

Network Working Group
Internet-Draft
Expires: December 30, 2004

W. Eddy
NASA GRC/Verizon FNS
J. Ishac
NASA GRC
M. Atiquzzaman
University of Oklahoma
July 2004

An Architecture for Transport Layer Mobility
draft-eddy-tlmarch-00

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with [RFC 3668](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on December 30, 2004.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

This document describes a generalized architecture for implementing mobility in the transport layer rather than the network layer. In addition, the document discusses the advantages of this approach, the basic mechanisms and interactions required to support transport layer mobility, and examples of how to enable mobility in various transport protocols, using this architecture.

1. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#).

[2.](#) Introduction

This document outlines an architecture that supports transport layer mobility, allowing nodes to remain reachable and retain existing connections while changing their point of attachment to the Internet. This architecture includes common mechanisms to support movement detection and location management, so that these problems do not need to be solved per transport protocol. Instead, the only requirement needed to add mobility support for a given transport protocol is some protocol-specific mechanism to update bindings to particular IP addresses. This architecture places mobility functions on the end hosts, rather than in the network. If use of mobile devices continues to increase, while deployment of network-layer support for mobility does not, a solution such as the one described in this document is potentially beneficial to end-users.

While this document describes an architecture placing the crux of mobility support at the transport layer, several solutions exist at other layers. For example, some wireless links are capable of handling link-layer mobility, allowing for seamless communication over a covered area. At the network layer, Mobile IP [[1](#)] provides mobility for both IPv4 and IPv6 based hosts by using the notion of a home network for a mobile node as an indirection point to the node's current location. Since, Mobile IP attacks a similar problem space to the transport layer mobility architecture this document describes, we compare the two approaches fairly extensively in [Section 6](#).

The architecture described in this document provides support for common services mobile hosts require. Mechanisms for movement detection are suggested to allow a host to know when its connectivity is changing. Movement detection events can then trigger the location updates that are another part of the architecture, and the binding updates which are transport-specific. Location management (via location updates) is the portion of the architecture that lets a host receive new connections, while binding updates are used by individual transport protocols to maintain existing connections during movement. The movement detection and location management functions are fully specified here, but only examples are provided of how binding updates may be implemented in common transport protocols. The exact format of binding updates is heavily protocol-dependent, and best scoped in documents specific to each protocol. This document does not define any wire protocols for location management or movement detection; the transport layer mobility architecture makes use of existing protocols for these features.

For proper operation, many transport protocols need to be aware of mobility anyway (transparent network layer mobility is fundamentally broken), so moving the bulk of a mobility solution into the transport

(where it may be edge-provided rather than network-provided) is a reasonable design [2]. While mobility support at the transport layer may offer significant benefits compared to solutions at other layers, it does not solve the entire mobile/wireless problem space. In particular the following issues are not dealt with fully in this document:

- o Interactions with NAT may affect binding updates in some transports and not others. NAT affects a host's global reachability and thus may break the location management functions of the architecture, but movement detection should be unaffected by the presence of NATs.
- o Security issues for binding updates in individual transports are not discussed. We give some examples of how these may be authenticated in some common transports.
- o Some transport protocols may not easily be able to implement the binding updates required by this architecture. Applications using these protocols may be better served by Mobile IP. In some cases, these transports typically service very short-lived connections, whose lifetimes are likely too short to worry about the macro-mobility this architecture deals with anyways. Such services are typically light on state and can inexpensively retry at the application or user level, so this is not a large consideration.

[Section 3](#) describes how movement detection may be performed within this architecture. [Section 4](#) then presents the architecture's location management mechanism. Examples of how binding updates may be implemented in common transport protocols are contained in [Section 5](#). [Section 6](#) is a comparison of the Mobile IP system to the transport layer mobility architecture. Layer interactions with other mobility architectures such as Mobile IP are outlined in [Section 7](#), and [Section 8](#) contains security considerations.

