At What Layer Does Mobility Belong?

Wesley M. Eddy
NASA GRC / Verizon FNS
weddy@grc.nasa.gov

Abstract– Internetworking is a complex problem, traditionally tackled by splitting responsibilities between several layers of protocols arranged in a stack. A shortcoming of the current Internet suite's layers is that the responsibilities of individual layers are somewhat ill-defined. The result is that frequently a feature may cause problems for higher layers when it unexpectedly exists lower in the stack, or may be unnecessarily and inefficiently implemented in multiple layers. Mobility is one such feature with no well-defined place in classical protocol stacks. If a link layer hands over between two distinctly administered networks, a network layer protocol will likely need to acquire a new address. Similarly if mobility is implemented at the network layer, such as with Mobile IP, then transport layer protocols must be prepared to deal with a slew of problems (rapid changes in available capacity and delay, the asymmetry of triangle routes, and security policies to name a few). Code for higher-level protocols (above the transport) is less frequently reused, so higher-layer mobility schemes fail to leverage the large base of TCP sockets code. We discuss the various strengths and weaknesses of implementing mobility at three different layers of the protocol stack, concluding that a transport layer mobility scheme is likely to suit today's mobile Internet users best, and that ideally there should be more communication between layers to avoid conflict and inefficiency.

1 Introduction

Since IP networking technology has become cheaply and readily available, and is able to span and connect diverse types of physical networks, it can be a cost-effective solution for networking mobile hosts, with the added benefit of enabling access to the global Internet. A problem with applying the Internet suite in mobile situations, however, is that the protocols were not designed to handle the challenges of mobility. A question then arises of how best to provide mobility support in the Internet suite.

The task of moving data across the Internet requires several subtasks to be performed. Addresses for hosts and services are required, along with some means for mapping human-readable names to protocol addresses. A route must be determined from the source to the destination. The data needs to be divided into chunks and packetized by the sender. The receiver then needs a way to put data chunks back together in order. Both sides must work together to ensure that data is not lost or corrupted and to prevent data from being sent at rates too fast for the receiver or points in the network itself to handle. None of these are particularly easy tasks, and this is far from a complete list of the various functions needed during a single Internet transfer.

The Internet suite's approach to implementing these tasks is to segregate them amongst a number of cooperating protocols arranged in a stack, so that each protocol builds on the services of those below it. For instance, point to point communication is handled by any number of link-layer protocols, while global addressing and routing is handled by IP, and reliability, ordering, and rate control are products of TCP. This leads to a nice division of labor, and a desirable modularity, such that TCP can be substituted for UDP at the transport layer depending on an application's desires, or Ethernet can be substituted for a serial line depending on available host interfaces. The building-block nature of the stack's design [Cla88] has proved to be a useful feature during the Internet's evolution.

Due to the modern Internet suite's loose descriptions of the duties of particular layers, some issues have arisen. Certain services might be provided at multiple layers (for example strong reliability is often present at both the link and transport layers), while others are not key features of any layer (mobility support is one example). Replication of services across multiple layers is harmful in that it may be wasteful of both computational and network resources. For example, link-layer retransmissions for reliability (as opposed to purely coding-based solutions) can be problematic to transport protocol performance, where retransmission-based reliability is traditionally done [FW02]. Services that do not clearly belong at any particular layer are even more problematic, in that their unexpected implementation in lower layers might cause problems at higher layers and be difficult to either detect or disable due to their absence from the scant defined interfaces between layers.

Host mobility is a problem not dealt with at all by the original Internet suite. IP hosts are configured with static addresses. Routing of IP packets is based on their destination IP address prefixes. The doubling of IP addresses as both identifiers and routing aids ties a particular host to the location its address routes to. Several approaches exist for integrating mobile hosts into this framework. A common set of goals among these approaches is that mobility between connection points should neither break existing connections nor render the mobile host unreachable for future connections. The notion of making mobility a feature of end-hosts rather than the core network design is also a key to us, since we do not want to require legacy Internet routers to change, nor is relying the network administration to support mobility desirable to end-users. We define three basic goals for evaluating Internet mobility frameworks:

1. Seamless Transitions - Movement between networks should not result in unacceptable loss of application data and should impose minimal disconnection times on the mobile node. If transport layer connections are broken, there should be a means for resuming them transparently to applications, which should not have to worry about the movement of the hosts they run on. Our emphasis is specifically on not breaking applications that use long-lived connection-oriented transport protocols. Connectionless protocols are typically used more for short-lived transactional applications, like DNS lookup, which should be less easily influenced by mobility.

