Fast Handovers and Context Transfers in Mobile Networks

Rajeev Koodli
Nokia Research Center
313 Fairchild Drive
Mountain View, CA 94043
rajeev@iprg.nokia.com

Charles E. Perkins
Nokia Research Center
313 Fairchild Drive
Mountain View, CA 94043
charliep@iprg.nokia.com

ABSTRACT

We describe recent work enabling fast handovers and context transfer between access routers offering Internet connectivity for mobile (often wireless) nodes. We present our framework for engineering general context transfer solutions, and a protocol which uses the framework to provide a simple yet general mechanism for carrying out context transfers during handovers. Since our mechanism operates at the network level, we expect that it will be the most expedient way to provide for seamless handovers between heterogeneous networks. We report our results which show that fast handovers with context transfer at the network layer can support uninterrupted voice over IP (VoIP). Thus, our context transfer framework will catalyze the arrival of a unified wireless telecommunications network, with voice and data connectivity anytime, anywhere. Towards that end, we describe how our results and context transfer framework relate to other work within the IETF.

Keywords

context transfer, mobile IP, mobile network, IPv6, fast handover

1. INTRODUCTION AND MOTIVATION

The idea of a unified telecommunications network is not particularly new. If we are to engineer such a unified network that is also able to deliver acceptable voice quality, major hurdles remain to be solved. In order to deliver an acceptable voice call, the telephone network has developed a number of sophisticated protocols and algorithms to efficiently utilize available transmission resources; collectively, these protocols and algorithms work together in what has been known as a *circuit-switched* network. For our purposes, the most important characteristic of circuit-switched networks is that the connections between the endpoints deterministically offer enough capacity to guarantee transmission of digitized voice. The connections are reserved for a particular voice call, and the capacity is not available for other uses, regardless of whether there is any conversation occurring. This may be viewed as guaranteeing a particular quality of service (QoS).

Often, there has been the implicit assumption that the major difference between the Internet and the telephone network springs from the difficulty to guarantee any particular level of QoS to an Internet application. Thus, Voice over IP (VoIP) applications have been vulnerable to Internet congestion and unpredictable delays. This often leads to unacceptable quality for voice calls, and thus many people believe that handling voice calls over the Internet cannot be scalably achieved without proven QoS protocols. Efforts along these lines (notably Integrated Services [25] and Differentiated Services [4]), while promising, have yet to achieve a wide enough deployed base to make the difference. Nevertheless, we believe that some combination of DiffServ for managing long-haul contractual services, and local management (e.g., using IntServ) for access of provisioned resources will eventually prevail.

The foregoing factors would already present a formidable set of obstacles inhibiting the development of the converged data and voice networks of the future. An even larger set of obstacles has come into play over the last few years with the deployment of hundreds of millions of radio-frequency cellular telephones. While not yet perfect, the voice quality attainable with such mobile phones is satisfactory enough to appeal to a very broad market. A handy telephone that goes with us, instead of us having to go to it, offers extraordinary convenience and can even save lives, especially during emergencies. The widespread acceptance of cellular telephony has caused a huge expansion in wireless infrastructure to support the necessary circuit-switched voice calls, with tightly engineered radio access networks (RANs) and base stations becoming a fact of life. Currently, however, the infrastructure and RANs do not utilize Internet protocols in any important way to establish and maintain network connections for voice calls.

This development complicates the network architecture, making it difficult to see just how or when the envisioned network convergence may occur, and yet most engineers assume that it eventually will happen. A major piece of the puzzle involves mobility management. We believe that the converged network will use Mobile IP for this purpose, suitably instrumented with interfaces into the existing subscriber profile management and authorization processes. In this article, we do not describe the latter operations in any detail (but see [3]), preferring to focus instead on ways to solve the problems of resource management caused by local mobility in access networks. We propose localizing the signaling and more careful security for fast handovers, increasing performance by context feature management.

All services of interest to us depend upon *Internet* connectiv-

ity. Thus, re-establishing routing paths to the Internet (i.e., IP connectivity) in the presence of user mobility becomes a crucial problem. Once a mobile node establishes basic IP network connectivity, we can also take steps to make sure that transport protocols, such as TCP and RTP, do not suffer performance degradation due to mobility. Our goal is to enable the network state information relevant to the mobile node to follow it. We assume that the network access nodes share security associations, so that the necessary signals between them will not be vulnerable to intervention by malicious third parties.

In this article, we describe fast network connectivity establishment and (subsequently) context transfer at the network protocol layer during handovers. This area of investigation has received significant attention, motivated by the envisioned convergence of telephone networks with the Internet. In order to understand the context transfer framework, we must first describe the underlying protocol mechanism for fast handovers. Since the motivation for transferring context is improved performance, it is only natural that context transfer and fast handover should be designed to operate well together.

The presentation in this paper is a reflection of our experiences (in the past 18 months of research and development) involving IPv6 mobile networking, as well as topics (particularly fast handovers in `[mobile-ip]` [17]) that appear to be gaining popularity in the IETF. We make use of our previous contributions to IETF on context transfers [14, 15] which we first presented to the `[mobile-ip]` and `[rohc]` [20] working groups, and our recent contributions [13, 24] to the `[seamoby]` [21] working group. We also make use of our work on fast handovers in [12], and participation in the `[mobile-ip]` design team effort on this topic [9].

2. PROBLEM DESCRIPTION

In this section, we provide a detailed problem statement. We begin with our reference model.

We assume that a Mobile Node (MN) attaches to its access network, typically through a wireless link, using an access point or a base station. The access point provides link connectivity, whereas an Access Router is what the mobile node perceives at the IP layer. Although the access point and access router are separable functional entities, they both could be integrated in a single physical device. Since we are interested in IP layer mobility, we focus on the Access Router which provides IP connectivity to the mobile node. We use Figure 1 as a reference in the following discussion. We assume that all nodes are addressable using IPv6, in conformance with recent standardization activity from the 3GPP [1] and 3GPP2 [2] standards development organizations.