3. Movement Detection

Movement detection in the transport layer mobility architecture is very similar to movement detection in Mobile IP. The major difference is that with transport layer mobility, in addition to detecting new networks, the mobile host MUST obtain an address on the new network.

With IPv6, a mobile node can use the built-in neighbor discovery [12] mechanism to detect when it has moved onto a new network. IPv6 address auto-configuration [13] and DHCPv6 [14] may then be used to obtain an address and configuration information in the new network. With IPv4, ICMP router discovery [15] messages may be used for mobility detection and DHCP [16] is employed for obtaining an address and configuration.

As in Mobile IP, movement detection and configuration of the mobile host on new networks occurs independently of the transport layer. An operating system implementing the transport layer mobility architecture MUST provide a means for transport protocols to recognize when the process of joining a new network is taking place. For example, this could be accomplished using message passing or by providing a pollable variable. This mechanism should provide distinct values for when a new network has been detected on an interface, and when it has been finally configured. In addition, the operating system MAY provide additional information about networks and interfaces to the transport layer. This might include media type, signal strength, error rate, or other metrics that might help a transport protocol decide if changing its address bindings would be advantageous.

This means of movement detection handles macro-mobility. Handling micro-mobility within networks is largely out of scope, however we discuss some problematic interactions. Micro-mobility based on local registration requires some anchor point in the network, which does not exist as part of the transport layer mobility architecture. Paging-based micro-mobility solutions may also be difficult to incorporate, as DNS provides the location management service for transport layer mobility, and DNS has no notion of paging nodes for address changes. Integrating micro-mobility into the architecture is possible future work that might be handled in a separate protocol, and lack of support for it does not significantly affect the applicability of the architecture in many conceivable cases.

[4.](#) Location Management

Locating a static host in order to access some service on it is generally accomplished using the Domain Name System (DNS), which maps human-meaningful domain names into IP addresses. This mapping is stored in a distributed database. The servers providing this database have become integral portions of the Internet architecture. The dynamic DNS [[17](#)] extension allows for updates to the database to be made without the intervention of administrators. This allows the DNS to be used as a location management system for mobile nodes.

In Mobile IP, location management is a service provided by the home agent (HA). A mobile host is always reachable at a static IP address. There is no need for DNS entries that point to it to be updated when it moves, as the HA will forward all packets directed to the mobile node's address through the Mobile IP tunnel.

With a dynamic DNS system already deployed as an integral portion of the Internet architecture, we may remove the mobile host's requirement of a fixed IP address and HA, and allow it to roam freely, acquiring addresses specific to the network it is currently attached to at any time. Location management is provided by updating its DNS records to reflect its current address. This approach leverages an existing piece of the Internet architecture (the DNS) rather than adding new pieces (home and foreign agents) to it. Additionally, this approach provides route optimization by always pointing requests at a mobile host's current location rather than forwarding them through some indirection architecture.

Securing the dynamic DNS against remote-redirection attacks can be accomplished using signatures to authenticate updates [[18](#)]. For the purpose of mobility, these signatures are most naturally generated by the SIG(0) method [[19](#)] which uses a public key stored in the DNS and a private key stored on the mobile node. The mobile node uses its private key to sign its updates in the specified manner and the public key available to the DNS server is used to verify that the signature on the update was generated by the private key, and thus the update is from the mobile node and not spoofed.

The security of this method relies on the protection of a mobile node's private key. In cases where physical security of a mobile device cannot be taken for granted, encrypting the private key with a short pass phrase may help. This would require the user to input the pass phrase in order to unlock the private key, either each time a DNS update needs to be sent or once for a session which may span several updates (similar to current passphrase agents).

