This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2011 proceedings

Economically Viable Support for Internet Mobility

Ping Zhang, Arjan Durresi
Purdue School of Science,
Indiana University Purdue University Indianapolis
Indianapolis, IN 46202, USA
{zhang58, durresi}@cs.iupui.edu

Raj Jain
Department of Computer Science and Engineering,
Washington University in St. Louis
St. Louis, MO 63130, USA
jain@cse.wustl.edu

***Abstract*—The support for mobility is a major requirement for the future Internet. Although several mobility solutions have been proposed, none of them has been largely deployed, mostly because they are not economically viable.**

We propose a business oriented mobility framework - Mobility Support Service (MSS) to be offered as a value-added service to paying mobile customers. MSS uses Identifiers (IDs) to locate the nodes and network addresses as points of attachment for the nodes. MSS is a scalable distributed service. MSS does not require changes in existing network architecture nor services. We present two MSS distributing algorithms that enable tradeoffs among costs and QoS. We show by simulations the advantages of MSS compared to existing mobility solutions, especially regarding scalability and service delay. However the main advantage of MSS is that it can be realistically offered, because it is economically viable.

I. Introduction

Mobility is a primary requirement for the future Internet; recent studies predict that mobile data traffic will double every year through 2014, increasing 39 times between 2009 and 2014 [1], [2]. Therefore, significant research efforts are dedicated to find solutions for mobility in the Internet [3]–[15].

The overloaded IP address is recognized as a major impediment for Internet mobility [6], [7] and various solutions to split ID and addresses have been proposed [8]–[11]. Other proposals include routing via invariant intermediate points [12], [13], or migrating connections from old addresses to new ones [14], [15]. However, none of the proposed solutions for Internet mobility has been largely deployed, and the only currently widely used method to access Internet in mobile manner is through cellular networks.

The Internet experience clearly indicates that no solution will be used in the Internet if it is not economically viable, regardless of how technically sound the solution is. For example, various QoS solutions, including Differentiated Services (DiffServ) and Integrated Services (IntServ), whereas considered "technically" scalable, after more than a decade of intense research, and implementation in almost all endpoints and routers, are not being used extensively, mostly because they are not economically viable in the Internet. On the other hand, *Skype* has been successful, because users pay for the QoS of their applications (voice and video) and the corresponding service providers generate revenues out of such services.

We believe that several critical economic flaws have impeded the use of many proposed mobility solutions in the Internet. *First*, existing solutions, based on static intermediary forwarding, such as Mobile IP [12], HIP [10], [16] and similar ones, require modifications to access networks. But, such ubiquitous deployment of network changes do not offer sufficient economic incentives, especially for service providers. *Second*, existing solutions require that all Internet users pay the cost of deployment and operations of the proposed mobility support, even though a large portion of users may not be willing to pay for mobility. *Finally*, in existing solutions, while technical collaborations among involved service providers are required, it is very difficult for them to split the revenues. For example, how can a home agent in MIP have contracts and share mobile revenues with random foreign agents?

We propose Mobility Support Service (MSS) that can manage mobility for its mobile customers. MSS will be offered by service providers dedicated to mobility, called Mobility Service Provider (MSP). MSP role could be played by existing service providers too. MSS will be offered as a value-added service to mobile customers who are willing to pay for it. Therefore, mobility support will generate its own revenue and justify the business and investments of MSPs. MSS does not require any change on access networks, existing network infrastructure, legacy applications, and operating systems.

MSS will be a distributed service over the Internet. Therefore, the quality of mobility support (latency and availability), as well as the cost of providing such a service will depend closely on the level of distribution of MSS. In this paper we propose two distribution algorithms of mobile profiles that enable the exploration of tradeoffs between the quality of mobility support and the network cost.

The paper is organized as follows: In Section II we review related works. Section III describes our MSS, including architecture and elements. In Section IV, we discuss two algorithms for location management based on the defined cost function. In Section V, we compare our algorithm to existing solutions by simulation. We conclude in Section VI.

