

Technical Report WINLAB-TR-415

**Network-Assisted Multihoming in the
MobilityFirst Future Internet Architecture**

Shreyasee Mukherjee, Akash Baid,
Ivan Seskar, Dipankar Raychaudhuri

Rutgers University

June, 2013

RUTGERS WIRELESS INFORMATION

NETWORK LABORATORY

Rutgers - The State University of New Jersey

Technology Centre of New Jersey

671 Route 1 South

North Brunswick, NJ 08902-3390

Phone: (732) 932-6857 FAX: (732) 932-6882

Technical Report WINLAB-TR-415
Network-Assisted Multihoming in the MobilityFirst Future Internet
Architecture

Shreyasee Mukherjee, Akash Baid,
Ivan Seskar, Dipankar Raychaudhuri

WINLAB PROPRIETARY

For one year from the date of this document, distribution limited to WINLAB personnel; members of Rutgers University Administration; and WINLAB sponsors, who will distribute internally when appropriate for their needs.

Copyright ©2010/2011 WINLAB, Piscataway, New Jersey

ALL RIGHTS RESERVED

Network-Assisted Multihoming in the MobilityFirst Future Internet Architecture

Shreyasee Mukherjee, Akash Baid,
Ivan Seskar, Dipankar Raychaudhuri

WIRELESS INFORMATION NETWORK LABORATORY

Rutgers - The State University of New Jersey

Technology Centre of New Jersey

671 Route 1 South

North Brunswick, NJ 08902-3390

Phone: 732-932-6857 FAX: 732-932-6882

ABSTRACT

This paper presents a novel network-assisted technique for enabling multihoming in the MobilityFirst Future Internet Architecture. The MobilityFirst architecture, a clean slate approach towards supporting the requirements of mobile nodes in the Internet, provides a name-based packet delivery mechanism which splits the role of IP addresses into two primitives: a permanent unique identifier per host, and multiple network addresses or locators per interface. We show how this architecture is particularly suitable for enabling several different ‘modes’ of multihoming. In particular, our approach shifts the burden of policy expression and data-stripping from end-nodes to in-network nodes, and utilizes the increased visibility at the routing layer to implement an efficient data-stripping algorithm. Results from detailed NS-3 simulations shows improvements of up to 30% in file transfer times when compared to TCP/IP when connecting through the best interface, and aggregate throughput of more than 90% of the maximum achievable value, when connecting through Wi-Fi and LTE simultaneously.

Table of Contents

1. Introduction	2
2. Overview of the MobilityFirst Architecture	5
2.1. Naming and Name Resolution	5
2.2. GSTAR Routing Protocol	6
2.3. Hop-by-hop Transport Protocol	6
3. Multihoming Algorithm in the MobilityFirst Architecture	8
3.1. Best Interface Data Transfer	8
3.2. Interleaved Data Transfer	8
3.3. Replicated Data Transfer	9
4. Evaluation	10
4.1. NS3 Simulation Setup	10
4.2. Best Interface Results	10
4.3. Interleaved Data Transfer Results	12
5. Conclusion	15
References	16

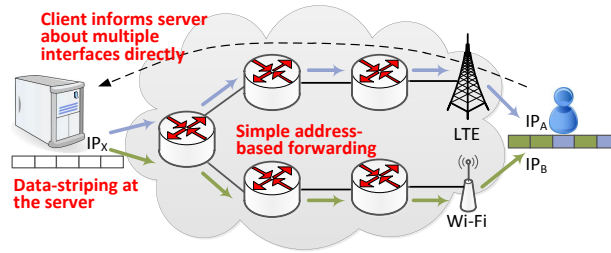
needed

1. Introduction

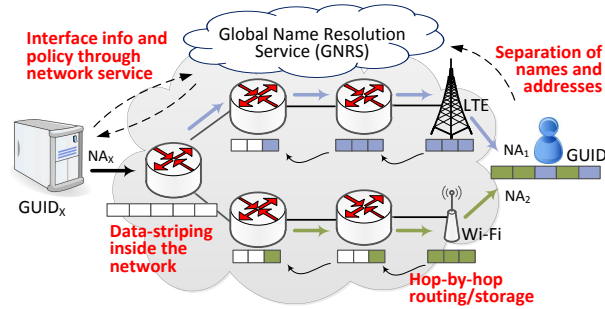
While the basic design of the Internet has remained largely the same since its inception, the manner in which devices connect to it has seen a dramatic change over the last decade. This change - from fixed, wired access to predominantly wireless access over Wi-Fi and 2G/3G/4G cellular technologies has led to several recent efforts towards a clean slate re-design of the Internet architecture [1, 2, 3]. As these works have shown, the first step towards supporting mobile devices and the ensuing mobility-related requirements, is to tailor the underlying routing and transport mechanisms for the wireless medium. However, beyond the obvious limitations of the properties of physical medium (e.g. variable bit-rate, packet loss, and disconnections), wireless access opens up new possibilities for improved communications quality - arguably the most important being simultaneous connection through more than one network interface.