Once a Mobile Node powers up, it first establishes link connectivity. It then performs Neighbor Discovery operations in order to establish IP connectivity [18]. These operations involve configuring a link-local address, which allows on-link communication only [7], using a well-known network prefix (`FE80::`) and the MN's interface identifier. The MN then has to ensure that its new link-local address is unique. In IPv6, address auto-configuration allows a node to configure

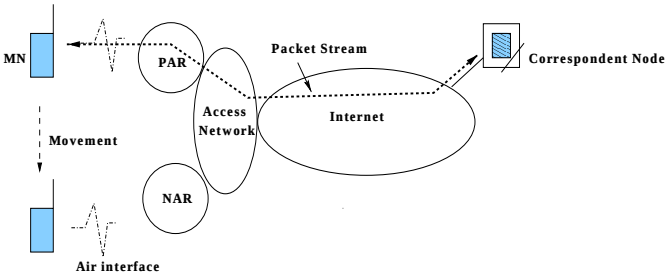

Figure 1: Mobility Reference Diagram

its IP address without requiring any stateful inspection on the network side [23]. This process, which simplifies address assignment, does however require each node to ensure the uniqueness of its address by performing Duplicate Address Detection (DAD) [23]. After configuring its link-local IP address, the MN performs router discovery, allowing it to identify its default router as well as to create a globally routable IP address. During this process, a router may require that the MN present its credentials for network access authentication and authorization [3]. After these operations, an IPv6 node (including a MN) is capable of sending and receiving IP packets using its new IP addresses.

Once a mobile node establishes IP connectivity, we assume that it runs some applications and establishes itself as an IPv6 endpoint. VoIP between the mobile node and correspondent node is one example of a useful application for which seamless connectivity is crucial. Other examples of interesting applications include streaming media and instant messaging. An application such as VoIP typically requires some Quality of Service (QoS) support from the network. This functionality, which reserves desirable *forwarding treatment* to certain distinguished packet streams, necessitates establishing some state or *context* for the particular packet stream. This context, for example, can include packet classification, packet metering and packet marking parameters [4]. The QoS state may also include an authorization token from a policy server to use network resources. In addition to QoS, a low-speed wireless link makes it highly advantageous to implement IP and transport header compression functionality, in order to use the expensive and limited bandwidth resources efficiently. The header compression process is stateful; a compressor and a decompressor both maintain reference state, with respect to which they communicate the differences in header fields. Thus, it also requires establishing and maintaining context on the access router. These examples and others show that context establishment is necessary for offering network features to a Mobile Node, once basic connectivity is established. Except for the network access authorization context (established when a MN first attaches to the network), the remaining contexts are typically specific to each packet stream.

Now, suppose that the MN undergoes handover from its current Access Router to another. We do not specify the means by which the MN or the AR determine the target router to which the MN attaches next. These mechanisms are dependent upon specific link layer characteristics as well as possibly the methods used for target router selection. For our

purposes here, we assume that the Previous Access Router (PAR) (i.e., the router to which the MN is attached to prior to handover) has the knowledge of the New Access Router (NAR) to which the MN will attach next. During the handover process, the MN has to relinquish its current link and establish a new one. After it regains its IP connectivity, the mobile node then allows its applications to use network features (QoS, header compression) again. This is our problem domain. We specifically intend to address the following two problems.

- how quickly the Mobile Node can *send* and *receive* IP packets subsequent to handover. We call this the *fast handover* problem.
- once the MN establishes IP connectivity, how to ensure that the disruption caused by the handover for network features is minimized. We call this the *context transfer* problem. It can also be characterized as re-establishing some connection state at the "last-hop" router (an edge router in the access network), typically over bandwidth-constrained wireless links.

The fast handover problem addresses basic routing of IP packets whereas the context transfer problem addresses the problem of making the handover process "seamless". We tackle both of these problems in the following sections.

3. ENABLING FAST HANDOVERS

There are two design points in the fast handover problem. First, the latency due to IP address acquisition and configuration subsequent to handover has to be reduced. This design point addresses the problem of how quickly the MN can transmit IP packets. We call this *connectivity latency* improvement. Second, the latency in forwarding IP packets to the MN's new IP address must be reduced. Observe that packets continue to arrive at the MN's previous IP address until the correspondent node or a regional mobility agent is notified, e.g., through a Binding Update [10]. These packets have to be re-routed to the MN's new IP address until the notification becomes effective. Even though the base Mobile IPv6 protocol allows for a MN to send a Binding Update to its previous router, it does not specifically address the latency involved in forwarding packets to the MN's new IP address. We call this design point *reception latency* improvement. A timeline illustrating the connectivity latency and the reception latency is is shown in Figure 2.

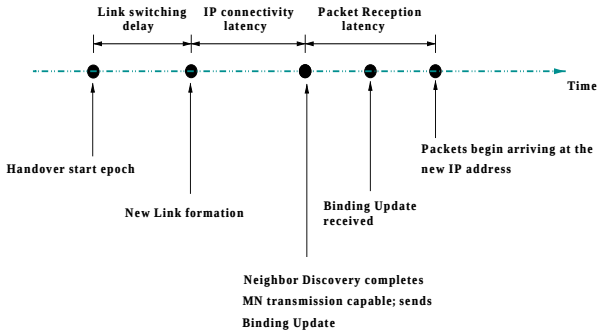

Figure 2: Handover Delay Timeline

3.1 Improving the Connectivity Latency

A MN needs to know its router's advertised network prefix in order to configure a globally unique IP address. With IPv6 stateless address auto-configuration, the MN also needs to perform Duplicate Address Detection to ensure address uniqueness. Both of these operations, which we called Neighbor Discovery procedures earlier, contribute to the connectivity latency. In order to illustrate the effect of these operations, we consider a cellular link with a Round-Trip Time (RTT) of 120 ms [5]. Performing DAD and router discovery in a sequence would mean a latency of 240 ms. Even if they could be made to overlap to some extent, we anticipate a minimum latency of one RTT and a typical latency of 1.5 RTT. A latency of 180 ms corresponds to 9 voice packets sampled at 20 ms intervals. Since the voice application continues to execute regardless of handover, these packets would be potentially discarded due to play-out deadline violations. Hence, it is crucial to minimize this connectivity latency.

The key idea behind minimizing the connectivity latency is to allow a MN to form a new IP address even before it attaches to its New Access Router. This means the MN would know its default access router and a prospective IP address prior to leaving PAR. The following sequence of events take place in order to facilitate this. We use Figure 3 for illustration, and messages outlined in [9].

