DNA WG   JinHyeock Choi
Internet-Draft   Samsung AIT
Expires: December 28, 2004   Erik Nordmark
                                                     SUN Microsystems
                                                        June 29, 2004

DNA with unmodified routers: Prefix list based approach
draft-jinchoi-dna-cpl-00.txt

Status of this Memo

Copyright Notice

Abstract

   Upon establishing a new link-layer connection, a host determines
   whether a link change has occurred to examine the validity of its IP
   configuration. This draft presents a way to robustly check for link
   change without assuming changes to the routers. Each link can be
   uniquely identified by the set of prefixes assigned to it. We
   propose that, at each attached link, the host generates the complete
   prefix list, that is, the prefix list contains all the prefixes on
   the link, and when it receives a hint that indicates a possible link

change, it detects the identity of the link by consulting the
existing prefix list. This memo describes how to generate the
complete prefix list and to robustly check for link change even in
the presence of packet losses.

Table of Contents

1. Introduction

   When a host establishes a new link-layer connection, it may or may
   not have a valid IP configuration for the link (for example, the
   subnet prefixes on the attached link and the default routers).
   Though the host has changed its network attachment point, it may
   still be at the same link. The term 'link' used in this document is
   as defined in [4].

   To examine its IP configuration, a host may check for link change,
   i.e. it verifies whether it is attached to the same or a different
   link as before [7]. The host can keep current IP configuration if
   and only if it remains at the same link.

   Usually a host receives the link information from RA (Router
   Advertisement) messages. But, as described in 3.2. [7], it's
   difficult for a host to correctly detect the identify of a link with
   an RA. None of the information in an RA can indicate a link change
   properly. Neither router address nor prefixes will do.

   It may be better to design a new way to represent the identity of a
   link, Link Identifier. Each link has its unique Link Identifier and
   all routers in the link advertise the same Link Identifier. In [11],
   Erik Nordmark proposed a new option, 'Location Indication Option',
   which can serve as Link Identifier. Also Brett Pentland and all
   submitted a draft about Link Identifier [12].

   However, even if some such new scheme is standardized and
   implemented, hosts would still need to cope with routers which do not
   (yet) implement such a scheme. Thus it makes sense to write down the
   rules for how to robustly detect a changed link without assuming
   changes to the routers.

2. Prefix list based approach

2.1 Approach

   Currently there is one thing which can represent the identify of a
   link,

   'The set of all the global prefixes assigned to a link.'

   If a host has the complete list of all the assigned prefixes, it can
   properly determine whether a link change has occurred. If the host
   receives an RA containing one or more prefixes and none of the
   prefixes in it matches the previously known prefixes for the link,
   then it is assumed to be a new link.

   This works because each and every global prefix on a link must not be
   used on any other link thus the sets of global prefixes on different
   links must be disjoint.

   For the purposes of determining the prefixes, this specification uses
   both 'on-link' and 'addrconf' prefixes [4], that is, prefixes that
   have either the 'on-link' flag set, the 'autonomous
   address-autoconfiguration' flag set, or both flags set. This is a
   safe approach since both the set of on-link and the set of addrconf
   prefixes must be uniquely assigned to a link.

   The difficulty lies both in ensuring that the complete prefix list
   for a single link is known and preventing prefixes from possibly
   different links to be viewed as the prefixes for a single link. This
   is challenging given that a single router advertisements is not
   required to include all prefixes for the link, router advertisements
   might be subject to packet loss, new routers and new prefixes (due to
   renumbering) might appear at any time on a link, and the host might
   attach to a different link at any time.

   If the prefix list determination is incorrect, there can be two
   different types of failures. One is detecting a new link when in
   fact the host remains attached to the same link. The other is
   failing to detect when the host attaches to a different link. The
   former failure is undesirable because it might trigger other
   protocols, such as Mobile IPv6 [4], to do unneeded signaling, thus it
   is important to minimize this type of failure. The latter type of
   failure can lead to long outages when the host is not able to
   communicate at all, thus these failures must be prevented.

