Mobility Support in IPv6 Based on the VIP Mechanism

Fumio Teraoka <tera@csl.sony.co.jp>
Keisuke Uehara <kei@wide.ad.jp>

Abstract

IPv6, the successor of current IP (IPv4), is now under development. Mobility support is one of the requirements for IPv6. This paper proposes a protocol for mobility support in IPv6. The proposed protocol is based on the mechanisms of Virtual Internet Protocol (VIP), which provides mobility in IPv4. Two Hop-by-Hop options are added to support mobility, one for user data and another for control. The proposed protocol provides routing optimization and authentication mechanisms as well as seamless mobility. This paper also proposes an extension of the state transition diagram of TCP for quick resumption of TCP communication when a mobile node reconnects to the network.

1 Introduction

IPv6 the successor of current IP (IPv4) is now under development. Mobility support is one of the requirements[2] for IPv6. There are still several problems in mobile computing from communication mechanisms in the physical layer to application programs. Focusing on the middle layers of the protocol stack, the following four problems can be identified. 1) Reassignment of network parameters: network parameters must be reassigned to a mobile node when the mobile node moves to another subnet. This procedure is troublesome and is prone to mistakes. 2) Seamless mobility: a mobile node must be recognized with an immutable identifier for seamless mobility. A mobile node becomes unrecognizable if its identifier changes upon moving. 3) Impersonation: it is possible to impersonate another node only in the same subnet of the proper node because the IPv6 unicast address specifies the location of the node. In a mobile computing environment, an attacker can impersonate any mobile node regardless of location. 4) Adapting TCP to a mobile computing environment: since TCP cannot distinguish network congestion from node disconnection, TCP communication does not resume immediately when a mobile node is reconnected to the network due to the TCP retransmission algorithm[1].

The first problem can be solved by using an auto-configuration protocol such as DHCPv6 while it is necessary to introduce a new protocol providing seamless node mobility to solve the second. This paper proposes a protocol for mobility support in IPv6, focusing on the second and third problems. The proposed protocol is based on the basic idea of *VIP*[9], which provides seamless node mobility in IPv4.

The fourth problem was found in our test operation of VIP. To solve this problem, this paper adds four control messages in TCP and extends the state transition diagram. Considering compatibility with existing TCP implementations, a simplified mechanism is introduced. The same mechanism is implemented in VIP and works well.

2 IPv6 Overview

This section gives a brief overview of the IPv6 specification to prepare for the description of mobility support in IPv6. Figure 1 shows the header format of IPv6. One of the features of IPv6 is improvement of the way to handle extensions and options. The *Protocol* field of the IPv4 header indicates the type of the next higher layer header. Option headers are added to the end of the base IPv4 header. The length of the entire IPv4 header including options are limited by the length of the *IHL* field. In contrast, the *Next Header* field of the IPv6 header, which corresponds to the *Protocol* field of the IPv4 header, indicates the type of the next inner header, not the next higher layer header, of the packet (see Figure 2). There is no limit on length of the IPv6 header.

IPv6 has six types of extension headers: Hop-by-Hop Options header, Destination Options header-1, Routing header, Fragment header, Authentication header, and Destination Options header-2. Except for the Hop-by-Hop Options header, extension headers are examined or processed only on the node specified by the *Destination Address Field*, while IPv4 options are examined by every node which forwards the IPv4 packet.

3 Mobility Support in IPv4

IETF (Internet Engineering Task Force) is the organization for standardization of protocols on the Internet. For mobility support in IPv4, several protocols including VIP were proposed to a

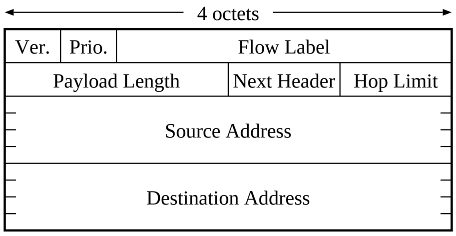

Figure 1: IPv6 header format

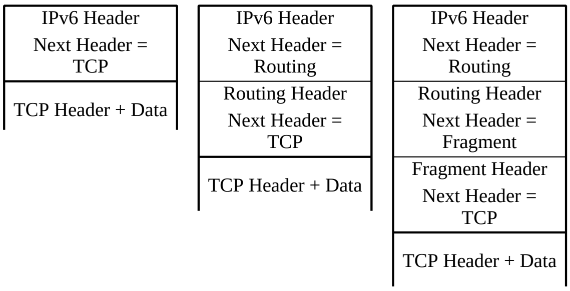

Figure 2: IPv6 options

working group of IETF. The working group did not choose one of them, but merged these protocols to create a draft protocol (*Mobile-IP*, hereafter). This section gives overviews of Mobile-IP and VIP.

