1ST QUARTER 2006, VOLUME 8, NO. 1

IEEE COMMUNICATIONS SURVEYS
The Electronic Magazine of Original Peer-Reviewed Survey Articles

www.comsoc.org/pubs/surveys

A REVIEW OF MOBILITY SUPPORT PARADIGMS FOR THE INTERNET

DEGUANG LE, XIAOMING FU, AND DIETER HOGREFE, UNIVERSITY OF GÖTTINGEN

ABSTRACT

With the development of mobile communications and Internet technology, there is a strong need to provide connectivity for roaming devices to continuously communicate with other devices on the Internet, at any time and anywhere. The key issue of this vision is how to support mobility in TCP/IP networks. In this article we review the TCP/IP protocol stack and analyze the problems associated with it in the mobile environment. We then investigate the mobility support techniques and existing solutions for providing mobility support on the Internet. We classify the proposed solutions based on the protocol layers and present paradigms for each category of layer. We also provide a comparison of the different solutions belonging to different categories, including their advantages and disadvantages. Results have shown that there is no single solution that perfectly addresses mobility support for the Internet. Finally, we conclude this survey with a recommendation of features that ought to be met in Internet mobility support.

With the rapid growth of wireless access technologies and the increasing number of mobile computing devices, two relevant scenarios other than traditional fixed networking have arisen. First, in a so-called *nomadic networking* scenario, a node requires access to the fixed network (for data or any other information services) at arbitrary times and from any location, without the need to continue the ongoing communication with their communicating peers during movements. In the second scenario, namely *mobile networking*, users require their services while roaming, preferably without interruption or the degradation of communication quality. In fact, the first scenario can be regarded as a special case of the second scenario. It is common that not only cellular devices, but also other types of computing devices (including PDAs and laptops) may desire to connect to the Internet in a nomadic or truly mobile fashion for various services, such as online gaming, video on demand, or stock trading. In this article we will focus on reviewing the existing mechanisms to support the second scenario, which provides true mobility support for roaming user devices.

As identified later, the traditional TCP/IP networks were originally designed for communications between fixed devices, and there are many issues that need to be resolved to support mobility. Given the importance of mobility support on the Internet in the last decade, studies that address these issues have arisen, coming up with a number of protocol proposals and IETF RFCs. Many of them have been designed and implemented, and some of them are starting to be deployed. Nevertheless, as analyzed in more detail below, they demonstrate both pros and cons in dealing with mobility support in terms of efficiency, functionality, security, etc. Therefore, a general comparison of different solutions is needed, including newly emerging alternatives, and a review and rethinking of the architectural aspect of Internet mobility support. Among previous work, Henderson [1] reviewed three host mobility solutions, namely, Mobile IP, TCP Migrate, and Host Identity Protocol (HIP)-based mobility, which operate in different layers, and compared them on various aspects of performance, security, deployment, scalability, robustness properties, etc. Eddy [2] discussed the strengths and weaknesses of implementing mobility at three different layers of the TCP/IP stack, suggesting that the transport layer is probably the best layer candidate to accommodate Internet mobility, and that there should be more collaboration between layers to avoid conflict and inefficiency. These exiting works did discuss some of existing and emerging mobility approaches and proposed some interesting metrics for comparison. Nonetheless, their reviews mainly focused on high-layer overview, while an in-depth analysis of the underlying properties of various proposals in introducing mobility to TCP/IP architecture is still

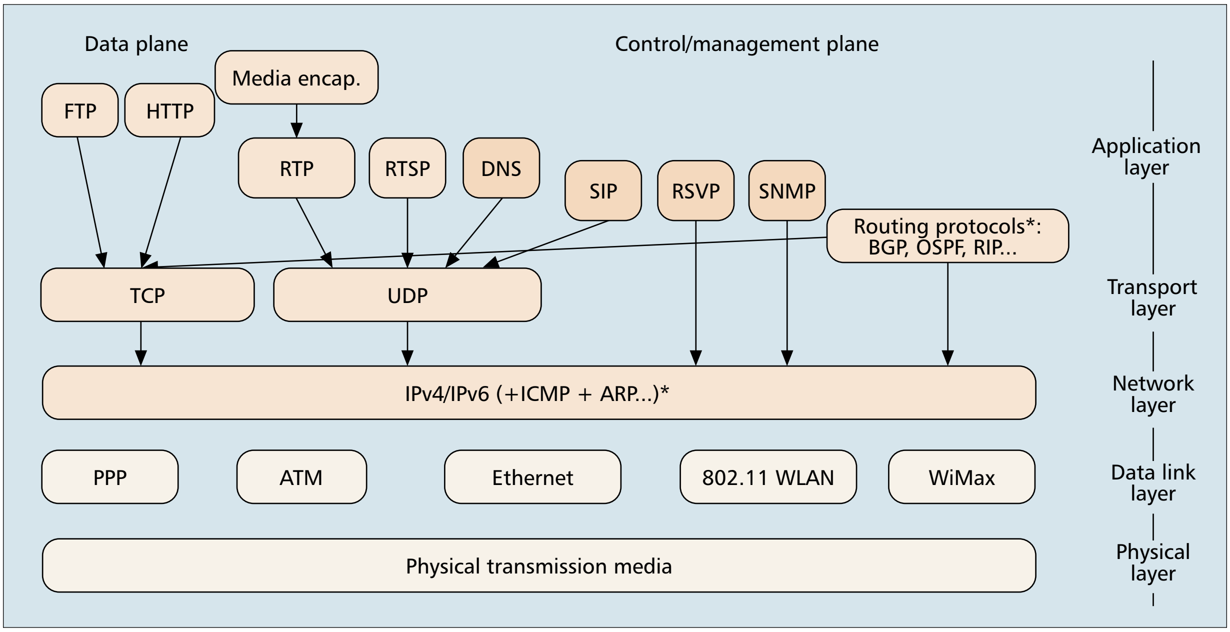


Figure 1. *TCP/IP stack.*

missing. At the same time, other approaches are not considered at all. The objective of this article is to investigate and compare existing Internet mobility support paradigms as comprehensively as possible, and to discuss what could be potentially deployable in terms of functionality, performance, changes to existing systems, etc. Interestingly, INFOCOM 2005 organized a panel discussion session on Internet mobility [3], and a number of issues discussed in this article were also discussed in the panel.

This article is organized as follows. We review the traditional TCP/IP stack, and present some general goals for any solution to mobility support for the Internet. In particular, we describe characteristics of communications in the mobile environment, the performance requirements for Internet mobility support, and why the traditional TCP/IP network is unable to support mobility. We present a detailed set of mobility support paradigms, each representing some specific changes to the existing protocol layer, and study the possible effect and impact, especially the integration of different mobile support paradigms. We summarize the advantages and disadvantages introduced by these different paradigms, and indicate that all existing solutions have different implications to their application scenarios. There is no single perfect solution so far; mobility support may require some rethinking of the Internet architecture, and there should be some general design considerations for any Internet 2 mobility support solution. Finally, we present our conclusion, which recommends features that ought to be provided for Internet mobility support.

The TCP/IP Stack and Why Mobility Support Is Difficult

In this section, starting with a review of the traditional TCP/IP, we describe the requirements, general goals for introducing mobility support, and problems in this stack.

TCP/IP Stack: A Review

For Internet communications, a number of protocols have to run in both end hosts and routers, utilizing a five-layer architecture (depicted in Fig. 1), where the Transmission Control Protocol (TCP), the User Datagram Protocol (UDP), and the Internet Protocol (IP) make up fundamental elements of the architecture, known as the TCP/IP stack.

In the TCP/IP stack, whereas most lower-layer (up to transport layer) functions are implemented in hardware devices and OS kernels, the application-layer protocols are implemented as user application daemon programs, interfacing with the transport layer to use the network service. The transport layer provides an end-to-end delivery service: TCP provides a connection-oriented service that allows for reliability, fragmentation, flow control, and congestion control, whereas the User Datagram Protocol (UDP) provides an unreliable datagram service that enhances basic network functions. The network layer is responsible for the routing and delivery of data from a source node toward a destination node across the same or different types of networks. The data link layer handles issues concerning the physical addressing, network topology, error notification, sequencing of frames, and flow control between neighboring nodes; link-layer protocols are typically specified by organizations such as IEEE, not the Internet community. The physical layer deals with the electrical/digital characteristics which are actually not part of the TCP/IP stack. (It is referred in the stack merely for compatibility with the OSI reference model.)

Basic Functional Requirements for Internet Mobility Support

Here, the term "Internet mobility support" refers to keeping ongoing communications continuity when an IP-based device moves (i.e., changes its topological point of attachment) to different networks. We exclude the case where the device just moves within a single network (or data link-layer mobility). In order to provide such support, a number of fundamental issues arise, which can be summarized as the following requirements for Internet mobility support.

Handover Management: The most important function needed to support mobility is to keep the ongoing communication alive while a mobile node (MN) moves and changes its point of attachment to the Internet. In order to continue to communicate, a core technology called handover management is required. The main objective of handover management is to minimize service disruption during handover.

Location Management: Another important function needed to support mobility is the reliable and timely notification of the MN's current location to those other nodes that need it. The technique to track the desired MN is called location management. Location management involves identifying the current location of the MN and also keeping track of their location changes as it moves on.

Multihoming: With a wide range of wireless access techniques such as GPRS, WCDMA/UMTS, IEEE 802.11x, etc. being introduced to provide access to the Internet, the future mobile environment will be characterized by diverse wireless access networks, and the MN will be equipped with multiple interfaces supporting different wireless techniques. Thus it is necessary to require multihoming support by which the MN can access the Internet through multiple links simultaneously and select and switch dynamic links while moving.

Applications: Internet mobility should also support current services and applications. That is to say, the mobility management mechanism should be transparent, without requiring changes to current services and applications.

Security: Any mobility solution must protect itself against misuses of the mobility features and mechanism, for example, stealing of legitimate addresses or flooding a node with a large amount of unwanted traffic. Therefore, security is an important concern when providing Internet mobility support. Motivated by these requirements, we argue that complete and useful Internet mobility should address these requirements as much as possible. In addition, there are performance requirements for mobile environments, as identified as below.

Performance Requirements for Internet Mobility Support

While developing an Internet mobility solution, the performance metrics also deserve special attention. The authors in [4] discuss the various subnetwork design issues that they consider relevant to efficient IP support in a general sense. In this subsection we discuss some performance metrics that are the most relevant for Internet mobility.