2.2 Assumptions

   In this approach, we assume that an interface of a host can not be

attached to multiple links at the same time. Though this kind of
multiple attachments is allowed in neither Ethernet nor 802.11b, it
may be possible in some Cellular System, especially CDMA.

2.3 Overview

Hints are used to tell a host that a link change might have happened.
This hint itself doesn't confirm a link change, but can be used to
initiate the appropriate procedures [7].

In order to never view two different links as one it is critical that
when the host might have attached to a link, there has to be some
form of hint. This hint doesn't imply that a movement to a different
link has occurred, but instead, in the absence of such a hint there
could not have been an attachment to a different link.

If the IP stack is notified by the link layer when a new attachment
is established (e.g., when associating to a different access point in
802.11), this will serve as the hint. It helps to reduce the risk
that the assignment of an additional prefix to a link will be
misinterpreted as being attached to a different link. Note that this
hint is merely a local indication and does not require any protocol
changes. For instance, in many implementations this would be an
indication passed from a link-layer device driver to the IP layer.

For implementations which do not provide a "link up" indication, the
solution is to instead rely on either the receipt of an RA that
contains disjoint prefixes from the prefixes that have been collected
on the previous link or the lack of scheduled RAs as described in
[4], as a hint of an attachment to a possibly different link.

Independent of the type of hint used, once a hint is received the
host will start to collect a new set of prefixes for the possibly
different link, and compare them the prefixes known from before the
hint. If there is one or more common prefixes it is safe to assume
that the host is attached to the same link, in which case the
prefixes learned after the hint can be merged with the prefixes
learned before the hint. But if the sets of prefixes are disjoint,
then at some point in time the host will determine that it is
attached to a different link. This process starts when the host is
powered on and first attaches to a link.

Since each router advertisement message isn't guaranteed to contain
all prefixes it is a challenge for a host to attain and retain the
complete prefix list, especially when packets can be lost on the
link.

The host has to rely on approximate knowledge of the prefix list

using RS/ RA exchanges. Just as specified in [4] the host sends an
RS (Router Solicitation) message to All-Router multicast address,
then waits for the solicited RAs. If there was no packet loss, the
host would receive the RAs from all the routers on the link in a few
seconds thereby knowing all the prefixes on the link. Taking into
account packet loss, the host may perform RS/ RA exchanges multiple
times to corroborate the result.

When a hint indicating a possible link change happens, if the host is
reasonably sure that its prefix list is complete, it can detect
whether it is attached to the same link on the reception of just one
RA containing one or more prefixes.

Otherwise, to make matters certain, the host may need at least wait
for more RAs than a single one, or additionally perform multiple RS/
RA exchanges after the hint. After each RS transmission, the host
waits for all RAs that would have been triggered by the RS as one
aspect of trying harder, and then sending multiple RSs (and waiting
for the resulting RAs) is a way to try even harder.

When hints are received frequently, this means that the host might
need to track more than two prefix lists at a time. Essentially,
each reception of a hint indicates the start of a new prefix list to
track, which may or may not turn out to be disjoint from a previously
known prefix list.

All tracking of the prefix lists must take the valid lifetime of the
prefixes into account, but apart from this limitation there isn't any
harm in the host remembering prefix lists from links it has attached
to earlier. The prefix list is maintained separately per network
interface.

3. DNA based on the complete prefix list

   To each link, the set of prefixes is uniquely assigned. Two separate
   links have two disjoint sets of prefixes. The two prefix lists of
   two separate links have no element in common.

   The identify of a link can be represented by the prefix list if it
   correctly includes all the assigned prefixes. We denote the complete
   prefix list as the the list of all the prefixes assigned to the link.
   Each link has its unique complete prefix list. We also say that the
   prefix list is complete if all the prefixes belong to it.