While devices with multiple interfaces are increasingly common (for example almost all smartphones have both Wi-Fi and 2G/3G/4G radio front-ends), the existing methods for utilizing these interfaces in a sequential manner do not take full advantage of parallel connectivity opportunities to improve users' quality-of-experience. In particular, conventional 'hetnet' access mechanisms simply connect the device through Wi-Fi whenever an access point is in-range and defaults to the cellular connection otherwise. Simultaneous access through both these interfaces can vastly improve the bit-rate and availability compared to one-interface at a time in such scenarios. The manner in which multiple interfaces should be used also depends on the application requirements. For example, video playback, which requires high reliability, can benefit from the same packets being sent through all available interfaces, while fast file download applications can make use of higher bandwidth that comes through striping a packet stream for interlaced transfer over multiple interfaces.

Earlier proposals on protocol extensions to support multiple interfaces rely on enhancements to the underlying end-to-end transport layer [4, 5, 6]. Specifically, these proposed mechanisms require a multi-homed end-point to inform the sending side about its multiple interfaces prior to the commencement of data-flow, and a data-striping algorithm on the sending side stack which adapts the ratio of packets sent to each interface according to the capacity of the end-to-end path. This approach results in rigidity in two key aspects: (i) Switching between different *modes*



(a) End-to-end Multihoming in IP based networks



(b) Network-assisted Multihoming in name-based networks

Figure 1.1: Key conceptual differences between conventional TCP/IP based multihoming and the proposed network-assisted approach

of making the best use of the multiple interfaces based on user policy or application requirements is difficult. For example specifying and switching between the ‘replication-mode’ and the ‘aggregation-mode’ needed for the video playback and file download applications mentioned above, requires complicated overlay hacks; (ii) Since all decision logic is implemented only at the end-nodes, in-network routers cannot adapt or buffer the flows in accordance with dynamic changes to the wireless channel quality.

We consider the problem of multihoming in the setting of a ‘name-based’ architecture. Name-based Internet architecture proposals (such as [1, 2, 7, 8]) address the problem of overloading of the function of IP addresses which serves as both host identifiers and locators. Instead, in name-based architectures, each host (*and not each interface*) is given a permanent, unique identifier, and each of its interface gets a new network address from whichever network it connects to. In this framework, hosts can use the permanent identifiers of their intended recipients when sending packets. This makes the name-based architecture intrinsically suitable for multihoming, since the destination identifier can be mapped to a set of network addresses based on its current connectivity status and the preferences of the destination.

In this paper, we describe a novel network-assisted approach for supporting multi-homed devices in MobilityFirst [1], a specific name-based Internet architecture, however the described

approach is applicable to other name-based architectures as well. In MobilityFirst, the permanent host-identifiers are called Globally Unique Identifiers (GUIDs), which are dynamically mapped to the network addresses (NAs) through a centralized mapping service called the Global Name Resolution Service (GNRS), as shown in Fig. 1.1. Our multihoming approach makes use of network-assistance in two important aspects. First, the GNRS is used by multi-homed nodes to specify the availability of multiple interfaces and the corresponding preference policies on how to use the interfaces. Second, the task of data-stripping is shifted from the end-host stack to the in-network routers which have a better view of the end-to-end path quality through the underlying routing layer. Specifically, our in-network data-stripping algorithm makes use of two types of information available at the routing layer - (i) Hop-by-hop backpressure propagation for determining path capacities on a slower time-scale; (ii) Link state advertisements for determining the ratio of packets to be sent on each interface, on a faster time-scale.

The key contributions of this paper are as follows:

- We describe a flexible, network-assisted approach to supporting multi-homed devices in the Internet. Several modes of utilizing multiple interfaces (e.g. unicast to most suitable interface, simultaneous transfer through all interfaces, etc.) is enabled through this approach.
- We describe a specific in-network data-stripping technique that relies on per-hop backpressure and link-state advertisements for splitting flows amongst different paths.
- We present evaluation results from detailed NS3 simulations, which show competitive performance against 'oracle', which an idealized application, achieving the best possible throughput (explained in detail in the evaluation section)

2. Overview of the MobilityFirst Architecture

The MobilityFirst architecture is a clean-slate design of the Internet, founded on the premise that in the near future, the number of mobile devices will far outnumber the number of stationary hosts/servers for which the Internet was initially designed for [9]. The design goals, key architectural concepts, protocol details, and prototyping efforts are described in detail in our earlier works [10, 11, 12, 13]. Here, we outline specific features of the MobilityFirst architecture that highlight how multihoming is supported in this architecture.

2.1 Naming and Name Resolution

the MobilityFirst architecture is built upon a new name-based service layer which serves as the “narrow-waist” of the protocol stack. The name-based service layer uses flat globally unique identifiers (GUIDs) for not just end-hosts but a broad range of communicating objects - from a simple device such as a smartphone, a person, a vehicle, a group of vehicles, a piece of content, and even context. GUIDs are assigned by one of multiple Name Certification Services (NCSs) and are mapped to a set of network addresses (NAs) or locators corresponding to the current points of attachment of the network object to the Internet.