II. Related work

Mobile IP (MIP) [12], [17], [18] and its enhancements [19], [20] are among the most popular solutions proposed to support mobility. MIP-like solutions require modifications on access networks and collaborative support from both home and foreign service providers [13], [21]. Therefore, both agents should have contracts that governs their collaboration, including sharing of revenues from their services, but such contracts are very difficult to implement among random pairs of agents (service providers.) Furthermore MIP suffers also from non optimal routing and triangulation.

In Host Identity Protocol (HIP) [10], [16], the IP address is split from the identifier, called Host Identity (HI). HI is initially acquired by DNS lookup [22], and mobile node keeps updating peers and DNS record during move. But DNS cannot support mobility, because its updates are slow [23]. For highly mobile nodes a Rendezvous Server (RVS) is proposed [24], but its distributed implementation is not discussed. Furthermore, HIP is proposed to be an Internet service, similar to DNS, therefore paid by all Internet users, mobile or not. Furthermore, MSS could be used as a framework to implement HIP.

The Locator/Identifier Separation Protocol (LISP) [11], [25] is a clean-slate architecture based on the separation of identifiers, called Endpoint Identifiers (EIDs), and addresses, called Routing Locators (RLOCs). EID-to-RLOC mapping are performed at the RLOC router and routing is accomplished by tunneling between RLOC routers. Several overlay mobility solutions have been proposed [9], [26]–[29]. Balakrishnan et al. [30], [31] propose two more levels of name abstraction, besides endpoint identifiers and IP addresses, namely user-level descriptors and service identifiers. FARA suggests to use rendezvous points to setup an initial connection to the mobile node, or use directory service (fDS) to lookup and keep track of the mobile node, though the mechanism is not discussed. Session Initiation Protocol (SIP) [32] is a signaling protocol used to set up and manage connections between end hosts, and it can be used to support mobility by setting up session and updating the IP address after the corresponding node moves [8], [33]. Similarly to MIP and HIP, all these proposed solutions are not economically viable, because (1) all Internet users have to pay the mobility service, and (2) when home and foreign service providers are required to collaborate, it is very difficult for them to have contracts and share revenues of the service.

III. Mobility Support Service (MSS)

MSS involves three entities: mobile subscribers, Mobility Service Providers (MSPs), and communicating peers. Subscribers or customers are mobile users who are provided with unique IDs and pay their MSP for the corresponding mobility services. An MSP is an organization that manages IDs, mobility and related functions for its subscribers. Communicating peers or simply users are Internet endpoints who use the MSP services to lookup for the current subscriber's address and related information. Besides mobility management, MSP could buffer traffic for sleeping devices, translate among communication protocols, as well as offer security and privacy support. Therefore, MSS will be offered as add-on services only to paying subscribers, and MSPs will generate their own revenues to sustain the MSS. A given MSP could have distributed presence in specific areas on the Internet in order to serve better its mobile subscribers and their communicating peers, or multiple MSPs could have contracts to collaborate and share revenues to serve better their subscribers, similarl to cellular providers's collaboration to provide rooming and split the revenues accordingly. The level of MSS distribution will depend on the desirable tradeoffs among QoS, security and associated costs.

A. MSS Architecture

In MSS, IP addresses are used as locators representing the current point of attachment. Fig. 1 illustrates the architecture of MSS. An MSS client application, called Mobility Support Layer (MSL), is installed on the nodes of mobile subscribers and users that would use MSS. Applications that want to utilize MSS are registered manually or automatically at the local MSL. When a registered application initiates a connection to a subscriber, MSL intercepts the connection setup system call and resolves (lookup) the destination ID to the corresponding current IP address by retrieving it from the MSS. Therefore, the ID resolution is transparent to applications. Furthermore, MSL can obtain further information from MSS, such as security keys to be used in communications with the corresponding mobile subscriber. Then initiating MSL will negotiate with the MSL on the mobile node, and will create the appropriate connection to current subscriber's IP address. Similar tasks are performed at the subscriber's side too. When both sides are mobile nodes, the ID resolution would be performed in reverse direction as well, and the two corresponding MSPs may not be necessarily the same.