   In case that a host has the complete prefix list, it can properly
   determine whether it is attached to the same link or not, when it
   receives a hint that a link change might have occurred.

   This section presents a procedure to generate the complete prefix
   list and a way to check for link change based on the existing prefix
   list even in the presence of packet losses.

3.1 Generating the complete prefix list

   To efficiently check for link change, a host always maintains the
   list of all known prefixes on the link. This procedure of attaining
   and retaining the complete prefix list is initialized when the host
   is powered on.

   Usually hosts execute and terminate the process of generating the
   complete prefix list (of a link) at an old attachment point, before
   handoff process begins. Though the procedure may take some time,
   that doesn't matter unless the host moves very fast. A host can
   generate the complete prefix list with reasonable certainty if it
   remains attached to a link sufficiently long, approximately 10
   seconds.

   To generate the complete prefix list, presently the host has to rely
   on approximate knowledge based on RS/ RA exchanges as follows.

   First the host sends an RS to All-Router multicast address. Assuming
   there is no packet loss, every router on the link would receive the
   RS and reply with an RA containing all the prefixes that the router
   advertises.

   After an RS transmission, the host waits for all RAs that would have
   been triggered by the RS. There is an upper limit on the delay of
   the RAs. MIN_DELAY_BETWEEN_RAS (3 Sec) + MAX_RA_DELAY_TIME (0.5 Sec)
   + network propagation delay is the maximum delay between an RS and
   the resulting RAs [4]. 4 seconds would be a safe number for the host

to wait for the resulting RAs. Assuming no packet loss, within 4
seconds, the host would receive all the RAs and know all the
prefixes.

In case of packet loss, things get more complicated. In the above
process, there may be a packet loss that results in the generation of
the incomplete prefix list, i.e. the prefix list that misses some
prefix on the link. To remedy this deficiency, the host may perform
multiple RS/ RA exchanges to collect all the assigned prefixes.

After one RS/ RA exchange, to corroborate the completeness of the
prefix list, the host may send one or more RSs additionally and wait
for the resulting RAs. Each RS/ RA exchange produces the set of the
prefixes on the link. These sets may or may not be identical
depending on whether there happened a packet loss or not. Assuming
no packet loss, these sets would be the same.

The host takes the union of all these sets to generate the prefix
list. The more RS/ RA exchange the host performs, the more probable
that the resulting prefix list is complete. Section 4 gives the
detailed analysis.

By performing multiple (multicast) RS/ RA exchanges and combining the
results, the host can reasonably sure that the existing prefix list
is complete.

To ascertain whether its existing prefix list is complete or not, the
host can set its own policy. The host may take into consideration
the packet loss rate of the link and the number of RS/ RA exchanges
it performed.

For example, the host may keep track of how may RS/ RA exchanges it
performed on this attachment point, and if this is 3 or more it
assumes that the resulting prefix list is complete. But if this is
only 1 or 2, the host doesn't assume the completeness of the prefix
list.

In general, the higher the error rate, the more RS/ RA exchanges are
needed to assure the completeness of the prefix list.

It may happen that the host fails to generate the complete prefix
list.

The host may not be sure that the prefix list is complete. Or worse,
the host may falsely assume the completeness of the prefix list while
it is not.

The host may generate either 1) the incomplete prefix list, i.e. the

prefix list misses some prefix on the link or 2) the superfluous
prefix list, i.e. the prefix list that contains some prefix that is
not assigned to the link.

It is noted that 1) and 2) is not exclusive. The host may generate
the prefix list that misses some prefix on the link but includes the
prefix not assigned to the link.

Severe packet losses during prefix list generation may cause the
incomplete prefix list. Or the host may have undergone a link change
before finishing the procedure of the complete prefix list
generation. Later we will deal with the case that the host can't be
sure of the completeness of the prefix list.

Even if the host falsely assumes that the incomplete prefix list is
complete, the pain of that assumption is that the host might later
think it has moved to a different link when in fact it has not.