3.1 Mobile-IP

The basic mechanisms of Mobile-IP are care-of-addresses and tunneling. Each mobile node has an immutable IP address as its home address. A mobile node is recognized with its home address regardless of location. A mobile node moves among subnets served by *foreign agents (FAs)*, special routers that act as gateways between the conventional internet and subnets for mobile nodes. When a mobile node moves to a subnet served by a FA, the mobile node is assigned the IP address of the FA as its care-of-address. Each mobile node has a *home agent (HA)*, a special node that maintains the current care-of-address of the mobile nodes it manages. The HA also advertises routing information of mobile nodes it manages.

When a node sends an IP packet to a mobile node, the IP packet reaches the HA of the mobile node due to the routing information advertised by the HA. The HA encapsulates the original IP packet within another IP packet. It assigns the care-of-address of the mobile node to the *Destination Address* field of the outer IP header, and then forwards the packet. This packet is delivered to the FA currently serving the mobile node. The FA decapsulates the received packet and forwards the original IP packet to the mobile node. The process between the HA and the FA is called tunneling because the original IP packet seems to enter a tunnel at the HA and go out of the tunnel at the FA.

3.2 VIP

The basic concept of VIP is separation between identifiers and addresses. VIP provides seamless node mobility by mapping between them. In the transport and higher layers, a node is recognized by an immutable identifier, not an address. The network layer resolves the identifier into an address and routes a packet in accordance with the address. In other words, an identifier is used for "what it is" while an address is used for "where it is". From network architecture viewpoint, this address resolution should exist in a sublayer in the network layer. VIP can be considered as a sublayer of IP. The VIP header is implemented as an IP option.

In VIP, the *VIP address* is introduced as an identifier in addition to the IP address. Since it was not intended to modify TCP/UDP and application programs, both the VIP address and the IP address must have the same format. Therefore, the VIP address can be used as the default IP address of the mobile node identified by the VIP address. The subnet specified by the VIP address of a mobile node is called the *home subnet* of the mobile node. The VIP address of the destination node is resolved into the corresponding IP address by the VIP sublayer, which is inserted between the TCP/UDP layer and the IP layer.

For efficient address resolution, each node has a cache called the *AMT (Address Mapping Table)*. When a packet is received or forwarded, an AMT entry for the source node is created or updated, and address resolution for the destination node may also be executed to the packet. A node that executes address resolution is called an *address resolver*. An address resolver in the home subnet of a mobile node is called a *primary address resolver* of the mobile node.

There are four routing cases when a node transmits a packet to a mobile node. (1) If the source node has the AMT entry for the mobile node, address resolution is executed on the source node, and then the packet travels the optimal route. If the source node does not have an AMT entry for the mobile node, the source node uses the VIP address of the mobile node as the default IP address. (2) An intermediate node which has the AMT entry for the mobile node executes address resolution, and then forwards the packet to the current location. (3) The packet reaches the home subnet of the mobile node, and then the primary resolver executes address resolution.

(4) The packet is forwarded to the previous subnet by an obsolete AMT entry for the mobile node. Since nodes in the previous subnet have the most recent AMT entry by receiving a VIP control packet when the mobile node moved, the packet is forwarded to the current location.

3.3 Discussion

Since Mobile-IP was intended to minimize changes in the current Internet, it has the following problems from the network architecture viewpoint: 1) violation of the subnet model: in Mobile-IP, a mobile node is identified by its home address. The IP routing mechanism cannot deliver IP packets in the subnet to which mobile nodes are connected because mobile nodes have unspecified different home addresses. 2) redundant routing: every packet destined to a mobile node reaches the HA of the mobile node first, and then it is forwarded to the current FA by using tunneling. Every packet travels this redundant route even if a mobile node and its peer node are connected to adjacent subnets. 3) single point of failure: if a HA crashes, communication with mobile nodes managed by the HA is disabled. 4) tunneling: an IP packet to a mobile node is encapsulated within another IP header on the HA of the mobile node. In a tunnel, the inner IP header is not processed on intermediate routers.

