Dynamic Internetworking Based on Late Locator Construction

Anders Eriksson
Ericsson Research
SE-164 80 Stockholm, Sweden
Anders.E.Eriksson@ericsson.com

Börje Ohlman
Ericsson Research
SE-164 80 Stockholm, Sweden
Borje.Ohlman@ericsson.com

***Abstract* — The legacy Internet technology is optimized for a semi-static inter-domain topology. Mobility or multihoming is handled by extending the legacy technology with new protocols. This paper describes a novel internetworking architecture with native support for a highly dynamic edge topology. The architecture distinguishes between a rather static core network on the one hand, and on the other hand edge networks forming an edge topology that may change on a short timescale due to mobility or re-homing events. The source host addresses the destination host directly with a hierarchically structured global locator. Indirection via a mobility agent is thus not needed. The name to locator resolution is based on a novel mechanism that constructs a global host locator on-demand that describes the current internetwork path from the core network to the host (late locator construction). This enables the resolution of the host name into a hierarchical and topologically significant host locator also with a highly dynamic edge topology where the path to the destination host traverses several moving and multihomed networks. The same locator construction mechanism is used to handle both node and network mobility as well as multihoming. Simulation results that verify the basic functionality of the late locator construction approach are reported.**

Keywords - internetworking, multihoming, mobility, name resolution, locator

I. Introduction

To be able to do routing in large networks it is generally regarded as helpful to have a hierarchical locator structure that reflects the network topology. On the other hand, if network mobility and multihoming is to be supported, renumbering is required after mobility or re-homing events to maintain the hierarchical structure of the locators. The alternative approach that often is taken is to introduce some kind of mobility agent that hides the mobility of nodes and networks from the correspondent node. The latter approach leads to suboptimal routing and adds complexity both to the control plane and the forwarding mechanisms of the network. It is therefore of interest to investigate whether a viable internetworking architecture for dynamic topologies can be designed with a native support for end-to-end forwarding without indirections via mobility agents.

This paper describes an internetworking architecture based on Late Locator Construction (LLC) for the support of a dynamic edge topology connected to a global core network as shown in Fig. 1. The dynamicity in the edge topology is caused by mobility and re-homing events occurring among the edge networks and hosts.

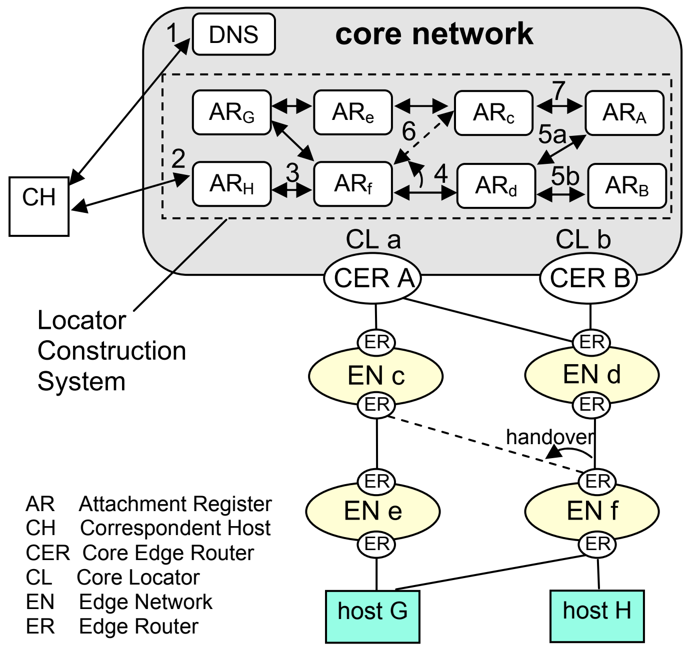

Figure 1. Internetwork with a semi-static core network, and a topology of edge networks that is dynamic due to mobility and re-homing events. The logical links between the Attachment Registers correspond to the attachments in the edge topology. The numbers indicate the steps to resolve the name of host *H* to a Global Locator.

When initiating a communication session, a host name is resolved to a topologically significant Global Locator (GL) that describes the current path between the core network and the host. The GL is an internetworking locator that bridges over different local addressing domains, e.g. public/private IPv4, or IPv6, and allows for end-to-end forwarding over an arbitrary edge topology without the use of mobility agents. Triangular or pin-ball routing can thus be avoided, and there is no need to restrict the topology to nested networks.