2. Location Management - A node must remain reachable via some static identifier regardless of its current location. Typically IP addresses have provided such identifiers, although host names in the DNS system also suffice and push the solution up a layer.

3. Infrastructure Free - A mobility solution that is implemented closer to the edges of the network is more desirable than one that requires support from network infrastructure. This allows end-users to move to networks regardless of whether they are specifically designed for mobility or whether network administrators choose to configure and support mobility infrastructure.

In this document, we start at the lower layers of the protocol stack and list some prerequisite support for using IP for mobile networking. We then discuss how our mobility goals can be met, moving to progressively higher layers in each section. Table 1 contains a brief summary of this discussion. A significant point is that no single-layer approach to mobility seems wholly adequate, that support and communication between multiple layers is required.

2 Sub-Network Layer Mobility

Protocol support is required below IP for detecting and joining new networks. At first, a host must be able to determine what types of link are available in a particular area. For example, while it may have interfaces for 802.11, Bluetooth, GPRS, etc, only a subset of these may be available in a particular area. Once this

	Seamless Transitions	Location Management	Required Infrastructure
Network Layer (MIP)	transport layer must deal with losses and path changes	included	deployment of HAs, and router support for fast/smooth handovers
Transport Layer	included	requires external location manager	little or none
Session Layer	included	may be included	little or none

Table 1: Summary of differences between mobility approaches at various layers

basic assessment has been made, a host then needs to attach itself to the network topology. On a wired Ethernet, this is as easy as plugging in, but may be more complex on a wireless network. For example, the DSDV routing scheme [PB94] can operate at the link layer and allow mobile hosts to form complete networks amongst each other in an ad-hoc fashion robust to their individual movements. However, this does not provide them with globally usable addresses for any form of location management, nor are remote hosts on separate network segments able to maintain existing connections when moving across physical networks based on this type of protocol alone. It is merely useful to determine the routing structure for the local network.

Adjacent to the link layer, the IP layer needs to be configured. Protocols like the Dynamic Host Configuration Protocol (DHCP) that allow for dynamic reconfiguration of hosts can provide some low level support for mobility by adapting IP configurations for new networks. For example, advertisements from DHCP servers might be used by a host to infer its movement to a new network and bootstrap its configuration, or Router Advertisements and IPv6 auto-configuration could be used as well. For mobility, there is a clear need for dynamic discovery and configuration of hosts and networking components to bootstrap the IP layer. However, such mechanisms alone cannot fully provide mobility as they reside too low in the stack to meet any of our criteria for mobility support. While they can reconfigure a host for new networks, they do not retain existing applications' connections nor perform any signalling of location changes to maintain global reachability. Low layer solutions are thus a part of any IP mobility scheme, yet are not able to completely constitute one. We assume that adequate mechanisms for determining connectivity are available below IP in all cases.

3 Network Layer Mobility

In the Internet suite, network layer (IP) addresses are administratively assigned and belong to a specific sub-network of the Internet. IP provides globally-usable addresses and is the layer at which routing is performed. Based on maintaining these two functions, there are two distinct approaches that might be taken for providing mobility support:

1. Use host-specific routes, updating them as each host moves.
2. Use routes to sub-networks and add indirection agents to the architecture. Let the indirection agents forward a mobile node's packets from the home network its address belongs to, to its current location.

Since the first option is clearly unscalable to the number of hosts on the Internet today, it can be disregarded. The second approach keeps with the present Internet routing structure and is that taken by the Mobile IP standard [Per02].

The major upside to implementing mobility support in the IP layer is that since it is at the waist of the protocol stack hourglass model, it is the one place where mobility support can benefit *every* higher layer. This is not only beneficial from the standpoint of minimizing reproduction of effort, but also in limiting potential bugs or security concerns.