Relying on the DNS for location management opens hosts using the

location management feature of this architecture up to a potential impersonation vulnerability, related to stale DNS information. For example, if a host (A) moves off of a network, but takes some time before reconnecting and updating its DNS record, another host (B) may acquire its old address. Another host (C) wishing to connect to some service on A, will look up A's address using the DNS, and begin communicating with B. If B can impersonate A suitably enough to fool C, then some sensitive information might be divulged. For typical mobile service users today, this scenario is not expected to cause many problems, but may be a concern for some. The problem can be avoided completely if A has notification that it is going offline for some time, then it might either preemptively remove its vulnerable DNS records, or update them to the address of some cooperating host that will provide its services in the interim.

5. Binding Updates

In the transport layer mobility architecture, movement detection and location management are provided outside the transport protocol. Triggers exist between the movement detection algorithm and the transport to initiate the transport's updating of its IP address bindings when an address is obtained in a new network, and likewise a trigger exists between movement detection and location management to update the mobile node's primary address in the DNS. The detail of this process which is not codified in this document, is the exact means a transport protocol uses to perform binding updates. This is dependent on the specific transport protocol, and the ability may be native in some transports, while requiring an extension to others. This section provides some examples of how various groups have implemented binding updates in the SCTP and TCP transport protocols, and an appendix shows an in-depth timing diagram of how the architecture fits together and functions with one particular transport protocol.

Due to SCTP's built-in ability to have transport associations bound to multiple addresses, binding updates in SCTP are a fairly natural extension. The only addition required to the base protocol is implementation of ASCONF chunks [20]. Several proposals have simultaneously appeared from different groups for using these chunks to do binding updates in a transport layer mobility scheme (TraSH [21], mSCTP [22], Cellular SCTP [23]). In particular, we shall describe the operation of the TraSH scheme.

With TraSH, handoffs begin after a mobile node enters a new network and receives a router advertisement and configures an address in the new network. In the architecture described in this document, these functions would occur under the responsibility of the movement detection procedures. TraSH then adds the new IP address into existing SCTP associations in addition to any old addresses in the association. As the mobile node moves further into the coverage area of the new network and discovers it is better connected there, it changes the primary destination address in its SCTP associations. At this point, the mobile node also uses the new address and new network to initiate new associations and updates its location management information. Finally, as connectivity with old networks degrades and is lost, the mobile node deletes the corresponding addresses from its SCTP associations. The architecture describe in this document provides the location management services, and the ASCONF extension to SCTP provides all that is needed for binding updates (new address addition, change of primary address, and removal of old addresses). A timing diagram of the entire procedure may be found in the appendix.

Unlike SCTP, TCP was not originally designed with the idea of being bound to multiple addresses simultaneously, so binding updates for TCP have generally been implemented in a completely different manner. While it would be possible to define TCP options similar to SCTP's ASCONF extension, to allow for the same granularity in adding, prioritizing, and deleting addresses, most TCP mobility work has simply focused on rebinding the currently used address (for example Migrate [24], and TCP-R [25]). This address replacement logically takes place at the same time as the change of primary address with TraSH, at the point where movement detection has observed and configured a new network AND determined that connectivity there is superior to that in the network whose address is currently used in the TCP connection. The old address is completely replaced by the new one in the transport control block and used to initiate new connections at this point.

Properly doing binding updates in TCP requires slightly more work, in that proposals must be careful to ensure that the binding updates are not spoofable, usually by some cryptographic means, and TCP lacks any standardized features that can provide this. For example the Migrate proposal for mobile TCP uses an elliptic curve Diffie-Hellman key exchange to authenticate the binding updates. The key data is negotiated at connection setup time via additional options. Other transports that do not currently include negotiation of cryptographic keys at startup will have to provide their own methods for this, or risk vulnerability to possible remote redirection attacks.

The architecture described in this document leaves the exact format and functioning of binding updates to be specifically defined for each transport protocol that wishes to use the architecture. What is provided, is a common means of movement detection and location updates.