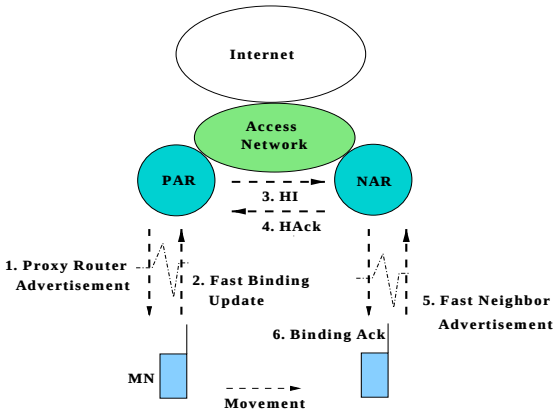

Figure 3: Fast Handover Signaling

1. Handover indication to the MN

   A handover trigger forces the PAR to send an IP message to the MN providing details of the new access router. The handover trigger may be network-generated, meaning that a handover control entity functioning in conjunction with the MN provides an indication to the PAR that the MN needs to undergo handover. The handover trigger may also arrive from the MN itself. The exact process that initiates the trigger, either from the MN or from the network, is dependent on the specific link layer. We provide a illustrative example and description below, while still emphasizing the need for generality at the network (IP) layer.

   Typically, a MN gets continuous signal strength measurements for its Access Point. When the signal strength from a target AP is better than the one from its cur-

rent AP, a MN may decide to switch its link connection. A MN usually makes this decision in conjunction with some entity on the network side; however, in some cases it may not engage with a network entity at all. During this *link probing* operation, a network control entity (such as a Radio Network Controller (RNC) in cellular systems) or the MN itself can determine the IP address of the target AR. This could be achieved on the network side, e.g., through a table lookup that matches the link-layer identifier (available in the link probing messages) of the target AR with its IP address. Another method is to embed the IP address in the link probing messages. This is preferable to the previous method, however, this would increase the message size and hence may not be feasible in bandwidth-limited systems. Yet another approach is to perform a dynamic (off-link) lookup of IP address of the target given the link layer identifier. This approach, similar to reverse-ARP [8], however requires extensions to or Neighbor Discovery [18].

A network control entity may use any suitable message to notify PAR about initiating a handover, or PAR may make this decision on its own. Alternatively, The MN may use a Proxy Router Solicitation message to request handover [9]. In response to either of these two messages, PAR sends a Proxy Router Advertisement message to the MN. This is essentially a Router Advertisement message from the target router to which the MN will attach to next. This message allows a MN to configure a potential IP address it will use next. Note however, that this address is not confirmed to be without conflict, since the New Access Router does not yet have information about the MN's new IP address.

2. Handover Indication to the New Access Router

   While the MN is configuring its IP address, PAR informs NAR regarding an impending handover. In the Handover Initiate (HI) message, PAR negotiates the validity of the new address with NAR by providing either directly the new IP address of the MN or the link-layer address of the MN which the NAR can use to compute the new IP address of the MN. In any case, PAR includes the previous IP address of the MN so that if the new IP address acquisition fails, a fallback host route with NAR can be established in order to route packets destined to the previous IP address of the MN.

3. New IP address negotiation

   In response to the HI message, the NAR has to ensure that it can grant the new IP address to the MN. One way to ensure the uniqueness of the new IP address is to perform DAD on behalf of the MN. Clearly, this has performance considerations, since DAD would involve messages over a potentially slow link. We propose that the NAR use its Neighbor Cache to avoid potential conflicts with the requested IP address. Note that the an access router's Neighbor Cache should contain the entries for all the currently reachable nodes on-link as well as for those nodes off-link for which the router is proxying IP addresses (such as the home address of a MN). It should, for all practical purposes, thus suffice to perform a local cache lookup in order to determine whether or not the NAR can support the new IP address. Such a lookup should be considerably faster than performing DAD over a slow link.

   If the NAR determines that granting the requested IP address would not cause a conflict, it then begins to proxy the address in order to avoid potential conflicts in future. It does this by sending a proxy Neighbor Advertisement message whenever necessary. When it is not permissible to use the new IP address, the NAR creates a host-specific (unaggregated) entry in its routing table to allow the use of previous IP address for the MN on its link, perhaps until the MN acquires a topologically correct IP address. It then sends a Handover Acknowledge (HAck) message back to the PAR in which it includes the result of negotiating the new IP address. Whether the new IP address can be granted or not, the sequence of HI and HAck messages creates the forwarding path from PAR to NAR, which we describe in section 3.2.

In summary, improving the connectivity latency includes messages to indicate handover to the MN, allowing it to form a new IP address, and negotiations between the access routers to support that new IP address. In the process, the access routers also set up a suitable forwarding path for the packets destined to the MN's previous IP address.

We briefly mention that if stateful address configuration is used for address assignment, the new access router (NAR) may have to send the newly allocated IPv6 address back to PAR in the HAck message. Even so, the above set of protocol messages are still valid. However, in such a case, the PAR delays the Proxy Router Advertisement message until the HI and HAck message sequence is completed. Detailed descriptions, including various error scenarios, are provided in [9].

3.2 Improving the Reception Latency

The forwarding path from PAR to NAR to MN must not be enabled until the MN explicitly authorizes PAR to do so. The MN sends this indication using a *"Fast"* Mobile IPv6 Binding Update message [10] to the PAR, only after receiving which the PAR must start forwarding the packets on the tunnel established earlier using the HI and HAck message sequence. The MN needs to receive a Binding Acknowledgment (BAck) before it is permitted to use the new IP address. Nevertheless, the MN must also be allowed, depending on link conditions, to leave PAR immediately after sending the Binding Update, and we also wish to maximize the overlap of IP handover signaling with the link establishment delay. This overlap processing has to be allowed even when the MN cannot maintain its current link with PAR after sending the Binding Update; thus, we propose that the mobile node be allowed to initiate the process for acquiring a new link with NAR without first waiting to receive a BAck. We propose buffering the BAck (or any packet meant for the previous IP address) at NAR for delivery once the MN establishes new link connectivity.

