VIP: A Protocol Providing Host Mobility

Fumio Teraoka, Keisuke Uehara, Hideki Sunahara, and Jun Murai

As of March 3, 1994

Abstract

"Mobile Computing" has been a buzzword in the area of computer communications. A technology vital to mobile computing is how to communicate with mobile hosts. This paper proposes a protocol, Virtual Internet Protocol (VIP), that allows hosts to move transparently within the Internet. VIP is defined as an instance of the virtual network protocol, the general protocol supporting mobile hosts. The virtual network protocol is based on the concept of a virtual network and the propagating cache method. VIP is scalable to the total number of mobile hosts and the scale of the Internet, is free from routing loops, and tolerant of loss of control packets. VIP is currently running on operating systems based on 4.3BSD UNIX. The VIP overhead is negligible when compared with IP.

1 Introduction

Computer networks such as the Internet have spread worldwide. Computers are getting smaller and smaller. Such an environment allows users to carry portable computers and use them to access networks. "Mobile Computing" has become a buzzword in the area of computer communications. A vital issue in the field of mobile computing is how to communicate with hosts that move between subnetworks[1]. In conventional network architectures such as TCP/IP and the OSI seven-layer model, the network address of a host denotes its identification as well as its location. Since host migration results in a change of its network address, network addresses returned by name servers cannot identify migrating hosts. In addition, communication channels cannot be preserved after migration.

There are several candidate protocols for supporting host mobility in the Internet[4, 8, 10, 16]. In addition to the schemes for the Internet, a scheme is proposed for ISO CLNP (Connectionless-mode Network Protocol) networks[1], while another scheme uses learning bridges[9]. Packet radio networks also support host mobility only within a subnetwork[6]. However, these protocols all suffer various

[1]In this paper, a *network* refers to the overall network, a *subnetwork* refers to an individual component network, and an *internet* refers to a network on which Internet Protocol (IP) is running as a network layer protocol.

problems, e.g. [4] and [9] cannot scale to the size of the network; network partitioning might disable communications between two hosts on the same Ethernet when at least one of the hosts is mobile in [16]; the schemes proposed in [8, 10] cannot be applied to datagram communication in the transport layer. In addition, a fundamental problem of these schemes is the requirement for a network address to identify the location of mobile hosts.

This paper proposes a protocol that provides host mobility in the Internet, *VIP (Virtual Internet Protocol)*. In contrast to other schemes, VIP introduces logical identifiers for mobile hosts. VIP was first proposed in 1991[12, 14] and improved in 1993[15]. VIP is defined as an instance of the *Virtual Network Protocol*, a network layer protocol that provides the transport layer with host migration transparency and which is applicable to any connectionless-mode network protocol such as IP[7], CLNP[5], and Xerox Internet Datagram[17]. VIP has the following features:

- VIP is scalable to the scale of the network and the total number of mobile hosts,
- VIP is free from routing loops with all packets, except for the first packet, traversing the optimum route, and
- VIP is also tolerant of the loss of control packets.

These features will be discussed in Section 6.

The structure of this paper is as follows. Section 2 defines two key services in the network layer to provide host mobility from the protocol layering viewpoint. In Section 3, the concept of a *virtual network* and the *propagating cache method*[14] are introduced as means of achieving these two key services. Section 4 proposes the *Virtual Network Protocol* based on the virtual network concept and the propagating cache method. Section 5 derives VIP by applying the Virtual Network Protocol to IP. Section 6 discusses the features of VIP. Currently, VIP is running on operating systems based on 4.3BSD UNIX. The implementation of VIP and its performance are given in Section 7. The measured performance proves that the overhead of VIP is negligible in comparison with that of IP. Section 8 concludes this paper.

2 The Key Services for Host Migration Transparency

2.1 Types of Host Migration

Host migration can be classified into two types: *off-line* and *on-line*. In off-line migration, the host is disconnected from the network before moving. On-line migration, in contrast, allows a host to access the network even during actual movement. In conventional wired networks, only off-line migration is possible due to physical restrictions. However, wireless networks such as digital cellular telephone systems allow on-line migration. In this paper, hosts that can migrate on-line are called *mobile hosts*, while those that can only migrate off-line are called *portable hosts.* The general term *migrating host* is used to denote these two host types.

2.2 The Layer To Provide Host Migration Transparency

From a user's viewpoint, it can be possible to achieve host migration transparency in the application layer, the transport layer, or the network layer. In case of the application layer approach, name server entries for migrating hosts are updated every movement. A host that wants to communicate with a migrating host knows the destination address by accessing the name server. In this approach, however, on-line migration cannot be achieved because communication channels are broken after movement.

In case of the transport layer approach, the transport layer re-establishes communication channels every movement with the assistance of the application layer approach. In this approach, if a host detects its movement, it notifies corresponding hosts of its new network address, and then the communication channels between these hosts are re-established. In general, communication channel management is complicated. This approach makes it more complicated. Besides that, the overhead of this approach increases in proportion to the number of active communication channels. In addition, this approach cannot provide host migration transparency in connectionless communication.

In contrast with these approaches, adding only two new services to the network layer allows host migration transparency as described below. From a protocol layering viewpoint, it is suitable that the network layer provides host migration transparency because it is responsible to host-to-host communication. In other words, the transport and higher layers do not perceive the notion of host and it is up to the network layer to maintain host location.

2.3 Key Services in the Network Layer

End-to-end communication services are provided by the transport and lower layers. To offer host migration transparency in end-to-end communication services, the following four services in the transport layer must be migration-transparent.

- (T-1) transport entity addressing. The transport layer must provide the session layer with a migration-independent identifier for the transport entity.

- (T-2) connectionless-mode communication. The transport layer must provide transport level connectionless-mode communication (datagram service), regardless of migration.

- (T-3) connection establishment. The transport layer must be able to establish transport level connections, regardless of migration.

- (T-4) connection-oriented communication. The transport layer must be able to preserve transport level communication during actual movement, offering transport level connection-oriented communication.