6. Comparison with Mobile IP

Mobile IP [1] has been adopted as a standard for enabling host mobility in the Internet architecture. Mobile IP provides mobility support at the network layer, allowing all transports and applications built on top of it to transparently work on mobile hosts. The current Mobile IP approach, in IPv4, suffers from a number of drawbacks, however.

For Mobile IP to function, a mobile node must have a home agent (HA) in its home network. In addition, foreign agents (FA) must be deployed in each foreign network the mobile node might attach to. These architectural elements are not commonly found in the Internet architecture. Thus, there is a strong reliance on network administrators to both deploy and configure these agents.

The triangle routes set up by Mobile IP add additional latency to the round-trip path that packets follow. The redirection and encapsulation of packets at the HA into a tunnel results in a waste of bandwidth and is especially expensive if the tunnel to the care-of address has high delay. If the network path between the HA and care-of address is disturbed, connectivity between the mobile node and corresponding hosts may be broken, even though the disturbance doesn't affect the direct paths between them. In addition, due to the use of spoofed source addresses by numerous large-scale distributed denial of service attacks and worm propagation, many routers implement ingress filtering [4], which blocks packets that seem to originate from a topologically impossible location. Since mobile nodes use their static home IP addresses as a source addresses in packets they send, these outgoing packets may be dropped by routers in the foreign network.

Triangular routing can be removed using either route optimization or reverse-tunneling. Route optimization procedures have been studied for Mobile IP. However, they are currently not standardized. Furthermore, implementing route optimization will either require support from routers or major changes to end-hosts, including those which are not necessarily mobile. Reverse-tunneling [5] allows mobile nodes to tunnel all network traffic through the HA. The downside to using topologically correct reverse tunnels is that the inefficient side of the triangle route is used in both directions rather than just one, further exacerbating the problems with bandwidth and delay.

Also, stateful firewalls and NATs generally rely on seeing a TCP SYN exchange to instantiate the state that allows them to process packets correctly. If a mobile host with an existing TCP connection changes network attachment points, firewalls and NATs are likely to

drop its packets and break the connection. Since many popular applications use TCP, this is a problem. A similar problem applies to services on the mobile host (using any transport protocol) when it moves behind a NAT.

As a mobile host moves between networks, it may be disconnected for a short period as the host detects a new network and registers the new location with the HA. This could result in packet loss at the higher layers if packets are sent to the old care-of address, and are not buffered or redirected by the network. Smooth handover methods have been investigated, but are not yet standard, and are likely to require additional work on the part of routers.

Since changes in network point of attachment occur transparently when using Mobile IP, transport protocols that keep state about the network path and base their behaviors on that state may not adapt to the new network in an efficient manner. For example, TCP keeps state including an estimate of the round-trip time [8] and the available capacity [7] of a network path. These properties may vary widely between the various points of attachment that a mobile node encounters. Without warning that a transition between networks is taking place, the transport layer can do nothing to help smooth the transition, such as pausing its transmissions, reprobng the network, or sending zero-window advertisements causing peers to do the same.

Mobile IP provides no protection against bogus HAs or FAs. Malicious users in a network may configure home and foreign agents that can be used to either compromise the privacy of a mobile node's data or deny it service. For example, a bogus HA might be set up to monitor a mobile node's traffic even when it was away in a foreign network. A bogus FA could likewise be configured to monitor the traffic of visiting mobile nodes. Either bogus HA or FA advertisements could be used to slow the registration process as a denial of service attack.

In a Mobile IP system, there is no location privacy for mobile nodes. Since mobile nodes always use a fixed IP address which is from their home network, foreign agents can use this address to infer where nodes are from both geographically (since addresses are used for routing) and organizationally (since address blocks are ownership is not private). Since an HA redirects packets to the care-of addresses of mobile nodes, HAs have similar information regarding a mobile node's current location.

Many of the downsides to the Mobile IP approach are artifacts of its place in the protocol stack. Since Mobile IP resides in the network layer, it's features are tied to services provided by the network infrastructure rather than the end hosts. This makes the end-users' capacity for mobility dependent upon the actions of their service

providers. Implementing mobility as a feature of the transport layer can move some power back to the users and offer mitigations to many of the listed drawbacks in the Mobile IP system.