In case that a link change happens, even if the host has the
incomplete prefix list, it will detect a link change. Hence the
incomplete prefix list doesn't cause a connection disruption. But it
may cause excessive signaling messages, for example Binding Update
messages in [8].

The superfluous prefix list presents a more serious problem.

Without the indication from the link-layer, the host can't perceive
that it has changed its attachment point, i.e. it has torn down an
old link-layer connection and established a new one.

So while generating the prefix list, without knowing it, the host may
be attached to multiple links in turn and accidentally end up with
prefixes from multiple links in the prefix list.

Here is an example. The host begins to collect the prefixes on a
link. But before the prefix list generation is completed, without
its knowledge, the host moves to a new link. Unaware that now it is
at the different link, the host keeps collecting prefixes with RS/ RA
exchanges to generate the prefix list. This results in the prefix
list containing prefixes from two different links. If the host uses
this prefix list, it fails to detect a link change.

Since, in the absence of link-up indications, a RA with a disjoint
set of prefixes is treated as a hint, the risk for confusion occurs
because the host can not tell from the RAs whether they were
solicited by the host. (RFC 2461 recommends that solicited RAs be
multicast.)

The safest approach is for the host to assume that any RA received
within 4 seconds after sending a RS, assuming no intervening link-up
indication, where solicited, that is, the host has not moved to a
different link in that time.

In case the host moves fast, this assumption may lead to the
superfluous prefix list generation. The protocol is robust against
such confusion when a link-up indication is delivered to the IP layer
any time the host might have changed it's attachment.

If the host doesn't have any link layer hints, for example link up/
down indications, it can't decide whether to use the prefixes it
receives for collecting information about the "old" link or about the
"new" link.

Maybe this is just something that should be stated as an assumption;
We may assume that when a host changes its attachment point, it will
be notified of the event and can distinguish the RAs arriving from
the old attachment point and the RAs from the new attachment point.

Thus we think the prefix-based approach has a stronger assumption
here than the Link Identifier-based approach, because the latter can
operate reliably without any link-layer hint.

3.2 Checking for link change

When a host receives a hint that indicates a possible link change, it
initiates DNA procedure to determine whether it still remains at the
same link or not. At this time, the complete prefix generation may
or may not be finished.

First, if the host has finished the complete prefix list generation
and can be reasonably sure of its completeness, the receipt of a
single RA (with at least one prefix) is enough to detect the identify
of the currently attached link.

Assume that, after the hint, the host receives an RA that contains at
least one prefix. Either by an RS transmission or by an arrival of
an unsolicited RA, the host can get an RA. The host compares the
prefixes in the RA with those in the existing prefix list. If the RA
contains a prefix that is also a member of the existing prefix list,
the host is still at the same link. Otherwise, if none of the
prefixes in that RA matches the previously known prefixes, it is at a
different link.

It may happen that the host can not be sure that the prefix list is
complete. In this case, the host may use another scheme that makes a
decision based on multiple solicited RAs, instead of one RA.

Suppose that before finishing the prefix list generation, the host
receives the hint that indicates a possible link change. Then the
host can't assume the completeness of the prefix list.

The host can generate another (complete) prefix list to compensate
the uncertainty of the existing prefix list. After the hint, it
performs one or more RS/ RA exchanges additionally to collect all the
prefixes on the currently attached link. With the resulting
prefixes, the host generates the second prefix list.

Then the host compares two prefix lists and if the lists are
disjoint, i.e. have no prefix in common, it assumes that a link
change has occurred. Usually this is done at a new attachment point.

For example, assume that the host keeps track of how many RS/ RA
exchanges it performed at a link. If this is 3 or more, the host
assumes that it have seen all the prefixes. Suppose that the host
has done a single RS/RA exchange, then it receives a link up
notification that causes it to initiate the DNA procedure. For a
better judgment, the host performs new RS/RA exchanges. If the host
tracks the list of prefixes received (from all received RAs) before
the link up notification and after the one, it can compare the lists
to check for link change. In case that the lists are disjoint, the
host can assume it has moved.