Off-line migration transparency, which must be provided to both portable and mobile hosts, can be supported by services *T-1* through *T-3*. On-line migration transparency, required by mobile hosts, requires all four services.

To support these services in the transport layer, the network layer must have the following six services.

- (N-1) Migration-transparent host addressing: provides the transport layer with a migration-independent identifier for the host.

- (N-2) Migration-transparent connectionless-mode communication: offers network level connectionless-mode communication regardless of migration.

- (N-3) Disconnect request: prepares for a host to leave a subnetwork when it is to be disconnected.

- (N-4) Disconnect indication: called by the data link layer to prepare to leave the current subnetwork and to enter another in on-line migration.

- (N-5) Connect request: prepares for joining to a subnetwork when a host is to be connected.

- (N-6) Connect indication: called by the data link layer to prepare for joining a host to a new subnetwork in on-line migration.

The services required in the transport layer are supported by the services in the network layer as follows: *T-1* is supported by *N-1*. *T-2* and *T-3* are supported by *N-1*, *N-2*, *N-3*, and *N-5*. *T-4* is supported by *N-2*, *N-4*, and *N-6*.

In the network layer, *N-3* through *N-6* are supported by the data link layer. Therefore, *N-1* and *N-2*, i.e. *migration-transparent host addressing* and *migration-transparent connectionless-mode communication*, are peculiar to the network layer and can be thought of as being key services in achieving host migration transparency. The relationships between these services are depicted in Figure 1[11].

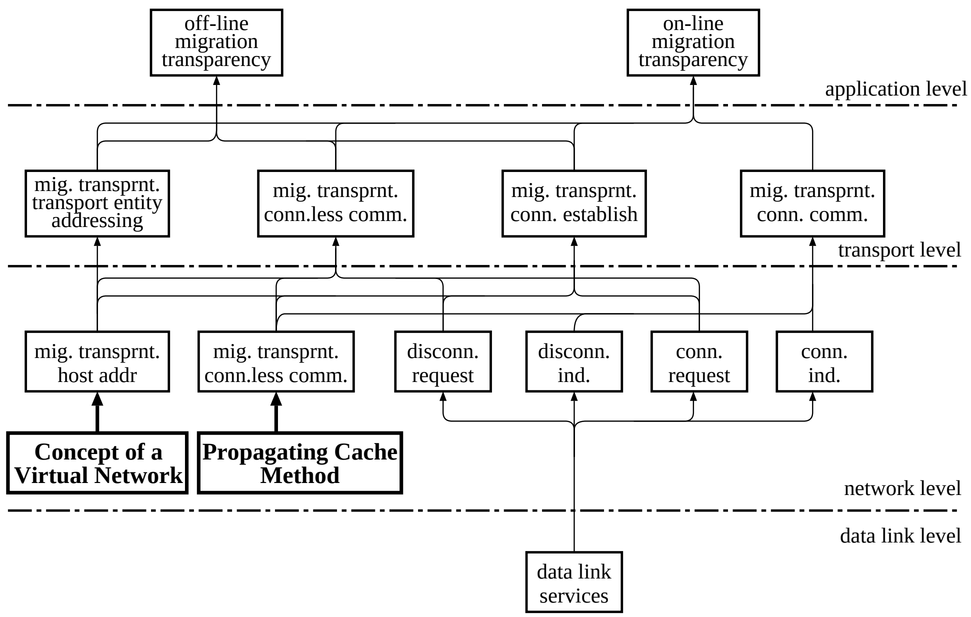

Figure 1: Relationships between services

3 The Virtual Network

The previous section defined two network layer services that are key to host mobility: 1) migration-transparent host addressing and 2) migration-transparent connectionless-mode communication. In order to achieve the former, the concept of a *virtual network* is introduced. To achieve the latter, the *propagating cache method* is proposed.

3.1 The Concept

The virtual network is a logical network that exists on top of the actual, or *physical*, network. Only the virtual network is visible to the transport layer. Each host is regarded as always being connected to the same virtual subnetwork, called the *home subnetwork* of the host even if it migrates to another physical subnetwork. A host has both a *virtual network address* (*VN-address*) and a *physical network address* (*PN-address*). Since a host never migrates in the virtual network, its VN-address never changes. The PN-address of a host indicates the location of the host in the physical network and is used for packet routing. The transport layer specifies the target host by its VN-address, no matter where it migrates to in the physical network. Basically, the PN-address is invisible to the transport layer. The VN-address and PN-address have the same format. For a host, the values of both addresses are the same when the host is connected to its home subnetwork.

The relationship between the VN-address and the PN-address of a host is analogous to that between the virtual and physical addresses in a virtual memory system. In a virtual memory system, the virtual address used in a program never changes, no matter where the program is located in physical memory. Similarly, a host can be indicated by its VN-address no matter where the host migrates in the physical network. From another viewpoint, the virtual network concept introduces *host identifiers* in addition to host addresses. The host identifier specifies “what it is” whereas the host address specifies “where it is”.

3.2 Layering

Since the transport layer uses the VN-address in the packet transmission request to the network layer, to deliver packets, the network layer must resolve the VN-address of a host into the corresponding PN-address. The network layer is responsible for host-to-host communication, including packet relaying, routing control, and so on. Since it is not appropriate to classify this address resolution into the traditional categories of basic network layer functions, the network layer should be divided into two sublayers, as follows (see Figure 2):

- The *virtual network sublayer* (*VN-sublayer*) conceals the physical location of hosts in the physical subnetworks. A host is assigned a VN-address, which the VN-sublayer resolves into the corresponding PN-address.

- The *physical network sublayer* (*PN-sublayer*) is the conventional network layer. According to the OSI layering model, this sublayer consists of the subnetwork independent sublayer, the subnetwork dependent sublayer, and the subnetwork access sublayer. The PN-sublayer is responsible for packet relaying, routing information exchange, and so on. A host is assigned a PN-address in the PN-sublayer.

