IndirectionDC: Using Data Centers as Internet Indirection point to tackle Mobility

Mamoon Raja
Tufts University

Abstract

Mobile devices have been very popular in today's world. But current internet infrastructure lacks the ability to deal with very basic mobility related issues. Several solutions have been proposed in past decade to solve such issues, but they are either not practical or need significant changes to internet infrastructure. We are going to overview these solutions, especially the ones using indirection to solve mobility issues, and we present a new approach that utilizes the datacenters to solve the mobility issues using indirection approach. Proposed approach uses datacenter nodes as the indirection point to solve the problems related to mobility.

Keywords

Mobility; Data Centers; Indirection

1. INTRODUCTION

These days mobile devices have become an important part of our lives, ranging from laptops to smart phones to wearable devices. More popular these devices will become, scenarios involving degradation in service due to mobility will increase too. Sales of android smart phones alone has reached 1 billion for year 2014 [8]. Similarly, global mobile data traffic has seen an increase of 74 percent in 2015. Which was increased from 2.3 exabytes per month in 2014 to 3.7 exabytes per month at the end of 2015 [5]. Looking ahead, we can see that the Internet of Things (IoT) will exponentially grow in the coming years [7], it is expected that we will have more than 30 billion IoT devices installed by 2020.

Current internet architecture lacks the ability to deal with very basic scenarios involving mobility. For instance, consider a simple client server scenario, where both client and server are using TCP/IP protocol, everything is working fine, suddenly client decides to move. The way current internet is designed, whenever a mobile device switches network, new IP address is assigned to it, the connection to the server is reset and any information about state is also lost. So client has to reconnect to server again and user will suffer with degradation of service.

Several solutions have been proposed to solve the mobility problems. Indirection based techniques have been used in past [13, 10, 20], we build on current solutions using indirection, and we propose IndirectionDC, an indirection based approach to assist end to end communication in mobility scenarios. We use data centers as indirection point to provide seamless handover while switching between different networks. In our approach, data center is handling all of user's connections to web services, and whenever user moves from one network to another it will not lose connection to the server because data center is doing state keeping for the user. And user can easily reconnect by going back to server via data center node. Our solution relies on the fact that availability of public data centers has been increased in last few years. Amazon, Google and Microsoft have data centers available all over the globe [1, 4, 2].

Current solutions to tackle mobility either lacks simplicity or need significant changes to current infrastructure. We are revisiting the indirection model, and instead of using indirection at home router like previous approaches, we are using data centers, which leads to an application layer solution. Our results suggest that we have improvements in end to end round trip times 40 percent of times, and even for cases where we have significant degradation, indirect route latency still lie under 100ms.

2. INTERNET AND MOBILITY

In current internet, IP addresses and domain names are primarily used to identify a particular end system. End system addresses can be associated with any autonomous systems, routers, computers or mobile devices. And to make end to end communication possible, most popular transport protocol used is Transmission Control Protocol (TCP) [14]. TCP is a connection oriented protocol, it relies on state keeping and needs information like IP address, port number of the end hosts. But TCP can not work if end host changes his IP address every now and then, because it can only support end to end communication if end host IP addresses are static.

In contrast, mobile devices are equipped with necessary tools to provide mobile support. A very common scenario is switching from user's home Wifi network to a cellular network. For scenarios like that, whenever a user moves it changes the IP address and server side TCP will fail to recognize the user anymore and the connection is lost. Due to this reason using legacy TCP/IP will result in poor performance.