The transport layer mobility architecture described in this document requires no additional network infrastructure aside from what IP networks currently provide.

With mobility anchored in the transport layer, there is implicit route optimization. Packets move directly from end source to end destination, with no indirection. There are no triangle routes.

Topologically correct endpoint addresses are always used with transport layer mobility, which make the system robust to ingress filtering.

In transport layer mobility, transport protocols are explicitly aware of changes in their network attachment status. This allows transports to take the proper action in order to smooth transitions into the new networks, like pausing transmissions during the handover and resetting congestion control state.

Stateful firewalls and NATs may still pose a problem for mobile transport protocols, although potentially less-so than with Mobile IP. For example, TCP can be extended for mobility by setting the SYN bit on the first segment in an ongoing mobile connection which comes from a new address. This pokes a hole for the connection in such devices. Other transport protocols that are affected by NATs may or may not find similar protocol-specific solutions.

The transport layer mobility architecture does not use HAs or FAs and so is immune to bogus agents. The system is however vulnerable to bogus DHCP servers. This problem is shared by the entire networks a bogus DHCP server resides on, however, harming both mobile and static nodes that use DHCP for configuration. In practice, non-authorized DHCP servers can be quickly located and removed from production networks.

Transport layer mobility provides a different amount of location privacy than Mobile IP. Corresponding nodes always know the mobile node's current location with transport layer mobility, although this information is more hidden from the home network, and the identity of the mobile node is more hidden from the foreign network. No active agents are charged with keeping state about a host. As the DNS is used for location management, a node's current location is always known and globally available information. Updating a mobile node's location in DNS is, however, only required if the node is running services that it desires be globally reachable for new sessions. If

this is not the case, or if its services can be located via some other mechanism (for example, via a connection to a peer-to-peer network that is persistent across the node's movements due to transport layer mobility), then a mobile node need not advertise its current location in the DNS. Encrypting binding and location updates would remove any identification of a mobile node's "home". Additionally, dynamic DNS service could be provided by any party - not necessarily in a "home" network - so some degree of anonymousness is possible using transport layer mobility.

Transport layer mobility enables "soft" handovers (not involving the teardown of connectivity at one point before it is re-established elsewhere) to be achieved for nodes having multiple interfaces or even single interfaces using technologies like software radios. This type of activity might also be possible in the Mobile IP framework.

This allows for the decoupling of registration on the new network's interface and ongoing data transfer on the old network's interface, further smoothing a node's handover between networks.

The current specification for Mobile IPv6 [3] mitigates some of the problems discussed for Mobile IPv4, although still lacks some features in comparison with transport layer mobility.

MIPv6 does not require FAs to be deployed in foreign networks. However, HAs are still needed as additional infrastructure in the Internet. The specification supports route optimization as a standard feature. As with transport layer mobility, location privacy from the the corresponding nodes is not provided, and as in version 4, there is no location privacy from the home network nor identity privacy from the foreign network.

MIPv6 can coexist with ingress filtering by using the Home Address Option to ensure topologically correct addressing. MIPv6 must still support packet encapsulation in its bidirectional tunneling mode. Also, HAs are still required to encapsulate data packets at least during the initial stage of the binding update procedure. This coupling of data forwarding and location management raises issues regarding scalability of MIPv6; the HA may still become a network bottleneck.

IPsec is a built-in part of MIPv6, which is used for securing the binding update, while in version 4, separate security mechanisms are used based on statically configured security associations [9].

NAT devices may still exist in an IPv4-IPv6 overlay network, at least in the near future. The incompatibility between NAT and IPsec [10] may create a problem for MIPv6. As in NAT traversal for Mobile IP

version 4 [[11](#)], some special protocol format will need to be defined to allow operation of MIPv6 through NATs.