The host GL is constructed by the Locator Construction System (LCS) at the latest possible stage, i.e. at the time of session initiation. Fresh topology information can then be used in the construction process. The GL of a host can thereby be constructed to describe a current internetwork path between the core network and the host across a highly dynamic edge topology.

Each network and host in the edge topology has an associated Attachment Register (AR) in the LCS, see Fig. 1. The host or network updates its AR with the names of the directly attached neighbors. Based on this attachment information distributed over the ARs, the LCS can construct a GL for the source and destination host at the time of session initiation.

A primary design goal of the LLC architecture is to keep the amount of update signaling to the LCS at a minimum, especially after changes in the edge topology that affect the GL of a great number of hosts, such as a network re-homing event or a mobility event of a large edge network. Topology changes due to mobility or re-homing events require state update signaling to the LCS only by the network entities that have seen changes of their directly attached neighbors. This can be compared with solutions based on dynamic DNS, where all hosts attached to a moving or re-homed network that changes its point of attachment have to update their locator registrations.

Another primary design goal is to keep the amount of routing and forwarding state to a minimum. Each entity in the edge topology keeps routing and forwarding state only for the directly attached neighbors. The amount of signaling to update such state is thus reduced compared to solutions that keep state for network entities beyond the directly attached neighbors, such as the legacy routing mechanisms in the fixed Internet, or forwarding based on flat labels [1][2].

The end-to-end routing is divided into routing across the edge topologies of the source and destination hosts on the one hand, and routing across the core network on the other hand. The edge routing system does not inject routing state into the core network routing system, which thus can be isolated from the scalability problems associated with a dynamic edge topology. For example, the legacy IPv4 Internet backbone can be used as a core network.

As the GL depends on the current network topology, it will change as the network topology changes. A host with an on-going session that detects that the destination GL is invalid requests the construction of a new GL. The separation of host identity from the locator, as defined in the HIP framework [3], is used to hide this locator change from the transport layer so that continuity of e.g. TCP connections can be maintained.

The LLC architecture is based on concepts developed in the Ambient Networks project [4]. For related work in the area of dynamic internetworking using a core network we refer to [1][2][5][6]. There exist a number of proposals for host mobility in IP networks, such as Cellular-IP [7] and HMIP [8]. These proposals rely on Mobile IP [9] for support of global mobility and thus depend on mobility agents and tunneling and do not address network mobility or network multihoming. The IETF Nemo WG [10][11] develops a framework for the support of network mobility. The Nemo Basic Support Protocol offers mobility transparency and location privacy for mobile hosts. However, additional route optimization mechanisms are needed to solve the pin-ball routing problem.

The LLC approach to multihoming uses the concept of multiple locators per interface as in the IETF Shim6 [13] and Monami [14] frameworks.

The GSE framework [12] addresses multihoming but not mobility. It has a global locator structure which includes a hierarchically structured core locator, a site locator, and an end system identity. Also, an upward delegation mechanism is used for name resolution. The LLC architecture modifies and extends these concepts to handle also host and network mobility.

The novel contribution of this paper is a new internetworking architecture, based on the late locator construction mechanism. It offers a unified way of supporting both node and network mobility as well as multihoming without having to introduce mobility agents. The Attachment Registers that are introduced require comparatively little state to be kept in the network. Initial simulations verify the feasibility of the approach. The LLC architecture is intended to support a range of use cases such as ISP selection and access network multihoming as well as vehicular networks and personal area networks.

The paper is organized as follows. Section II describes the structure of the edge topology and the GL. Section III describes the late locator construction concept, and its application to mobility management and multihoming. Section IV gives an overview of the end-to-end routing framework. The scalability characteristics of the LLC architecture are discussed in Section V, and some early simulation results are reported in section VI. Finally, conclusions are drawn in section VII.