End-to-end authentication and symmetric key generation are performed for each address change or timeout, using the public key acquired from MSS. No special routing requirement is needed to deliver the packet to the destination IP address, and the routing mechanism is completely unaware of the mobility support.

B. Mobility Service Provider (MSP)

Mobility Service Provider (MSP) is the key part of MSS. MSP is a dedicated entity that provides mobility management and other related services. A given MSP will have a number of servers distributed at specific subnets or backbones according to its commercial interest. Various MSPs could have agreements among them in order to increase the distribution of their businesses, similar to roaming agreements among cellular providers. The QoS of MSS offered by MSPs will be measured mostly by two components: (1) the delay to update a new location for a given mobile subscriber and (2) the delay experienced by an Internet user when looking up for the location (current IP address) of a given mobile subscriber (ID).

The level of distribution of MSS servers and their network topology will depend on the tradeoffs among the number of subscribers, their mobility characteristics, offered QoS and service costs. More servers with replicas of ID - current IP and with more connections will cost more, but will be able to provide a better QoS to more subscribers and users.

Therefore, such factors will be decided by each MSP based on its business goals and have to be balanced with the revenues from mobile subscribers. For example, a larger number of MSS servers would lead to better availability and QoS, but at the same time to higher costs. Furthermore, as it is well known in distributed systems, more replicas will lead to more synchronization messages among them. Therefore, the level of MSS distribution and of mobility data replication among them are the key decisions to be made by an MSP. In the next section we will introduce a basic cost function to be used in the optimization of the replication process.

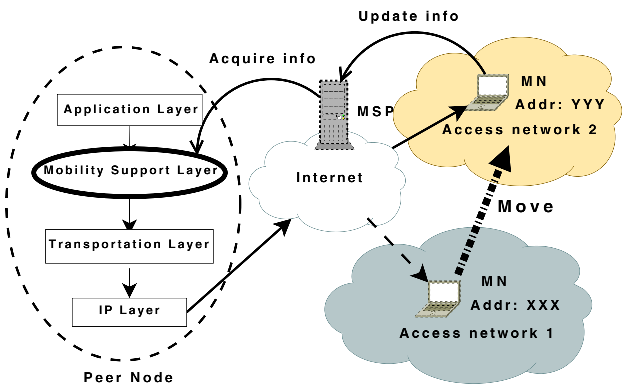

Fig. 1. The architecture of MSS

IV. Replicated Location Management

Location management, as a crucial building block of Internet mobility support, is evaluated by two major criteria: availability and latency to access the profiles of subscribers. The key to make location management reliable and efficient is to distribute profiles properly, that means as close as possible to the place where they are used.

We use the terms: "lookup" for profile query and "update" for profile modification. For a given lookup request, it would be desirable to get the reply from a replica near by rather then from one far away. Furthermore, when we have multiple replicas of a given profile, we need to update all of them when a modification is made, which is called "synchronization." Each synchronization will induce certain amount of overhead.

Ideally if we have enough resources and do not consider economic factors, we may setup replicas at all possible locations so that any lookup requests from all places can be served by nearby replicas. However, this solution is impracticable due to the cost of servers and the cost of synchronizing all replicas. Therefore, when there are enough accumulated lookup requests, such that the cost of synchronization would be surpassed by savings, a replica should be created closer to its users. So, the decision of profile replica setup and removal is determined by the number of lookups and update requests at specific places.