It might seem that Mobile-IP separates an identifier (the home address) and an address (the care-of-address). However, it does not. The care-of-address specifies the location of the FA, not the mobile node, in the Internet. In the subnet to which the mobile node is connected, the packet must be routed in accordance with the home address. A subnet may have mobile nodes which have several unspecified network numbers. As a result, it becomes impossible to route a packet in such a subnet by using the normal IP routing mechanism.

VIP is more elegant than Mobile-IP from the network architecture viewpoint. The cache mechanism of VIP allows a node to transmit a packet through the optimal route to the mobile node. Although the current location of a mobile node is managed by nodes in its home network in VIP, the cache mechanism also allows a node to communicate with a mobile node even if the home network crashes.

4 Proposed Protocol

The basic concepts and mechanisms of VIP can be applied to IPv6. The protocol derived from IPv6 by applying VIP is called *VIPonV6*. As mentioned in Section 3.2, address resolution functions should exist in a sublayer in the network layer. It is appropriate to incorporate mobility functions in IPv6 by using extension headers.

4.1 Header Formats

As shown in Section 2, there are six types of extension headers. Extension headers except for Hop-by-Hop Options header are not examined until the packet reaches the node specified by the *Destination Address* field. An advantage of the VIP mechanism is routing optimization by intermediate routers. To achieve this feature in VIPonV6, the Hop-by-Hop Options header is suitable for the header for VIPonV6.

There are two header types in VIP: data and control. In VIPonV6, two options are added in the Hop-by-Hop Options header. One option is for user data while another is for control[1]. Figure 3 depicts the formats of these headers. In both options, the *Option Type* field has the value $2x_{16}$. This value means that this option is excluded from integrity computation of the authentication header because the option data may change along the path. The *Option Data Length* field has value 64 in the option for user data, or 108 in the option for control.

The option for user data has the following fields:

- Source Identifier: a 128-bit number that uniquely identifies the source node regardless of location, i.e., regardless of the source address in the IPv6 header.

- Source Address Version: a 32-bit number that specifies the version of the source identifier/address pair. This number must be incremented monotonously when a new address is assigned.

- Holding Time: a 32-bit number that specifies the time in second a node should keep the AMT entry for the source node of this packet.

- Timestamp: a 32-bit number that specifies the time in second when the source node transmits this packet. This field is used to prevent replay attack.

- Authentication Data: the result of keyed MD5 calculation. A 128-bit number that authenticates the source node of this packet and guarantees the integrity of the option data.

[1] hereafter, an IPv6 packet with the mobility option for user data is called a *VIPonV6 data packet*, and an IPv6 packet with the mobility option for control is called a *VIPonV6 control packet*.

4 octets

Option Type	Opt Data Len
Source Address Version	
Holding Time	
Timestamp	
Source Identifier (16 octets)	
Authentication Data (16 octets) (keyed MD5)	
Destination Identifier (16 octets)	
Destination Address Version	

(a) Mobility option for user data

4 octets

Option Type	Opt Data Len
Address Version	
Holding Time	
Timestamp	
Identifier (16 octets)	
Address (16 octets)	
Authentication Data (64 octets) (RSA digital signature)	

(b) Mobility option for control

Figure 3: IPv6 mobility options

Status	reserved
Address Version	
Holding Time	
Timestamp	
Identifier (16 octets)	
Address (16 octets)	
Authentication Data (16 or 128 octets)	

Figure 4: AMT entry format

- Destination Identifier: a 128-bit number that uniquely identifies the final destination node regardless of location, i.e., regardless of the destination address in the IPv6 header.

- Destination Address Version: a 32-bit number that specifies the version of the destination identifier/address pair.

In the option for control, the *Identifier, Address, Version, Holding Time*, and *Timestamp* fields specify the information of a mobile node for which an AMT entry should be created or updated. For authentication, RSA digital signature is used in the option for control.

4.2 AMT Entry Format

Similar to VIP, each node should have an Address Mapping Table (AMT) for address resolution. When a VIPonV6 packet is received or forwarded, an AMT entry for the node specified by the *Source Identifier* field (in case of a VIPonV6 data packet) or the *Identifier* field (in case of a VIPonV6 control packet) is created or update. When a packet is transmitted or forwarded, the destination address is modified if an appropriate AMT entry for the destination node exists. Figure 4 shows the format of AMT entry in VIPonV6.