- **Handover Latency** refers to the elapsed time from the last packet received via the old network to the arrival of the first packet along the new network during a handover.
- **Packet Loss** is defined as the number of packets lost while maintaining communication during a handover.
- **Signaling Overhead** is defined as the number of messages for the handover and location procedures.
- **Throughput** is the amount of data transmitted over a mobile Internet in a given period of time.

Deployment Requirements for Internet Mobility Support

In addition to functional and performance requirements, there are some considerations that one should take into account to successfully deploy a mobility mechanism in the Internet. Below is a summary of those that seem most prominent.

- Minimum changes to the applications. It is desirable not having to change every application when the mobility mechanism is applied in the Internet.
- Avoid adding third-party. Adding a third-party device into the network usually incurs additional management overhead and security vulnerabilities, and should be avoided if possible.
- Easy integration into the existing infrastructure. Changes to allow integration into the existing infrastructure should be kept simple, as a well-deployed infrastructure implies a significant amount of investment, operational, and administrative/maintenance efforts if it is necessary to make updates to software or hardware in routers.

Limitation of Traditional TCP/IP for Internet Mobility

The traditional TCP/IP was designed for fixed computer networks. This subsection will analyze some of the limitations of TCP/IP for Internet mobility.

Limitation of the Link Layer — To the maximal possibility, wireless access techniques only provide the mobility of homogeneous networks at the link layer [5], which is not appropriate for Internet mobility across heterogeneous networks. In general the nature of network heterogeneity requires mobility support functions provided in higher layers. Besides, in mobile environments, the data link layer is based on wireless access technologies (such as 3G, WLAN etc.), which are characterized by low bandwidth, high bit error rates, faded and interfered signal with the radio channel, etc. These wireless link features are encountered by the moving terminals, which may degrade the transport performance of high layers.

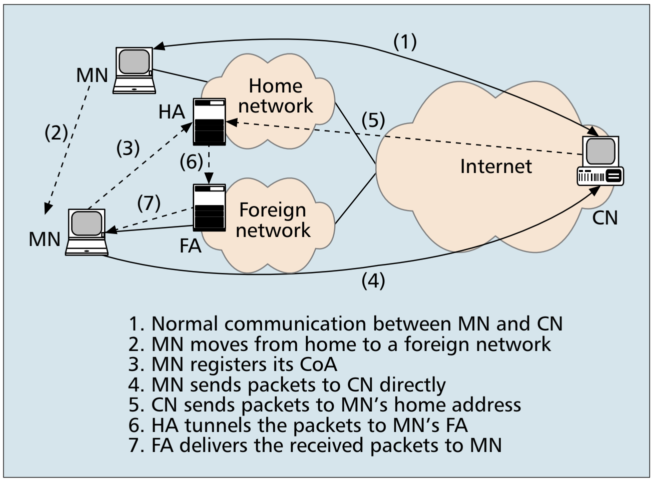
Limitation of the IP Address — The IP address of the network layer plays the roles of locator and identifier. In the mobile environment, the IP address of the MN has to be changed to represent the change of its point of attachment to the network when it moves from one network to another. In traditional TCP/IP, a change of the IP address makes it impossible for other devices to contact the device using a constant IP address. In addition, even if the device is able to obtain a new IP address dynamically, the transport connections established in the previous network will be broken after the change of IP address.

Lack of Cross-Layer Awareness and Cooperation — For example, the design of traditional transport-layer protocols relies on the services provided by the network layer, and does not consider wireless link properties and mobility. Thus, the congestion control of TCP [6] does not distinguish the packet loss caused by handover of mobility and wireless link properties from the normal packet loss in wired networks, which degrades transport performance [7]. Besides, TCP congestion control is based on the assumption that the end-to-end path of a connection is relatively stable after connection establishment. In the mobile environment, the MN will change its access point of Internet attachment without notifying TCP of its moving, and thus the existing end-to-end connection path has to be changed accordingly, which may violate this assumption and cause TCP to make congestion control decisions based on invalid information [8].

Limitation of Applications — Many applications based on traditional TCP/IP architecture are also limited in use in the mobile environment. For example, in Domain Name System (DNS), the Fully Qualified Domain Name (FQDN) is usually statically bound to a node's IP address. Thus the tight binding between the FQDN and the IP address will be invalid because of the dynamic change of IP addresses of the MN.

Extending TCP/IP to Support Mobility

As mentioned in the previous section, the traditional TCP/IP is not appropriate for Internet mobility. Therefore, various solutions have been developed to address it. Among them,



1. Normal communication between MN and CN
2. MN moves from home to a foreign network
3. MN registers its CoA
4. MN sends packets to CN directly
5. CN sends packets to MN's home address
6. HA tunnels the packets to MN's FA
7. FA delivers the received packets to MN

Figure 2. *MIPv4 architecture and its operations.*

those representing the network layer are Mobile IPv4 (MIPv4) [9], Mobile IPv6 (MIPv6) [10], and Location Independent Network Architecture for IPv6 (LIN6) [11]. In the transport layer, a wide range of studies have been undertaken to provide mobility support for TCP [12–18], the Stream Control Transmission Protocol (SCTP) [19], and the Datagram Congestion Control Protocol (DCCP) [20]. Session Initiation Protocol (SIP) [21], Dynamic DNS (DDNS) [22], and IKEv2 Mobility and Multihoming (MOBIKE) [23], [24] provide mobility support in the application layer.

Some researchers were interested in introducing a new protocol layer between the classic network layer and transport layer to provide Internet mobility, such as Host Identity Protocol (HIP) [25] based mobility [26], and Multiple Address Service for Transport (MAST) [27].

In this section we investigate the solutions for improving mobility of TCP/IP in more detail.

Mobility Support in the Network Layer

Because IP is the ubiquitous internetworking layer for the Internet, solutions that build on the existing network layer are considered a natural approach. Mobile IPv4 (MIPv4), proposed by Perkins [9], Mobile IPv6 (MIPv6), proposed by Johnson *et al*. [10], and various enhancements to the performance of MIPv4/v6 proposed in [28–34] have represented "classic" means for supporting mobility on the Internet. The Location Independent Network Architecture for IPv6 (LIN6) proposed by Teraoka *et al*. [11] provides an alternative to mobility support to MIPv6. These protocols apply techniques such as proxy, tunneling [35], and locator/identifier separation [36] to deal with mobility.

Mobile IPv4/IPv6 and Its Enhancement — MIPv4 defines a home network where the MN is assigned a permanent IP address called the home address that identifies the MN. MIPv4 also defines foreign networks that the MN visits. It introduces two new entities, the home agent (HA) and the foreign agent (FA), to relay the packets between the MN and the correspondent node (CN).

In MIPv4, when the MN is on its home network, it acts like any other fixed node (FN) of that network and requires no special mobile IP features. Each time it moves out of its home network and accesses a foreign network, it obtains a care of address (CoA), e.g. through Dynamic Host Configuration Protocol (DHCP) [37], and informs its HA of the new address by sending a Registration Request message to the HA. Upon the HA receiving the Registration Request message, it replies to the MN with a Registration Reply message. The HA then assumes the MN and once packets destined to the MN arrive at the home network, the HA intercepts these packets by using the Proxy Address Resolution Protocol (ARP) [38, 39] and forwards them to the MN with the CoA via a tunneling technique. When the FA receives packets, it removes the IP encapsulation of the packets and delivers them to the MN. When the MN wishes to send packets back to the CN, the packets are routed directly from the MN to the destination, where the MN uses the FA as its default router. Figure 2 shows the MIPv4 architecture and its operations.

Packets of MIPv4 to the MN travel via the HA, whereas the packets from the MN are routed directly to the destination, which incurs triangular routing. MIPv4 registration clearly takes a long time, which significantly increases handover latency. Also, since packets destined for the MN are not delivered until registration is completed at the HA, this interruption may cause packet loss. Furthermore, the Agent Advertisement messages and Registration messages that are sent when the node is traversing also introduce overhead over the Internet. As the number of wireless users grows, the signaling overhead will increase. To avoid these drawbacks, a number of techniques such as routing optimization technique [40], anticipation technique [29], hierarchical technique [28], and paging technique [41] etc., have been developed to enhance the basic protocol. In specific environments where the MNs frequently change their point of attachment to the network and the number of mobile users grows simultaneously, a number of micro-based mobility protocols (such as regional registration [28], Low Latency Handover in MIPv4 [39], Hawaii [30], and Cellular IP [31]) have been proposed to improve the performance of MIPv4.

Security Considerations for MIPv4 — First, the MN may suffer from the router's ingress filtering. A foreign network protected by a firewall may reject the packets when the MN sends the packet directly to the CN using its home address as the source address. Ingress filtering can be avoided by using reverse tunneling. Second, a major risk is associated with the authentication of the MN. If a bogus CoA was registered with the HA, it could prevent all connections to the MN, or even worse, cause all packets to be redirected to some attacker. To prevent this, the registration messages must be authenticated. Therefore, RFC 3344 [9] specifies the authentication extensions that are supplied with MIPv4 registration messages. The authentication extension contains the Type, Length, Security Parameter Index (SPI), and a "message digest," which is calculated using HMAC-MD5 [42] (and keyed MD5 [43] for backward compatibility with older MIPv4 implementations). Third, without replay protection the attacker could perform valid but unwanted operations afterward by resending old registration messages. Therefore, MIPv4 proposes to add some information (e.g., timestamps) to the registration messages by the message sender, and then the receiver can check the validity of the message. To avoid the latency and clock resynchronization issues, an optional nonce-based replay-protection approach is also suggested [9].

Because the traditional IP protocol has a variety of limitations for the next-generation Internet [44], the IETF defines a new network layer protocol, i.e. IPv6, attempting to replace the current IP protocol. The IPv6 is inherent in supporting Internet mobility management via MIPv6 [10]. MIPv6 follows the same basic principles as MIPv4, including home address, CoA, HA, and tunneling. The main difference is that an FA no longer exists and the security aspect has been improved. In addition, route optimization has also been incorporated into MIPv6.