The MN at least needs to receive the network prefix of NAR before it can prepare for fast handover. Even if the MN fails to send a BU for any reason, e.g., due to lost connectivity,

it should still be able to benefit from the information it has gathered. Note that this case is similar to the MN not receiving the BAck prior to leaving the PAR[1]. In any case, the MN announces its presence on the new link with a Neighbor Advertisement message [9]. Since the MN does not yet know if its new IP address is valid, it has to ensure its uniqueness prior to using that IP address. This is the purpose of the new *Fast* Neighbor Advertisement message, in which the MN presents its both old and new addresses, and the NAR responds back with a message indicating which IP address to use. In order to improve the reception latency, IP packets are forwarded to the mobile node as soon as it establishes a link presence. Enabling the forwarding path from PAR to NAR accomplishes this; however, since packets may arrive before the MN establishes a new link, those packets should be buffered at the NAR. Once the MN announces its presence, e.g., through a Fast Neighbor advertisement message, the NAR starts forwarding buffered packets (including the Binding Acknowledgment) to the MN.

Observe that the sequence of Fast Neighbor Advertisement message and a corresponding reply message again incurs a delay equivalent to an RTT over the air interface. During this time, a VoIP application may be generating packets which risk being discarded due to isochronous timing requirements[2]. We propose the following enhancement which could eliminate the need for an additional RTT involving Fast Neighbor Advertisement message and an associated reply.

After establishing a new link with NAR, the MN sends packets directly to NAR. However, observe that the IP source address of the MN has not been positively confirmed[3] in these packets. In this case, the MN maintains a TENTATIVE state variable in its Neighbor Discovery (ND) cache entry for its default router [18]. This state variable is set to ON prior to sending any IP packets on the new link with NAR, and is set to OFF upon receiving positive confirmation from NAR. When TENTATIVE == ERROR, as would happen if the access router denies the availability of the newly chosen IP address, the MN must not send any packets using that new IP address.

If the application running on the mobile node tries to transmit any IP packets, for transmission when TENTATIVE is ON, the ND module encapsulates the IP packet in a new ND message type called "Neighbor Cache Validation" and does normal transmission. This message type must include the MN's new tentative IP address and its link-layer address. When the access router (NAR) receives this ND packet, its ND module treats the new message type according to the following rules.

- In the very rare case when a conflict of IP address is noticed, (i.e., for the IP address selected for use by the mobile node, the access router has a neighbor cache entry with different link layer address), NAR generates an error message, and discards the packet. When sending the error message, the NAR may indicate to the MN to use its old address (assuming it would have received the HI message) or to initiate a address configuration procedure [23]. The NAR may forward packets addressed to the mobile node's previous address. Once it receives an error message, the MN must not send any more IP packets using the new CoA. It must set the TENTATIVE state variable to ERROR. Once the MN successfully determines the IP address to use, the MN then sets TENTATIVE to OFF, allowing normal operation.
- NAR is proxying the address for the MN. In this case, the NAR overrides its existing cache entry with the information supplied by the MN. The NAR may generate a unicast Neighbor Advertisement message to the MN confirming its IP address. Alternatively, it may simply forward any queued packets as an indication to confirm the new address. The NAR then removes the encapsulation and forwards the inner IP packet towards destination IP address specified in the inner IP packet. Once the MN receives confirmation that its IP address is valid, it sets the TENTATIVE state variable to OFF, and proceeds to behave like any other node. The MN *may* send a Neighbor Advertisement message to all nodes on-link to advertise its presence.

Note that NAR is never proxying the address for some other node. The relevant node would not use the Neighbor Cache Validation message.

Link-layer triggers have been proposed as a possible means of improving the RTT latency [11]. For this, the NAR has to be able to determine the presence of the MN using appropriate link-layer messages. This mechanism is specific to each link connecting the MN to its AR. If there are no packets at the NAR, the MN still has to determine which IP address to use. Thus, link-layer triggers do not eliminate the RTT latency when there are no packets waiting at the NAR to deliver to the MN. In the meanwhile, as we mentioned earlier, an application on the MN may continue to generate IP packets.

Together, the above solutions for connectivity and reception latency design points allow a MN to quickly establish IP connectivity at the New Access Router (NAR). Figure 4 shows the handover delay with the proposed improvements. As soon as the new link is established (see Section 3.1), the MN can start sending *and receive* IP packets immediately. This is the essence of fast IP handover. The MN may then send a normal Binding Update [10] to its mobility agent and the correspondent node(s) so that they could begin packet transmission directly to the MN's new IP address. In the following section, we describe how the handover support can be extended in order to facilitate *seamless* operation of transport layer protocols.

4. CONTEXT TRANSFERS

Recall that a MN typically establishes IP and sub-IP state after it establishes connectivity. This state includes network

[1]the only difference is that when the BU is actually lost, the forwarding path from PAR to NAR is not enabled

[2]We observe that VoIP packets can be lost due to link switching delay which is inevitable. For IP layer handovers, the design criterion is to minimize or even eliminate the packet loss due to IP layer handover messages

[3]The likelihood of choosing an address that is already in use, however, is very small; 1 in 2^{64}.

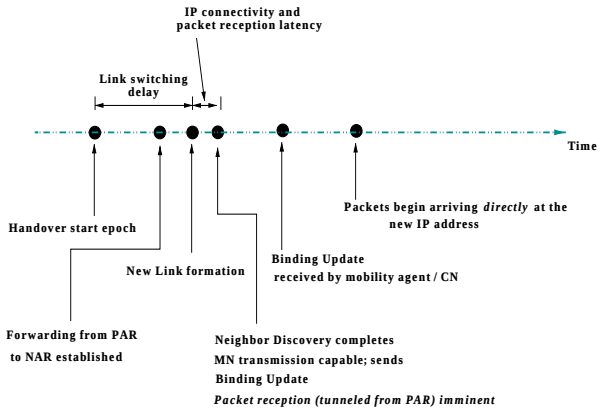

Figure 4: Improved Handover Delay Timeline

access authorization, QoS and header compression among others. It should be possible for the transport protocols to operate without having to re-establish this state (or context) once basic connectivity is established. Whereas fast handover supports establishing IP connectivity, transferring feature contexts facilitates other protocols to operate without the need for context re-establishment.