4.3 Authentication

The goal of authentication in VIPonV6 is to prevent impersonation by using the identifier of another node. To achieve this goal, in case of the VIPonV6 data packet, the identifier/address pair of the source node must be authenticated on a node that receives or forwards the packet before creating or updating the AMT entry. In case of the VIPonV6 control packet, the identifier/address pair of the target node must be authenticated.

Authentication mechanisms are based on cryptography. There are two kinds of cryptosystems: the secret-key cryptosystem (or the symmetric cryptosystem) and the public-key cryptosystem (or the asymmetric cryptosystem). In the former, both the sender and the receiver share the same secret key. In the latter, the encryption key and the decryption key are different. The encryption key is made public while the decryption key must be kept secret. DES[4] is categorized as a secret-key cryptosystem, while RSA[7] is categorized as a public-key cryptosystem. Other than cryptography, a kind of checksum algorithm with a secret key such as keyed MD5[6] can also be used for authentication.

One of the advantages of VIPonV6 is to allow AMT entries to propagate across the Internet for routing optimization. To create authenticated AMT entries on intermediate nodes, it is necessary for each intermediate node to share the session key of each source node if the secret-key cryptosystem is applied. However, this is impractical. Therefore, in VIPonV6, keyed MD5 is applied only to end-to-end authentication while the public-key cryptosystem (RSA digital signature) is used for authentication on intermediate nodes as shown in Figure 3. In Figure 3-(a), the hatched fields and the *Source Address* field of the IPv6 header are included in calculation of keyed MD5. In Figure 3-(b), the hatched fields are included in

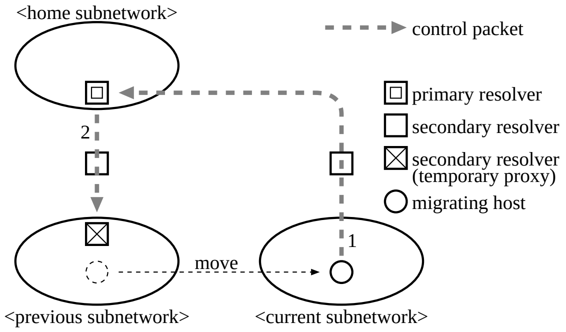

Figure 5: Packet flows upon node moving

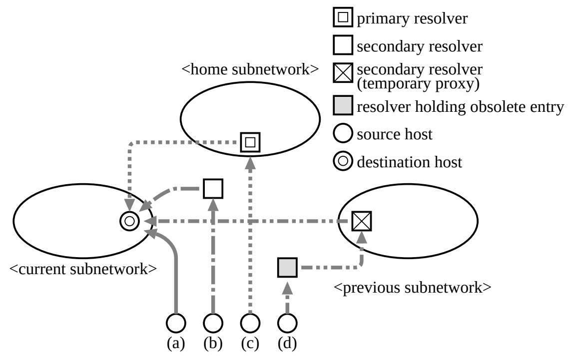

Figure 6: Packet flows in data communication

calculation of digital signature.

4.4 Connecting to a Subnet

A mobile node obtains a temporary address when it is connected to a subnet, and then it transmits a VIPonV6 control packet to its home subnet to register the current address with the home subnet. This packet includes RSA digital signature in the *Authentication Data* field. An intermediate node forwarding this packet creates or updates the AMT entry for the mobile node after authenticating the packet. The primary address resolver in the home subnet also forwards the packet to the subnet to which the mobile node was previously connected (see Figure 5). A mobile node periodically transmits a VIPonV6 control packet to its home subnet.

4.5 Data communication

Similar to VIP, there are four routing cases when a node transmits a packet to a mobile node (see Figure 6). (1) If the source node has the AMT entry for the mobile node, address resolution is executed on the source node, and then the packet travels the optimal route. If the source node does not have an AMT entry for the mobile node, the source node uses the identifier of the mobile node as the default address. (2) An intermediate node which has the AMT entry for the mobile node executes address resolution, and then forwards the packet to the current location. (3) The packet reaches the home subnet of the mobile node, and then the primary resolver executes address resolution. (4) The packet is forwarded to the previous subnet by an obsolete AMT entry for the mobile node. Since nodes in the previous subnet have the most recent AMT entry as described above, the packet is forwarded to the current location.

5 Discussion

5.1 Overhead

The basic IPv6 header is 40 octets long. The Hop-by-Hop options header including only the mobility option for user data is 72 octets[2] long. This increase of the header length will reduce the throughput of application layer data.