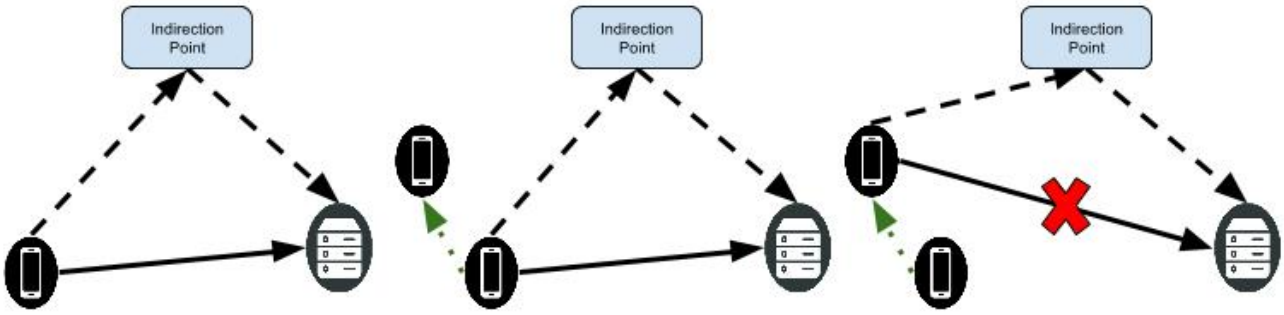

Figure 1: Indirection Model a) Mobile device talking with server via direct route and indirect route using indirection point b) Mobile device moves to new location c) Mobile device can not talk to server using direct path anymore

3. INDIRECTION MODEL

Our design consists of a very simple setup, where we have a fixed node which is used as an Indirection point and any mobile device can connect to peers using the indirection points. First, we have a simple scenario as shown in in Figure 1-a, where a mobile user residing in home network can talk to server using two alternate paths, either the direct (path shown by solid line) or the indirect path (shown by dashed line). Direct route is the route used by the end points for normal communication, whereas indirect route is the additional path supported by our indirection model.

Later, user decides to move and as a result network is switched from from home network to some new network (shown by green dotted line). Now, end to end nature of current internet will not allow the user to talk directly with the web server, because user has switched network and the IP address of user's mobile device has been changed and server does not recognize the IP address of mobile end host anymore. Instead, if we are using the indirection points to talk to server, our connection will remain establish and we can reconnect via indirection point again.

3.1 Indirection in Mobile IP

Mobile IP takes advantage of indirection and uses Indirection model to tackle mobility, it is important to briefly discuss Mobile IP before we move on to our solution. Mobile IP relies on home agent as an indirection point to tackle mobility [13], where home agent is a router on a mobile node's home network. Home agent will route traffic to mobile nodes whenever it is away, it also maintains the location information. Another key feature of Mobile IP is Foreign agent, which is a router on a network visited by mobile node, it cooperates with the home agent to keep mobile node connected to server. Mobile IP mainly consists of three main functions:

- **Agent Discovery**, Home agent and/or Foreign agent advertise its availability on each link.
- **Registration**, When the mobile node is away from home, it registers foreign agents address with its home agent.
- **Tunneling**, Home agent tunnels the traffic for mobile node to the foreign agent.

3.2 Using a Data Center as an Indirection point

Even though Mobile IP is a complete solution, it still involves making changes to IP substrate. So, we propose to use data center nodes as the indirection point, using a data center node as an indirection point remove the need for changing IP substrate. Our approach is simplistic, it does not alter anything underneath application layer. User can simply send every data packet to data center with server's information and data center can route the traffic to destination server. Various cloud providers have made their data centers available for public use [1, 2, 4], which motivates our case for using data center as an indirection point.

4. CAN INDIRECTION HELP?

In this section we will evaluate our strategy using set of carefully designed experiments. To test our indirection model we have used the same setup as shown in Figure 1. Azure Data centers at 4 locations around USA are used as indirection point. We have used Planet lab nodes [6] to run our experiments, planet lab nodes are selected based on their availability, responsiveness and geographic location, and after careful filtering 38 healthiest nodes available are selected. Selected nodes are spread around various locations in USA.

Main goal of our experiments is to calculate the latency overhead of using data centers as indirection point. We are calculating round trip times using simple Ping experiments on Planet Lab Nodes, whereas for Azure nodes we are using TCP ping (an open source variant of ping). We are interested in calculating the latency stretch, which is defined as the ratio of indirect path latency and direct path latency, for every path we are calculating it by dividing indirect path latency by direct path latency. Since we are using *four* Azure data centers at different locations, we need a strategy for client to decide which data center to use as indirection point. We are

using two methods to do that:

Nearest Data Center

Simplest way for a user to select a data center is by selecting the nearest data center. Out of all the data center nodes available, we select the one with the minimum latency from source node.

Best possible Route

Selecting nearest data center strategy is agnostic from data center to destination path. So we propose a more sophisticated approach, user considers the path between data center to destination too and selects the best possible route available.