Higher layer protocols, such as reliable transports like TCP and SCTP, will still have no information about handovers, since all the MIPv6 operation is transparent to the transport layer.

Some proposals in the IETF MIPSHOP group (particularly FMIPv6 and HMIPv6) exist to lessen the latency of the handover and the amount of packet loss that may occur.

7. Layer Interactions

Typically, the division of labor between protocols is clear in the Internet architecture. This is true of the basic requirements for packet service such as routing, addressing, ordering, and reliability. The proper place in the stack for more advanced features like mobility is unclear, because such a feature was not required when the stack layers were initially created. The mobility extension of the architecture defined in this document stretches across several layers. It uses network and application layer protocols for movement detection (IP and DHCP), an application protocol for location management (DNS), and handles binding updates in individual transport protocols (examples given for TCP and SCTP).

The literature is full of diverse approaches to mobility. Particularly, Mobile IP is a proposed standard [1]. While section [Section 6](#) lists a number of reasons why the approach outlined in this document may be preferred over Mobile IP (versions 4 and 6), it is possible to use both approaches simultaneously. This would enable mobility support through Mobile IP for transport protocols that have not been extended to use this document's mobility architecture, and would simultaneously allow a more pleasant mobility experience to transport protocols that do interface with this architecture.

For example, in the current MIPv6 specification, a mobile node can initiate communications to another node using either the mobile node's home address or its current care-of address. By using the care-of address, the mobile node can establish an efficient communication channel to any other node, not simply those that support route optimization and have established the proper network bindings. However, the connection duration is limited to the lifetime of the care-of address. Transport layer mobility can be used as a means to extend the connection over multiple care-of addresses. Thus, transport layer mobility might provide a valuable benefit to applications requiring long-lived connections.

Thus, mobility at both the transport and network layers may be able to co-exist when both residing on the same node. However, complications arise when a mobile node acts as a mobile router (MR), providing network mobility (NEMO) [26][27]. With NEMO, the MR builds a bi-directional tunnel to its home agent, thus providing mobility to hosts and networks attached to it.

As a result, nodes behind a mobile router are:

- o Unaware of any changes in network attachment.
- o No longer the mobile endpoint and thus, do not own a unique care-of address on the foreign network.

- o Subject to all packets being transparently tunneled to the MR's home network.

By masking movement and providing nodes with addresses from the MR's home network, the algorithms outlined in this document for movement detection and location management are no longer applicable for the duration that node stays attached to that MR. Thus, attachment to a mobile router is treated as a single foreign network. Should the node detach from the mobile router and move to a different mobile or foreign network, it behaves normally following the architecture outlined in this document. All this is done without any special requirements by either entity.

The mobile router MAY wish to notify hosts with mobility awareness of changes in attachment, so that transport mobility aware hosts can take appropriate action to reprobe the network paths. However, specification of any such communication does not currently exist. Also, mobile routers MUST tunnel packets that follow the architecture outlined in this document and MUST NOT split those connections as an attempt to provide route optimization.

8. Security Considerations

Security and privacy concerns have been addressed, where relevant, throughout this document. The security of the transport layer mobility architecture is mainly dependent upon the authentication of the location updates and binding updates. The location updates specified use dynamic DNS with the SIG(0) extension for authentication, and are thus vulnerable to any holes this system's design or implementation may possess. The exact format of the binding updates is specific to each transport protocol. The examples presented for SCTP and TCP take different approaches to authenticating the binding updates, each of which has its own security considerations.