We motivate the need for context transfers using an example. Consider IPv6/UDP/RTP [7, 19, 22] header compression [5] involving a mobile node with an IPv6 address. The size of a *Full Header* in this case is 84 bytes. If we consider voice sampled at 9.6 Kbps with 20 ms packetization interval and an average compressed header size of 4 bytes, each Full Header packet is equivalent to 3 voice packets. On a cellular link with 120 ms RTT, it takes at least 6 Full Header packets to be transmitted before a positive acknowledgment from the decompressor can arrive and confirm compression state establishment. Even in an optimistic approach, where a compressor does not necessarily wait for an acknowledgment, several Full Header packets are deemed necessary in a lossy link. Typically this threshold is higher during handovers since the "fringe areas" across cellular boundaries are prone to extremely high error rates. So, we assume the number of Full Headers needed for context establishment to be between 3 - 6, corresponding to 9 - 18 voice packets. Observe that 18 voice packets is 360 ms of voice activity with 20 ms packetization. Added to this number of packets is the compression algorithm latency, which we assume to be about 40 ms. Conservatively, this means, a total of 400 ms is needed for context establishment during each handover, which directly affects the quality of voice.

An alternative to context re-establishment is context relocation during handovers. In the above case, header compression context(s) are relocated from one access router to another during handovers so that compression can continue *seamlessly* and alleviate the need for expensive context re-establishment. In what follows, we describe a framework for context transfers that different features could make use of. This framework is the result of our (on-going) research on context transfers. We make use of our previous work in [14, 13, 15].

Our framework addresses the following key components.

1. data structure representation of various contexts for inter-operability across access routers
2. encapsulation of context data structures for transfer and processing, and
3. use of handover signaling for effecting context transfers

We discuss each of these in the following sections. Target router selection mechanism is an open area of research; for our purposes here, we assume that the IP address of the target router is known during context transfers.

4.1 Data Structure Representation

The need for a suitable data structure representation arises from the diversity of various feature contexts and their realizations through different protocols and mechanisms. Typically, there are multiple features associated with each unidirectional packet stream, including QoS, header compression, security etc. Each of these features may support different mechanisms. For example, an access router may support IPv6/UDP/RTP and IPv4/TCP header compression, and IntServ and Diffserv QoS. Hence, a feature context needs to specifically refer to the particular feature realization. One or more feature contexts belong to a packet stream context (or *microflow* context), and one or more packet stream contexts belong to a mobile node's context. This hierarchy, consistent with the nomenclature we proposed in [16], is shown in Figure 5.

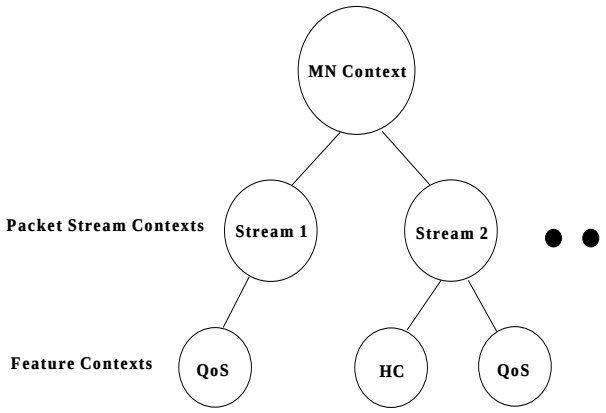

Figure 5: Context Hierarchy

We observe from Figure 5 that a feature context is the basic unit of context transfer. Hence, we define a *Generic Profile Type* or GPT as an object that uniquely identifies the structure of the data related to a feature context. Each instance of a GPT clearly defines the state variables associated with the particular feature context. For example, a QoS Profile Type (QPT) for Diffserv defines the packet classification, packet metering and packet marking control variables, whereas a Compression Profile Type (CPT) for IPv6 defines all the static and changing fields in the IPv6 header. A QoS Profile Type could provide the necessary reservation parameters for voice or multimedia applications. A GPT serves the following purposes. First, for each feature realization, it provides a definition for inter-operability. This definition would include all the necessary and sufficient

state parameters pertaining to the context that can be used for recreating the feature subsequent to handover. Second, a GPT provides a standard programming object that can be used to request context transfers (through appropriate signals) or to initialize feature contexts (via suitable APIs). As part of this initialization, a data object conforming to the profile type would provide a convenient way to structure part of a request for authorization. Finally, the same object can be used for verifying if a target router provides support for a desired feature. Such a verification would constitute the target router selection process. Given an appropriate GPT, each feature context itself then consists of a tuple of the form [GPT, associated state parameters].

4.2 Context Data Structure Encapsulation

The context data structure needs appropriate encapsulation for communication and processing. The context transfer messages involve mobile node - access router communication and inter - access router communication. The MN uses Seamless Handover Initiate (SHIN) option to request context transfer, and a router (optionally) replies back with a Seamless Handover Acknowledge (SHAck) option. The access routers use Seamless Handover Request (SHREQ) and Seamless Handover Reply (SHREP) options. These options are described in detail in [13]. All these options are carried in suitable handover signaling messages, discussed in the next section, to effect context transfers. All the packet options have the format identical to the one shown in Figure 6.

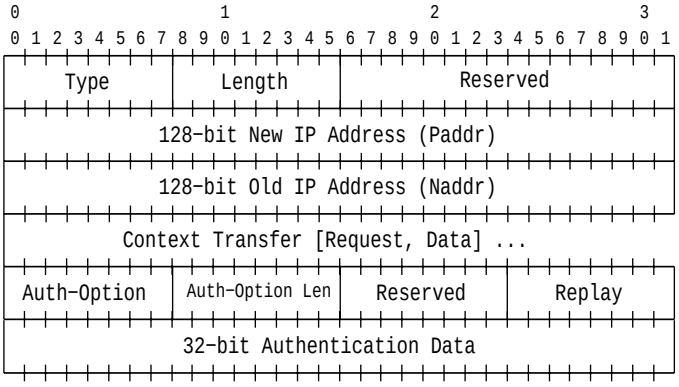

Figure 6: Generic Packet Format for Context Data

The Type and Length fields identify the type of the context transfer option (e.g., request, response) and its length respectively. The previous IP address (`Paddr`) of the MN is needed when the New Access Router needs to fetch the context from the Previous Access Router. The New IP address)`Naddr`) is needed when the New Router has to associate the received contexts (corresponding to previous IP address) to the MN's new IP address. Furthermore, the MN may supply the Previous Access Router's address when context transfers take place subsequent to handover. Following the IP addresses, various context transfer requests or the contexts themselves are enumerated in the [type, length] format. Finally, an authentication option is included to protect against malicious or bogus nodes that may attempt to disrupt communications by prematurely providing a false indication that the mobile node arrived at the new access router. The 32-bit authentication data includes all the context data protected using appropriate security association between the MN and the access router.