Mobile IP requires a home agent (HA) in the network a mobile node's address belongs to. Location updates are sent by the mobile node to the HA as its connectivity changes, and the HA forwards any packets that it sees for the mobile node on the home network through an IP tunnel to the mobile node's physical location. No matter where it is physically attached, the mobile node always sends packets using its IP address in the home network. In this way, existing connections are maintained since the address bindings do not change (thanks to the HA), and location management is implicit since the mobile node retains its global address and the HA maps that address to the mobile node's current location.

There would be serious security problems if the location updates were not authentic able as coming from the mobile node. Specifically it would be difficult to prevent remote-redirection or connection hijacking attacks, where a malicious host impersonates another host and takes over as the endpoint of the traffic destined for its victim. Mobile IP uses cryptographically-based authentication of location updates to prevent this. Given the difficulty in correctly and efficiently implementing cryptographic algorithms, and assuring the validity of the authentication scheme, an advantage of Mobile IP is that it minimizes the possibility of multiple different, and possibly flawed, authentication schemes being implemented in separate higher layer mobility protocols.

Despite the advantages from its location in the stack, Mobile IP has several drawbacks.

- Triangle routes are created between the mobile node and remote hosts. While packets go directly *from* the mobile node to a remote host, packets sent *to* the mobile node first go through the HA in the home network, which may be geographically distant from the mobile node's current location. This leads to an increase in delay and can exacerbate problems of path asymmetry which are well known to be troublesome to higher layers like TCP [BPFS02].

- The HA and home network are a point of failure for the mobile node's connectivity, even when it is attached elsewhere. Furthermore, servicing mobile nodes increases the load on the home network in both ingress and egress directions and burdens the HA with sniffing packets, maintaining tunnels to the mobile hosts, and validating location updates.

- A mobile node always uses its static home address in the source field of IP packets it sends. When the mobile node is outside its home network, routers and firewalls may assume the node is spoofing its address as a part of some form of attack, rather than due to mobility, and block its packets. Thus, using Mobile IP in an attempt to maintain connectivity, may instead cause a loss of connectivity. The solution to this problem is to use another tunnel back to the HA for sending packets, however, this uses the less here efficient side of the triangle route in *both* directions and increases the load at the HA.

- The interface between IP and transport protocols is not rich enough for upper layers to be notified when mobility is taking place, and when a transition between networks has occurred. TCP's round-trip time and congestion window estimates may be invalid after a movement across networks and need to be reset. Ideally, the transport could also pause transmissions during the handover to minimize potential losses. However no mechanisms exist for these signals to be exchanged between layers, and so problems like large loss events and spurious retransmission timeouts are bound to occur.

- Several steps are involved in a Mobile IP handover between networks. The mobile node must detect its motion, find a new means of connecting, and update the HA with its current location, presumably performing authentication procedures at each step. While this transition is occurring, the mobile node may be completely disconnected and the network will lose any packets that are destined for the mobile node. There are proposed methods for mitigating this, however these require modifications to Internet routers.

The large number of problems which Mobile IP is prone to make its general usefulness less than compelling, despite its blanket approach providing mobility support to all higher layers. By our three simple evaluation criteria, location management is built in, while seamlessness is only accomplished with a cooperating transport protocol, and infrastructure changes are required. The infrastructure requirement may be the largest problem, as end users suffer in places where network administrators are not willing or able to support Mobile IP.

4 Transport Layer Mobility

To implement mobility at the transport layer, there must first be means for a host to detect new networks that it moves to, and obtain new IP addresses in them. DHCP or Router/Neighbor Discovery and IP autoconfiguration already provide these services and are widely deployed. After new addresses are obtained, the transport layer bindings at remote hosts must be updated for existing connections. Since routing is handled below the transport layer, it must make use of a higher level service (like a name to address mapping) for location management. The readily available dynamic DNS extension may be employed for this purpose.

Lower layer protocols like (e.g. DHCP) can take care of reconfiguring the host for its new network, while higher layer protocols (e.g. DNS) maintain its reachability for new connections. The remaining task that the transport layer must implement is providing a means for dynamic rebinding of a connection's IP address(es). Currently, no standard means exists for this in any commonly deployed transport, although several solutions have been researched.