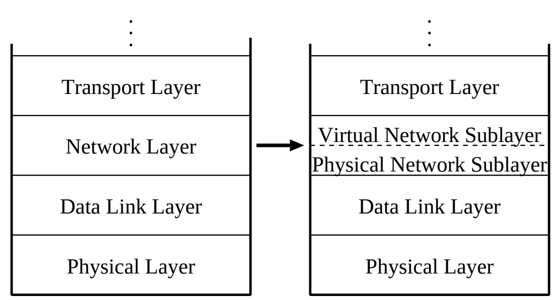

Figure 2: Protocol layering for a virtual network

This protocol layering affects the packet header format. According to this division, the VN-sublayer header is added between the (physical) network layer and transport layer headers. If the (physical) network layer header of an incoming packet specifies the VN-sublayer as the next higher layer, the packet is passed to the VN-sublayer. Otherwise, the packet bypasses the VN-sublayer and is passed to the transport layer. Thus, the VN-sublayer can be bypassed if an incoming packet does not have the VN-sublayer header, or if the receiving host does not support the VN-sublayer and silently ignores the VN-sublayer header.

3.3 The Propagating Cache Method

The basic idea of the propagating cache method takes advantage of a cache mechanism for address resolution, i.e. each router or host holds a cache for address resolution. Address resolution is executed either on the source host or an intermediate router. If the source host does not know the PN-address of the destination host, it assumes that the destination host is physically connected to its home subnetwork, i.e. the sender assumes the PN-address of the destination host to be equal to its VN-address. The transmitted packet then travels to the home subnetwork of the destination host. However, if a router on the path holds a cache entry for the destination host, it resolves the VN-address of the destination host into the PN-address. Then, the packet is forwarded to the actual physical location of the destination host without traveling the entire route to the home subnetwork. A cache is created or updated using the packets received. Cache information propagates across the wide area network along the communication paths between migrating hosts, such that routing to a migrating host becomes more efficient as communication between migrating hosts occur.

A host is called a *family host* on its home subnetwork. The relationship between the VN-address and the PN-address of a family host is always maintained by the hosts or routers in its home subnetwork. In practice, having routers in the home subnetwork maintain the relationship for the family hosts is sufficient for address resolution.

4 The Virtual Network Protocol

The *Virtual Network Protocol (VN-protocol)* is a general protocol that provides host mobility on the top of connectionless-mode network layer protocols. The VN-protocol exploits the virtual network concept and the propagating cache method to achieve the two key services defined in Section 2.3. The cache in the propagating cache method is called the *Address Mapping Table* (*AMT*). The AMT consists of entries, each of which corresponds to a migrating host. The VN-protocol has two control packet types: *CAMT* (create an AMT entry) and *InvAMT* (invalidate an AMT entry). A data packet as well as the *CAMT* packet creates or updates an AMT entry when relayed or received. A router that executes address resolution is called an *address resolver.* Address resolution for a packet destined to a migrating host is executed by either the source host or an address resolver.

4.1 Address Resolvers

There are three types of address resolvers for each migrating host: *primary address resolver*, *secondary address resolver*, and *temporary proxy.* A primary address resolver is an address resolver in the home subnetwork of a migrating host. The primary address resolvers for a migrating host always hold the most recent AMT entry for the migrating host. A secondary address resolver is an address resolver in the subnetworks other than the home subnetwork of the migrating host. Obsolete AMT entries might remain on secondary address resolvers. A temporary proxy is a kind of secondary address resolvers in the subnetwork to which the migrating host was previously connected. The temporary proxy was introduced for re-mapping for packets that suffered incorrect mapping for the destination host by an obsolete AMT entry.

4.2 Connecting to a Subnetwork

Figure 3 depicts the packet flow when a migrating host connects to a subnetwork. (1) The migrating host transmits a *CAMT* packet to its home subnetwork. Upon relaying this *CAMT* packet, intermediate routers create or update the AMT entry for the migrating host, i.e. such intermediate routers become *secondary address resolvers* for the migrating host. While connected to a subnetwork other than the home subnetwork, the migrating host continues to periodically transmit a *CAMT* packet. (2) If an intermediate router holds an obsolete AMT entry for the migrating host, it broadcasts an *InvAMT* packet to all the connected subnetworks to invalidate any obsolete entries that might remain on other routers. When a router receives such an *InvAMT* packet, if it holds an obsolete AMT entry, it invalidates that entry and also broadcasts the *InvAMT* packet. Note that the *InvAMT* packet is not broadcast throughout since a host or router does not forward an *InvAMT* packet if it does not have the corresponding entry or holds a newer entry. (3) When a primary address resolver of the migrating host receives this *CAMT* packet, it also creates or updates the AMT entry for the migrating host. If it holds an obsolete AMT entry for the migrating host, it transmits another *CAMT* packet to the subnetwork to which the migrating host was previously connected. As a result, the previous subnetwork holds the most recent AMT entry for the migrating host until it times out, i.e. the previous subnetwork becomes a *temporary proxy* for the migrating host. (4) On the path of the *CAMT* packet transmitted by a primary address resolver, the same processing as (2) is executed.

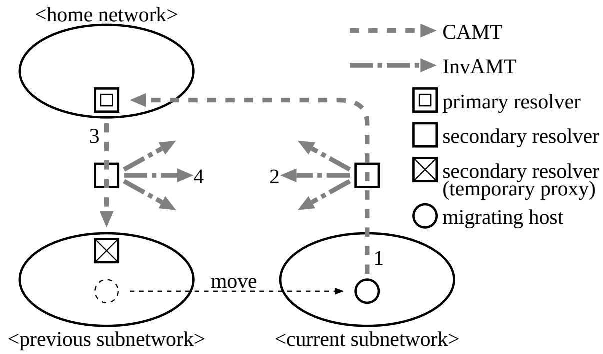

Figure 3: Host connection

4.3 Data Communication

Figure 4 depicts the flow of packets in data communication. There are four cases: (a)-(d). (a) If the source host has the AMT entry for the destination host, it executes address resolution. The packet will traverse the optimum route. (b) If the source host does not have the AMT entry for the destination host, it assumes that the PN-address of the destination host is equal to its VN-address. The packet heads for the home subnetwork of the destination host. If an intermediate router holds an AMT entry for the destination host, it executes address resolution and forwards the packet. (c) A primary address resolver ultimately receives the packet heading for the home subnetwork of the destination host. It executes address resolution and forwards the packet. (d) A router that holds an obsolete AMT entry for the destination host executes address resolution for the packet heading for the home subnetwork of the destination host. The packet is forwarded to the previous subnetwork of the destination host. A temporary proxy of the destination host in the previous subnetwork executes address resolution and forwards the packet to the correct location. Note that once the source host receives a packet from the destination host, it creates the AMT entry for the destination host.