The proposed MSS enhances the performance and availability of location management system by replicating profiles of a mobile subscribers at multiple locations, and dynamically adjusting the replica distribution to balance the cost of implementing replication and overall performance. The primary benefit of replication is that MSS subscribers and their peer communicators could have a guaranteed service performance, such as latency bound for address update and lookup. However, the replication has to be handled carefully as a tradeoff among the improvements in user's QoS (lookup), subscriber updates, and replica synchronization and replica distribution costs.

Our approach in MSS replication is based on previous work in distributed systems, databases and cellular mobility systems [34], [35]. To be able to optimize the replication process, we developed a cost function that captures various network costs and user's QoS.

A. Cost Functions

We develop a cost function to be used by replication algorithms in exploring tradeoffs among performance, QoS and various costs (communication, computing, and infrastructure). The cost function reflects the goal of incorporating the effect of replications in improving user's QoS, as well as the cost in replica implementation and synchronization. Using this model, we then outline its possible use for exploring strategies to realize different objectives that balance user goals and network costs.

The whole network is modeled as a tree. Each node of the tree represents a MSS server, and the requests from subnetworks without MSS servers are combined into the nearest ancestor network that has a server. For a replica update or lookup request originated in subnet x to subnet y, the cost $C(x, y)$ is represented as:

$$C(x,y) = \sum_{\forall}(\kappa_1 metric_1(x,y) + \kappa_2 metric_2(x,y) + ...) \quad (1)$$

The value of each metric is normalized and converted to a neutral unit, and then summed together, where $metric_1(x, y)$ is the neutral unit conversion of cost based on *metric* or *policy 1*, and κ_1 is the weight coefficient of $metric_1(x, y)$. *Metrics* will include communication delay, processing delay, monetary costs of processing and communications in Internet.

A simple model for the cost of a lookup request for a mobile node A, l^A, consists of three parts: cost $c(x, y)$ for the request message traveling from requesting node in subnet x to corresponding server in subnet y; cost $c(y, x)$ for response message traveling back in the reverse direction; and possible propagation cost $Prop_l^A(y)$ of notifying necessary servers and possibly adjusting replica states, if any, originating from subnet y. Therefore:

$$l^A(x) = c(x,y) + c(y,x) + Prop_l^A(y) \quad (2)$$

Similarly, the cost u^A of an update request sent by node A can also be divided into three parts: cost $c(m, n)$ for the request message traveling from A in subnet m to corresponding server in subnet n; cost $Sync^A(n)$ of all sync messages traveling to synchronize profiles where the first one originates from subnet n, and cost of propagation $Prop_u^A(n)$ which is similar to $Prop_l^A(y)$. Therefore:

$$u^A(m) = c(m, n) + Sync^A(n) + Prop_u^A(n) \quad (3)$$

We define L^A and U^A as sums of serving all lookup and update requests for mobile node A during a given period of observation, that is:

$$L^A = \sum_{\forall i} l^A(x_i) = \sum_{\forall i} \left(2 \times c(x_i, y_i) + Prop_l^A(y_i)\right) \quad (4)$$

and

$$U^A = \sum_{\forall j} u^A(m_j) = \sum_{\forall j} \left(c(m_j, n_j) + Sync^A(n_j) + Prop_u^A(n_j)\right) \quad (5)$$

Therefore $L^A + U^A$ is the cost function or optimization goal of the location management system for node A. By summing up the cost functions for all mobile nodes we derive the total cost function of the system C_t.

We have developed two MSS replication algorithms that maximize C_t: (1) The offline algorithm that computes optimal replica distributions for request statistics of a known past period from a global view, and (2) the online algorithm that adjusts local replica status according to limited local knowledge obtained from online request for lookup from Internet users and location updates from mobile MSS subscribers. These problems are NP-hard problems so optimization is usually conducted on reduced problems or targeted at sub-optimal solutions.