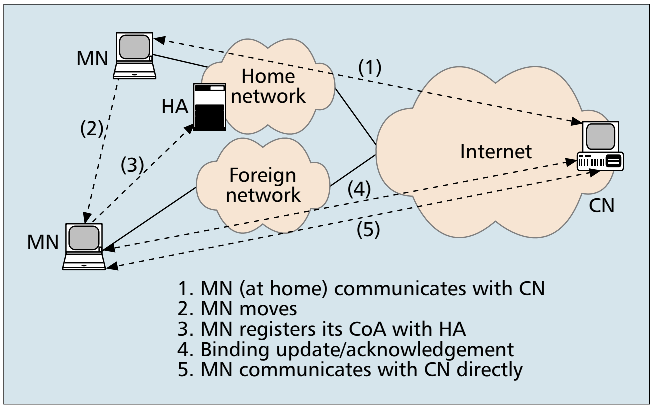

MN
HA
Home network
(1)
(2)
(3)
Internet
Foreign network
(4)
(5)
CN
MN
1. MN (at home) communicates with CN
2. MN moves
3. MN registers its CoA with HA
4. Binding update/acknowledgement
5. MN communicates with CN directly

Figure 3. *MIPv6 architecture and its operations.*

In MIPv6, when the MN moves to another network it acquires the CoA through either stateful [45] or stateless [46] address auto-configuration. After obtaining a new CoA, the MN registers to the HA and also to the CN with Binding Update messages (BUs). The HA and the CN record this binding in its binding cache. After this, packets from the CN can be routed directly to the CoA of the MN with the CN's home address in the Routing header. Similarly, the MN sends all packets to the CN directly using the Home Address Destination option, which eliminates the triangle routing. In the event that the CN wants to communicate to the MN for the first time, the first packet is tunneled through the HA as in MIPv4. The HA intercepts any packets addressed to the MN's home address and tunnels them to the MN's CoA using IPv6 encapsulation. For discovering the HA, MIPv6 defines the Dynamic Home Agent Address Discovery (DHAAD) mechanism [10]. Figure 3 shows the MIPv6 architecture and its operations.

Because BUs are transferred between the MN and the CN, as well as the HA, this incurs significant extra overhead, especially when MNs move quickly or increase proportionally. Thus, the IETF developed the Hierarchical Mobile IPv6 (HMIPv6) [32] protocol to reduce overload and improve handover speed by separating the mobility management local mobility from global mobility. HMIPv6 proposes a multi-level hierarchical network architecture and defines a site as any level of the hierarchical architecture. Inside the visited — or foreign — network, a new entity called the mobility anchor point (MAP) is introduced. It acts like a local HA. When the MN moves within the foreign network, it will only register its new local CoA to the MAP. The local mobility can be completely hidden from all nodes outside the site. When the MN moves between inter-sites, the mobility will be handled by MIPv6.

Fast Handovers for Mobile IPv6 (FMIPv6) [33] is another proposal aiming at optimization for MIPv6. FMIPv6 attempts to acquire information that is needed to join a new link before disconnecting communication at the old link. It utilizes cooperating access routers that can request information from other access routers that are possible candidates for a handover. This is done by establishing a tunnel between the two access routers that allows the MN to send packets as if it was connected to its old access point while it is completing its handover signaling at its new access router. Therefore, it reduces the procedure time of movement detection, new CoA configuration, and binding updates, etc. during handover, and eliminates packet loss. Jung *et al*. [34] propose a combination of both approaches of HMIPv6 and FMIPv6, which is designed to combine the advantages of both and provide additional improvements to reduce signaling overload, packet loss, and handover latency.

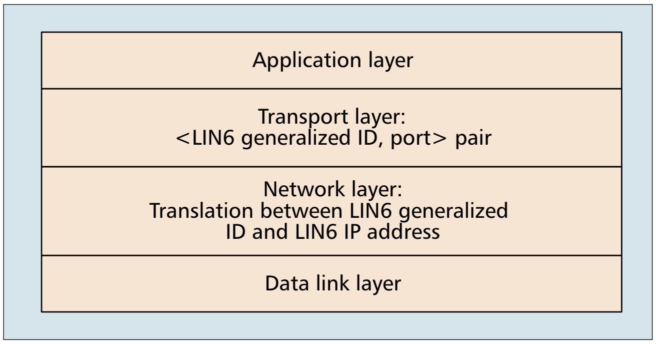
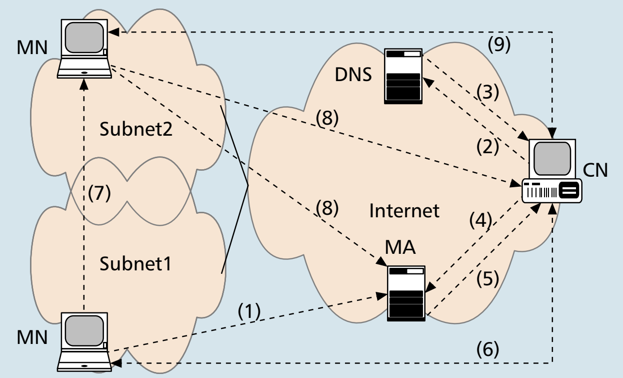

Application layer
Transport layer:
<LIN6 generalized ID, port> pair
Network layer:
Translation between LIN6 generalized ID and LIN6 IP address
Data link layer

Figure 4. *LIN6 architecture.*

Security Considerations for MIPv6 — In MIPv6, the security features are integrated and provided as an extension to headers. The traffic can be protected by IP security protocol (IPsec) [47] Authentication Header (AH) [48] and Encapsulating Security Payload (ESP) [49] extension headers. Furthermore, IPsec is also suggested to protect MIPv6 BUs and BAs between the MN and the HA from forgery of the data originator and replay attacks. The MN and the HA are required to establish IPsec security association (SA) either by manual configuration or automatic key management protocols. BUs and BAs between the two entities can then be protected using the IPsec ESP in transport mode or the AH extension headers. For the protection of the registration messages of BUs and BAs between the MN and the CN, MIPv6 uses the return routability procedure to assure that the right MN is sending the messages. The detailed procedure is specified in [10], based on the idea of relying on the routing infrastructure to check that the MN is reachable both at its claimed home address and its claimed CoA. The advantage of this method is that it limits the potential attackers to those having an access to one specific path on the Internet, and avoids forged BUs from anywhere else on the Internet. The weakness of the method is that it doesn't defend against attackers who can monitor the path between the home network and the correspondent node. The return routability procedure is therefore subject to active attacks such as the Man-in-the-Middle attack launched by such attackers. This weakness has been investigated and some improvements have been proposed [50–52]. In addition, MIPv6 develops route optimization as an alternative to reverse tunneling. The MN uses the home address in a packet with the Home Address Destination option. MN uses its CoA as the source address in the IP header and sends the packet directly to the CN, so it can avoid ingress filtering and pass through the firewall. Unfortunately, there have not been enough security considerations for HMIPv6 and FMIPv6, and further security and operational issues with regard to MIPv6 and its extensions are still not yet addressed, for example, interacting with the AAA infrastructure, bootstrapping, and general stateful packet firewall traversal. Some of these issues have been discussed in recent IETF proposals and research investigations (e.g., [53–56]).

LIN6 — LIN6 proposes an alternative Internet mobility solution for the IPv6 protocol. Its basic idea is separating the identifier and locator in the IPv6 address. LIN6 introduces the LIN6 ID for each node as the node identifier so that each node can be identified by its LIN6 ID no matter where the node is connected and no matter how many interfaces the node has. In addition, it defines two types of network addresses: the LIN6 generalized ID and the LIN6 address. The LIN6 generalized ID is formed by concatenating a constant value called the LIN6 prefix before the LIN6 ID. It is used in the transport layer to identity the connection. The LIN6 address



1. Register MN network prefix<->LIN6 ID at bootstrap
2. Query MN FQDN
3. Response MA network prefix+LIN6 ID
4. Query MN LIN6 ID
5. Response MN network prefix
6. Establish connection and start data transfer
7. Move
8. Update the mapping of LIN6 ID<->MN network prefix
9. Go on communication

Figure 5. *LIN6 mobility and its operations.*

is formed by concatenating the network prefix and LIN6 ID. It is used to route packets over the network layer. The network prefix will then change according to the network where the MN attaches. Figure 4 illustrates the LIN6 architecture.

In LIN6, on packet transmission, the network layer extracts the LIN6 ID from the LIN6 generalized ID and concatenates the network prefix and LIN6 ID to create the LIN6 address of the destination node. On packet reception, the network layer removes the network prefix part of the LIN6 address, and then attaches the LIN6 prefix to create the LIN6 generalized ID of the source node. When the MN moves to another network and obtains the network prefix of the new network, the MN updates its location with the CN in one of two ways. If the MN has a security association, it sends the Mapping Update Request message to the CN. In this case, the Mapping Update Request message must include the Authentication Header. If the MN has no security association, the MN sends the Mapping Refresh message to the CN to inform the CN that the MN has moved. As a result, the CN re-queries the mapping agent (MA) to obtain the new network prefix of the MN. The MN also sends Mapping Update Request message to the MA to inform the current network prefix.

In order to track the current location of the MN, LIN6 employs the MA to maintain the mapping of the LIN6 ID and the network prefix, and makes use of the DNS to locate the MAs of the MN. Each MA shall be assigned a predefined 64-bit value called MA IFID as the interface identifier. When the MN is powered on and attaches to a network for the first time, it registers its current location with its MAs. When the CN wants to communicate with the MN for the first time, the CN sends a query to the DNS sever and obtains the Authentication, Authorization, Accounting and Auditing (AAAA) record, which consists of the network prefix of the MA and the LIN6 ID of the MN. Then the CN generates the IPv6 address of the MA by concatenating the upper 64 bits of the AAAA record and the MA IFID, which is used as the lower 64 bits of the IP address. Then it queries the MN's MA for the network prefix of the MN and receives the IP address of the MN. When the MN moves to a new network, it registers the new network prefix with the MA and the CN by sending Mapping Update messages (MUs) with the Authentication Header or Mapping Refresh message. Figure 5 shows the LIN6 network architecture and its operations.

Security Considerations — In LIN6, location registration with the DNS/MA is authenticated by IPsec or exchanged cookies. Thus, the security level is almost the same as in MIPv6.