If a host does not have the functions for the VN-protocol, it transmits conventional network layer packets. When a router relays such a packet, if it has the AMT entry for the destination host of the packet, it adds the VN-sublayer header to the packet and forwards it. In the worst case, such a packet reaches the primary address resolver of the destination host and is converted into a VN-protocol packet. When such a host receives a VN-protocol packet, if it silently ignores the header of the VN-sublayer, it correctly recognizes the VN-address of the source host. Section 6.6 will discuss such a case in the IP environment.

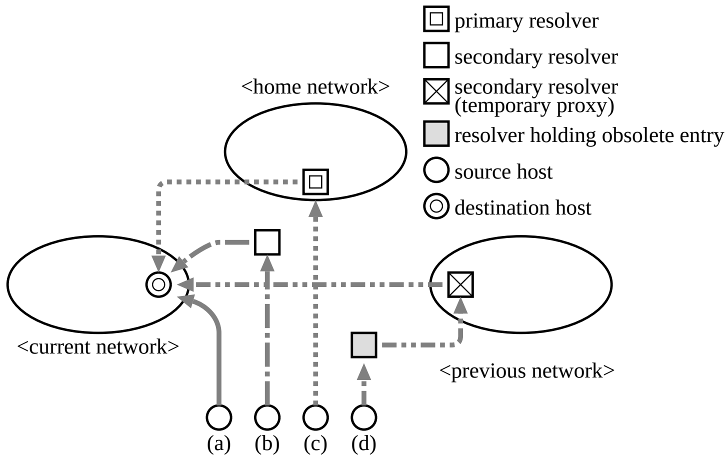

Figure 4: Data communication

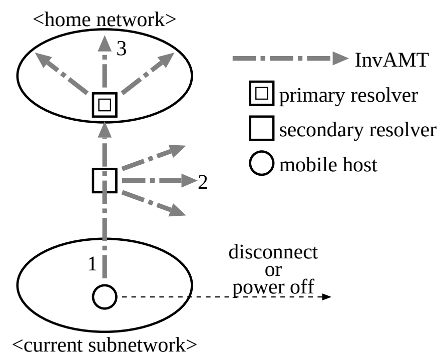

Figure 5: Host disconnection or power off

4.4 Disconnecting from a Subnetwork

Figure 5 depicts the packet flow when a migrating host disconnects from a subnetwork or is powered off. (1) The migrating host may transmit an *InvAMT* packet to its home subnetwork. (2) On the path of the *InvAMT* packet, if an intermediate router has an obsolete AMT entry for the migrating host, the router broadcasts another *InvAMT* packet to all connected subnetworks. When a router receives such an *InvAMT* packet, if it holds an obsolete AMT entry, it invalidates that entry and also broadcasts the *InvAMT* packet. (3) When a primary address resolver of the migrating host receives the *InvAMT* packet, if it has an obsolete AMT entry for the host, it broadcasts another *InvAMT* packet to all connected subnetworks.

4.5 The VN-Protocol Header

As mentioned in Section 3.2, the network layer header includes both the VN-sublayer header and the PN-sublayer header. The PN-sublayer header is equal to the conventional network layer header. The VN-sublayer header must have the following 13 fields: *Flags, Source VN-address/PN-address/Address-Version, Destination VN-address/PN-address/Address-Version, Holding Time, Resolver PN-address, Target VN-address/PN-address/Address-Version*, and *Target Holding Time*. The *Flags* field specifies the packet type and other information. There are three packet types: *data*, *CAMT*, and *InvAMT*. The *Holding Time* field in the data packet specifies the timeout value of the AMT entry created or updated by this packet. The *Resolver PN-address* field in the data packet specifies the PN-address of the address resolver that executed address resolution for this packet. The *Target VN-address/PN-address/Address-Version* fields in the *CAMT/InvAMT* packet indicate those values of the target machine of the AMT entry to be updated. On the other hand, in case of the data packet, the *Source VN-address/PN-address/Address-Version* fields are used for AMT update.

4.6 AMT Entry

The AMT must hold the information that enables 1) resolution of the VN-address of the destination host into the corresponding PN-address, 2) keeping the entry consistent with the most recent information, and 3) deletion of itself when it has not been refreshed for a certain period. Thus, an AMT entry must have the following five fields: *Virtual Network Address*, *Physical Network Address*, *Address Version Number*, *Status*, and *Holding Time*.

An AMT entry takes one of three states: *free, valid*, and *invalid*. In the *free* state, the entry does not hold any information. When relaying or receiving a packet, if the host does not have an AMT entry for the source host of the packet or it has an older entry for the source host, it creates or updates a *valid* entry. In the *valid* state, the entry holds a valid mapping between a VN-address and a PN-address. A *valid* entry overwrites the destination PN-address field when relaying a packet which holds an older mapping for the destination host. When the entry times out, it returns to the *free* state. A *valid* entry changes its state to *invalid* when receiving an *InvAMT* packet which has an equal or newer timestamp than the entry. A *valid* or *invalid* entry detects an obsolete mapping for the destination host which might be caused by an obsolete AMT entry on another machine. Invalid entries are used not for address resolution but for detecting packets that have an obsolete mapping between the destination VN/PN-addresses. When a host or router detects an obsolete mapping in a packet, it transmits an *InvAMT* packet to the address resolver indicated by the *Resolver PN-address* field of the packet. The reason why the *invalid* state was introduced is to increase the probability of