B. *Offline Replication Algorithm*

In the offline algorithm, we assume universal knowledge of all updates and lookup requests during a given period of time. The value of offline algorithm is mostly theoretical, to show the performance upper bound. Furthermore, the offline algorithm can be used for planning and periodic cost/performance auditing. Since the replica set of offline algorithm is static, there is no cost associated with replica change, i.e., there is no need to propagate lookup requests to any other servers nor to propagate update requests to non-replica servers. Thus, $Prop_l^A$ and $Prop_u^A$ are both zero in the offline algorithm and the corresponding simulation statistic.

Similar topics, such as facility location analysis have been researched in operations research area for many years. Such techniques have also been adapted by cellular and PCS network research [35]. These problems are NP-complete problems for a general graph, since it needs to consider all possible solutions and replica layout are highly dependent (i.e., the set up of a replica at a specific location is based on the existence and position of all other replicas) [36]. Therefore, optimization is usually conducted on reduced topology, i.e., a tree abstracted from general graph, and this optimization problem can be reduced to a $p-median$ problem in Discrete Location Theory, which can be solved by using existing dynamic programming (DP) solutions. A brief algorithm is shown below, in which $\beta(i, j)$ is the cost to serve lookup requests originated from node j at node i; $\alpha(i, T_j)$ is the optimal cost of serving subtree T_j, including all lookup and update requests originating from T_j, when served by a replica at node i; and $\alpha(T_j)$ is the optimal cost of serving subtree T_j when replicas are all within. After having optimal cost computed for each subtree, the optimal replica set can be retrieved.

Algorithm 1 Offline replication algorithm

Sum up update $N_U(x)$ and lookup stats $N_L(x)$ for each node x.
Calculate β matrix for each pair of nodes.
Compute α of each leaf node.
Starting from bottom nodes until root, do:
for all j As a non-leaf node **do**
    for all i As each other node **do**
        Compute $\alpha(i, T_j)$
    end for
    Compute $\alpha(T_j)$
end for

C. *Online Replication Algorithm*

The proposed online algorithm runs in a distributed manner and each MSS server node only has local knowledge and can only communicate with its neighbors. Therefore, the relation of each pair of neighbor nodes can be represented as status of edges, as well as the relation change represented as edge status change. Edge algorithm is the core of the online algorithm.

An offset vector is used to store the status of an edge, and is changed when either side of edge receives a request according to a transition table. The offset vector is used to represent all past information based on optimal replica selection, and will not be affected by the actual replica status, i.e., it only captures relative statistic of optimal request history. We built on Lund's algorithm [37] to adjust replica status.

The edge algorithm, offset and replica update, is performed at each node involved in a request processing. When a request arrives at a subnet server, either lookup or update, the server will propagate the request to the appropriate neighbors according to offset vectors, if it cannot serve the request locally. Synchronization messages and propagation messages are also delivered in the same way.

V. Simulation and Comparison

The major concern in replication is its scalability. Therefore, we conducted a simulation to evaluate the signaling overhead of our algorithms. For simplicity, only network hop count of message traveling is used as network cost in this simulation. HIP rendezvous server (RVS) model [24] is used as reference: an unique and fixed server is responsible for storing a mobile node's address, and consequently no additional cost other than serving lookup and update, such as synchronization will occur, as there is no distributed structure proposed for HIP.

Algorithm 2 Online replication edge algorithm

```
For request submitted locally or propagated from the other
  end of edges, adjust the offset vector F.
if lookup request then
  if having a local replica then
    Return profile from incoming edge
    Propagate notification to all other edges, if F changes
  else
    Propagate request towards replica set from one edge
      indicated by F value
  end if
else
  if having a local replica then
    Update local profile
    Propagate to neighbors which have replica, indicated
      by F value
    Propagate notification to all other edges, if F changes
  else
    Propagate request towards replica set from one edge
      indicated by F value
  end if
end if
Set new replica state S if needed, base on new offset
  vector F.
```

For MSS, request messages are usually served by the nearest MSS server, and then synchronization messages are further propagated along the replica tree, plus possible replica status change. An abstract network topology is used that consists of subnets and links connecting subnets and comprises of ten tier-1 subnetworks, 21 tier-2 subnetworks and 77 tier-3 subnetworks, as shown in Fig. 2.