A mobile node periodically transmits a control packet to its home subnet. The control packet is 152 octets long. Assume that a mobile node transmits a control packet every t second. The control packet communication consumes $152 * 8/t = 1216/t$ bps bandwidth. If t is five minutes, the control packet communication consumes only 4 bps per mobile node.

Upon forwarding or receiving a VIPonV6 packet, the node may authenticate the source node of the packet for AMT update. In case of the VIPonV6 data packet, the forwarding node must calculate keyed MD5 if it has the secret key of the source node. In case of the VIPonV6 control packet, the forwarding node must decrypt the digital signature. This processing requires much time in comparison with communication time. If authentication checking is executed before packet forwarding on each router on the path, communication time will become much larger. This overhead can be avoided if authentication checking and AMT update are executed as a background job. For example, assume that a daemon process is running in the user space of the operating system for authentication checking and AMT update. Upon forwarding a VIPonV6 packet, the protocol handler immediately forwards the packet after triggering the daemon.

[2] although the length of the entire mobility option for user data is 68 octets, each extension header of IPv6 must be an integer multiple of 8 octets long.

5.2 Robustness of Authentication

Assume that a mobile node is connected to a subnet and communicates with other nodes. An attacker can obtain examples of a pair of plain data and its MD5 from VIPonV6 data packets. Therefore, the robustness of the VIPonV6 data packet authentication depends on the robustness of keyed MD5 against "known-plain-text attack".

The mobile node periodically transmits a VIPonV6 control packet to its home network. Since the mobile node inserts a random number in the plain data for the RSA digital signature, an attacker cannot find out the pair of plain data and its encrypted data. Therefore, the robustness of the VIPonV6 control packet authentication depends on the robustness of RSA against "cipher-text-only attack". The robustness of the cryptosystem itself is beyond the scope of this paper.

5.3 Key Distribution

VIPonV6 does not include a key distribution protocol. For the RSA digital signature, public keys must be obtained on demand. The name server is a candidate of the public key distribution server. For this purpose, it is necessary to add a new record type to the name server system, which indicates the public key of a node. In addition, communication with the name server must be secure. Since IETF is working on secure communication with the name server, its result will be employed for public key distribution by the name server. In keyed MD5, the secret key must be shared between peer nodes. For this purpose, the algorithms in [3, 8, 10] can be used.

6 Quick Resumption of TCP Communication

TCP cannot distinguish node disconnection from network congestion. A correspondent node of a mobile node cannot recognize disconnection/reconnection of the mobile node. If retransmission of TCP packet occurred while the mobile node was disconnected, the correspondent node does not resume transmission of data packets until its retransmission timer expires. This section proposes extending the TCP state transition diagram to include host mobility.

6.1 Extending the State Transition Diagram

Figure 7 depicts the TCP state transition diagram[5]. This diagram does not take node mobility into account. To allow TCP communication to resume quickly upon node reconnection, the *ESTAB* state of the diagram should be extended. In Figure 8, the *ESTAB* state is expanded to six states: *ESTAB, R-DISCONN, DISCONN, DISC-ACK WAIT, DISC-IND WAIT*, and *CONN-ACK WAIT*. State transition among these six states occurs due to node connection and disconnection. For the state transition among these six states, four new control messages should be added in TCP: *CONN, CONN-ACK, DISC, and DISC-ACK*.

There are many possible scenarios for node connection/disconnection. This paper chooses the following two cases and describes the state transitions according to Figure 8.

Case 1. Node X is suddenly disconnected from the network, and is then reconnected to the network when nodes X and Y are communicating with a TCP connection. This event will occur when node X is connected to an Ethernet and moves to another.

Case 2. Before disconnection from the network, node X prepares for it, and is then reconnected to the network when nodes X and Y are communicating with a TCP connection. This event will occur when node X enters the area overlapping between two radio cells, and it changes from the current cell to the new cell.

In case 1, the state of node X transfers from *ESTAB* to *DISCONN* when node X is disconnected from the network. The state of node Y remains unchanged (*ESTAB*) because there is no means for node X to notify node Y of its disconnection. When node X is reconnected to the network, it transmits a control packet *CONN* to node Y and its state transfers from *DISCONN* to *CONN-ACK WAIT*. When node Y receives this *CONN* packet, it transmits a control packet *CONN-ACK* to node X and re-initializes the parameters of the TCP connection to nodes X. When node X receives the *CONN-ACK* packet, the state of node X transfers from *CONN-ACK WAIT* to *ESTAB*, and node X re-initializes the parameters of the TCP connection to node Y.