Analysis of Network Layer Mobility — MIPv4 provides network-layer mobility and transparency to the higher layers. However, there are a number of problems associated with it, e.g. that triangular routing introduces higher latency and extra overhead to the network. In addition, all packets to the MNs pass through the HA, which induces heavy load for the HA, and in the event of an HA failure, all the desired traffic for MNs using that HA will be interrupted. Thus, MIPv4 is vulnerable to single point of failure. Although many enhanced techniques and micro-mobility protocols can improve MIPv4 performance, MIPv4 still has weakness in terms of efficiency and complexity. MIPv6 has the advantages of inherent mobility, security support, and routing optimization compared with MIPv4. BUs and Binding Acknowledgment messages (BAs) are authenticated using IPsec AH and ESP. The CN learns the MN's CoA dynamically and sends packets directly to the MN by using the IPv6 routing header. However, as in MIPv4, MIPv6 has the same problem of the third device, which increases failure probability of communication, and it has additional header overhead. The enhancements of HMIPv6, FMIPv6, and their combination improve the performance by minimizing signaling overhead, packet loss, and handover latency, but their scalability and complexity are a concern.

In comparison with MIPv4/MIPv6, LIN6 is more tolerant of defects/errors because the HA in MIPv4/MIPv6 cannot be replicated to the subnet other than at the home link, while the MA introduced in LIN6 can be replicated anywhere on the Internet. Also, LIN6 has less overhead due to its avoidance of the extension header and tunneling. That is, LIN6 does not use any packet interceptor or forwarder such as the HA, so its routing is the same as traditional IP-based routing. Conceptually, LIN6 adds a transient "presence" service to DNS lookup for dynamic locator mapping (in this sense, LIN6 can also be considered as the introduction of a new layer), but it is only limited to IPv6.

Mobility Support in the Transport Layer

Because the transport layer is subject to the impact of mobility, much work has been carried out over the past few years on TCP performance improvement and mobility enhancement [12–18], including efforts to enhance UDP for mobile environments (e.g., [57]). Recently, the mobility support for the new transport layer protocols of SCTP and DCCP has been proposed. The basic idea of enabling transport-layer mobility is to remove network-layer dependences by using indirection, migration, tunneling, multihoming techniques, etc.

Extending TCP — Much focus has been placed on the TCP as it is the most widely used transport-layer protocol. We classify the different proposals into two categories: improving TCP performance for the mobile Internet and TCP mobility support extension.

Improving TCP Performance for the Mobile Internet — TCP is a reliable transport protocol tuned to perform well in traditional wired networks where network congestion is the primary factor of packet loss. However, networks with wireless links and mobile hosts induce significant increases in losses due to high bit error rates, temporary disconnection, limited bandwidth, etc., which violate many of the assumptions made by traditional TCP, causing TCP to not adapt well to these

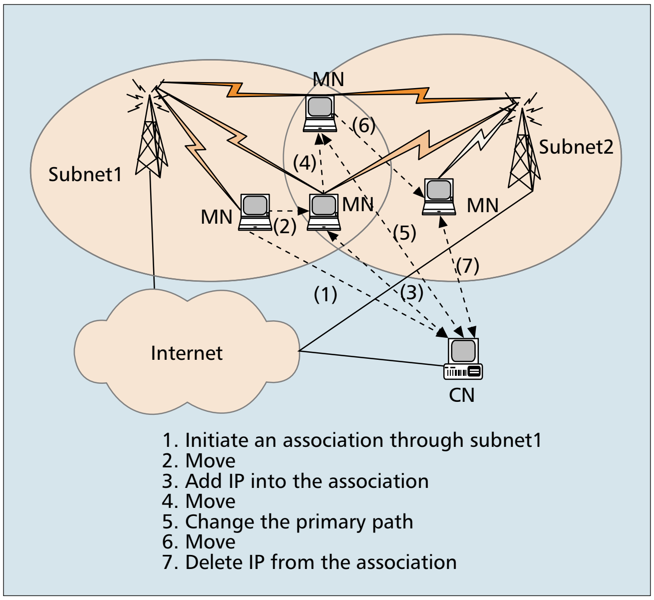

MN
(6)
Subnet1
Subnet2
(4)
MN
(2)
MN
MN
(5)
(7)
(1)
(3)
Internet
CN
1. Initiate an association through subnet1
2. Move
3. Add IP into the association
4. Move
5. Change the primary path
6. Move
7. Delete IP from the association

Figure 6. *MSCTP mobility and its operations.*

environments. Therefore, a number of researchers have aimed to improve TCP performance for the mobile Internet. Indirect TCP (I-TCP) [12] and Mobile TCP (MTCP) [13] focus on the bit error rate (BER) problem of wireless links. In ITCP and MTCP, a TCP connection between the MN and the FN is split in two with a device called the mobile support station (MSS), and the connection between the MSS and MH is optimized for the wireless link. Both I-TCP and MTCP achieve better throughput than standard TCP. Caceres and Iftode used a fast retransmission mechanism [14] to address the problem of short disconnections during handover. Haas developed an asymmetric transport-layer protocol called Mobile-TCP [15] to minimize communication overhead on the MN. In Mobile-TCP, functions through algorithms and procedures are implemented with the lower complexity on the MN than the FN without sacrificing performance and features. To avoid the invalid TCP congestion control, decisions incurred by the change of TCP connection path in the mobile environment, Y. Swami *et al*. [8] implement a Lightweight Mobility Detection and Response (LMDR) TCP option that allows the MN to inform the CN when it detects the location change which can be assisted by other layers such as the neighbor discovery of MIPv6. Based on the notification, the proper congestion control behavior can take place and react to correct the performance.

The above proposals optimize the transport performance of TCP over networks with wireless links. Although they cannot support real mobile networking, they provide mobility enhancement for the nomadic networking scenario.

Mobility Extension to TCP — Other researchers have considered the issue of how to maintain the ongoing TCP connection when an interruption occurs due to a change in the IP address.

Funato [16] develops a simple and secure redirection mechanism called TCP Redirection (TCP-R) to maintain active TCP connections. The concept of TCP-R is to revise the pair of addresses in the ongoing TCP connection when the IP address associated to the TCP connection is changed by TCP redirection options extension. In TCP-R, when the MN initiates a new connection, it ascertains if the CN is TCP-R aware or not, and then may perform a redirection operation. When the MN moves and is assigned a new IP address, it sends a Redirect message with the RD REQ option to the CN. Upon the CN receiving the message, it validates the connection authenticator with AT REQ and AT REP. If correct, it revises the pair of addresses of the ongoing TCP connection with the new MN's IP address. Simultaneously, the MN also revises its own pair of IP addresses. Finally, they resume communicating with the revised TCP connection.

Snoeren and Balakrishnan [17] propose an end-to-end approach to support TCP mobility through a migrating technique. TCP Migrate is similar to TCP-R. It differentiates from TCP-R through its different implementations by specifying different TCP migrate options.

MSOCKS [18] presents an alternative TCP mobility support by split-proxy mechanism and extension to SOCKS [58]. In MSOCKS, when the MN changes the IP address that a TCP connection uses to communicate with the MSOCKS proxy, it opens a new connection to the proxy and sends a RECONNECT message with the connection identifier of the existing connection. Upon receiving the RECONNECT message, the proxy separates the old connection between MN and Proxy (MN-Proxy) from the connection between the Proxy and the CN (Proxy-CN), and then concatenates the new MNProxy connection to the Proxy-CN connection in place of the old MN-Proxy connection. Finally, the proxy closes the old connection. Once the concatenation is setup, the proxy sends an OK message to the MN.

As TCP is one of the primary protocols in the TCP/IP stack, the performance of TCP over mobile environments is still a hot topic today after almost 10 years of study. For example, Elaarag surveyed and compared different approaches to improve the performance of TCP over mobile wireless networks [7]. Jaiswal and Nandi [59] evaluated the impact of MIPv6 on TCP variants. These researchers give some new guidance for improving TCP performance in the mobile environment or mobility enhancement by itself.

M-UDP — Since wireless links tend to be susceptible to BERs and UDP will also be sustained to a large percentage of packet loss, Brown and Sigh [57] proposed a Mobile UDP (M-UDP). M-UDP aims at reducing packet losses in wireless links. The idea is similar to I-TCP and M-TCP, namely, to split UDP connections in two at some node close to the mobile user. This node (called the "supervisor host," or SH) attempts to use any free bandwidth to retransmit packets lost during a fade, thus ensuring that the number of lost packets is kept small. Every UDP packet is buffered at the SH; the SH discards a packet if it has run out of buffer space or if it has been observed to have been transmitted certain times. This approach is simply a straightforward solution and does not consider security in the first place. Further details are described in [57, 60].

MSCTP — A recently developed IETF transport-layer protocol, the Stream Control Transmission Protocol (SCTP) [19], provides another potential approach for mobility support due to its multihoming feature. Using SCTP's ADDIP extension [61], Mobile SCTP (MSCTP) has been proposed [62].

In MSCTP, the MN initiates an SCTP association with the CN by negotiating a list of IP addresses. Among these addresses, one is chosen as the primary path for normal transmission, and the other addresses are specified as active IP addresses. When the MN reaches a new network and obtains a new IP address, it sends an Address Configuration Change (ASCONF) Chunk with an Add IP Address parameter to inform the CN of the new IP address. Upon receiving the ASCONF, the CN adds the new IP address to the list of association addresses and returns the ASCONF-ACK Chunk to the MN. While the MN is moving, it may change the primary

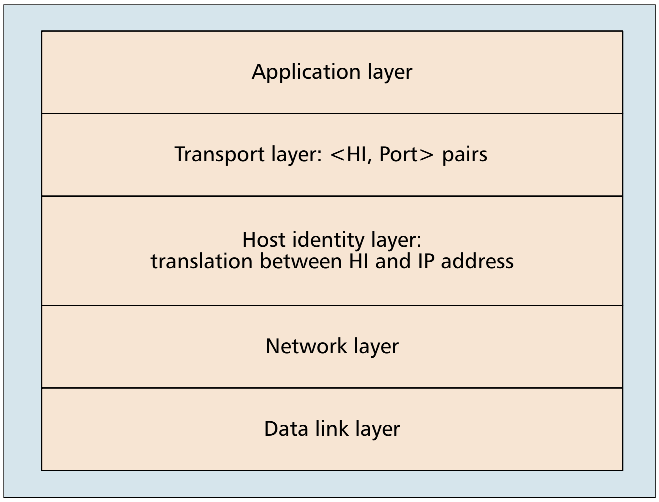

Application layer
Transport layer: <HI, Port> pairs
Host identity layer:
translation between HI and IP address
Network layer
Data link layer

Figure 7. *Introducing HIP into the TCP/IP stack.*

path to the new IP address via the path management function [19]. The SCTP association, therefore, can continue data transmission while moving to the new network. The MN also informs the CN to delete the IP address of the previous network from the address list by sending an ASCONF Chunk with a Delete IP Address parameter when it confirms that the previous network link has permanently failed. Figure 6 illustrates the operations of MSCTP.