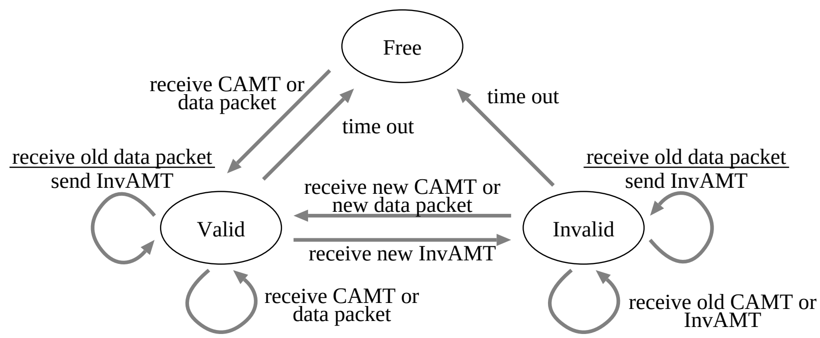

Figure 6: State transition diagram

detecting obsolete mappings. An *invalid* entry also times out, and then changes its state to *free*. Such deletion of the AMT entry (*valid* → *invalid* → *free*) is called the *two-phased deletion*. Figure 6 depicts the state transition diagram of an AMT entry.

5 VIP: Virtual Internet Protocol

VIP (Virtual Internet Protocol) is derived from IP by applying the VN-protocol. *VIP addresses* are introduced as VN-addresses in the VN-protocol, while IP addresses are thought of as PN-addresses. Table 1 shows the correspondence of the VN-protocol to VIP.

Table 1: Correspondence of the VN-protocol to VIP

VN-protocol	**VIP**
VN-addresses	VIP addresses
PN-addresses	IP addresses
home subnetwork	the subnetwork indicated by the network number of the VIP address
primary address resolvers	routers connected to the home subnetwork of a migrating host (home routers)
secondary address resolvers	routers connected to the subnetworks other than the home subnetwork of a migrating host
temporary proxies	routers connected to the previous subnetwork of a migrating host

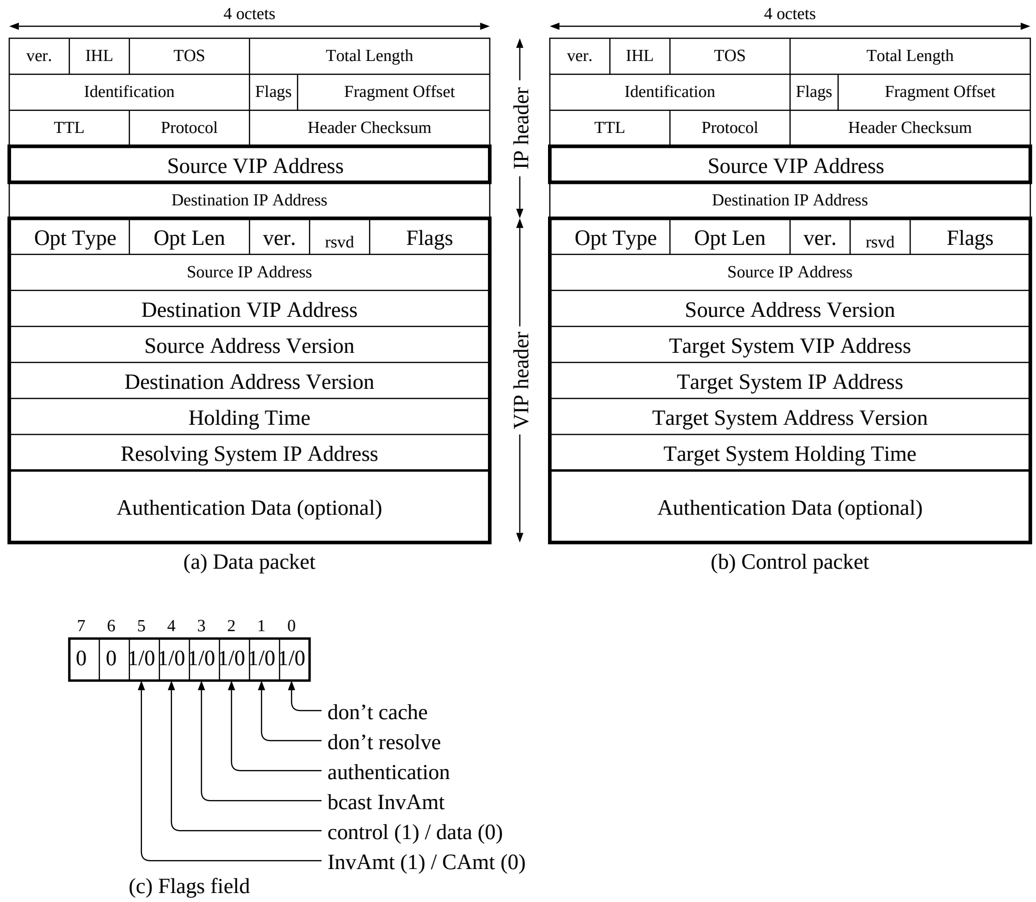

Figure 7: VIP header formats

5.1 VIP Header and AMT Entry

Figure 7 depicts the VIP header formats. Figure 8 shows the format of an AMT entry for VIP. Since the length of the VIP header becomes too great if it includes all mandatory fields for the VN-sublayer header, the control packet is separated from the data packet. To ensure backward compatibility with IP, the VIP header is implemented as an IP option, instead of the higher sublayer header of IP[13]. In Figure 7-(a) and (b), the top 20 octets construct the IP header, while the remainder is the VIP header as an IP option. In accordance with the IP option format, the VIP header includes the *Option Type* and *Length* fields. Note that the *Source VIP Address* field is included in the IP header and the *Source IP Address* field is included in the VIP header to ensure backward compatibility with IP (see Section 6.6).