In case 2, let us assume that the physical layer or the data link layer can detect the event which is node X entering the area overlapping between the two cells. When node X receives a signal for disconnection from the lower layer, it transmits a control packet *DISC* to node Y and the state of node X transfers from *ESTAB* to *DISC-ACK WAIT*. When node Y receives this *DISC* packet, it transmits a control packet *DISC-ACK* to node X and the state of node Y transfers from *ESTAB* to *R-DISCONN*. When node X receives the *DISC-ACK* packet, the state of node X transfers from *DISC-ACK WAIT* to *DISC-IND WAIT*. When node X disconnected from the net-

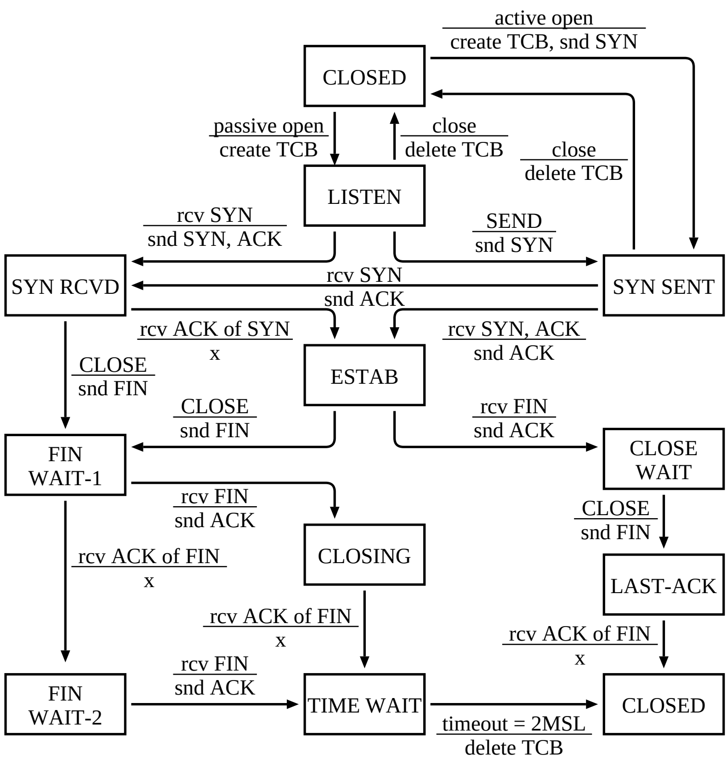

Figure 7: State transition diagram of a TCP connection[11]