The difference with the previous case is that the host doesn't use
the first RA received after the hint to make its decision. Instead
it waits for the timeout and then use the received set of prefixes to
determine whether a link change has occurred. Though slower, this is
more robust.

In summary, first a host makes the complete prefix list. When a hint
occurs, if the host decides that the prefix list is complete, it will
check for link change with just one RA (with a prefix). Otherwise,
in case that the host can't be so sure, it will perform additional
RS/ RA exchanges to corroborate the decision.

When the host is sure that the prefix list is complete, a false
movement assumption may happen due to renumbering when a new prefix
is introduced in RAs. We may solve the renumbering problem with
minor modification like below.

1) When a router starts advertising a new prefix, for the time being,
every time the router advertises a new prefix in an RA, it includes
at least one old prefix in the same RA. The old prefix assures that
the host doesn't falsely assume a link change because of a new
prefix. After a while, hosts will recognize the new prefix as the
one assigned to the current link and update its prefix list.

2) To make matters more certain, we may mandate hosts to assume a
link change only after a new link-layer connection. It's reasonable
to assume that a new link-layer connection precedes a link change.

In this way, we may provide a fast and robust solution. If a host
can make the complete prefix list with certainty, it can check for
link change fast. Otherwise, it can fall back on a slow but robust
scheme. It is up to the host to decide which scheme to use.

[4](#). **Performance Analysis**

   In this section, we compute the probability that a host fails to
   generate the complete prefix list and consequently assumes a link
   change erroneously.

   Suppose, in a link, there are N routers, R[1], R[2],...., R[N].

   Each R[i] advertises the Router Advertisement RA[i] with the prefix
   P[i].

   It is the worst case that each router advertises the different
   prefix. It is necessary to receive all the RA[i] to generate the
   complete prefix list.

   We assume there is a host, H, and when the host sends a Router
   Solicitation, let P be the probability that it fails to receive a
   RA[i], whether because of a RS loss or a RA loss.

   So when the sends a Router Solicitation, the probability that it will
   receive all RA[i] is $(1-P)^N$.

   Let's assume the host performs RS/ RA exchange T times, 1,2,..,T.

   Let S[k] be the set of all RAs which the host H successfully receives
   at k-th RS/RA exchange. The probability that R[i] belongs to S[k] is
   (1-P).

   Let PL[k] be the set of prefixes which are made from S[k], i.e. the
   set of P[j] such that RA[j] belongs to S[k]. Obviously, the
   probability that P[i] belongs to PL[k] is also (1-P).

   Let PL be the union of all PL[k], from k=1 to k=T. PL is the prefix
   list made from performing RS/ RA exchange T times.

   1) The probability of the complete prefix list generation

   First the probability that P[i] belongs to PL is $1-P^T$. The
   probability that the prefix list PL is complete is $(1-P^T)^N$.

   For example, assume the error rate is 1 % and there are 3 routers in
   a link, then, with 2 RS/ RA exchanges, the probability of generating
   an accurate Complete Prefix List is roughly 99.97 %.

   At this point, assume that the host H receives a hint that a link
   change might have happened and consequently initiates the procedure
   of checking a link change.

2) The false DNA probability if the host checks for link change with one RA.

Assume one RA, whether solicited or unsolicited arrives. If the host H makes a decision based solely on the RA and the prefix list, the probability that it falsely assume a link change is P^T.

For example, given the error rate is 1%, with 2 RS/ RA exchanges, the probability of false movement detection is 1/ 10000.

3) The false DNA probability if the host checks for link change with additional RS/ RA exchanges.

Instead of depending on the single RA, the host H performs additional RS/ RA exchange U times, 1,2...U. Then the probability that H falsely assumes a link change is

$[P^T + P^U - P^{(T+U)}]^N$.