II. Network Entities and Global Locators

A. Terminology

The edge topology consists of Core Edge Routers (CER), Edge Networks (EN), and Hosts, see Fig. 1. We refer to these entities as Edge Entities. Edge Routers (ER) are used at the internetwork interfaces of the Edge Networks. Core Edge Routers act as gateways between Edge Networks and the core network. Edge Entities form a dynamic edge topology by attaching to and detaching from each other. Attachments are strictly bilateral relations between the attached entities, which are referred to as neighbors. An Edge Entity registers its neighbors with its associated Attachment Register.

In the LLC architecture, the packet forwarding across the edge topology is performed by the Edge Entities using the Global Locator as an internetworking locator. The internetworking layer of the LLC architecture is overlaid on the network technologies of the core network and the edge networks in the same fashion as the IP layer is overlaid on local network technologies in the traditional Internet.

A Core Edge Router is addressed with a semi-static GL that can be mapped one-to-one to an address that is routable in the core network technology at hand. This type of GL is called a Core Locator (CL).

The Edge Network Identifier (EID) identifies an Edge Network, and a Node Identifier (NID) identifies a host.

B. Structure of the Global Locator

Fig. 2 shows an example of an edge topology with multihomed edge networks. Also shown are examples of host GLs that are used for packet forwarding across the internetwork.

The semantic content of a host GL can be described by a dot-separated list that starts with the CL of a CER that is near to the host. The list is extended with a sequence of EIDs which describes the path across the edge networks between the CER and the host. The Node Identifier of the host is the final item in the dot-separated list. Any contiguous subset of this list that starts with the Core Locator is referred to as a GL prefix. The byte-level syntax of the GL is beyond the scope of this paper.

Fig. 2 illustrates the use of host GLs, NIDs, EIDs, and GL prefixes. Four examples of GLs for host *G* are indicated, each of them describing a specific internetwork path from the core network to this host. An instance of a GL for host G is *a.c.e.G,* where *a* is the CL of the CER, and *c* and *e* are the EIDs of the Edge Networks that are traversed from the CER to the host having NID *G*. For host *H*, two GLs are indicated, *a.c.f.H*, and *b.d.f.H*, each corresponding to a specific path.

Note the difference between a GL and an explicit source route. A GL is a true locator in the sense that it describes the location relative to a root object, which in this case is the core network. On the other hand, an explicit source route describes the path from the source to the destination.

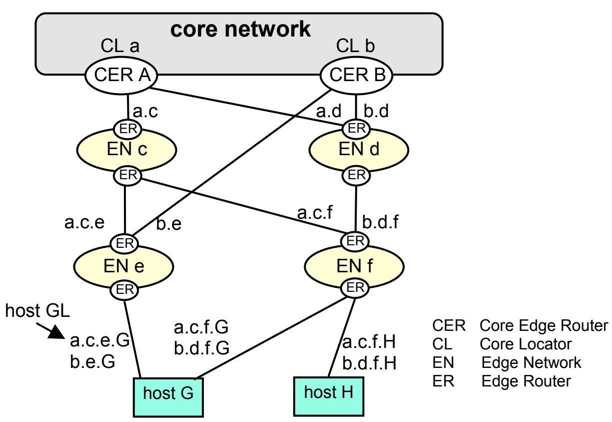

Figure 2. Example of a dynamic edge topology with Global Locators describing paths between the core network and the hosts.

A GL is globally unique if it consists of globally unique components. The first component of the GL, the CL of the CER, is globally unique since it is used for routing in a global core network. Statistical uniqueness can be accomplished for the EID and the NID components of the GL by generating a sufficiently long bit string using cryptographic methods. Centralized administration for locators and identifiers is therefore not needed, except for the assignment of the CL of the CER. However, assuming that the core network is IPv4 or IPv6, the CL of a CER can be assigned using traditional methods and existing address administrations.

To reduce the size of the host GL, and to avoid revealing too much information about the edge topology, a method based on IPv6 prefix delegation could be used to compress the topological information in the GL into a regular IPv6 address. This is an area for future research.

III. Late Locator Construction, Mobility Management and Multihoming

A. Name Resolution and Late Locator Construction

The Locator Construction System consists of a set of Attachment Registers, see Fig. 1. Each Edge Entity has an associated Attachment Register where it stores the EID of the Edge Entity itself, as well as the EIDs of its neighbor Edge Entities. The AR of an Edge Entity also stores the GL of the AR of each neighbor Edge Entity, which is learnt during the attachment procedure. The entries in the ARs represent logical links that correspond to the attachments between the neighboring Edge Entities.