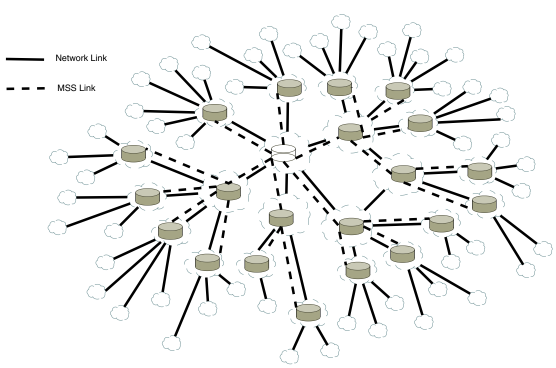

Fig. 2. Simulation network topology

For the simulation duration, each mobile node moves 20 times, only at leaf level subnets, and during simulation one mobile node will be looked up for 200 times , for a 1:10 update/lookup ratio. The update and lookup pattern for each mobile node is random generated, and then 200 rounds of trace are generate based on the pattern. Each mobile node moves random distance complying to Binomial distribution with $p = 0.5$ every time, and random lookup request locations comply to a uniformed distribution across all leaf subnets. Requests in each generated round have random variance. We use 86400 seconds (or 24 hours) as total timestamp for a round. Lookup timestamp variance is 1800 (equivalent to 41.7%) and update is 900 (equivalent to 208%). Location variance is 2.81% for update request and 1.40% for lookup requests.

One hundred mobile nodes are simulated, and the result shown in the graphs are numerical average of the sum of 100 node traces, to reduce interference of random errors. The unit of time in the graphs corresponds to events of lookup requests, i.e., 10 units of time roughly contains 10 lookup and 1 update requests.

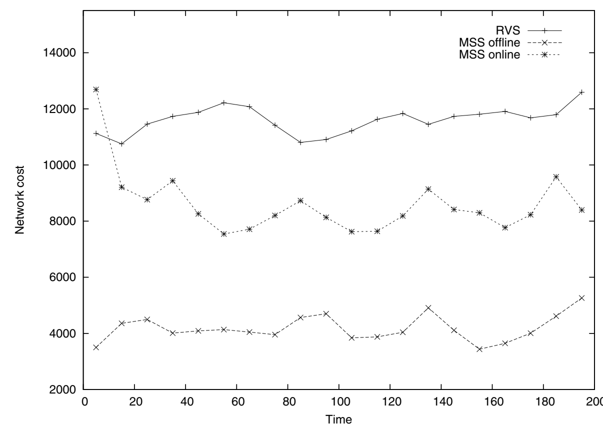

Fig. 3. Comparison of total signaling overhead.

From the data shown in Fig. 3 we can see that both offline and online algorithms outperform the RVS/HA solution. The overall network cost of the offline algorithm defines the performance boundary of 3-competitive online algorithm.

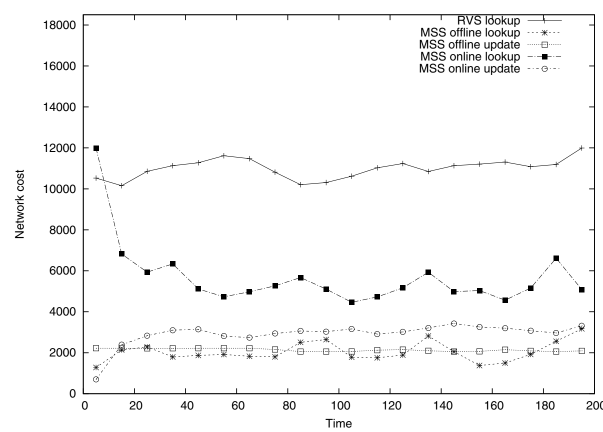

Fig. 4. Comparison of update and lookup overheads.