Security Considerations — Unlike TCP, SCTP uses a fourstep negotiation process to initiate an association, which can prevent Denial of Service (DoS) attacks such as an SYN attack. IPsec is then used to secure the SCTP communication. The addition/deletion of an IP address to an existing association during mobility does provide an opportunity in which existing associations can be hijacked. The attacker is then able to intercept and alter the packets sent and received in the association. For this reason, MSCTP suggests using IPsec or Transport Layer Security (TLS) [63, 64] to protect against this insecure/threatening environment.

DCCP — The Datagram Congestion Control Protocol (DCCP) [20] provides integrated mobility and multihoming support by defining the DCCP-Move packet type and two new DCCP features: the mobility capable feature and the mobility ID feature. DCCP specifies mobility support as optional and the default to be off, thus DCCP nodes must enable mobility support with the mobility capable feature.

First, the MN sends a Change L option of the mobility capable feature to inform the CN that it would like to enable changing its address during connection. Then the CN sends a Change R option to confirm the MN. After that, the MN sends a value of the mobility ID feature that is used to identify connection. The value of the mobility ID feature is selected randomly for security reasons, and a new value is chosen after each move of the MN. The CN confirms the value of the mobility ID feature by sending a Conform L option. When the MN reaches a new network and obtains the new IP address, it sends a DCCP-Move packet containing a mobility ID value that was chosen for connection identification. Upon receiving the DCCP-Move packet, the CN sends a DCCP-Sync message to the MN, and changes its connection state, using the new MN address.

Security Considerations — DCCP does not provide cryptographic security guarantees. Nevertheless, by sequence number validity checks, DCCP can protect against some attacks. For example, attackers cannot hijack a DCCP connection unless they can guess valid sequence numbers, which are randomly chosen according to the guidelines in [65].

Analysis of Transport Layer Mobility — The TCP extensions proposed for improving transport performance on the mobile Internet cannot deal well with mobility on their own. Their main purpose is merely to minimize degradation of transport performance. The mobility enhancements of TCPR, TCP Migrate, and MSOCKS to TCP can handle mobility and keep all features of the standard TCP. Their operations are done in a secure way.

MSCTP provides an alternative solution in the transport layer. It can support seamless handover and improve transport performance. However, the current MSCTP proposal only illustrates the basic requirements for Internet mobility. Some essential issues, such as when and by which criteria the primary path is changed, or the addition and deletion of the IP addresses mapped to the SCTP association, should occur during handover and are open to further study. Moreover, MSCTP by itself does not handle location management, thus a proposal on reusing MIP for location management in MSCTP is proposed in [66].

Similarly, the current specification of DCCP is at its primitive stage. There are many problems unsolved. For example, DCCP has no support for simultaneous movements of both communicating endpoints, i.e. DCCP supports mobility of only one endpoint, while the other endpoint remains stationary.

Providing Mobility Support in a New Layer

Traditional TCP/IP protocols are already heavily loaded down with functionalities that have been added over the years. Optimization and adding new functionalities to support mobility are very difficult. A new idea has therefore emerged for Internet mobility that supports introducing a new layer, such as HIP and MAST, where Internet mobility is deployed.

HIP — The Host Identity Protocol (HIP) [67] is being designed by the IETF to establish secure communication and to provide continuity of communication. Similar to LIN6, HIP is based on the idea of separating location from identity by an interposed host identity protocol layer that operates between the network and transport layers (Fig. 7). HIP introduces a new host identity namespace called the host identifier (HI), which is a public key. The transport-layer connection is bound to HI instead of the IP address, and the IP address becomes a pure routing message. The HI is dynamic, mapped to one or more IP addresses in the HIP layer. In practice, HIP uses a host identifier tag (HIT) to represent HI. The HIT can be obtained by taking the output of a hash function over the HI, and truncating it to the IPv6 address size.

In HIP, the dynamic binding between HI and IP address is achieved by using the update packet with HIP readdress packets (REA) parameter. In addition, HIP employs the rendezvous sever (RVS) to provide location management. Upon HIP initiation, the initiator retrieves the RVS IP address by looking up the domain name of the peer from DNS with a HIP RVS resource record (RR), and sends I1 with destination HIT packet to the RVS. The RVS then forwards the initial HIP packet to the peer at its current location. After receiving I1, the peer completes HIP initiation directly without the help of RVS. Throughout ongoing communication, the MN moves and acquires a new IP address, sending an HIP update packet with REA to inform the CN of the new IP address, and the CN responds to the ACK. Due to security concerns, the CN may verify that the MN is available through

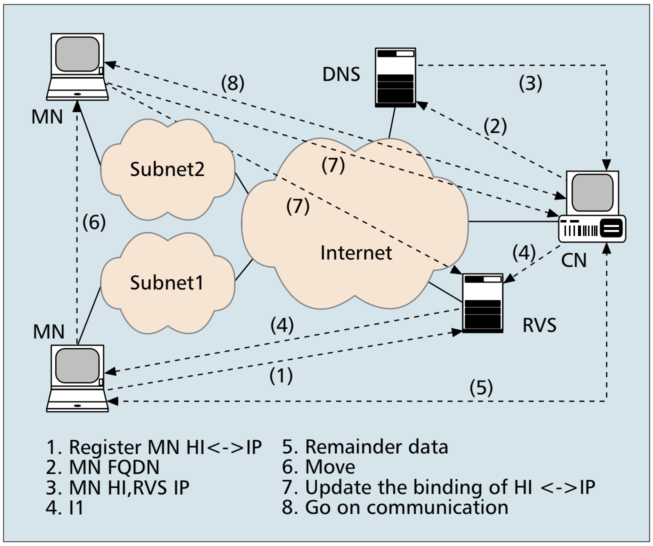


Figure 8. *HIP mobility and its operations.*

the new IP address. Once the CN has successfully verified this, the new IP address becomes active and the old address is removed, so that the CN can communicate through the new IP address. Figure 8 illustrates the operations of HIP.

Security Considerations — In HIP, the connection establishment procedure includes four steps instead of the traditional three in TCP, thus preventing DoS attacks. Communications are bound to the public keys of the HI, as opposed to IP addresses, and are encrypted with ESP, so the hijack attempt would also be unable to reveal the contents of communications. The REA message is also signed with the sender's public key, so it is impossible to hijack communications through the use of the REA message.

MAST — Multiple Address Service for Transport (MAST) was proposed by Crocker [27] for Internet mobility and multihoming. Like HIP, MAST defines a layer between the network and transport layers without creating a new namespace by using the existing IP addresses, in which the initial IP address assigned for the transport layer connection/association is used for the identifier of the MN, and other IP addresses added dynamically while moving are used as the locator of the MN. Thus, the basic idea of MAST in providing mobility is simple: it maps different IP addresses to the single initial IP address.

MAST defines a mechanism that supports multiple IP address association. The MAST association is manipulated with Request/Response messages, which are used to initially establish the MAST association, update the set of valid IP addresses, query association status, convey error information, terminate the association, etc.

In MAST, when the MN moves across the Internet, the IP addresses of the MN locator may be added and removed, while the initial IP address continues to be bound to the transport layer. Other addresses of the MN locator are mapped to that initial IP address of the MN identifier by MAST control exchange. Over the life of the association, the different MN locator addresses might be active at different times. To find the MN, MAST uses DNS to provide the information of dynamic presence service relating to the MN. The DNS SRV [68] record is defined to reference a dynamic presence service through which an endpoint can register its current set of IP addresses. MAST specifies that the MN registers its current address with the dynamic presence service available through the Extensible Messaging and Presence Protocol (XMPP) [69]. Figure 9 illustrates the MAST-based approach for mobility management.

Security Considerations — To resist the attacks of hijacking an association, MAST uses association-specific weak authentication [70], which ensures that later packets come from the same source as the initial packet. In addition, IPsec or TLS is also suggested for other security issues sich as spoofing, redirection, etc.

Analysis of New Layer Mobility — HIP supports multihoming by dynamic mapping from one HI to multiple IP addresses. It also resolves the problem of simultaneous movement of endpoints by resending the HIP Readdress message to the RVS if no reply is received. However, the RVS also changes the basic property by replacing the IP addresses of their client nodes in the DNS with their own addresses. Thus, the IP address in the DNS entry no longer directly designates the endpoint. It suffers from failures because the I1 packet must be relayed by the RVS when initializing a connection. In addition, the applications that have followed the structure of traditional layers have to be modified to it. MAST does not define any new namespace or addressing structure, and requires no change to IP modules or transport modules. In addition. it has no additional packet header overhead and minimal additional packet-processing overhead. Hence, MAST has a low barrier to adoption and use. However, as its development is still in its preliminary stages there are many open issues to be resolved. For example, the optimal locator selection can imply some design difficulties.

Mobility Support in the Application Layer

Attempts have also been made to support Internet mobility in conjunction with the application layer. This section discusses Internet mobility support using SIP, DDNS, and MOBIKE in the application layer.