Each Attachment Register can be located in a separate node. Alternatively, a multitude of Attachment Registers can be co-located in the same node. Also, the Attachment Registers can be implemented using distributed hash table techniques. The pros and cons of these approaches are outside the scope of this paper.

An Attachment Register belongs to the associated Edge Entity. The LCS can thus be distributed on all service providers owning the Edge Entities. An AR must be located so that it can be assigned a semi-static GL that can be retrieved via e.g. DNS. Therefore, ARs should be located in the core network (as shown in Fig. 1), or in fixed Edge Networks.

The GL of a host is constructed by querying the specific sequence of ARs that corresponds to an internetwork path between the host and the Core Edge Router. A correspondent host that needs to resolve the domain name of host H into its GL finds the AR of host H via DNS, which returns the GL of the AR (step 1 in Fig. 1). This AR is the first in a sequence of ARs that form a resolution path corresponding to the internetwork path from host *H* to the Core Edge Router. The resolution path is found by a routing protocol operating on the attachment registration information in the ARs as will be described in more detail in section IV. For example, the resolution path of host *H* in steps 2-5b in Fig. 1 is the sequence of Attachment Registers associated with host *H*, EN *f*, EN *d*, and CER *B* (before the hand-over). Each AR along the resolution path returns the EID or NID of the associated Edge Entity.

The GL of host *H* is constructed from the sequence of identifiers retrieved along the resolution path. This sequence is represented by a dot-separated list which is prefixed with the Core Locator of the CER. Finally, the construction of the GL of host *H* is completed by adding the NID of host *H* to the list. In the example of Fig. 1, the GL resulting from steps 1-5b is *b.d.f.H*. Using this GL, the correspondent host can send user data directly to host *H* without any indirection via a mobility agent, and using a minimum of routing and forwarding state in the edge topology.

When a host needs to resolve its own name, it does not have a source GL to insert in the resolution request so that the name resolution system (e.g. DNS) can return a response. To construct a temporary source GL, a record route mechanism is used. Initially, the source GL consists only of the NID of the source host. As the request is forwarded along the default path to the core, each traversed Edge Network adds its EID to the source GL. Finally, the source GL is completed by the Core Edge Router adding its CL. The mechanism for establishing a default path to the core network is described in section IV.

The AR of a not so mobile host may cache the host GL to avoid unnecessary locator constructions, and to reduce the response time of the LCS. A stale GL will result in a forwarding failure that is reported to the source host, which in turn requests the construction of a fresh GL. To adjust to variations in the dynamicity of the internetwork, the timeout for cached entries could be adapted based on the hit rate.

B. Mobility Management and Hand-over

The hand-over event of Edge Network *f* shown in Fig. 1 results in a new path from the core network to host *H*. A new GL for host *H* describing this path must thus be constructed. To this end, network *f* first registers its new neighbor in its Attachment Register. Edge Network *f* thus changes the registration in AR_f from pointing at network *d* to pointing at network *c*. The new GL can now be constructed by the LCS. The construction of the new GL is illustrated in Fig. 1 (replacing steps 4 and 5 with steps 6 and 7). As a result of the updated attachment registration, the correspondent host will be able to resolve the name of host *H* into its new GL, which is *a.c.f.H*.

If there is a change in topology during a session, host *H* requests its new GL from the LCS, and sends a GL update to the correspondent host. Host *H* can detect topology changes as it uses its GL as the source locator in the data packets sent to the correspondent host, as described in section IV. When the path described by the source GL no longer exists, an error message will be returned to host *H* by the first Edge Router that detects that the next hop entity described by the source GL is not attached. Host *H* then queries the LCS for its new GL. Updates of the GL between peer hosts, including the case when both hosts move simultaneously, the so called double jump situation, can be handled in a secure manner based on the same type of mobility and multihoming mechanisms as described in the HIP framework [3].

C. Multihoming

Multihoming is based on the GLs described in section II, and the network and host mobility mechanism described in this section. The LLC architecture handles host and network mobility as well as multihoming using one common set of mechanisms.