9 References

- [1] Perkins, C., "IP Mobility Support for IPv4", [RFC 3344](#), August 2002.
- [2] Eddy, W., "At What Layer Does Mobility Belong?", to appear in IEEE Communications, 2004.
- [3] Johnson, D., Perkins, C. and J. Arkko, "Mobility Support in IPv6", [RFC 3775](#), June 2004.
- [4] Ferguson, D., "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", [RFC 2827](#), May 2000.
- [5] Montenegro, G., "Reverse Tunneling for Mobile IP, Revised", [RFC 3024](#), January 2001.
- [6] Hain, T., "Architectural Implications of NAT", [RFC 2993](#), November 2000.
- [7] Allman, M., Paxson, V. and W. Stevens, "TCP Congestion Control", [RFC 2581](#), April 1999.
- [8] Paxson, V. and M. Allman, "Computing TCP's Retransmission Timer", [RFC 2988](#), November 2000.
- [9] Glass, S., Hiller, T., Jacobs, S. and C. Perkins, "Mobile IP Authentication, Authorization, and Accounting Requirements", [RFC 2977](#), October 2000.
- [10] Aboba, B. and W. Dixon, "IPsec-Network Address Translator (NAT) Compatibility Requirements", [RFC 3715](#), March 2004.

- [11] Levkowitz, H. and S. Vaarala, "Mobile IP Traversal of Network Address Translation (NAT) Devices", [RFC 3519](#), April 2003.
- [12] Narten, T., Nordmark, E. and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", [RFC 2461](#), December 1998.
- [13] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", [RFC 2462](#), December 1998.
- [14] Deering, S., Bound, J., Volz, B., Lemon, T., Perkins, C. and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3315](#), July 2003.
- [15] Deering, S., "ICMP Router Discovery Messages", [RFC 1256](#), September 1991.
- [16] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), March 1997.
- [17] Vixie, P., Thomson, S., Rekhter, Y. and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", [RFC 2136](#), April 1997.
- [18] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", [RFC 3007](#), November 2000.
- [19] Eastlake 3rd, D., "DNS Request and Transaction Signatures (SIG(0)s)", [RFC 2931](#), September 2000.
- [20] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., Rytina, I., Belinchon, M. and P. Conrad, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", September 2003.
- [21] Fu, S., Atiquzzaman, M., Ma, L., Ivancic, W., Lee, Y., Jones, J. and S. Lu, "TraSH: A Transport Layer Seamless Handover for Mobile Networks", University of Oklahoma Technical Report OU-TNRL-04-10, January 2004.
- [22] Koh, S., Lee, M., Riegel, M., Ma, M. and M. Tuexen, "Mobile SCTP for Transport Layer Mobility", February 2004.
- [23] Aydin, I. and C. Shen, "Cellular SCTP: A Transport-Layer Approach to Internet Mobility", October 2003.
- [24] Snoeren, A. and H. Balakrishnan, "An End-to-End Approach to Host Mobility", Proc. of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking,

August 2000.

- [25] Funato, D., Yasuda, K. and H. Tokuda, "TCP-R: TCP Mobility Support for Continuous Operation", International Conference on Network Protocols (ICNP), October 1997.
- [26] Devarapalli, V., Wakikawa, R., Petrescu, A. and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", [draft-ietf-nemo-basic-support-03](#) (work in progress), June 2004.
- [27] Ernst, T. and H-Y. Lach, "Network Mobility Support Terminology", [draft-ietf-nemo-terminology-01](#) (work in progress), February 2004.