To control address resolution and the manipulation of AMT entries, VIP defines four more flags in the *Flags* field of the VIP header in addition to the flags indicating the packet type (see Figure 7-(c)). If the *don't cache* flag is set, routers other than the home routers of the source host do not create the

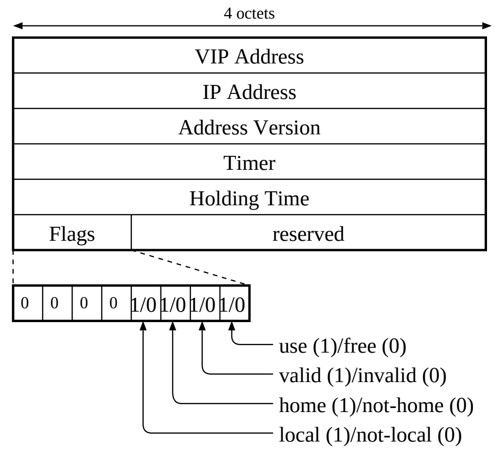

Figure 8: AMT entry for VIP

AMT entry for the source host. If the *don't resolve* flag is set, routers other than the home routers of the destination host do not execute address resolution for the destination host. If the *authentication* flag is set, the packet includes authentication data. If the *broadcast InvAMT* flag is not set, routers do not broadcast an *InvAMT* packet upon detecting obsolete AMT entries.

5.2 Examples

Figure 9 shows a simple Internet consisting of both VIP and IP machines. In the figure, a VIP host *Host:A*-a moves from its home subnetwork *Net:A-b* to *Net:B*-a (1). The VIP address of *Host:A*-a remains unchanged while a temporary IP address is assigned to the host by, say, DHCP (Dynamic Host Configuration Protocol)[2]. *Host:A*-a transmits a *CAMT* packet to its home subnetwork *Net:A-b*, so that intermediate VIP routers (*GW:B*, *GW:BB-ab*, and *GW:A*) and its home routers (*GW:A-ab* and *GW:A-bc*) create an AMT entry for *Host:A*-a. Consider packet transmission from a VIP host *Host:C-a* to *Host:A*-a. Since *Host:C-a* does not have the AMT entry for *Host:A*-a, it assumes that the IP and VIP addresses of *Host:A*-a are the same. Therefore, the VIP packet heads for *Net:A-b*. Upon relaying the packet on *GW:BB-ab*, since it has the AMT entry for *Host:A*-a, it executes address resolution and forwards the VIP packet to *Net:B*-a. An IP host *Host:C-b* transmits an IP packet to *Host:A*-a with the VIP address of *Host:A*-a as the destination IP address. This IP packet is converted into a VIP packet by *GW:BB-ab* and forwarded to *Net:B*-a. Note that once *Host:C*-a receives a VIP packet from *Host:A*-a, it creates the AMT entry for *Host:A*-a, such that it executes address resolution at the transmission time. However, packets transmitted by *Host:C-b* always travel the redundant route.

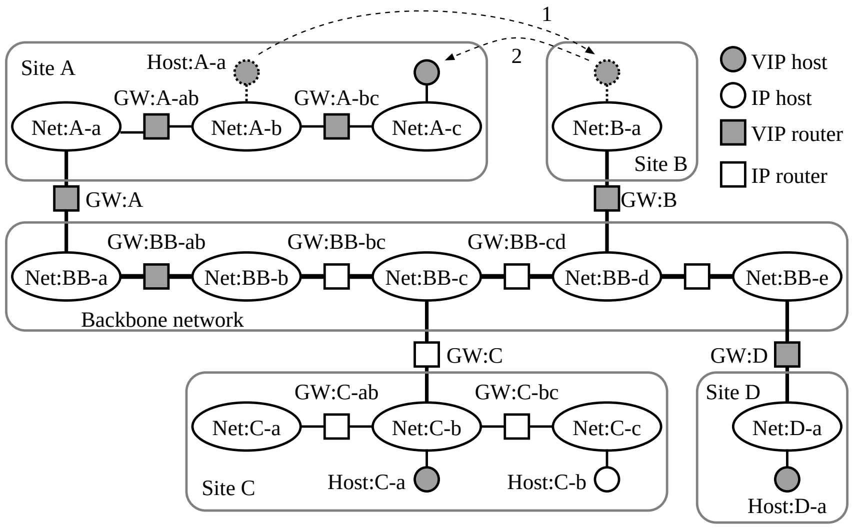

Figure 9: A sample network

Suppose that *Host:A-a* moves again from *Net:B-a* to *Net:A-c* (2) and that its home routers and *GW:D* hold an AMT entry for *Host:A-a*, which indicates that the current subnetwork of *Host:A-a* is *Net:B-a*. When *Host:A-a* transmits a *CAMT* packet to its home subnetwork, *GW:A-bc* also transmits another *CAMT* packet to *Net:B-a* because *GW:A-bc* held an obsolete AMT entry. As a result, *GW:B* updates the AMT entry for *Host:A-a*, i.e. it becomes the temporary proxy of *Host:A-a*. However, an obsolete AMT entry for *Host:A-a* remains on *GW:D*. Consider the packet transmission from a VIP host *Host:D-a* to *Host:A-a*. The VIP packet transmitted by *Host:D-a* is forwarded to *Net:B-a* because *GW-D* has an obsolete AMT entry for *Host:A-a*. Since *GW:B* has the most recent AMT entry for *Host:A-a*, it executes correct address resolution and forwards the packet to *Net:A-c*. *GW:B* also transmits an *InvAMT* packet to *GW:D* to invalidate the obsolete AMT entry.

6 Discussion of VIP Features

6.1 Scalability on Memory Usage

Although VIP makes use of a cache called the AMT for address resolution, the number of AMT entries a host or router must maintain is independent of the scale of the internet or the total number of migrating hosts. It is sufficient for a host to have AMT entries for other hosts with which the host communicates. If a host communicates with up to 100 hosts simultaneously, it is enough for the host to have 100 AMT entries, which occupy only 2,800 octets[2].