4.1 Experiment 1: Top 100 Web Servers

First, we test our methodology on traffic from 38 Planet lab nodes acting as end users to top 100 Alexa web servers, we are using *four* Azure data center nodes as indirection points. We are calculating direct path latency of every path by pinging all of the web servers directly from every planet lab node. For indirect paths, we are running two stages of pings from every Azure data center to planet lab nodes and to web servers.

Cumulative Distribution Function (CDF) of Latency stretch is shown in figure 2. We can see that best route strategy performs better than the nearest DC strategy. We are even getting benefits in latency for 30% of times and for 90% of times it stays below 10, and we also observe a tail after latency stretch get larger than 10.

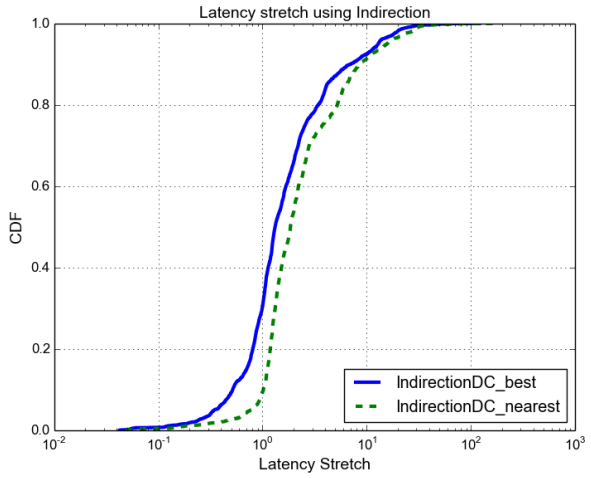

Figure 2: Latency stretch using Data centers as Indirection point and Top 100 Alexa's Websites

4.2 Experiment 2: Planet Lab to Planet Lab

Using top 100 web servers to estimate latency stretch has an inherent bias because we are only using Azure data centers, so setting in our first experiment will favor web servers using Azure's cloud, and similarly can provide bad estimation for websites hosting on other Cloud services. Since only 30% of web providers are using azure's platform [15], so we need to test our approach in more neutral settings. For our second experiment we are using each of 38 Planet lab nodes as client, and for every planet lab node we are using other 37 nodes as web servers, and we are still using Azure data center nodes as indirection point. As Figure 5 shows, we are getting better performance now. Around 40% of times we are getting improved performance for this setting and tail has also been decreased.

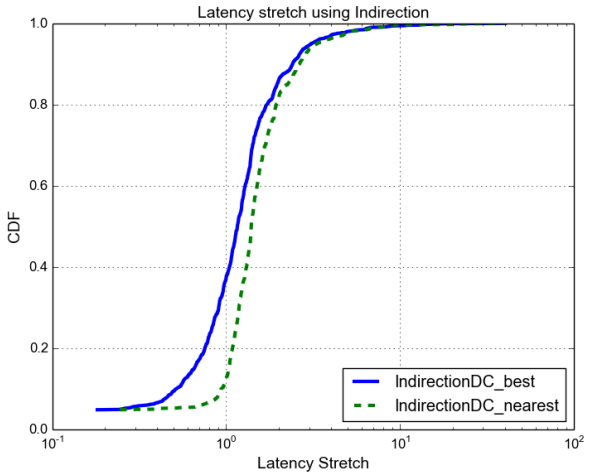

Figure 3: Latency stretch using Data centers as Indirection point and Planet Lab nodes as servers

4.3 Experiment 3: IndirectionDC vs MobileIP

Now we introduce the notion of mobility and compare our approach with Mobile IP. We have four pairs of planet lab nodes, pairs of nodes are used to introduce the notion of mobility. Mobility conditions are satisfied by selecting each pair on following criteria, for every node pair P with nodes n_1 and n_2:

- Both n_1 and n_2 are in USA.
- Both n_1 and n_2 are in same city/geographical area, but does not belong to same institution.
- Both n_1 and n_2 are active.

Finding pairs satisfying above criteria is a difficult task, because not all the planet lab nodes are active all the time. Furthermore, with geographic constraints, it becomes more difficult. For this reason we are only using five pairs for this experiment.