4.3 Using Context Transfer Options with Handover Signaling

In this section, we describe how context transfers can be used together with fast handovers in order to support seamless handovers. We use Figure 7 as the reference in the following discussion.

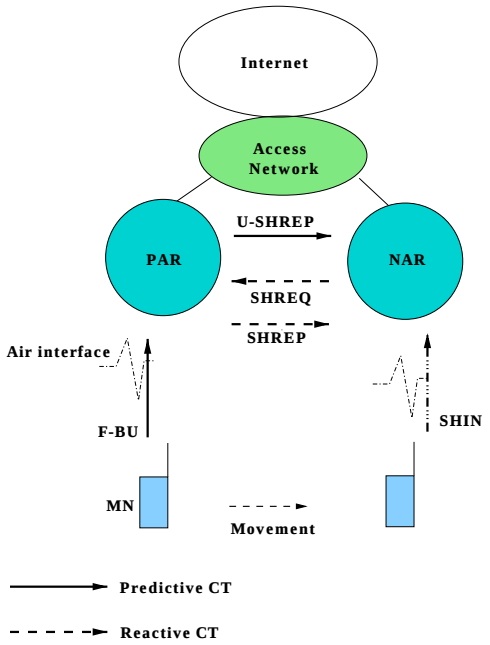

Figure 7: Context Transfers with Handover Signaling

Subsequent to forming its new IP address, a MN sends a *Fast* Binding Update to its PAR. In this message, the MN indicates its desire for context transfer using an appropriate bit. The PAR must wait until it receives the Binding Update before transferring contexts. From the discussion in Section 3, PAR enables the forwarding path towards NAR only after it receives a Binding Update, and then after that it stops forwarding packets to the MN on its previous link. Hence, in order to avoid stale contexts, the PAR must transfer context after it receives the Binding Update. For this, PAR uses the SHREP option in an "unsolicited" fashion (U-SHREP in Figure 7), in the HI message, and includes all the relevant feature contexts as well as the authentication option. When the MN attaches to the NAR, it always sends a message containing SHIN option requesting context transfer. With fast handovers, the required context should already be present at NAR when SHIN message arrives. The NAR must verify that the authentication data present in the SHIN message matches that supplied by the PAR in U-SHREP, before activating the feature contexts for the MN. The NAR may send a SHREP-Ack option back to PAR in the HAck message.

Sometimes when fast handover signaling fails or is unavailable, retrieving contexts *reactively* from the PAR is the only recourse. If PAR never receives a Binding Update for example, then it does not transfer contexts (and does not enable

the forwarding path). In such a case, the NAR engages in retrieving the contexts from PAR using the SHREQ message when it receives the SHIN message. The PAR uses "Previous IP address" to retrieve the requested feature contexts and the "New IP address" to set up the forwarding path for packets arriving at the previous IP address.

5. PERFORMANCE STUDY

In this section, we describe the performance study regarding fast handovers for Mobile IPv6 [9], as well as reactive context transfers. We have implemented fast handover messages and associated processing in Access Routers running a FreeBSD derivative, and in a Mobile Node running Linux IPv6. An Access Point (AP) capable of 11 Mbps offers IEEE 802.11b Wireless LAN connectivity to the MN, and is connected to one of the interfaces on the AR. Each AP is connected to a separate AR.[4] The MN changes its AP, and hence the AR, upon receiving a handover command from the user. Specifically, the MN issues a Proxy Router Solicitation message to PAR supplying the target AP's base station identifier (bssid). The PAR resolves the bassid to the IP address of the NAR[5], and sends a Proxy Router Advertisement message. The MN, upon receiving the Proxy Router Advertisement message, forms a new IP address and sends a Fast Binding Update message. After this, the kernel code instructs the device driver to "attach" to the target AP. The MN may receive a Fast Binding Acknowledgment (F-BAck) message before it actually switches to the new AP.

Once the MN establishes connectivity with the new AP, the device driver detects the new link and passes control to the code that sends either a Fast Neighbor Advertisement (F-NA) message (if F-BAck was not received on the old link) or a Neighbor Advertisement message (if F-BAck was received on the old link). Recall that the PAR and NAR engage in handover messaging (HI/HAck) after PAR sends the Proxy Router Advertisement message to the MN. Thus, when the NAR processes F-NA message, it immediately drains the buffer containing packets for the MN, almost always including a Fast Binding Acknowledgment (F-BAck) message. The NAR then sends a Router Advertisement message containing the option which indicates the IP address to use.

We use *tcpdump* to monitor the times when the signals are sent and received on the MN. The total handover delay is the difference in times between departure instance of Proxy Router Solicitation message and the arrival instance of a packet from NAR (typically the F-BAck) on the new link. The "link switching delay" is the difference in times between the transmission of F-BU packet on the old link and transmission of F-NA or NA packet on the new link. This delay includes the latency incurred in the IP stack as well as the delay involved in acquiring the new link (MAC and physical layer components).

In Table 1, Table 2 we list various signaling delays in *milliseconds*, after a total of 50 different readings were taken.

[4]Even though our wireless interfaces operate at a relatively high speed, we are targeting our work towards deployment on much slower devices. Thus, eliminating unnecessary signaling and message components remains crucial

[5]we maintain a static database of base station identifiers and the corresponding IP addresses of the Access Routers

	minimum	maximum	median	mean
PrAdv-PrSol	2	3	3	3
FBU-PrAdv	3	4	3	3.44
FNA-FBU "link-switching"	28	126	64	66.4
FBAck-FNA	2	4	3	3.4
FBAck-PrSol "total delay"	38	135	73	75.8

Table 1: Fast Handover Latencies (in milliseconds) - FBAck not received on old link

	minimum	maximum	median	mean
PrAdv-PrSol	2	3	3	2.7
FBU-PrAdv	3	4	3	3.4
FBAck-NA "link-switching"	28	138	91	85
NA-PrSol "total delay"	36	146	99	93.5

Table 2: Fast Handover Latencies (in milliseconds) - FBAck received on old link

Table 1 includes the scenario when the F-BAck is not received on the old link, and Table 2 includes the scenario when the F-BAck is received on the old link. Each entry on the first column indicates the difference in time between the signals mentioned.

