software is installed on the NTMobile node (NTM node). General applications that run in the node employ a connection based on a virtual IP address. A packet addressed to the virtual IP address is encapsulated in a real IP address and routed to the real network by a UDP tunnel. The UDP tunnel is created between the NTM nodes when the NTM node initiates communication. By using the above method, general applications can communicate without being affected by the existence of NAT, by switching between IPv4 and IPv6 networks, and/or by the differences of these networks. In addition, NTM nodes can perform tunnel communication via the most optimal route because the UDP tunnel is established on an end-to-end basis except for certain specific cases.

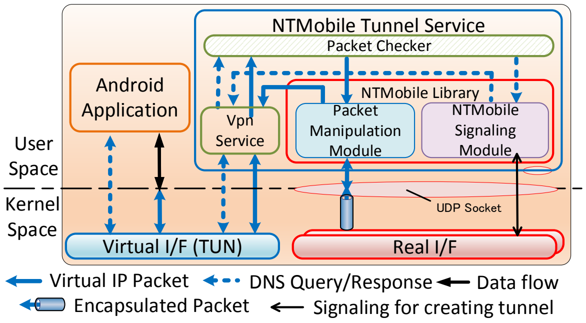

Fig. 1. Implementation overview and packet processing flow.

NTMobile consists of NTM nodes, direction coordinators (DCs), and relay servers (RSs). DC is a coordinator that manages the address information of NTM nodes and provides instructions to NTM nodes for tunnel establishment. RS is a server that relays communications under specific cases, such as communication between IPv4 and IPv6 networks.

III. Proposed Method

Android 4.0 and later versions provide VpnService, a standard VPN API with which application developers can build a VPN. Accordingly, developers can create a virtual interface (I/F) for use in the VPN and arbitrarily configure the routing table for the virtual I/F without requiring root privileges.

The proposed method achieves the NTMobile architecture by using the VpnService. Fig. 1 shows an implementation overview and the packet processing flow. The NTMobile tunnel service is an Android service that runs in the background. It was developed using VpnService and a conventional NTMobile library. The NTM node registers its own real IP address to a DC and obtains a virtual IP address at boot time. A virtual IP address is assigned to the virtual I/F. The routing table of the virtual I/F is configured to receive packets destined for virtual IP addresses and name resolution packets.

When a general application resolves the name of a corresponding NTM node, a DNS query message is obtained by VpnService and passed to the NTMobile tunnel service. That service initiates a signaling process for tunnel establishment and exchanges address information with the corresponding NTM node through DC. The address information includes both the real and virtual IP address. The NTM tunnel service records the relationship between this address information in a tunnel table. From that point, the service creates a DNS response packet and returns the obtained virtual IP address of the corresponding NTM node to the general application through the name resolution process.

As described above, the general application establishes a connection with the corresponding NTM node using a virtual IP address. The packets destined for the virtual IP addresses are received by the NTMobile tunnel service through the VpnService API. The NTMobile Tunnel Service obtains the real IP address of the corresponding NTM node from the tunnel table. It sends the received packet as data to the corresponding NTM node with a UDP (datagram) socket. In this way, the packet destined for the virtual IP address of the corresponding NTM node is encapsulated by UDP and IP headers containing the real IP address of the corresponding NTM node. It is then routed over real networks.

When the corresponding NTM node receives the encapsulated packet from the real I/F, the packet is decapsulated by the usual data reception process of the socket communication. The NTM tunnel service passes the obtained packet destined for its own virtual IP address to the general application through the VpnService API. Consequently, general applications running on each NTM node can communicate based on the established virtual connection. Our proposed method thereby achieves NTMobile in only the user space.

IV. Evaluation

We implemented the NTMobile tunnel service based on our proposed method and evaluated the throughput performance. We constructed four kinds of networks—private IPv4, global IPv4, IPv6, and dual-stack—in a local environment, as shown in Fig. 2. DC and RS were respectively constructed on virtual machines and connected to the dual stack network. In addition, we installed the implemented NTMobile tunnel service on two Android smartphones. Galaxy Nexus was used as the NTM node and was connected to AP_{NAT}, AP_{IPv4}, or AP_{IPv6} using IEEE 802.11n.

As a measurement method, we conducted TCP communication between NTM nodes for 30 seconds using Iperf in the three cases shown below. Each case was measured ten times. In addition, we measured throughput performances of general communication without NTMobile and NTMobile applied in the communication by using a kernel module based on the conventional implementation method in the same cases.

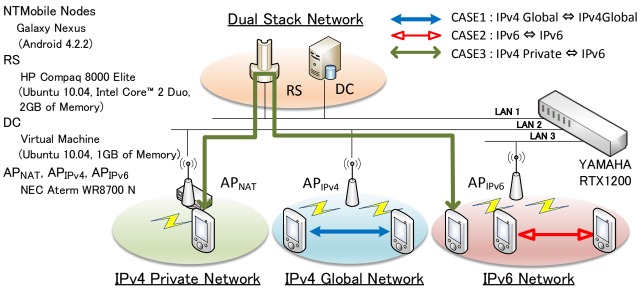

Fig. 2. Evaluation environment and device specifications.

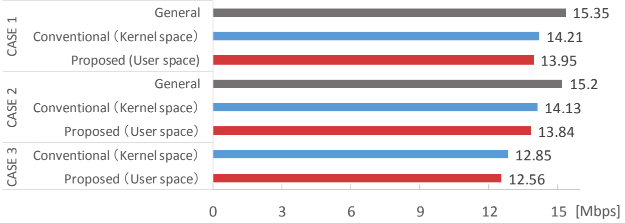

Fig. 3. Measurement results of throughput.

- Case 1 : Communication between global IPv4 networks
- Case 2 : Communication between IPv6 networks
- Case 3 : Communication between private IPv4 network and IPv6 network

Fig. 3 shows the average values of the measurement results. In our proposed method, throughput values decreased by only approximately 1.83% to 2.26% compared to the conventional implementation method. Compared to general communication, throughputs of the NTMobile-applied communication based on the conventional implementation method decreased by approximately 7.04% to 7.43%, and throughputs based on the proposed method decreased by approximately 8.95% to 9.12%. The conventional implementation method manipulated packets for encapsulation and decapsulation in the kernel space. On the other hand, the proposed method manipulated packets in the user space. Therefore, the proposed method produced extra memory between the user space and kernel space through the VpnService. However, a large difference was not confirmed in the evaluation experiments. Based on the results, we confirmed that the proposed method causes no practical problems.

V. Conclusion

The proposed method is applicable to NTMobile and other mobility protocols in the IP layer that adopt a tunneling scheme for packet transmission. Using the proposed method, we can achieve full connectivity and mobility in IPv4 and IPv6 networks with commercially available Android devices without kernel reconfiguration.

References

[1] K. Naito, et al.,"Proposal of seamless ip mobility schemes: Network traversal with mobility (ntmobile)," in *Proc. IEEE GLOBECOM 2012*, pp. 2572–2577, 2012.

[2] K. Kamienoo, et al., "Implementation and evaluation of ntmobile with android smartphones in ipv4/ipv6 networks," in *Proc. IEEE GCCE 2012*, pp. 125–129, 2012.

[3] H. Suzuki, et al., "Ntmobile: new end-to-end communication architecture in ipv4 and ipv6 networks," in *Proc. ACM MobiCom 2013*, pp. 171–174, 2013.