Fig 4 reveals further interesting results. The lookup overhead dominates the overhead of RVS. Although RVS has so few updates, its lookup cost, which also represents latency here, is two to five times higher than that of our solutions.

VI. Conclusion

Mobile Support Service MSS is a framework that aims to offer economically viable support for Internet mobility. MSS is provided as a value-added service, and does not require changes on access and network infrastructure. MSS will be offered by Mobility Service Providers that will be payed by their mobile subscribers. Various MSPs could collaborate and share corresponding revenues, based on contracts. We developed a cost function that can be used to explore the desirable tradeoffs among user QoS and network costs. We proposed two algorithms that can manage the profiles of subscribers by optimizing the corresponding cost function.

Acknowledgments

This work was supported by the National Science Foundation under Grants No. 1019120, 1019119 and by CACR IU.

References

[1] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2009-2014," available at http://www.cisco.com, Jun. 2010.

[2] ——, "Hyperconnectivity and the Approaching Zettabyte Era," available at http://www.cisco.com, Jun. 2010.

[3] T. R. Henderson, "Host mobility for IP networks: a comparison," *IEEE Network*, vol. 17, pp. 18–26, November 2003.

[4] W. M. Eddy, "At what layer does mobility belong?" *IEEE Communications Magazine*, vol. 42, pp. 155–159, October 2004.

[5] D. Le, X. Fu, and D. Hogrefe, "A review of mobility support paradigms for the internet," in *Communications Surveys and Tutorials, IEEE*, vol. 8, August 2006, pp. 38–51.

[6] D. Clark, R. Braden, A. Falk, and V. Pingali, "Fara: reorganizing the addressing architecture," in *FDNA '03: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*. New York, NY, USA: ACM, 2003, pp. 313–321.

[7] D. D. Clark, K. Sollins, J. Wroclawski, and T. Faber, "Addressing reality: an architectural response to real-world demands on the evolving Internet," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 4, pp. 247–257, 2003.

[8] E. Wedlund and H. Schulzrinne, "Mobility support using SIP," in *WOWMOM '99: Proceedings of the 2nd ACM international workshop on Wireless mobile multimedia*. New York, NY, USA: ACM, 1999, pp. 76–82.

[9] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," in *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2002, pp. 73–86.

[10] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host Identity Protocol," RFC 5201 (Experimental), Internet Engineering Task Force, Apr. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5201.txt

[11] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "Locator/ID Separation Protocol (LISP), draft-ietf-lisp-03.txt," July 2009. [Online]. Available: http://www.ietf.org/id/draft-ietf-lisp-03.txt

[12] C. Perkins, "IP Mobility Support for IPv4," RFC 3344 (Proposed Standard), Internet Engineering Task Force, Aug. 2002, updated by RFC 4721. [Online]. Available: http://www.ietf.org/rfc/rfc3344.txt

[13] M. Buddhikot, A. Hari, K. Singh, and S. Miller, "Mobilenat: a new technique for mobility across heterogeneous address spaces," *Mob. Netw. Appl.*, vol. 10, no. 3, pp. 289–302, 2005.

[14] A. C. Snoeren and H. Balakrishnan, "An end-to-end approach to host mobility," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2000, pp. 155–166.

[15] I. Aydin, "Cellular SCTP: A Transport-Layer Approach to Internet Mobility," in *Computer Communications and Networks, 2003. ICCCN*, 2003, pp. 285–290.