Multihoming of networks or hosts will result in multiple paths between the core and the Edge Network or host. Each path will be represented by a specific GL. The decision on which GL to use is taken within the LCS based on user or operator policies.

A re-homing event results in new paths, with new GLs becoming available, while other paths become unavailable. To maintain an ongoing session, a new GL is requested by the host having its GL affected by the event, and the correspondent host is updated with the new GL in the same fashion as for a hand-over event.

IV. Routing and Forwarding

A. Routing and Forwarding via the Core Network

End-to-end routing and forwarding is divided into two independent realms, the core network and the edge topology. To send a packet from a source to a destination, the source host queries the LCS for its own GL, and for the GL of the destination host. These GLs are used as source and destination locators in the packet header.

The path between the core network and the source host described in the source GL is used by the Edge Routers for forwarding from the source host to the core network. In the core network the CL in the destination GL is used to forward the packet, using legacy mechanisms, to a Core Edge Router near the destination host. Finally, the dot-separated list of EIDs and the NID in the destination GL is used to forward the packet to the destination host.

The next hop Edge Entity described by the GL can be resolved to an address internal to an Edge Network (e.g. an IP address) by the local routing system. The operation of this local routing system is based on traditional IP routing principles and is beyond the scope of this paper.

To reduce the amount of routing state and signaling within the edge topology, Edge Entities only keep internetworking routing state for their neighbors. Routing to destinations beyond the neighbors is performed by the routing protocol in the LCS for the subset of the end-to-end path that traverses the edge topology, and independently by the core network routing system for the subset of the path that traverses the core network. Since the Edge Entities are not visible in the core network routing system, it can be strictly isolated from the routing in the edge topology and employ legacy routing mechanisms.

B. Integrated Locator Construction and Routing

The GL describes an internetwork path between a host and the core network across the edge topology. When the LCS constructs a GL, it thus performs a routing operation by finding an internetwork path. This path has a one-to-one correspondence to the resolution path along a sequence of Attachment Registers. Whenever an AR points at several other ARs, a routing protocol is needed to select the next Attachment Register in the resolution path, e.g. to select between steps 5a and 5b in Fig. 1. This routing protocol finds the resolution path within the LCS, and thus the internetworking path and we therefore refer to it as a LCS Routing (LCSR) protocol. The Attachment Registers run this routing protocol between themselves, and thereby relieve their associated Edge Entities from many of the traditional routing tasks.

In the simplest form, the LCSR protocol is a distance vector protocol of the same type as is used among the Edge Entities to establish a default path to the core network, see section IV.E below. However, more powerful LCSR protocols can be considered that support policy routing and short-cut routing.

C. Short-cut Routing

When the source and destination hosts are attached to Edge Networks close to each other, forwarding via the core network is clearly non-optimal. For example, in Fig. 1, traffic between host *G* and host *H* can be forwarded along a short-cut path via Edge Network *f* without having to pass the core network.

To perform short-cut forwarding the LCS resolves a source-destination host pair of names into a source route describing the short-cut path across the edge topology. This source route is then used instead of the source and destination GL when forwarding traffic.

D. Policy and QoS Routing

The LCSR protocol can be designed to support policy and QoS routing. Policy and QoS parameters associated with each attachment to a neighbor Edge Network can be registered in the Attachment Registers. When performing policy routing, the LCSR protocol calculates the optimal resolution path based on the parameters registered in the ARs.

If needed, a home agent can be introduced at a fixed location in the end-to-end path. The decision to use a home agent would be based on policy criteria and could be taken to support location privacy of the destination host, to reduce the round-trip delay for hand-over signaling, or to off-load the correspondent host from the signaling caused by various mobility or re-homing events. If a home agent is desired for a specific session, the Attachment Register of the destination host returns the GL of the home agent representing the destination host, instead of returning the GL of the destination host. Note that the LLC architecture does not introduce such home agents by default, but only after explicit policy decisions.

E. Default Paths to the Core Network

When a host needs to resolve its own name, the request must be forwarded along a default path to the core as described in section III. The Edge Entities can establish this default path using a simplistic distance vector routing protocol. The only destination that is announced by this routing protocol is the core network. Each Core Edge Router thus announces the presence of the core network to its neighbor Edge Networks, which propagate the announcement using a distance vector algorithm to their neighbor Edge Networks and so forth until a host is reached. Each Edge Router or host can then calculate the next hop along the shortest path to the core network.