SIP — The Session Initiation Protocol [21] was initially developed by the IETF as an application-layer multimedia signaling protocol. Nonetheless, it demonstrates potential capabilities for Internet mobility through its ability to define a number of specific entities and specify SIP messages. In SIP the main entities are the user agent, the redirect server, and the proxy

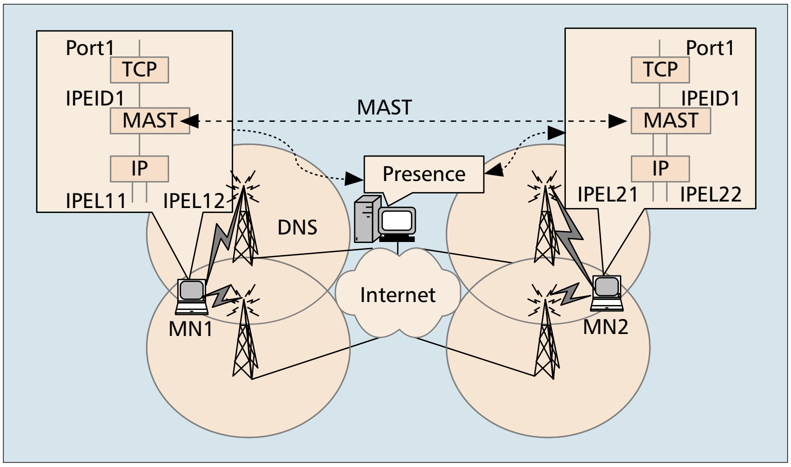


Figure 9. *MAST-based approach for mobility management.*

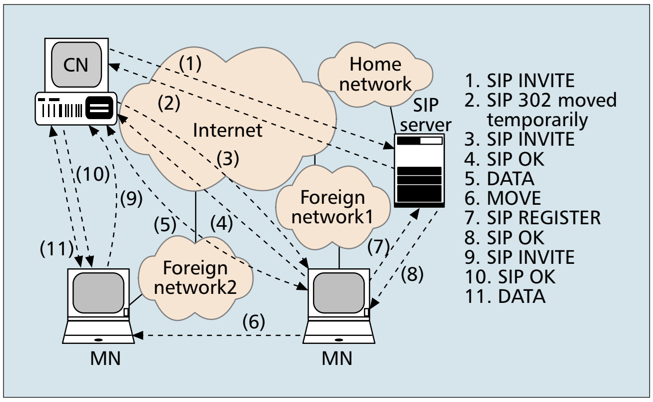


Figure 10. *SIP mobility and its operations.*

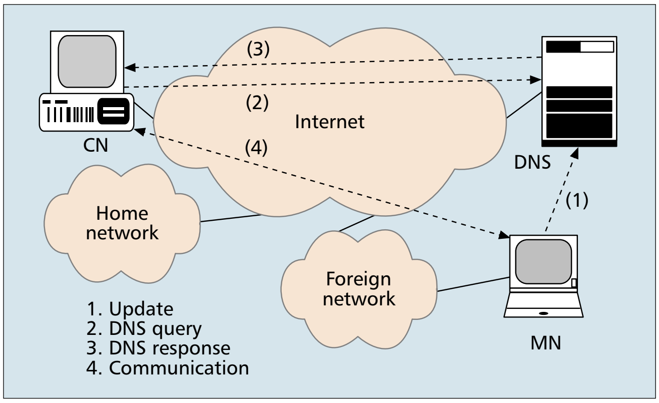


Figure 11. *DDNS location management.*

server. Generally, the user agent is the only element where media and signaling converge. It identifies incoming SIP messages from the user and tracks SIP messages according to user actions. The redirect server receives SIP messages and identifies the current location of the node. The proxy server relays SIP messages. Both the redirect and proxy servers can be used for location management, and they accept location registrations from users. Typically, the SIP server denotes both the redirect and proxy server. The SIP messages defined in SIP include INVITE, ACK, BYE, OPTIONS, CANCEL, REGISTER, etc.

In recent years, there have been several proposals for SIP mobility support [71–76]. The basic idea can be summarized as follows. When the CN initiates a session with the MN, it sends an INVITE message. The SIP server in the home network of the MN has current information about the MN's location and redirects the INVITE message there. Then the normal SIP signaling procedure is performed to establish the session. If the MN accesses a new network and obtains a new IP address via DHCP while the session is ongoing, it will send a RE-INVITE message with updated session description. This maintains the same Call-ID of the existing session but replaces the Contact field of the SIP header with the new IP to inform the CN where it wants to receive future SIP messages, as well as replacing the c field of the Session Description Protocol (SDP) [77] header with the new IP address to redirect the packets to its new location. After receiving the RE-INVITE message, if the CN runs a session over UDP, it will send packets directly to the MN's new IP address. However, if the CN runs a session over TCP, it will send packets to the MN by the tunneling technique [74]. When the MN receives the encapsulated packets, it in turn removes them from IP encapsulation. Similarly, the MN also tunnels packets to the CN. Finally, the MN sends a REGISTER message to the home SIP server to update the location information stored there, so that the new call can be correctly redirected. Figure 10 illustrates SIP mobility and its operations.

Generally, the handover procedure using SIP may introduce handover latency for the signaling messages procedure and overhead for IP encapsulation [71], [78]. To improve SIP mobility performance, Dutta [78] optimizes SIP mobility management by using the intra-domain solution, which limits the movement indication to within the domain to reduce handover latency and minimize packet loss. Kim *et al*. [73] propose a mechanism of Predictive Address Reservation with SIP (PAR-SIP), which reduces handover latency by proactively processing the address allocation and session update using link-layer information of wireless networks.

Security Considerations — In SIP there is support for both authentication and encryption of SIP messages, using either challenge-response or private/public key cryptography.

DDNS — As mentioned earlier, traditional DNS is restricted in the mobile Internet. To resolve the problem, Vixie *et al*. [24] propose a method for dynamic updating of RRs or RRsets from a specified zone by specifying the UPDATE messages. Because most applications ubiquitously resolve FQDN to an IP address at the beginning of communication, DDNS can be considered for location management in the mobile environment where the MN acts as a server and other nodes actively originate communication with the MN.

To locate the MN as it moves to a new network, the MN dynamically registers and updates its FQDN-to-IP entry with the new IP address to DNS servers by sending DNS UPDATE messages. Then whenever the CN wants to communicate with the MN, it will query the DNS sever with the FQDN of the MN, and the DNS sever responds with the current IP address of the MN. Finally, the CN can initiate and establish communication with the MN directly. Figure 11 illustrates the location management of DDNS in the mobile Internet.

Security Considerations — The dynamic UPDATE messages are based on authenticated requests [79] and transactions are used to provide authorization by Secret Key Transaction Authentication for DNS (TSIG) [80] or DNS Request and Transaction Signatures (SIG(0)) [81, 82]. Only authorized sources are allowed to make changes to a zone's contents.

MOBIKE — The Internet Key Exchange version 2 (IKEv2) [83] signaling protocol is part of IPsec. In IPsec the IKE Security Association (SA) and IPsec SA are established between the IP address pair and maintained by IKEv2. The IP address pair is tied to the IKE SA and IPsec SA. Therefore, in the mobile environment, when devices move and IP addresses change during IPsec communication, the existing IKE SA and IPsec SA become invalid and must be rekeyed. Rekeying the SAs for user interaction and the authentication process often occurs too slowly [24]. To deal with these mobility challenges, IKEv2 is being extended by the MOBIKE working group of the IETF for the mobility extension called MOBIKE, which aims to keep the established IKE SA and IPsec SA alive throughout a session so that there is no need to rerun the initial IKEv2 exchange. In this sense MOBIKE can also be regarded as a network-layer solution, although it operates based on a procedure located in the higher layer. MOBIKE provides mechanisms to detect dead peers for connectivity check, and updates the IP address stored with IKE SA and IPsec SA by specifying message exchange of the IP address update notification.

In MOBIKE, multihoming support is integrated by allowing a peer address set to be stored in the IKE SA during initial IKEv2 exchange. In addition, MOBIKE uses Vendor ID

Category	Network layer		Transport layer				A new layer		Application layer		
	MIP	LIN6	TCP	UDP	SCTP	DCCP	HIP	MAST	SIP	DDNS	MOBIKE
Handover	√	√	√	√	√	√	√	√	√		√
Location	√	√					√	√	√	√	
Multihoming					√	√	√	√	√		√
Applications	√	√	√	√	√	√				√	
Security	√	√	√		√	√	√	√	√	√	√

■ Table 1. *Functions of proposed paradigms: a comparison.*

Payload or Notify payload during initial IKEv2 exchange to signal the support for MOBIKE. This ensures that a MOBIKE-capable node knows whether its peer supports MOBIKE or not. When the MN moves to another network, MOBIKE uses the IKEv2 Dead Peer Detection (DPD) mechanism for connectivity testing between address pairs. Once the MN detects a dead address/path, it then sends an authenticated address update notification with a different preferred address. Changing the preferred address also has an impact for IPsec SAs. To allow the IPsec protected data traffic to travel along the same path as the MOBIKE packets, the outer tunnel header addresses ought to be modified according to the preferred address pair. MOBIKE suggests two methods by which the IPsec SAs are changed to use the new address pair. With one method, when the IKE SA address is changed, it automatically moves all IPsec SAs associated with it to the new address pair. The other method requires a separate exchange to move the IPsec SAs separately.

Security Considerations — In MOBIKE all the messages are already authenticated by the IKEv2, so there is no possibility that any attackers would modify the actual contents of the packets. However, the IP addresses in the IP header of the packets are not authenticated, which might cause vulnerability in the remote redirection.

Analysis of Application-Layer Mobility — SIP provides Internet mobility support without any modifications of lower-layer protocols, which are then easily deployed. Because it functions independently of IP addresses, this makes SIP appropriate for use with a heterogeneous network. Nonetheless, it is adverse to real-time applications since considerable handover latency and overload occur with certain procedures, such as the acquisition of DHCP IP address renewal, location registration, and the transmission of the RE-INVITE message from the MN to the CN. In addition,, overload also occurs through the IP encapsulation of TCP connections.

DDNS utilizes existing DNS for location management, which does not require special servers, as with MIP. However, the DNS registration delay needs to be optimized. In addition, as DDNS cannot maintain ongoing communication within the mobile Internet, it is used for location management along with other solutions as candidate approaches.

With MOBIKE, when an IP address changes due to mobility, the IP source and destination address obtained via the configuration payloads within IKEv2 and used inside the IPsec tunnel remain unaffected, i.e., applications do not detect any change at all. However, MOBIKE cannot deal with the rendezvous problem, in which both peers move and obtain the new IP address at the same time without being able to communicate this to one another.

Comparison of Different Paradigms for Internet Mobility Support

In this section we will qualitatively evaluate the mobility solutions on the layer category level summarized above from three aspects of functional requirements, performance metrics, and required changes of existing systems. We would like to emphasize that the comparison is not complete for solutions in question, but the main issues are discussed according to comparative approaches.