The number of AMT entries a router must maintain is equal to the number of family hosts for the router, because the correct information for a migrating host is always maintained by its home router(s). A router may also have AMT entries for intermediate address resolution. AMT entries for intermediate address resolution may be lost without harm, because the home router of a migrating host always holds its AMT entry. If a router is connected to two Ethernet segments, each of which has up to 500 hosts, the router must have 1,000 AMT entries for its family hosts, which occupy 28,000 octets.

6.2 AMT Entry Management

The integrity of an AMT entry is maintained by a version number mechanism, i.e. the *Address Version* field of the AMT entry is compared with the *Destination Address Version* field of the VIP header for address resolution, or with the *Source Address Version* field for update of an AMT entry. Since the two version numbers compared upon address resolution and AMT entry update are created by the same host, the comparison of version numbers always works correctly provided each host ensures that its address version number increases monotonically.

6.3 Routing Loops

VIP is a protocol which resolves a VIP address into a corresponding IP address. Packet routing is calculated in the IP layer. VIP never affects the routing calculation in the IP layer. Since the comparison of version numbers works correctly as mentioned above, the destination IP address of the packet is updated monotonically as the packet is relayed by routers. Therefore, the packet reaches a host regardless of whether the receiving host is the correct destination host. Consequently, as long as the routing control in the IP layer is free from routing loops, VIP will never exhibit routing loops.

6.4 Loss of Control Packets

There are two control packet types for VIP: *CAMT* and *InvAMT*. VIP is tolerant of losing control packets. Since a migrating host continues to transmit *CAMT* packets to its home subnetwork periodically, the loss of a single *CAMT* packet is not fatal. An *InvAMT* packet is transmitted whenever an obsolete AMT entry is found. It may also be transmitted when a host is about to disconnect from a subnetwork or the host is powered off. If an *InvAMT* packet is lost, obsolete AMT entries will time

[2] According to Figure 8, the size of an AMT entry is 24 octets. In our implementation, one more four-octet field is added for AMT entry management.

out. Even if an obsolete AMT entry delivers a VIP packet to an incorrect host, the receiving host can detect the misdelivery because the destination VIP address of the packet does not match the VIP address of the host.

6.5 Network Partitioning

There is the possibility that network partitioning disables communication with a migrating host even if an available physical link exists. Suppose that the network is partitioned between a host which is about to send a packet to a migrating host and the home subnetwork of the migrating host. If neither the sending host nor intermediate routers have the AMT entry for the migrating host, the transmitted packet will be lost. However, if the migrating host sends packets, AMT entries for the migrating host propagate along the communication path. Thus, network partitioning is not fatal to VIP but VIP is not tolerant of all kind of network partitioning.

6.6 Compatibility with IP

Suppose that a VIP subnetwork is connected to a traditional IP network and that a VIP host, whose home subnetwork is the VIP subnetwork, migrates to an IP subnetwork. The VIP host can communicate with any IP host with its VIP address. When the VIP host transmits a VIP packet to an IP host, the values in the *Destination IP* and *VIP* fields of the VIP packet will be the same. The VIP packet travels the optimal route to the destination IP host. IP routers and IP hosts ignore the VIP header as an unknown IP option[3]. IP routers correctly relay the VIP packet and IP hosts correctly handle the VIP packet. In addition, since the *Source IP Address* field of the VIP packet holds the VIP address of the source VIP host instead of its temporary IP address, the destination IP host recognizes the correct source address of the VIP packet.

When the IP host transmits an IP packet to the VIP host, the *Destination IP Address* field of the IP packet is assigned the VIP address of the VIP host. This packet heads for the home subnetwork of the VIP host. If an intermediate router has an AMT entry of the VIP host, it converts the IP packet into a VIP packet. In the worst case, the IP packet reaches the home subnetwork of the VIP host, then is converted into a VIP packet. Thus, packets from an IP host to a VIP host might travel along redundant routes.

7 Implementation and Performance

7.1 Implementation

In general, there are two ways of incorporating a new protocol into existing systems: 1) to write programs from scratch and 2) to modify existing programs. For the sake of ease of implementation and future software distribution, the latter was chosen to implement VIP. VIP was implemented by modifying 4.3BSD UNIX based operating systems (NEWS-OS on SONY NEWS workstations and BSD/386 on PCs with Intel 386/486). The VIP functions were added to the IP layer in the kernel. The added functionality requires approximately 900 lines of C code. Figure 10 shows a block diagram of the code. In the figure, thin boxes indicate IP modules while thick boxes indicate VIP modules.

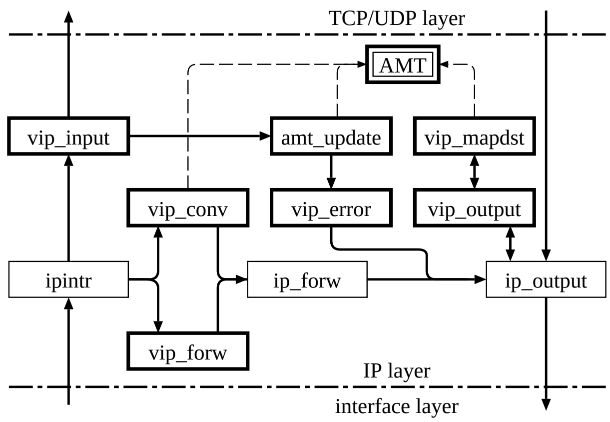

Figure 10: Block diagram

7.2 Performance

The VIP processing time was measured by using a special timer board with a 1-μsec granularity. Our test environment consists of seven SONY NEWS workstations and four Ethernets. Each workstation has a MC68030/25MHz CPU and 4-8 Mbytes of main memory.