For example, given the error rate is 1 % and there are 3 routers in a link, if the host H performs 2 RS/ RA exchanges before the hint and 1 RS/ RA exchange after one, the probability of false movement detection is roughly 1/1000000.

In the above formula, the result goes to $P^{(U*N)}$ as T goes infinity. The term $P^{(U*N)}$ results from the probability that the host receives no RA during U RS/ RA exchange after the hint. To see that it still remains at the same link, a host needs to receive at least one RA.

We think it is reasonable to assume that the RS will be retransmitted until at least one RA arrives. If we take a one more assumption that the host receives at least one RA, the probability will be

$[[P^T + P^U - P(T+U)]^N - P^{(U*N)}]/ [1- P^{(U*N)}]$

The above converges to zero as T approaches infinity.

5. IANA Considerations

   No new message formats or services are defined in this document.

6. Security Considerations

   Because DNA schemes are based on Neighbor Discovery, its trust models
   and threats are similar to the ones presented in [10]. Nodes
   connected over wireless interfaces may be particularly susceptible to
   jamming, monitoring and packet insertion attacks. Use of [9] to
   secure Neighbor Discovery are important in achieving reliable
   detection of network attachment. DNA schemes SHOULD incorporate the
   solutions developed in IETF SEND WG if available, where assessment
   indicates such procedures are required.

   The threats specific to DNA is that an attacker might fool a node to
   detect the attachment to a different link when it is in fact still
   attached to the same link, and conversely, the attacker might fool a
   node to not detect the change of a link.

   The first form of attack is not very serious, since at worst it would
   imply some additional higher-level signaling to register new
   (care-of) address. The second form of attack can be more serious,
   especially if the attacker can prevent a host from detecting any new
   link. The protocol as specified would require an attacker to be
   on-link and be authenticated and authorized to send router
   advertisements when Secure Neighbor Discovery [9] is in use. But
   even without SEND, an attacker would need to send router
   advertisements containing the prefixes to which it wants the host to
   be unable to detect movement. This can be done for a small number of
   prefixes, but it isn't possible for the attacker to completely
   disable DNA for all possible prefixes on other links.

7. **Examples**

   This section contains some example packet flows showing the operation
   of prefix based DNA.

7.1 **Example with link-up indication**

   Assume the host has seen no link-up indication for a long time and
   that it has the prefixes P1, P2, and P3 in its prefix list for the
   interface.

   The IP layer receives a link-up indication. This hint makes it
   multicast a Router Solicitation and start collecting the received
   prefixes in a new list of prefixes.

   The host receives a Router Advertisement containing no prefixes.
   This has no effect on the algorithm contained in this specification.

   The host receives a Router Advertisement containing only the prefix
   P4. This could be due to being attached to a different link or that
   there is a new prefix on the existing link which is not announced in
   RAs together with other prefixes, and a spurious hint. In this
   example the host decides to wait for another RA before deciding.

   One second later a router advertisement arrives which contains P1 and
   P2. As a result the "new" prefix list has P1, P2, and P4 hence is
   not disjoint from the "old" prefix list with P1, P2, and P3. Thus
   the host concludes it has not moved to a different link and its
   prefix list is now P1, P2, P3, and P4.

   Some time later a new link-up indication is received by the IP layer.
   Triggers sending a RS.

   A Router Advertisement containing P5 and P6 is received by the host.
   Based on some heuristic (the assumed frequence of prefixes being
   added to an existing link) this time the host decides that it is on a
   new link.

   One second later a Router advertisement with prefix P7 is received.
   Thus the prefix list now contains P5, P6, and P7.

7.2 **Example without link-up indication**

   Assume the host has collected the prefixes P1, P2, and P3 in its
   prefix list for the interface.

   The host receives a Router Advertisement containing only prefix P4.
   The fact that P4 is disjoint from the prefix list makes this be

treated as a hint. This hint makes the host multicast a Router
Solicitation and start collecting the received prefixes in a new list
of prefixes, which is initially set to contain P4.