Functional Aspects

We first summarize and compare the mobility support solutions based on requirements for handover management, location management, multihoming, applications, and security. Table 1 summarizes how the requirements are supported by the solutions presented above. From the table, we can conclude that none of these solutions fulfill all requirements. The network layer does not yet support multihoming. New layer solutions of HIP must define new a API for the HI, which requires modification of current applications. The transport layer by itself cannot track nodes, so it is short of the location management function. They depend on other layers for location management, as in DDNS, MIP, etc. Application-layer solutions are only appropriate for specific applications, such as SIP for real-time multimedia, DDNS for location management, and MOBIKE for higher-layer protocols and applications using IPsec. For the security issue, most paradigms (except M-UDP) address it to some degree. Some paradigms like MIP, MOBIKE, etc. specified some potential threats. However, the security considerations of some paradigms are still primitive. For example, in the transport layer, the MSCTP suggested using IPsec or TLS to prevent hijacking attacks, but similar to most paradigms developed so far, it does not specify the security mechanism in detail. DCCP selects the value of the Mobility ID feature randomly to protect against attackers, which is not secure enough in fact because it does not specify how to guarantee the randomicity of the value of the Mobility ID feature. Moreover, DCCP also does not provide cryptographic security guarantees.

Performance Aspects

With the above functional comparison it is easy to derive qualitatively the performance of different layer solutions based on the metrics of handover latency, packet loss, signaling overhead, and throughput. The handover mechanism of the network layer suffers from large handover latency and considerable packet loss caused by proxies and a lack of support for multihoming, although many techniques, such as make-before-break or anticipated handovers [84], have been

Category	Network layer		Transport layer				A new layer		Application layer		
	MIP	LIN6	TCP	UDP	SCTP	DCCP	HIP	MAST	SIP	DDNS	MOBIKE
Host	√	√	√	√	√	√	√	√	√	√	√
Routers	√	√									
Third device	√	√	√	√			√	√	√		
TCP/IP layering							√	√			

Table 2. *Required changes to existing systems: a comparison.*

developed to address the problems. Advantages to transport-layer mobility include inherent route optimization, no dependence on the third device, multihoming support, etc., which make seamless handover and minimization of packet loss possible with the ability to pause transmissions in expectation of a mobility-induced temporary disconnection. The new layer, as in HIP, employs RVS/DNS for location management, which might take quite some time to query and update a node's current IP address, by which time it may result in handover latency and packet loss. In the application layer, the resigning of an entire zone of DDNS whenever the IP addresses of one entry change creates a high cost to globally converge the DNS server, which also impacts handover latency and packet loss.

Mobility solutions in the network layer also involve signaling overhead problems caused by tunneling and extension headers, etc. Transport-layer solutions seem to alleviate the problem because they manage mobility by negotiating and switching connections directly between endpoints. In the new layer, the updates of the MN's interface status must be signaled to the CN as in HIP using REA. Similarly, the solutions of the application layer also suffer from signaling overhead for IP address updating or redirecting.

Besides, considering the impact of throughput in the mobile environment, transport-layer mobility improves the performance of throughput effectively by implementing policies that reset congestion control after reattachment. Other layers' solutions by themselves cannot guarantee that the efficiency of transport connections is maintained and cannot handle the degradation of throughput caused by congestion control.

Required Changes to Existing Systems

In order to maintain backward compatibility, the network and protocol infrastructure concern is another important factor in deployment, including required changes to the endpoint and the intermediate router, and the addition of a third entity such as a proxy, an agent, etc., for the network infrastructure, as well as changes to the protocol infrastructure. Table 2 illustrates the required changes compared for different solutions. Network-layer solutions are based on routing mechanisms, so they require changes to the endpoint and router for addressing binding. In addition, they need a third device of agents for packet forwarding and location management. Because transport-layer solutions are based on an end-to-end model, they require no change to intermediate routers, and they are absent from location management by themselves, so there is no deployment of a third device. Therefore, transport-layer solutions require very little infrastructure change. New layer solutions need modifications of the endpoint, and they employ RVS/DNS for location management, so they also need the addition of a third device. In addition, the introduction of a new protocol layer also destroys the traditional TCP/IP infrastructure. Similarly, the application solution of SIP employs a proxy server to relay flows and redirect servers to locate the MN; it needs to add a third device and change the endpoint.

Conclusion

In this article, we analyzed the problems of the traditional TCP/IP stack caused by the mobility of nodes and their wireless links, and we illustrated that many layers of the TCP/IP stack have a negative effect on Internet mobility. We also presented a survey of different mobility support paradigms for the Internet. From our comparisons and the discussion of the advantages and disadvantages of each paradigm, we concluded that current mobility solutions do not solve all general problems related to Internet mobility, and it is hard to dictate which one is most suitable. The individual layer contributes to Internet mobility, and while the technology is important, the market will decide. Link-layer mobility support is fundamental in a mobile Internet, but it constrains within a limited domain and cannot preserve higher-layer connections. Although the network-layer solutions can handle most requirements, they have been slowly deployed because they are ineffective and complex. Transport-layer solutions can perform handover management efficiently, but they lack the ability to perform location management by themselves. New layer solutions voilate the traditional TCP/IP structure, which has been deployed widely, so it is difficult to deploy or modify the current infrastructure of the Internet. On the other hand, application-layer approaches are restricted to specific applications.

To provide an effective solution, keeping in mind the issues of basic functional requirements, performance requirements, and deployment for Internet mobility support, we conclude with the features that need to be satisfied in the mobile Internet:

- Can it efficiently deal with handover, for example, by using the anticipated technique of radio triggers, etc. to detect handover and perform the routing/path update and location registration process in advance.
- Can it handle various mobile scenarios at the endpoints, including the client-server scenario, where the MN only originates the sessions, and the point-to-point scenario, where the sessions may be originated at either one of the endpoints of communicating peers. Will it accomplish this by enhancing location management as with DDNS.
- Will it provide end-to-end mobility and avoid third party entities or tunneling mechanism that increase complexity and reduce mobility.
- Will it take advantage of multihoming, which can simultaneously make for seamless handover and improve mobility with its redundancy and load sharing features.
- Will avoid erroneously triggering congestion control mechanisms, which could arise from the handover of mobility, wireless link characteristics (e.g., lossy and

bursty high BER), and communication path change, in the transport layer, e.g., by extending TCP mobility support features such the LMDR TCP option and enhancing signaling mechanisms between the transport layer and other layers such as the link layer, network layer, etc.

- Will it preferably provide compatibility (and thus allow easier market adoption). That is, it should not impact current applications, network architecture, TCP/IP structure, or add additional entities.
- Will it take into account the security factors in mobile environments. Efficient Internet mobility management is a more challenging issue. In order to satisfy the features recommended above, it needs all the layers' participation in a highly cooperative way. Therefore, we anticipate a multi-layer architecture for advanced mobility support, and we suggest the transport layer as the main candidate, assisted with other layers together, for Internet mobility support.

Acknowledgment

The authors would like to thank Wesley Eddy, Hannes Tschofenig, and Antonio Skarmeta for their insightful comments. In addition, anonymous reviewers, as well as Martin Reisslein, Editor-of-Chief of *IEEE Communication Surveys and Tutorials*, provided constructive suggestions for improvements and clarifications of the article.

References

[1] T. R. Henderson, "Host Mobility for IP Networks: A Comparison," *IEEE Network*, Nov. 2003, pp. 18–26.

[2] W. M. Eddy, "At What Layer Does Mobility Belong?," *IEEE Commun. Mag.*, Oct. 2004, pp. 155–59.

[3] INFOCOM 2005 Mobility Panel, "How Does Mobility Fit into the Internet Layering Scheme?," Mar. 2005, available: http://roland.grc.nasa.gov/_weddy/papers/mobility-panel.html

[4] P. Karn *et al.*, "Advice for Internet Subnetwork Designers," RFC 3819, July 2004.

[5] K. Kuladinithi *et al.*, "Mobility Management for an Integrated Network Platform," *Proc. IEEE MWCN 2002*, pp. 621–25.

[6] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," RFC 2581, Apr. 1999.

[7] H. Elaarag, "Improving TCP Performance over Mobile Networks," *ACM Computing Surveys*, Sept. 2002, pp. 357–74.

[8] Y. Swami, K. Le, and W. Eddy, "Lightweight Mobility Detection and Response (LMDR) Algorithm for TCP," Internet draft (work in progress), draft-swami-tcp-lmdr-06, Aug. 2005.

[9] C. Perkins. "IP Mobility Support for IPv4," RFC 3344, Aug. 2002.

[10] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775, June 2004.

[11] F. Teraoka, M. Ishiyama, and M. Kunishi, "LIN6: A Solution to Multihoming and Mobility in IPv6," Internet draft (work in progress), draft-teraoka-multi6-lin6-00, Dec. 2003.

[12] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," *Proc. ICDCS'05*, Vancouver, Canada, June 1995, pp. 136–43.

[13] R. Yavatkar and N. Bhagawat, "Improving End-to-End Performance of TCP over Mobile Internetworks," *Proc. IEEE WMCSA'94*, Santa Cruz, CA, 1994.

[14] R. Caceres and L. Iftode, "Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments," *IEEE JSAC*, 1995, pp. 850–57.

[15] Z. J. Haas, "Mobile-TCP: An Asymmetric Transport Protocol Design for Mobile Systems," *IEEE ICC'97*, Montreal, Canada, 1997.

[16] D. Funato, K. Yasuda, and H. Tokuda. "TCP-R: TCP mobility support for continuous operation," *Proc. ICNP 1997*, pp. 229–36.

[17] A. C. Snoeren and H. Balakrishnan, "An End-to-End Approach to Host Mobility," *MOBICOM 2000*.

[18] D. A. Maltz and P. Bhagwat, "MSOCKS: An Architecture for Transport Layer Mobility," *INFOCOM 1998*.

[57] K. Brown and S. Singh, "M-UDP: UDP for Mobile Networks," *ACM SIGCOMM Comp. Commun. Rev.*, Oct. 1996, pp. 60–78.

[60] K. Brown and S. Singh, "A Network Architecture for Mobile Computing," *INFOCOM 1996*.

[19] R. Stewart, Q. Xie, and K. Morneault, "Stream Control Transmission Protocol," RFC 2960, Oct. 2000.

[20] E. Kohler, "Datagram Congestion Control Protocol Mobility and Multihoming," Internet draft (work in progress), draft-kohler-dccp-mobility- 00, July 2004.

[21] J. Rosenberg *et al.*, "SIP: Session Initiation Protocol," RFC 3261, June 2002.

[22] P. Vixie *et al.*, "Dynamic Updates in the Domain Name System (DNS UPDATE)," RFC 2136, Apr. 1997.

[23] T. Kivinen and H. Tschofenig, "Design of the MOBIKE Protocol," Internet draft (work in progress), draft-ietf-mobike-design-05, Nov 2005.