For 20% of times, our approach competes with Mobile IP but Mobile IP performs slightly better for the rest. 80% of times Mobile IP has latency stretch of 1.5, whereas IndirectionDC has latency stretch of 3. One reason behind the difference is that we are using a relatively smaller data set

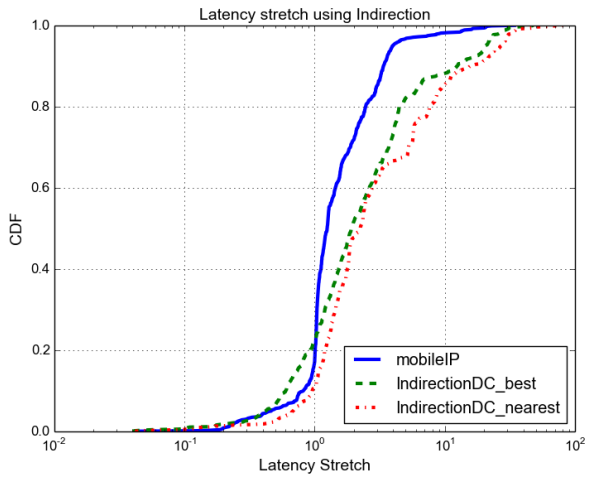

Figure 4: Latency stretch using Data centers as Indirection point and Planet Lab nodes as servers

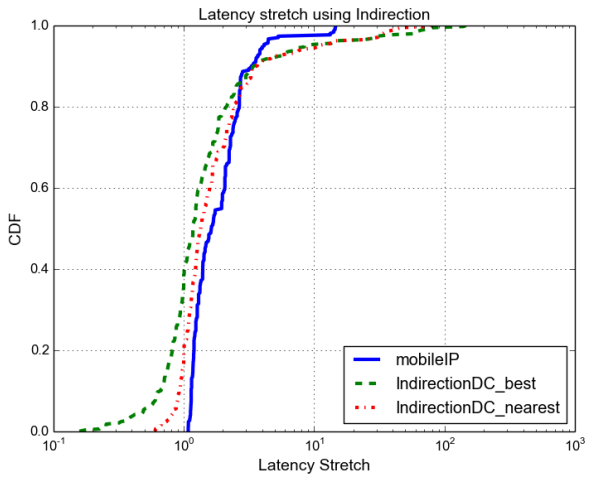

Figure 5: Latency stretch using Data centers as Indirection point and nodes from iPlane datset

and pair of nodes in Michigan are affecting the results considerably, because we are getting very small direct latencies (less than 1ms) from these nodes. Next, we will provide experiments for large scale setup in order to compare the both approaches.

4.4 Experiment 4: Using Iplane dataset

It is a challenging task to design a large scale experiment that mimics the mobility of real networks. Using Planet Lab nodes is not sufficient because of the limited node pairs availability in same geographical area. Using publicly available IP address databases is also not a viable solution, because we can not access those remote hosts to run experiments to estimate latency between remote nodes. King [9] provides a way to estimate latency between arbitrary end hosts, but it does not work for most of IP addresses.

Best possible way to estimate latency between arbitrary end hosts is provided by iPlane [12]. iPlane provides a scalable service for accurate predictions of Internet path performance, it uses opportunistic measurement techniques by participating in several Bittorrent torrents, and passively monitoring TCP connections to infer properties of edge links. iPlane also provides a very rich data set of latency estimates between millions of arbitrary hosts all over the world, which is updated daily.

4.4.1 Filtering iPlane data set

iPlane provides around 200 Mb of data consisting of inter-IP links and inter-POP mappings. To filter data set and get list of pairs we use several filtering criterion based on AS information, geo location and response to ping and TCP ping. To get the location of IP addresses, we are using GeoIP2 Python API [3].

4.4.2 Experimental design and Results

For this experiment we are using planet lab nodes as servers, because we do not have remote access on iPlane nodes we are using. Similarly, node pairs are selected from iPlane data set and hence we can fetch the latency between these node pairs from data set. Results suggest that best route selection strategy performs slightly better than Mobile IP most of the time, but IndirectionDC also has a longer tail as compared to Mobile IP.

5. OVERHEAD OF USING INDIRECTION

So far, we have briefly talked about the latency stretch observed by a user of IndirectionDC. Next we will talk in detail about the overheads in latency in more detail. We are using data from Experiment 2 to gain further insights on scenarios where we observe significant latency stretch. We know that for around 60% percent we are observing latency stretch of more than one.