The host receives a Router Advertisement containing no prefixes.
This has no effect on the algorithm contained in this specification.

The host receives a Router Advertisement containing only the prefix
P4. This could be due to being attached to a different link or that
there is a new prefix on the existing link which is not announced in
RAs together with other prefixes. In this example the host decides
to wait for another RA before deciding.

One second later a router advertisement arrives which contains P1 and
P2. As a result the "new" prefix list has P1, P2, and P4 hence is
not disjoint from the "old" prefix list with P1, P2, and P3. Thus
the host concludes it has not moved to a different link and its
prefix list is now P1, P2, P3, and P4.

Some time later the host receives a Router Advertisement containing
prefix P7. This is treated as a hint since it is not part of the
current set of prefixes. Triggers sending a RS and initializing the
new prefix list to P7.

A Router Advertisement containing P5 and P6 is received by the host.
This is disjoint with both of the previous prefix lists, thus the
host might be attached to a 3rd link after very briefly being
attached to the link with prefix P7.The host decides to wait for more
RAs.

One second later a Router advertisement with prefix P7 is received.
It still isn't certain whether P5, P6, and P7 are assigned to the
same link (and without a link-up indication such uncertainties do
exist).

A millisecond later a Router Advertisement with prefixes P6 and P7 is
received. Now the host knows that P5,P6, and P7 are assigned to the
same link.

Four seconds after the RS was sent and no RA containing P1, P2, P3,
or P4 has been received the host can conclude with high probability
that it is no longer attached to the link which had those prefixes.

8. References

8.1 Normative References

[1]   Bradner, S., "IETF Rights in Contributions", BCP 78, RFC 3667,
      February 2004.

[2]   Bradner, S., "Intellectual Property Rights in IETF Technology",
      BCP 79, RFC 3668, February 2004.

[3]   Bradner, S., "Key words for use in RFCs to Indicate Requirement
      Levels", BCP 14, RFC 2119, March 1997.

[4]   Narten, T., Nordmark, E. and W. Simpson, "Neighbor Discovery for
      IP Version 6 (IPv6)", RFC 2461, December 1998.

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

[6]   Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6)
      Addressing Architecture", RFC 3513, April 2003.

[7]   Choi, J., "Detecting Network Attachment in IPv6 Goals",
      draft-ietf-dna-goals-00 (work in progress), June 2004.

8.2 Informative References

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

[9]   Arkko, J., Kempf, J., Sommerfeld, B., Zill, B. and P. Nikander,
       "SEcure Neighbor Discovery (SEND)", draft-ietf-send-ndopt-05
       (work in progress), April 2004.

[10]   Nikander, P., Kempf, J. and E. Nordmark, "IPv6 Neighbor
       Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.

[11]   Nordmark, E., "MIPv6: from hindsight to foresight?",
       draft-nordmark-mobileip-mipv6-hindsight-00 (work in progress),
       November 2001.

[12]   Pentland, B., "Router Advertisement Link Identification for
       Mobile IPv6 Movement Detection",
       draft-pentland-mobileip-linkid-00 (work in progress), May 2003.

[13]   Choi, J., "Router Advertisement Issues for Movement Detection/
       Detection of Network Attachment", draft-jinchoi-ipv6-cra-00
       (work in progress), October 2003.

   [14]   Daley, G. and J. Choi, "Movement Detection Optimization in
          Mobile IPv6", draft-daley-mobileip-movedetect-01 (work in
          progress), May 2003.

Authors' Addresses

   JinHyeock Choi
   Samsung AIT
   i-Networking Lab
   P.O.Box 111 Suwon 440-600
   KOREA

   Phone: +82 31 280 9233
   EMail: jinchoe@samsung.com

   Erik Nordmark
   Sun Microsystems
   17 Network Circle
   Menlo Park, CA 94043
   USA

   Phone: +1 650 786 2921
   EMail: erik.nordmark@sun.com