F. Disconnected Operation

To support operation in a part of the edge topology that becomes disconnected from the core, an Edge Network capable of handling a local DNS and LCS could be elected as a temporary core network. Using broadcast, the temporary core network solicits each Edge Entity to register its name with the local DNS, and initiate an AR associated with that name in the LCS.

V. Scalability Considerations

To maintain on-going sessions, each host that has a GL that turns stale due to a mobility or re-homing event must query the LCS for a new GL, and then register this GL with the correspondent host. Both of these actions have scalability implications as discussed below.

The LCS will receive locator construction requests from hosts with on-going sessions that have their GLs affected by a mobility or re-homing event. The signaling capacity required by the nodes in the LCS grows linearly with the number of hosts that request a new GL. A caching mechanism in the Attachment Registers would alleviate this scalability issue.

The GL update signaling to the correspondent host is not processed by the Edge Entities, or the LCS. Instead, each such signaling event is processed by the peer hosts involved in a session. There is thus only one update signaling event per host pair, and no scalability problem with regard to signaling processing capacity in the Edge Entities.

Both the signaling to request a new GL, and the end-to-end GL update signaling, require capacity for the transport of the burst of signaling that occurs after a major topology change. Again, the volume of this signaling grows linearly with the number of hosts with ongoing sessions that have their GLs affected by the topology change. The transport capacity must be dimensioned accordingly.

The core network routing system of the LLC architecture does not receive any routes from the Edge Networks. Therefore, the LLC core network has the potential to scale better than the current Internet backbone, which receives route updates due to topology changes in the edge domains.

VI. Simulation Results

To investigate the feasibility of the LLC architecture we have simulated the behavior of the basic mechanisms described in this paper using a proprietary packet-level simulator. A simplistic mobility model was used. The moving Edge Networks and hosts are initially positioned randomly within a rectangular space, and then start moving in random directions and with random speeds between zero and a maximum speed v. The hosts attach to moving Edge Networks that are within a specific range r, and each moving Edge Network also attaches to other moving Edge Networks and Core Edge Routers that are within a range r. The time varying topology that was formed with this simulation model had an average hop count of three and a maximum hop count of seven from a host to a Core Edge Router along the default path. The simulated topology consisted of 20 Core Edge Routers, 200 moving Edge Networks, and 1000 hosts.

The time to reach the connected state from an initial detached state was simulated. In the initial detached state all moving Edge Networks and hosts are located at random positions, and have neither attached to neighbor Edge Entities nor registered any neighbors with the ARs. In the connected state a source host has established a connection with a destination host located at a random position in the topology. To reach the connected state from the detached state, each Edge Entity detects its neighbors, establishes a default path to the core network, and registers the neighbors with the Attachment Register. The source host requests the construction of its source GL and then requests the resolution of the name of the destination host into the GL of its Attachment Register. Finally, the source host requests the construction of the GL of the destination host, and upon reception of this GL starts sending user data packets to the destination host. In parallel with the registration of the neighbors, the ARs of all edge entities establish a default path to the ARs of the Core Edge Routers. The distance vector protocol for default path routing to the core described in section IV.E was used with an announcement rate of the shortest path distance to the core between each pair of neighbors once every 50 ms. The add-on mechanisms for QoS, policy, and shortcut routing were not included in the simulations.

Fig. 3 shows the fraction of the host population that has reached the connected state as a function of time after the initial state where all edge entities are detached. Traces are shown for four different values of the maximum speed parameter *v*. This speed parameter is related to the range limit *r* described above so that *v* = 1 corresponds to a speed of 1 % of the range limit per second. At this speed the topology changes at such a rate that 15 % of the hosts need to update their source or destination GL every second. For larger values of *v* a substantial fraction of the host population may have to update a GL more than once before the connected state is reached, which is reflected in the simulation results.