We look at direct route latency for three scenarios: All of the observed paths; Outliers, with latency stretch greater than 5; and paths where latency stretch is less than 1. Figure 6 shows that the most of outliers have smaller direct route latencies as compare to other paths, which suggests that we are getting performance degradation for cases when end host is located very close to source. The reason is that some of the planet lab nodes are in same regions and have very small round trip time between them. We observe a very evenly distributed trend for the cases where we are getting latency gains, which is good.

Figure 7 provides a good insight on what kind of overheads we are observing when using IndirectionDC. It shows the distribution of indirect route latency for scenarios with overhead greater than 2. These are the cases where user potentially faces double or more latency as compared to the direct route. But our results suggest that indirect path still have latency of less than 100ms most of the time.

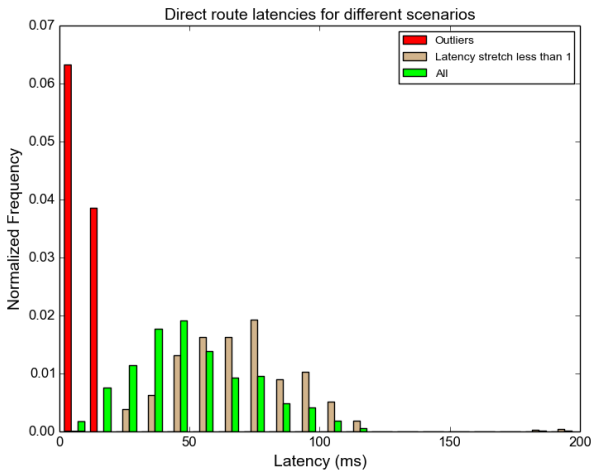

Figure 6: Direct route latencies for outliers

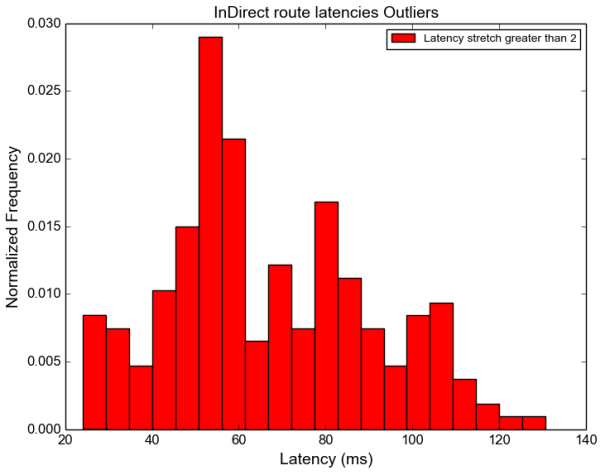

Figure 7: Indirect route latencies for cases where latency stretch is greater than 2

6. RELATED WORK

Techniques using indirection have been proposed in the past to solve the problems related to mobility. Perkins and Johnson [13] proposed MobileIP, which introduced the notion of home agent and mobile node, where home agent was fixed and traffic was rerouted using the home agent whenever you are away. Mobile IP is already discussed in Section 3.

There has been a significant interest in solving the mobility related issues during the past decade. Several variants of Mobile IP were proposed [10], but problem with MobileIP is that it requires changes to IP substrate. This led towards development of solutions that can work on top of IP layer. In 2000, Snoeren and Balakrishnan [18], [17] proposed an end to end approach to solve the problem of mobility in IP. Their approach relies on designing a new TCP option, which involves suspending and reactivating TCP connection. Since their solution proposed changes to transport layer, it was not very practical and involved significant changes to legacy TCP.

In 2002, Stoica et al. [19] proposed overlay based extension to the simple indirection scheme in Internet Indirection Infrastructure (i3), which can support problems like multicast too. Zhuang et al. [20] later implemented a Robust Overlay Architecture for Mobility (ROAM), which was build on top of i3. In 2005, Le et al. [11] surveyed wide range of solutions for mobility problems in internet, their conclusion was that all of the solutions are solving specific problems related to mobility. Also, most of the solutions require changing the infrastructure and are costly.

Most recently, Sharma et al. in [16] proposed a new approach that resolves identities to network locations using a massively scalable global name service. They used demand-aware replica placement engine that intelligently replicates name records to provide low lookup latency, low update cost, and high availability.

7. DISCUSSION AND IMPROVEMENTS

We have discussed the issues related to mobility along with possible solutions and we proposed a very simple solution to solve these issues. Let's talk about some of the potential issues in our work and improvements we meed to make.