Authors' Addresses

Wesley M. Eddy
NASA GRC/Verizon FNS

EMail: weddy@grc.nasa.gov

Joseph Ishac
NASA GRC

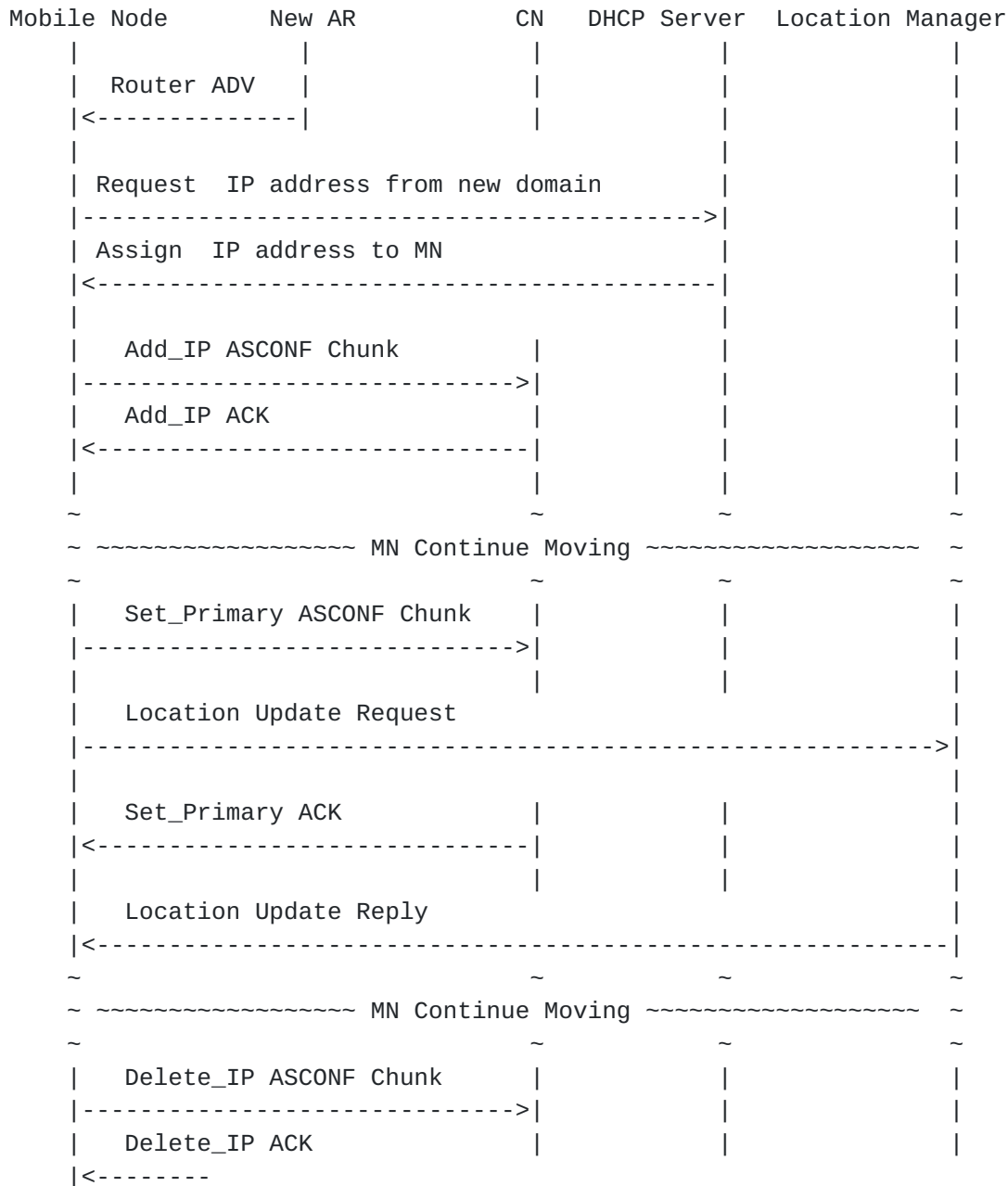
EMail: jishac@grc.nasa.gov

Mohammed Atiquzzaman
University of Oklahoma

EMail: atiq@ou.edu

Appendix A. Mobility Example Using TraSH

In this diagram, "New AR" is the access router in a new network that the Mobile Node is moving into, while maintaining a connection with corresponding node CN, and Location Manager is a dynamic DNS server. This example uses the ASCONF chunks that are part of a proposed SCTP extension, as conceived by the TraSH adaptation of SCTP for mobility support. The steps that TraSH goes through in order to perform movement detection, binding updates, and location updates are shown in order.



Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