In order to get a set of results in which the MN *always* receives the F-BAck on the old link (see Table 2), we had to introduce a delay of 20 ms between F-BU and the time when the device driver is instructed to perform link switching. When this artificial delay was 10 ms, the MN received F-BAck some times and it did not receive it at other times.

From these results, we make the following observations. First, the link switching delay constitutes most of the total handover delay (87.5%, and 90% mean values). Since this delay is inevitable, the results indicate that the overheads incurred by the IP layer messages are very small. Even though more rigorous testing is needed, this should be quite encouraging for fast handover implementors. Second, it is very likely to be more efficient for the MN to leave PAR as soon it transmits the F-BU, without having to wait for the F-BAck on the old link. The above results indicate that a mobile node should typically initiate link switching immediately after transmitting a Fast Binding Update. We are very encouraged that the total delay is under 100 ms, since that makes our techniques quite promising for supporting handover for real-time applications such as VoIP.

We have conducted some preliminary performance measurements involving *reactive* context transfers. Subsequent to establishing connectivity at NAR, the MN sends a SHIN message requesting its header compression context be relocated from PAR to NAR. In response to SHIN, NAR sends a SHREQ message to PAR requesting compression context relocation. The PAR responds with a SHREP message that contains the requested compression contexts. Finally, NAR sends a SHREP-Ack message acknowledging successful instantiation of received compression contexts. The delay be-

	mean delay
SHREQ-SHIN (on NAR)	88
SHREP-SHREQ (on PAR) "context bundling"	118
SHREP_Ack-SHREP (on NAR) "context instantiation"	95
SHREP_Ack-SHIN "total CT delay"	1110

Table 3: Reactive CT latencies (in microseconds)

tween receiving SHREQ and responding with SHREP on PAR constitutes the delay involved collecting the requested context parameters and marshaling them for transfer. We refer to this as "context bundling" delay. The delay between receiving SHREP and responding with SHREP-Ack constitutes the delay in instantiating the received context on NAR. We refer to this as "context instantiation" delay.

The header compression context itself includes a Full IPv6/UDP/RTP header with Mobile IPv6 home address option, and the MN's "signature", which includes the source and destination IPv6 addresses and port numbers that identify flow, and other fields maintained by the header compression algorithm. Together, the size of the context is approximately 150 bytes.

In Table 3, we list the delays in *microseconds* involving the various signals. It is evident that the total delay due to context transfer is extremely small, about 1.1 ms. Even though this result pertains to reactive context transfer, we can easily relate the components of this delay to those in context transfers with fast handover signaling. With fast handovers, the delay (on PAR) between receiving F-BU and transmitting U-SHREP is essentially the "context bundling" delay. Similarly, the delay (on NAR) between receiving U-SHREP and transmitting SHREP-Ack is essentially the "context instantiation" delay. Hence, we are able to conclude that context transfers with fast handover signaling will provide similar (if not even better) results as in reactive context transfers. More importantly, we are able to demonstrate that the overall delay incurred in fast handovers and context transfers is closer to the "link-switching" delay. As a comparison, relating the result in Table 3 to that in Table 1, the overall delay is about 77 ms, and the "link-switching" delay is 66.4 ms. The resulting overhead due to IP layer handover signaling and context transfers, in this experiment, is less than 14%. We believe that this is an important result (which itself can be improved even further) that demonstrates the feasibility of network layer fast handovers and context transfers.

6. FUTURE WORK

We believe that, as long as wireless bandwidth is expensive, and the growth of the wireless Internet continues unabated, there will be irresistible pressures to improve the efficiency of the air link between the mobile node and access network. This improved efficiency will come about partially by the use of *All-IP* cellular infrastructure, but mostly by developing link and network-layer context features such as we have described in this article. From this perspective, there are many opportunities for future elaboration of the context transfer framework presented here.

For instance, we suggest that most of the work of maintaining QoS can be viewed as a context transfer operation. This area alone may drive a fundamental revision of QoS management. We believe that a *fish-eye* approach to QoS management is appropriate. This would guide one to design a system so that local state nearby the access routers is very detailed, but remote state inside the core network and infrastructure is much less detailed. The loss of detail would be a by-product of information aggregation according to some sensible criteria. This might lead to a sort of hierarchical or expanding-domain approach to QoS state management; the degree to which a particular context transfer for QoS state interacts with the rest of the network will depend on whether the transfer occurs across domain boundaries, of whatever variety.

Even though it is perhaps the most complicated, we view the QoS context transfer as the archetype for our framework. For one thing, if the framework enables good solutions for QoS context transfer, it is likely to do so for transferring other context features. However, we also believe that there will be distinct limitations on the kinds of context transfers that should be associated with the HI/HAck mechanism [9]. Stated broadly, the appropriate context transfers are those which can be characterized as related to the connection state of the mobile node at the network layer. Other sorts of context associated with the mobile node should be transferred by different mechanisms, perhaps controlled by the application. For instance, if the user has a collection of web pages cached at a local web proxy (for example, as with WebExpress [6]), that collection should be transferred to a new web proxy by some means not related to seamless handovers at the network layer. It is probably the case, however, that the higher-level context transfers will be triggered by actions at the network layer – just as actions at the network layer are likely to be triggered by actions at the link-layer and below. Determining the precise criteria for selecting the appropriate contexts for transfer at the network layer will be a challenging and very rewarding area of research.

Transferring security contexts, which is necessary to maintain secure network-layer connectivity, is a new research area. We foresee a need for a great deal of improvement in technology, which will result in greatly improved user convenience. The local context transfer mechanisms should be kept separate from more global establishments of security associations. As an example, the initial security associations between a mobile node and local access router may be set up by a remote key distribution authority, but re-establishing the security association with a new access router (NAR) could often be handled without bothering the remote agent. This happy circumstance would occur whenever PAR and NAR have some pre-existing security association with each other.