[24] P. Eronen, "IKEv2 Mobility and Multihoming Protocol (MOBIKE)," Internet draft (work in progress), draft-ietf-mobike-protocol-07, Dec 2005.

[25] R. Moskowitz and P. Nikander, "Host Identity Protocol Architecture," Internet draft (work in progress), draft-ietf-hip-arch-03, Aug. 2005.

[26] T. Henderson, "End-Host Mobility and Multihoming with the Host Identity Protocol," Internet draft (work in progress), draft-ietf-hip-mm-02, July 2005.

[27] D. Crocker, "Multiple Address Service for Transport (MAST): an Extended Proposal," Internet draft (work in progress), draft-crockermast-proposal-01, Sep. 2003.

[28] E. Gustafsson, A. Jonsson, and C. Perkins, "Mobile IP Regional Registration," Internet draft (work in progress), draft-ietf-mip4-reg-tunnel-00, Nov. 2004.

[29] K. Malki, "Low Latency Handoffs in Mobile IPv4," Internet draft (work in progress), draft-ietf-mobileip-lowlatency-handoffs-v4-11, Oct 2005.

[30] R. Ramjee *et al.*, "HAWAII: A Domain-based Approach for Supporting Mobility in Wide-area Wireless Networks," *ICNP 1999*.

[31] A. Valko, "Cellular IP: A New Approach to Internet Host Mobility," *ACM SIGCOMM Comp. Commun. Rev.*, Jan. 1999, pp. 50–65.

[32] H. Soliman *et al.*, "Hierarchical Mobile IPv6 mobility management (HMIPv6)," RFC 4140, Aug. 2005.

[33] R. Koodli, "Fast Handovers for Mobile IPv6," RFC 4068, July 2005.

[34] H. Y. Jung *et al.*, "Fast Handover for Hierarchical MIPv6 (F-HMIPv6)," Internet draft (work in progress), draft-jungmobileip- fastho-hmipv6-04, June 2004.

[35] C. Perkins, "IP Encapsulation within IP," RFC 2003, Oct. 1996.

[36] B. Aboba, "IAB Considerations for the Split of Identifiers and Locators," Internet draft, Mar. 2004.

[37] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131, Mar. 1997.

[38] D. Plummer, "Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware," STD 37, RFC 826, Nov. 1982.

[39] J. Postel, "Multi-LAN Address Resolution," RFC 925, Oct. 1984.

[40] D. B. Johnson and C. Perkins, "Route Optimization in Mobile IP," Internet draft (work in progress), draft-ietf-mobileip-optim-11, Sep. 2001.

[41] J. Kempf, "Dormant Mode Host Alerting ('IP Paging') Problem Statement," RFC 3132, June 2001.

[42] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," RFC 2104, Feb. 1997.

[43] R. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321, Apr. 1992.

[44] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, Dec. 1998.

[45] R. Droms *et al.*, "IPv6 Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 3315, July 2003.

[46] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration," RFC 2462, Dec. 1998.

[47] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, Nov. 1998.

[49] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)," RFC 2406, Nov. 1998.

[48] S. Kent and R. Atkinson, "IP Authentication Header," RFC 2402, Nov. 1998.

[50] H. Zhu, F. Bao, and R. H. Deng, "Securing Return routability Protocol against Active Attack," VTC 2004-Fall, Los Angles, California, Sept. 2004.

[51] F. Zhao, J. Zhou, and S. Jung, "Improvement on Security and Performance of MIP6 Return Routability Test," Internet draft (work in progress), draft-zhao-mobopts-rr-ext-00, July 2005.

[52] F. Dupont and J.-M. Combes, "Using IPsec between Mobile and Correspondent IPv6 Nodes," Internet draft (work in progress), draft-ietfmip6- cn-ipsec-01, June 2005.

[53] F. Le *et al.*, "Mobile IPv6 and Firewalls: Problem Statement," Internet draft (work in progress), draftietf- mip6-firewalls-03, Oct. 2005.

[54] X. Fu *et al.*, "Enabling Mobile IPv6 in Operational Environments," *Proc. 10th IFIP Int'l. Conf. Pers. Wireless Commun. (PWC 2005)*, Colmar, France, Aug. 2005.

[55] G. Giaretta, J. Kempf, and V. Devarapalli, "Mobile IPv6 Bootstrapping in Split Scenario," Internet draft (work in progress), draft-ietf-mip6- bootstrapping-split-01, Oct. 2005.

[56] W. Haddade and S. Krishnan, "Combining Cryptographically Generated Address and Crypto-Based Identifiers to Secure HMIPv6," Internet draft (work in progress), draft-haddad-mipshop-hmipv6-security-01, Oct. 2005.

[58] M. Leech *et al.*, "SOCKS protocol version 5," RFC 1928, Apr. 1996.

[59] S. Jaiswal and S. Nandi, "Simulation-based Performance Comparison of TCP-variants over Mobile IPv6-based Mobility Management Schemes," *29th Annual IEEE Int'l. Conf. Local Comp. Net.*, Nov. 2004, pp. 284–91.

[61] R. Stewart *et al.*, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration," Internet draft (work in progress), draft-ietf-tsvwg-addip-sctp-13, Oct. 2005.

[62] M. Riegel and M. Tuexen, "Mobile SCTP," Internet draft (work in progress), draft-riegel-tuexen-mobile-sctp-05, July 2005.

[63] T. Dierks and C. Allen, "The TLS Protocol Version 1.0," RFC 2246, Jan. 1999.

[64] A. Jungmaier, E. Rescorla, and M. Tuexen, "Transport Layer Security over Stream Control Transmission Protocol," RFC 3436, Dec. 2002.

[65] D. Eastlake, S. Crocker, and J. Schiller, "Randomness Recommendations for Security," RFC 1750, Dec. 1994.

[66] S. J. Koh and Q. Xie, "Mobile SCTP with Mobile IP for Transport Layer Mobility," Internet draft (work in progress), draft-sjkoh-mobilesctp- mobileip-04, June 2004.

[67] P. Nikander, J. Ylitalo, and J. Wall, "Integrating Security, Mobility, and Multihoming in a HIP Way," *Proc. NDSS'03*, San Diego, CA, Feb. 2003, pp. 87–99.

[68] A. Gulbrandsen, P. Vixie, and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)," RFC 2782, Feb. 2000.

[69] P. Saint-Andre and J. Miller, "Extensible Messaging and Presence Protocol (XMPP): Core," RFC 3920, Oct. 2004.

[70] J. Arkko and P. Nikander, "Weak Authentication: How to Authenticate Unknown Principals without Trusted Parties," *Proc. Security Protocols Wksp. 2002*, Cambridge, UK, Apr. 2002, pp. 5–19.

[71] E. Wedlund and H. Schulzrinne, "Mobility Support using SIP," *Proc. 2nd ACM Int'l. Wksp. Wireless Mobile Multimedia*, Aug. 1999, pp. 76–82.

[72] R. Shacham *et al.*, "Session Initiation Protocol (SIP) Session Mobility," Internet draft (work in progress), draft-shacham-sipping-session-mobility-01, July 2005.

[73] W. Kim *et al.*, "Layer Assisted Mobility Support Using SIP for Real-time Multimedia Communications," *ACM MobiWac 2004.*

[74] F. Vakil, A. Dutta, and J-C. Chen *et al.*, "Supporting Mobility for TCP with SIP," Internet draft (work in progress), draft-itsumo-sippingmobility-tcp-00, June 2001.

[75] F. Vakil, A. Dutta, and J-C. Chen, "Supporting Mobility for Multimedia with SIP," Internet draft (work in progress), draft-itsumo-sippingmobility-multimedia-01, July 2001.

[76] N. Banerjee, S. K. Das, and A. Acharya, "SIP-Based Mobility Architectgure for Next Generation Wireless Networks," *Proc. IEEE Int'l. Conf. Pervasive Computing and Commun. (PerCom 2005)*, Mar. 2005, pp. 181–90.

[77] M. Handley and V. Jacobson, "SDP: Session Description Protocol," RFC 2327, Apr. 1998.

[78] A. Dutta *et al.*, "Implementing a Testbed for Mobile Multimedia," *Proc. GLOBECOM 2001*, pp. 25–29.

[79] B. Wellington, "Secure Domain Name System (DNS) Dynamic," RFC 3007, Nov. 2000.

[80] P. Vixie *et al.*, "Secret Key Transaction Authentication for DNS (TSIG)," RFC 2845, May 2000.

[81] D. Eastlake, "DNS Request and Transaction Signatures (SIG(0)s)," RFC 2931, Sept. 2000.

[82] D. Eastlake, "Domain Name System Security Extensions," RFC 2535, Mar. 1999.

[83] C. Kaufman, "Internet Key Exchange (IKEv2) Protocol," Internet draft (work in progress), draft-ietf-IPsec-ikev2-17, Sept. 2004.

[84] R. Bless *et al.*, "Quality of Service Signaling in Wireless IP-based Mobile Networks," *VTC 2003-Fall*, Orlando, FL, Oct. 2003.

Biographies

Deguang Le (le@cs.uni-goettingen.de) received the B.S. degree in Electronic Engineering from Huaqiao University, Fujian, China, in 1998, and then entered the postgraduate study in Xiamen University, Fujian, China. He was a recipient of the China Council Scholarhsip during 2004–2005, and is currently working toward the Ph.D. degree in the Institute of Informatics, University of Göttingen. His research interests include mobile and wireless communications and network technologies

Xiaoming Fu (fu@cs.uni-goettingen.de) received a Ph.D. degree in Computer Science from Tsinghua University, Beijing, China, in 2000. He was a research member at the Technical University Berlin before joining the University of Göttingen as an assistant professor in 2002. His research interests include network architectures, mobile networks, protocol design, and performance evaluation. He is a co-author of RFC 4094 and aproximately 40 research papers. He is currently an expert of ETSI STFs on Internet protocol testing.

Dieter Hogrefe (hogrefe@cs.uni-goettingen.de) received his Diploma degree and Ph.D. from the University of Hannover, Germany. His research activities are directed toward computer networks and protocol engineering. In these fields he has published several books and numerous papers. After serving for several years in research positions at Siemens, he held professorships at the Universities of Dortmund, Berne, and Luebeck. Since 2002 he has been a Professor of Telematics at the University of Göttingen. He is chairman of the ETSI Technical Committee on Methods for Testing and Specification (MTS).