[16] J. Laganier, T. Koponen, and L. Eggert, "Host Identity Protocol (HIP) Registration Extension," RFC 5203 (Experimental), Internet Engineering Task Force, Apr. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5203.txt

[17] C. Perkins, P. Calhoun, and J. Bharatia, "Mobile IPv4 Challenge/Response Extensions (Revised)," RFC 4721 (Proposed Standard), Internet Engineering Task Force, Jan. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4721.txt

[18] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775 (Proposed Standard), Internet Engineering Task Force, Jun. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3775.txt

[19] I. Akyildiz, X. Jiang, and S. Mohanty, "A Survey of Mobility Management in Next-generation all-IP-based Wireless Systems," in *Wireless Communications, IEEE*, vol. 11, August 2004, pp. 16– 28.

[20] E. Perera, V. Sivaraman, and A. Seneviratne, "Survey on network mobility support," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, no. 2, pp. 7–19, 2004.

[21] J. Kempf, "Goals for Network-Based Localized Mobility Management (NETLMM)," RFC 4831 (Informational), Internet Engineering Task Force, Apr. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4831.txt

[22] P. Nikander and J. Laganier, "Host Identity Protocol (HIP) Domain Name System (DNS) Extensions," RFC 5205 (Experimental), Internet Engineering Task Force, Apr. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5205.txt

[23] J. Day, *Patterns in Network Architecture: A Return to Fundamentals*. Prentice Hall, December 2007.

[24] J. Laganier and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension," RFC 5204 (Experimental), Internet Engineering Task Force, Apr. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5204.txt

[25] B. Quoitin, L. Iannone, C. de Launois, and O. Bonaventure, "Evaluating the benefits of the locator/identifier separation," in *MobiArch '07: Proceedings of first ACM/IEEE international workshop on Mobility in the evolving internet architecture*. New York, NY, USA: ACM, 2007, pp. 1–6.

[26] D. ANDERSEN, "Improving end-to-end availability using overlay networks," 2005. [Online]. Available: citeseer.ist.psu.edu/andersen05improving.html

[27] S. Zhuang, K. Lai, I. Stoica, R. Katz, and S. Shenker, "Host mobility using an internet indirection infrastructure," *Wirel. Netw.*, vol. 11, no. 6, pp. 741–756, 2005.

[28] S. Guha and P. Francis, "An end-middle-end approach to connection establishment," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 193–204, 2007.

[29] Y. Mao, B. Knutsson, H. Lu, and J. M. Smit, "DHARMA: Distributed Home Agent for Robust Mobile Access," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies 2005 (INFOCOM 2005)*, vol. 2, 2005, pp. 1196–1206.

[30] A. C. Snoeren, H. Balakrishnan, and M. F. Kaashoek, "Reconsidering internet mobility," in *In Proc. HotOS-VIII*, 2001, pp. 41–46.

[31] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish, "A layered naming architecture for the internet," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, Aug 2004, pp. 343–352.

[32] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261 (Proposed Standard), Internet Engineering Task Force, Jun. 2002, updated by RFCs 3265, 3853, 4320, 4916, 5393. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt

[33] R. Shacham, H. Schulzrinne, and S. Thakolsri, "Session Initiation Protocol (SIP) Session Mobility, Request for Comments: 5631," October 2009. [Online]. Available: http://tools.ietf.org/html/rfc5631

[34] O. Wolfson, S. Jajodia, and Y. Huang, "An adaptive data replication algorithm," *ACM Transactions on Database Systems*, vol. 22, no. 2, pp. 255–314, June 1997.

[35] K. Q. Tian and D. C. Cox, *Mobility Management In Wireless Network: Data replication strategies and applications*. Boston: Kluwer Academic Publishers, December 2004.

[36] O. Wolfson, S. Jajodia, and Y. Huang, "An adaptive data replication algorithm," *ACM Transactions on Database Systems*, vol. 22, pp. 255–314, 1997.

[37] C. Lund, N. R. J. Westbrook, and D. Yan, "Competitive on-line algorithms for distributed data management," *SIAM Journal on Computing*, vol. 38, no. 3, pp. 1086–1111, March 1999.