Therefore, we expect that there will be a need for active research in order to enable localities of access routers to establish security associations with each other. This will first involve determining the nodes which are neighboring. The way that the security associations can be established may depend on administrative requirements, but general results

can be obtained. In typical situations, where the access routers are statically located, additional flexibility may be available for design choices. This is because malicious nodes do not typically stay in any one place for a long time, for fear of discovery and reprisal. Thus, very stable access nodes could possibly use protocols that depend on that stability for more economical operation.

As a final suggestion, we believe that there is an intriguing relationship between context transfers for seamless handovers, and active networks. It is likely that network-layer context transfers will trigger additional actions at the application layer, and these additional actions would have some direct effects on the way that active network modules are loaded within the access networks. It could also be the case that interprocess communication between active network modules will be affected by the mechanisms for seamless handover and context transfer.

7. CONCLUSION

We have described and demonstrated new technologies for handling seamless handovers at the network layer. Compared to legacy layer-two based cellular telephone systems, these technologies will pave the way for more universal Internet access from multi-modal wireless devices, and easier roaming across administrative domains. There are two major components for seamless handovers: speed, and reliable packet delivery. The fast-handover protocol specification from the IETF [9] goes a long way toward improving the speed of transfer, and buffering packets to protect against loss helps to smooth the transfer. Bundling feature context transfer on the fast handover signals can greatly improve the "handover experience", especially in the areas of maintaining QoS, header compression and avoiding unnecessary delays due to re-establishment of security associations.

We believe that as time unfolds, the role of context transfer will continue to grow, and that these technologies, under the control of the mobile nodes themselves, will lead to improved efficiencies and yet at the same time greater functionality available to the mobile wireless network nodes. This will be visible at greater convenience and less cost to the user. We hope that this paper will lead others to develop new research areas associated with improving the effectiveness of mobile networking and nomadic Internet access.

8. ACKNOWLEDGMENTS

We wish to thank our colleagues at Nokia Research Center in Mountain View for their valuable input all along during this work. We thank Vijay Devarapalli for providing the performance results reported in Section 5. We also wish to thank the reviewers whose comments have helped improve this paper.

9. REFERENCES

[1] The Third Generation Partnership Project. http://www.3gpp.org

[2] The Third Generation Partnership Project 2. http://www.3gpp2.org

[3] Asokan, N. and Flykt, P. and Perkins, C. and Eklund, T. AAA for IPv6 Network Access (work in progress) draft-perkins-aaav6-03.txt, March 2001.

[4] Blake, S. et al. An Architecture for Differentiated Services Request for Comments (Informational) 2475, Internet Engineering Task Force, December 1998.

[5] C. Bormann et al. RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed. Request For Comments 3095, Internet Engineering Task Force, July 2001.

[6] C. Brooks, M. Mazer, S. Meeks and J. Miller. Application-Specific Proxy Servers as HTTP Stream Transducers. 4th International World Wide Web Conference, 1995.

[7] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. Request for Comments (Draft Standard) 2460, Internet Engineering Task Force, December 1998.

[8] Finlayson, R. and Mann, T. and Mogul, J. C. and Theimer, M. Reverse Address Resolution Protocol Request for Comments (Standard) 903, Internet Engineering Task Force, June 1984.

[9] George Tsirtzis (ed.). Fast Handovers for Mobile IPv6 (work in progress). draft-ietf-mobileip-fast-mipv6-00.txt, February 2001.

[10] D. Johnson and C. Perkins. Mobility Support in IPv6 (work in progress). draft-ietf-mobileip-ipv6-14.txt, July 2001.

[11] J. Kempf (editor). Requirements for Layer 2 Protocols to Support Optimized Handover for IP Mobility. draft-manyfolks-l2-mobilereq-00.txt, July 2001.

[12] R. Koodli and C. Perkins. Fast Handovers in Mobile IPv6 (work in progress). Internet Draft, Internet Engineering Task Force. draft-koodli-mobileip-fastv6-01.txt, October 2000.

[13] R. Koodli and C. Perkins. A Context Transfer Framework for Seamless Mobility IPv6 (work in progress). Internet Draft, Internet Engineering Task Force. draft-koodli-seamoby-ctv6-01.txt, July 2001.

[14] R. Koodli and C. Perkins. A Framework for Smooth Handovers with Mobile IPv6 (work in progress). Internet Draft, Internet Engineering Task Force. draft-koodli-mobileip-smoothv6-00.txt, July 2000.

[15] R. Koodli, C. Perkins, and M. Tiwari. Header Compression State Relocation in IP Mobile Networks (work in progress). Internet Draft, Internet Engineering Task Force. draft-koodli-rohc-hc-relocate-00.txt See also draft-koodli-seamoby-hc-relocate-01.txt, July 2001.

[16] Levkowetz, O. et al. Problem Description: Reasons For Doing Context Transfers Between Nodes in an IP Access Network (work in progress) draft-ietf-seamoby-context-transfer-problem-stat-01.txt, May 2001.

[17] IP Routing for Wireless/Mobile Hosts (mobileip) Working Group, Internet Engineering Task Force. http://www.ietf.org/html.charters/mobileip-charter.html

[18] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6). Request for Comments (Draft Standard) 2461, Internet Engineering Task Force, December 1998.

[19] J. Postel. User Datagram Protocol. Request for Comments (Standard) 768, Internet Engineering Task Force, August 1980.

[20] Robust Header Compression (rohc) Working Group, Internet Engineering Task Force. http://www.ietf.org/html.charters/rohc-charter.html

[21] Context Transfer, Handoff Candidate Discovery, and Dormant Mode Host Alerting (seamoby) Working Group, Internet Engineering Task Force. http://www.ietf.org/html.charters/seamoby-charter.html

[22] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. Request for Comments (Proposed Standard) 1889, Internet Engineering Task Force, January 1996.

[23] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. Request for Comments (Draft Standard) 2462, Internet Engineering Task Force, December 1998.

[24] C. Westphal, R. Koodli, and C. Perkins Context Relocation of QoS Parameters in IP Networks (work in progress) Internet Draft, Internet Engineering Task Force. draft-westphal-seamoby-qos-relocate-00.txt, July 2001.

[25] J. Wroclawski. The Use of RSVP with IETF Integrated Services. RFC 2210, Internet Engineering Task Force.