Latency Measurements

Different experiments have been designed to test latency overheads of proposed approach. Overall, we can say that using data center as an indirection point has some overheads in term of latency, but total indirect route latencies are still acceptable. Comparison with Mobile IP still needs more insights and in depth analysis.

Throughput Missing

Evaluation of our proposed scheme still lacks one main aspect, we still need to perform tests to look at the overhead in terms of throughput. We can use the same experimental set up and instead of sending pings, we will send large files.

Availability of Data Centers

As we discussed earlier there are bunch of different cloud services available, while Microsoft azure provides us with a good baseline, there is still a need for improvement by using multiple providers. Our best route strategy can optimize the performance even more, if we have more options in terms of data centers.

But this leads to another important question: cost of using the data center? Every cloud operator has a different price model. Looking into the variety of providers and gaining more insights sounds like a way to way forward.

Design Decisions

Looking at a bigger picture, we made a case for using data center nodes as an indirection point to assist mobility scenarios. But there is a need to look at the issues beyond the implementation of such approach, issues like scalability, availability and security.

8. CONCLUSION

We are moving closer to the Internet of Things paradigm, which presents many unique challenges. And mobility is one of the main problem internet faces now and in coming future. But current internet is not ready to deal with these issues, IndirectionDC is a realistic solution with minimal changes to infrastructure. We still need to figure out the detailed design of our scheme, but increase in availability of data centers strengthen the case for solutions relying on new technologies with minimum changes to current internet.

Acknowledgments

9. REFERENCES

[1] Amazon AWS http://aws.amazon.com/.

[2] Google Cloud https://cloud.google.com/.

[3] MaxMind GeoIP2 API.

[4] Microsoft Azure http://azure.microsoft.com/.

[5] M. Anand. Cisco visual networking index: Global mobile data traffic forecast update, 2015 - 2020 white paper.

[6] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. Planetlab: An overlay testbed for broad-coverage services. *SIGCOMM Comput. Commun. Rev.*, 33(3):3–12, July 2003.

[7] CiscoBlogs. Mobility's place in the internet of things.

[8] Gartner. Annual smartphone sales surpassed sales of feature phones for the first time in 2013.

[9] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: Estimating latency between arbitrary internet end hosts. In *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment*, IMW '02, pages 5–18, New York, NY, USA, 2002. ACM.

[10] A. Hanson, E. Sturniolo, A. Menn, E. Olson, and J. Savarese. Method and apparatus for providing mobile and other intermittent connectivity in a computing environment, Nov. 14 2006. US Patent 7,136,645.

[11] D. Le, X. Fu, and D. Hogrefe. A review of mobility support paradigms for the internet. *Commun. Surveys Tuts.*, 8(1):38–51, Jan. 2006.

[12] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iplane: An information plane for distributed services. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, OSDI '06, pages 367–380, Berkeley, CA, USA, 2006. USENIX Association.

[13] C. E. Perkins and D. B. Johnson. Mobility support in ipv6. In *Mobicom, 1996*, New York, NY, 1996.

[14] J. Postel. DoD standard Transmission Control Protocol. RFC 761 (Historic), Jan. 1980. Obsoleted by RFCs 793, 7805.

[15] RighScale. Cloud Computing Trends: 2016 State of the Cloud Survey.

[16] A. Sharma, X. Tie, H. Uppal, A. Venkataramani, D. Westbrook, and A. Yadav. A global name service for a highly mobile internetwork. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 247–258, New York, NY, USA, 2014. ACM.

[17] A. C. Snoeren, D. G. Andersen, and H. Balakrishnan. Fine-grained failover using connection migration. In *Proceedings of the 3rd Conference on USENIX Symposium on Internet Technologies and Systems - Volume 3*, USITS'01, pages 19–19, Berkeley, CA, USA, 2001. USENIX Association.

[18] A. C. Snoeren and H. Balakrishnan. An end-to-end approach to host mobility. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, pages 155–166, New York, NY, USA, 2000. ACM.

[19] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. In *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '02, pages 73–86, New York, NY, USA, 2002. ACM.

[20] S. Zhuang, K. Lai, I. Stoica, R. Katz, and S. Shenker. Host mobility using an internet indirection infrastructure. *Wirel. Netw.*, 11(6):741–756, Nov. 2005.