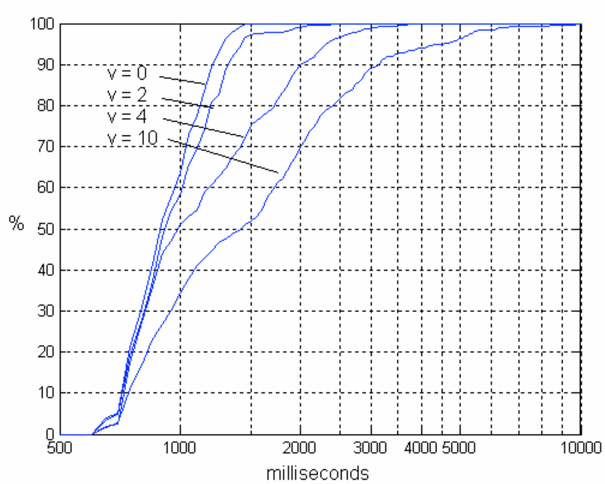

Figure 3. Fraction of the host population that has reached the connected state as a function of time after the initial state where all edge entities are detached from each other. Traces are shown for four different values of the speed parameter *v*.

The simulator does not model the impact of limitations in link bandwidth, or limitations in the capacity for processing of network signaling. The simulation results thus reflect the impact of packet loss and changes in the topology due to mobility events, but not the impact of link or processing capacity limitations.

VII. Conclusion

This paper has presented the LLC architecture, a new internetworking architecture designed to provide native support for multihoming as well as network and host mobility in dynamic edge topologies. Transport of user data via mobility agents is not needed, and the pinball routing problem can thus be avoided. A key feature of the proposed architecture is that the routing system for the edge realm keeps a minimum of routing and forwarding state. For basic connectivity without short-cut or policy routing, each Edge Entity only needs to keep state for the directly attached neighbors.

The need for future research has been identified in the areas of locator caching strategies, reduction of locator overhead, and disconnected operation. Also, the introduction of QoS, policy, and shortcut routing in the LLC architecture should be investigated further. For large access networks, optimization schemes may be needed to address capacity requirements that grow linearly with the number of edge entities.

Finally, simulation results that verify the basic functionality of the proposed architecture were presented.

Acknowledgment

We would like to thank András Méhes, Howard Green, and our colleagues in the internetworking work package of the Ambient Networks project for their feedback on this work.

References

[1] B. Ahlgren, J. Arkko, L. Eggert and J. Rajahalme, “A Node Identity Internetworking Architecture”, 9th IEEE Global Internet Symposium, Barcelona, Spain, April 28-29 2006.

[2] M. Caesar et al, “Routing on Flat Labels”, ACM SIGCOMM 2006.

[3] R. Moskowitz and P. Nikander, “Host Identity Protocol Architecture”, RFC 4423, May 2006.

[4] Ambient Networks Project web page, http://www.ambient-networks.org/

[5] Ambient Networks Project deliverable, “D11-E.1 Basic Functionality and Prototype”, FP6-CALL4-027662-AN P2/ D11-E.1, December 2006.

[6] D. Farinacci, V. Fuller, D. Oran, “Locator/ID Separation Protocol”, Internet-Draft, January 2007, work in progress.

[7] A. Valkó, “Cellular IP: A New Approach to Internet Host Mobility,” *ACM SIGCOMM Comp. Commun. Rev*., vol.29, no. 1, Jan. 1999.

[8] H. Soliman, C. Castelluccia, K. El Malki, L. Bellier, “Hierarchical Mobile IPv6 Mobility Management (HMIPv6)”, RFC 4140, Aug. 2005.

[9] C. Perkins, Ed., “IP Mobility Support for IPv4”, RFC 4721, Jan. 2007.

[10] V. Devarapalli, R. Wakikawa, A. Petrescu, P. Thubert, “Network Mobility (NEMO) Basic Support Protocol”, RFC 3963, January 2005.

[11] C. Ng, F. Zhao, U.C. Davis, M. Watari, P. Thubert, “Network Mobility Route Optimization Solution Space Analysis”, Internet-Draft, September 2006, work in progress.

[12] M. O´Dell, “GSE - An Alternate Addressing Architecture for IPv6”, Internet-Draft, February 1997, work in progress.

[13] E. Nordmark, M. Bagnulo, “Level 3 multihoming shim protocol”, Internet-Draft, November 2006, work in progress.

[14] N. Montavont et al, “Analysis of Multihoming in Mobile IPv6”, Internet-Draft, February 2007, work in progress.