Transport layer approaches requires more cooperation between layers than the network layer approach, as the location management functions are handled separately. In some sense, this is more of an cross-layer or inter-layer approach than purely a transport layer approach. The transport layer, however, is the only place where protocols may require significant modifications.

Since SCTP has native support for multiple addresses per host, there is a proposed ADDIP extension that would allow dynamic addition and deletion of addresses to and from an existing association. This clearly provides a means of transport layer mobility as several proposals describe (e.g. [FAM$^+$04]). Cellular SCTP [AS03] is a further refinement that provides for smoother handovers by sending duplicate data to a hosts addresses both on the old and new networks during a transition. This lowers the potential for losses. Cellular SCTP also defines a means of location management using SIP.

Since TCP only supports bindings to single IP addresses, for mobility support it needs an extension for changing the bound address of a connection. Several similar mechanisms have been proposed [FYT97, SB00], although none are currently standards. Many other attempts have been made at providing TCP mobility by transparently splitting connections through some form of proxy (for example MSOCKS [MB98]). Such proxy-based approaches tend to suffer from the same drawbacks as Mobile IP, with few of the benefits that direct transport protocol enhancements have. For this reason we do not discuss connection-splitting schemes here, but use transport layer mobility to describe end-to-end binding update based schemes.

It is, both technically and philosophically, mandatory that the transport layer be aware of mobility. A purely network layer scheme that hides mobility from the transport layer is problematic because the transport layer is the Internet suite's layer traditionally tasked with congestion control. Good congestion control requires keeping data on the end-to-end path between hosts. If this path changes due to mobility, then the transport layer needs to be aware so that it may adjust. For instance, its sending rate before a movement between networks may be too fast for the new network path, causing substantial packet loss if the transport does not reinitialize its congestion control state for the new network path. Protocols that use an AIMD scheme similar to TCP, may have the opposite problem, in that a slow start threshold can prevent the transport from efficiently utilizing a new path of greater capacity, where additive increases do not probe the available capacity quickly enough. These problems indicate that for mobility schemes at any layer, congestion-controlled transport protocols require at least some modifications if their connections are to persist across attachment point changes.

Advantages to transport layer mobility include inherent route optimization (triangle routes never occur), no dependence on the concept of a home network or additional infrastructure beyond DHCP and DNS, the possibility of smooth handovers if the mobile node has multiple interfaces, and the ability to pause transmissions in expectation of a mobility-induced temporary disconnection. Since most common applications use TCP, a mobility support extension to TCP has most of the benefit of inheritability that Mobile IP does, and less of the problems since some of Mobile IP's drawbacks are due directly to the transport. Furthermore, since topologically correct source addresses are always used by a transport layer mobility scheme, there is less potential for security mechanisms in the network to inhibit mobile nodes.

One problem with a transport layer approach is the dependence on other layers for location management. For example, if dynamic DNS is employed, it may take quite some time to globally converge to a host's current address, by which time it may be ready to move again. Another problem is that if each individual transport protocol is to implement binding updates, then each one requires an authentication scheme to prevent spoofing. Ensuring the security of each individual authentication scheme could be tedious and error-prone if they are significantly different between transport protocols.

From the standpoint of our three evaluation criteria, transport layer mobility can allow for seamless transitions between networks, by pausing transmissions pro-actively to minimize losses during the handoff, and by implementing policies that reset congestion control after re attachment. Lack of integrated location management is a problem, but can be solved by a higher layer. A major advantage to the transport layer approach is that it requires very little infrastructure. At most, facilities like DHCP and dynamic DNS are required, but since these are already a well-deployed part of the infrastructure, this represents no additional requirement for change.

5 Session Layer Mobility

Although rarely used by present Internet applications, the concept of a session layer can leverage the same advantages of a transport layer mobility scheme without requiring the difficulty of a rebinding mechanism to be implemented in an already well-established transport protocol. For example, after a new address is obtained, the session layer may simply initiate new transport connections to replace the existing ones. Alternatively, if there were transport layer protocols that had mobility support in the form of dynamic address rebinding, the session layer might still be better poised as a place to monitor movement and trigger binding updates. Such division of labor could ease implementation, and place updating the location management system at a more logical level, as location management is a system or service-wide function. It could be troublesome if individual transport connections were able to change system-wide location information, as some inconsistencies might arise depending on the metrics different connections could use when comparing signal strengths, error rates, etc and preparing for a handover between networks.