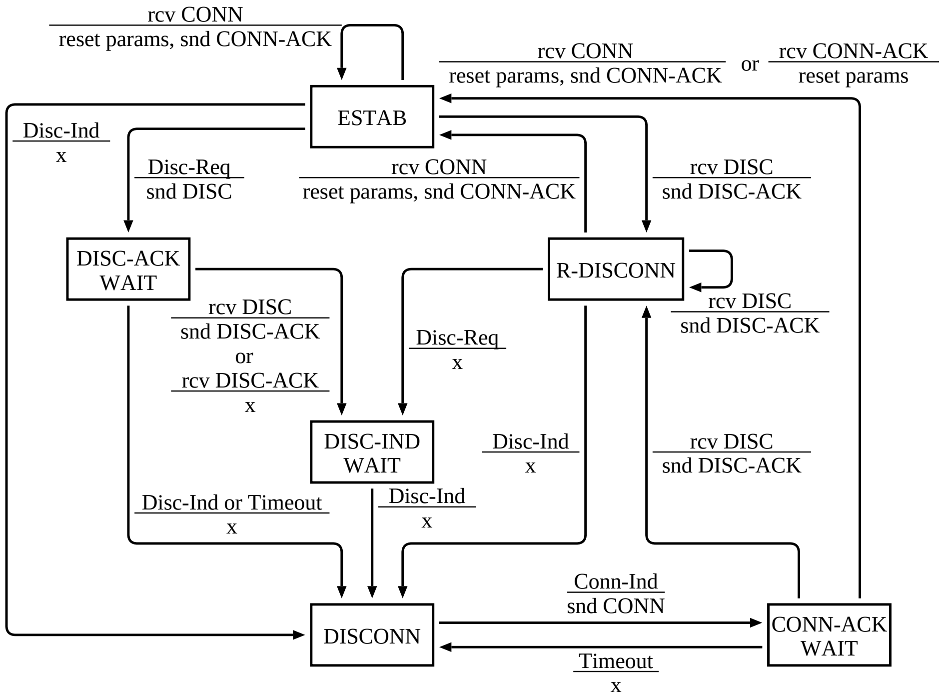

Figure 8: State transition diagram of an established transport connection

work, its state transfers from *DISC-IND WAIT* to *DISCONN*. When node X is reconnected to the network, it transmits a control packet *CONN* to node Y and its state transfers from *DISCONN* to *CONN-ACK WAIT*. When node Y receives the *CONN* packet, it transmits a control packet *CONN-ACK* to node X and its state transfers from *R-DISCONN* to *ESTAB*, and it then re-initialize the parameters of the TCP connection to node X. When node X receives the *CONN-ACK* packet, its state transfers from *CONN-ACK WAIT* to *ESTAB*, and it then re-initializes the parameters of the TCP connection to node Y.

In case 1, node Y continues to transmit packets while node X is disconnected. However, node Y can resume the communication as soon as node X is reconnected. In case 2, since node Y can detect the disconnection of node X, node Y can suspend the transmission while X is disconnected. In addition, node Y can resume the communication as soon as node X is reconnected.

6.2 Simplified Mechanism

Considering compatibility with existing TCP implementations, a simplified mechanism should be introduced in VIPonV6, instead of implementing the proposed mechanism in TCP. When a mobile node is reconnected to the network, it transmits a VIPonV6 control packet to the peer node of each active TCP connection and re-initializes its parameters. When a node receives a VIPonV6 control packet and finds that the peer node has just been reconnected to the network, the receiving node re-initializes the parameters of active TCP connections to the mobile node.

7 Conclusions

This paper proposes VIPonV6, a protocol providing node mobility in IPv6. VIPonV6 adds two options in the Hop-by-Hop extension header as the VIPonV6 headers. One is for user data and another is for control. Authentication mechanisms of VIPonV6 allow each node to have authenticated cache of mapping information. This cache mechanism achieves routing optimization. This paper also proposes extensions of TCP to resume communication immediately when a mobile node is reconnected. We are implementing IPv6 on an operating system based on BSD-UNIX. We plan to implement VIPonV6 on our IPv6 implementation.

Acknowledgment

The authors thank Prof. Jun Murai and the members of the WIDE Project for their valuable discussion on the design of VIPonV6 and its authentication mechanisms.

References

[1] R. Cáceres and L. Iftode. The Effects of Mobility on Reliable Transport Protocols. In *Proceedings of the 14th International Conference on Distributed Computing Systems*, pp. 12–20, June 1994.

[2] F. Kastenholz and C. Partridge. *Technical Criteria for Choosing IP:The Next Generation (IPng)*, December 1994. RFC 1726.

[3] R. M. Needham and M. D. Schroeder. Using Encryption for Authentication in Large Networks of Computers. *CACM*, Vol. 21, No. 12, pp. 993–999, December 1978.

[4] National Bureau of Standard. *Data Encryption Standard*, January 1977. Federal Information Processing Standards Publications, 46.

[5] J. Postel. *Transmission Control Protocol*, September 1981. RFC 793.

[6] R. Rivest. *The MD5 Message-Digest Algorithm*, April 1992. RFC 1321.

[7] R. Rivest, A. Shamir, and L. Adelman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystem. *CACM*, Vol. 21, No. 2,, February 1978.

[8] J. G. Steiner, C. Neuman, and J. I. Schiller. Kerberos: An Authentication Service for Open Network Systems. In *Proceedings of USENIX Winter Conference*, February 1988.

[9] F. Teraoka, K. Uehara, H. Sunahara, and J. Murai. Vip: A protocol providing host mobility. *CACM*, Vol. 37, No. 8, pp. 67–75, August 1994.

[10] S. Yamaguchi, K. Okayama, and H. Miyahara. Design and Implementation of an Authentication System in WIDE Internet Environment. In *Proceedings of the 5th IEEE Region 10 International Conference*, pp. 653–657, September 1990.

Author Information

Fumio Teraoka is a researcher of Sony Computer Science Laboratory Inc. Current research interests include computer networks, operating systems, and distributed processing.

Keisuke Uehara is a researcher of Keio University. Current research interests include computer networks, operating systems, distributed file systems and programming languages.