Figure 11 shows the processing time required for the transmission, relay, and reception of IP and VIP in the stable state. “Stable state” means that the source host has the AMT entry for the destination host, intermediate routers have the AMT entry for the source host and for the destination host, and the destination host has the AMT entry for the source host. Therefore, the IP address for the destination host is resolved at the source host. Intermediate routers and the destination hosts update only the *Holding Time* field of the entry for the source host. Intermediate routers do not modify any header field related to the destination host, although they search the AMT for the entry

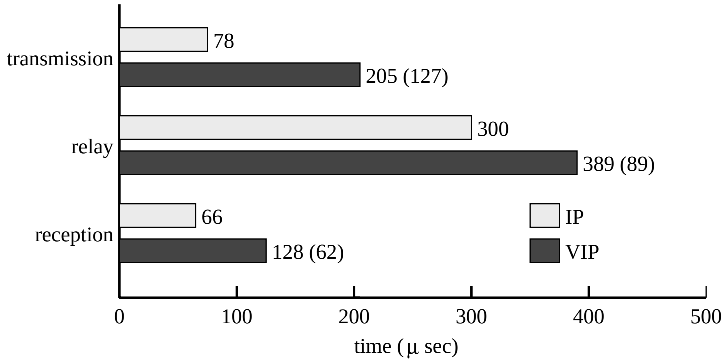

Figure 11: Comparison of three protocols

for the destination host. The numbers in parentheses represent the overhead relative to that of IP. The host-to-host overhead of VIP can be formulated as a function of the hop count, N_{hops}, as follows.

$$overhead = 89N_{hops} + 189 \ (\mu\text{sec})$$

The VIP overhead is approximately 0.6 msec if N_{hop} is 5, 1.0 msec if N_{hop} is 10, and 2.0 msec if N_{hop} is 20. Since IP transmission in the Internet takes time on the order of 100 msec, the VIP data packet transmission overhead is negligible from the user's viewpoint.

8 Conclusions

This paper proposed VIP, a protocol for mobile hosts. VIP is an instance of the VN-protocol, the general protocol providing host migration transparency to the transport layer. The key point of the VN-protocol is to separate logical identifiers from physical identifiers. In the conventional network architecture, each host is identified by its location in the network, i.e. by a physical identifier. However, each host should be recognized by a logical identifier because it is no longer stationary. This is the base of the virtual network concept. Such a notion should be applied to other identifiers such as group identifiers for multicast communications.

Currently, VIP is running on a test network. We plan to deploy VIP in the WIDE internet[3] and in a campus network of Keio University[4] to determine the optimum values for parameters such as the interval time of the *CAMT* packet transmission, the number of AMT entries, and the value of the *Holding Time* field of the VIP header.

[3]The WIDE internet is an academic internet in Japan. It is the infrastructure network of the WIDE project, a research project for a widely distributed environment.

[4]Most students (more than 2,000) have portable computers.

Some problems, however, still remain unsolved. One is security. To prevent malicious hosts from impersonating others, a host authentication mechanism must be implemented in VIP. Security is an area for future study. We also plan to apply the mechanisms of VIP to other connectionless-mode network layer protocols such as ISO CLNP.

Acknowledgments

We express our thanks to Dr. Mario Tokoro of Sony Computer Science Laboratory Inc./Keio University and Prof. Hideyuki Tokuda of Keio University/Carnegie Mellon University, for giving us many valuable suggestions on the basic concept of the virtual network protocol. We acknowledge the assistance of the members of Sony Computer Science Laboratory Inc. and the WIDE project, especially the members of the WIDE VIP working group, for their contributions to the design of the protocols of VIP.

References

[1] Carlberg, K. A Routing Architecture That Supports Mobile End Systems. In *Proceedings of MILCOM 92*, 1992.

[2] Droms, R. *Dynamic Host Configuration Protocol*, October 1993. RFC1531.

[3] Braden, R. *Requirements for Internet Hosts – Communication Layers*, October 1989. RFC1122.

[4] Ioannidis, J. and Maguire, G. Jr. The Design and Implementation of a Mobile Internetworking Architecture. In *Proceedings of 1993 Winter USENIX*, January 1993. pp.489 – 500.

[5] ISO. *Information processing systems – Data communications – Protocol for providing the connectionless-mode network service*, 1988. ISO 8473.

[6] Leiner, B., Nielson, D. and Tobagi, F. (Editors). *Proceedings of The IEEE – Special Issue on Packet Radio Networks*, Vol. 75, No. 1, January 1987.

[7] Postel, J. *Internet Protocol*, September 1981. RFC791.

[8] Rekhter, Y. and Perkins, C. *Loose Source Routing for Mobile Hosts*, February 1994. Internet Draft (draft-rekhter-lsr-mobile-hosts-00.txt).

[9] Sincoskie, W. and Cotton, C. Extended Bridge Algorithms for Large Networks. *IEEE Network*, Vol. 2, No. 1, January 1988.

[10] Sunshine, C. and Postel, J. *Addressing Mobile hosts in the ARPA Internet Environment*, March 1980, IEN135.

[11] Teraoka, F. VIP: A Protocol Providing Host Migration Transparency, *Internetworking: Research and Experience*, Vol. 4, No. 4, December 1993.

[12] Teraoka, F., Claffy, K. and Tokoro, M. Design, Implementation, and Evaluation of Virtual Internet Protocol. In *Proceedings of the 12th International Conference on Distributed Computing Systems*, June 1992. pp.170 – 177.

[13] Teraoka, F. and Tokoro, M. Host Migration Transparency in IP Networks: The VIP Approach. *ACM Computer Communication Review*, Vol. 23, No. 1, January 1993.

[14] Teraoka, F., Yokote, Y. and Tokoro, M. A Network Architecture Providing Host Migration Transparency. In *Proceedings of ACM SIGCOMM 91*, September 1991. pp.209 – 220.

[15] Uehara, K., Teraoka, F., Sunahara, H. and Murai, J. Enhancement of VIP and Its Evaluation. In *Proceedings of INET'93*, August 1993.

[16] Wada, H., Yozawa, T., Ohnishi, T. and Tanaka, Y. Mobile Computing Environment Based on Internet Packet Forwarding. In *Proceedings of 1993 Winter USENIX*, January 1993. pp.503 – 517.

[17] Xerox. *Internet Transport Protocols*, December 1981. XSIS 028112.