A purely session layer approach has basically the same advantages of a transport layer scheme, with the added benefit of not requiring difficult to make changes in mature transport protocols. The key disadvantage to a session layer approach is that the idea of using session layers has not been popular amongst application developers who may just as easily (with a similar amount of code) create and use transport layer connections, for example. The lack of any real deployment of session layers among common Internet applications prevents this solution from having much impact.

Since the session layer is traditionally unused, its pre-existing functions and responsibilities are much less than those of a reliable transport protocol like TCP. The lower amount of complexity in the session layer makes it easier to add features there. For example, SLM [LLIS99] is a session layer based mobility scheme that unlike many transport layer mobility methods, provides location management. Such a session layer might be effectively paired with a mobility-enhanced transport protocol to divide the labor of maintaining connections and maintaining reachability most logically.

By our three evaluation criteria, session layer mobility approaches seem quite favorable. At the session layer, transitions can be made as smooth as at the transport layer, perhaps even slightly more so, if both sides can preset some state before that transition that will allow them to resume more quickly. The required infrastructure for session layer mobility is also as low as that of transport layer mobility schemes. A further advantage is that some complexity might be removed from the required changes to already complex transport protocols, by commonalizing location management services for multiple transport connections.

6 Conclusions

The host mobility problem may be attacked from many layers. Link layer support is mandatory in any case, but can do very little to either preserve higher layer connections or provide location management when movement is across administrative domains. The common network layer solution is Mobile IP, which while effective, has several limitations in practice. Most of Mobile IP's problems can be tackled by a higher transport or session layer approach. Due to cultural unacceptance of a session layer, the transport layer approaches to mobility are likely the strongest, despite requiring modifications to well-established protocols like TCP. By deploying mobility-enabled TCP implementations, applications that use TCP may transparently gain mobility support just as they do with Mobile IP, with less potential problems. Although the question of what layer mobility should properly be provided at is largely an open question, we suggest the transport layer as the strongest candidate and have presented common strengths and weaknesses of approaches at various levels.

References

[AS03] I. Aydin and C. Shen. Cellular SCTP: A Transport-Layer Approach to Internet Mobility, October 2003. Internet Draft (work in progress).

[BPFS02] H. Balakrishnan, V. N. Padmanabhan, G. Fairhurst, and M. Sooriyabandara. TCP Performance Implications of Network Path Asymmetry , September 2002. RFC 3449.

[Cla88] David D. Clark. The Design Philosophy of the DARPA Internet Protocols. *ACM Computer Communications Review*, 18(4), August 1988.

[FAM$^+$04] S. Fu, M. Atiquzzaman, L. Ma, W. Ivancic, Y. Lee, J. Jones, and S. Lu. TraSH: A Transport Layer Seamless Handover for Mobile Networks, January 2004. University of Oklahoma Technical Report OU-TNRL-04-10.

[FW02] G. Fairhurst and L. Wood. Advice to Link Designers on Link Automatic Repeat ReQuest (ARQ), August 2002. RFC 3366.

[FYT97] Daichi Funato, Kinuko Yasuda, and Hideyuki Tokuda. TCP-R: TCP Mobility Support for Continuous Operation. In *IEEE International Conference on Network Protocols*, October 1997.

[LLIS99] B. Landfeldt, T. Larsson, Y. Ismailov, and A. Seneviratne. SLM, A Framework for Session Layer Mobility Management. In *Proc. IEEE ICCCN*, October 1999.

[MB98] David A. Maltz and Pravin Bhagwat. MSOCKS: An Architecture for Transport Layer Mobility. In *IEEE INFOCOM*, 1998.

[PB94] Charles Perkins and Pravin Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *ACM Computer Communications Review*, 24(4), October 1994.

[Per02] C. Perkins. IP Mobility Support for IPv4, January 2002. RFC 3220.

[SB00] Alex C. Snoeren and Hari Balakrishnan. An End-to-End Approach to Host Mobility. In *Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, August 2000.