This dynamic mapping of GUIDs to NAs is done through a logically centralized but physically distributed Global Name Resolution Service (GNRS) as shown in Figure 2.1. This is one of the key features utilized for multihoming. When a host X is connected to the Internet through more than one interface, the NA corresponding to each of its connection point is entered in the GNRS mapping table, along with a preference policy which specifies how the multiple interfaces should be used. Another host Y which wants to send packets to X can query the GNRS to get the addresses of each of its interfaces and decide on which interface to send to, or whether to stripe data across multiple interfaces. The details about this decision process and the data-stripping algorithm is provided in Sec. 3.

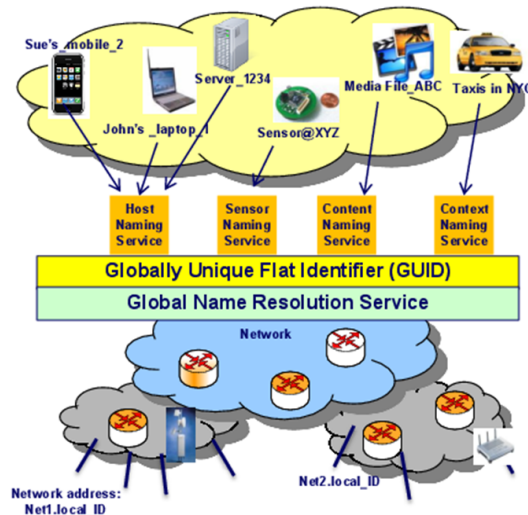


Figure 2.1: Separation of Identification and Network Location in the MobilityFirst Architecture

2.2 GSTAR Routing Protocol

At the routing layer, MobilityFirst uses the Generalized Storage Aware Routing (GSTAR) protocol [10]. GSTAR unifies key techniques from mobile ad-hoc networks (MANET) and disruption-tolerant networks (DTN) routing protocols, which makes it particularly suitable for mobile nodes. In GSTAR, all nodes periodically broadcast fine grained link quality information, in the form of flooded link state advertisements (FLSAs), that contain the short term and long term Expected Transmission Time (SETT and LETT) of their current 1-hop neighbors, which is then used to calculate the overall path quality to all other nodes using Dijkstra's shortest path algorithm [12]. The autonomous unit of data transmission at the network layer is called a *chunk*, which is composed of a large number of smaller data packets. Chunks can be stored at any hop if the path quality to the destination-node is estimated to be significantly below its long term average value.

The estimate of the end-to-end path quality provided by GSTAR is also utilized for data-stripping across multiple paths for multi-homed end-hosts - when interleaving packets between two or more paths, the multicasting algorithm (described in the next section) tries to match the ratio of packets sent on each path to the ratio of the quality of the paths.

2.3 Hop-by-hop Transport Protocol

The transport protocol used in MobilityFirst is based on the Hop protocol [14]. Hop is a clean slate transport protocol that transfers large blocks of contiguous data reliably in a hop-by-hop fashion. Before sending a block B to its next hop, a sender sends a control message $BSYN$,

on receipt of which the receiver sends a *BACK*, which contains a bitmap of the packets of the block B that it has correctly received. Hop routers utilize in-network caching to temporarily cache in-transit blocks, to reduce re-transmission overhead. Hop also uses an ack-withholding mechanism, wherein at any router, if the difference between the number of received blocks for a source/destination pair and the number of blocks successfully transmitted to its downstream hop exceeds a value H , the router stops sending *BACKS* to its upstream neighbor for newer blocks for the same source/destination flow.

The backpressure congestion control mechanism used in Hop is utilized in our multihoming algorithm to modify the number of block sent on each interface. However, as we show in our evaluations, relying only on backpressure propagation for making the striping decision results in a long delay between when the link quality changes on the last hop and when the sending end-node gets to know about it. Thus we adopt a two-level approach in which the backpressure is used at a lower time-scale and the path quality information from the routing layer is used at a more granular time-scale, to efficiently stripe packets across multiple paths.

3. Multihoming Algorithm in the MobilityFirst Architecture

The key features of MobilityFirst, as highlighted, clearly shows that it can natively support multihoming, with seamless data transfer across multiple interfaces. One of the key advantages that MobilityFirst offers over current internet architecture is the separation of the naming and addressing space, which provides seamless switching of interfaces as well as, increased resilience to temporary disconnections. In this section, we would like to highlight what are the best ways we can utilize these features to perform a variety of multihoming applications namely, 1) data transfer on the best interface, 2) interleaved data on all the interfaces and 3) replicated data on all the interfaces. The flexibility to choose any of the above methods lies in the hands of the user/application and the application can specify its policy accordingly through GNRS updates.

3.1 Best Interface Data Transfer

Different interfaces may have different bandwidths and fluctuating link qualities and as such, a user can choose to use only the best interface available at any point of time to send/receive data. Through GSTAR routing the source router already has the best route to each GUID in its forwarding table, and as such choosing the best interface is straightforward.

3.2 Interleaved Data Transfer

The atomicity of striping is a chunk, wherein for every incoming chunk, the router at the source takes a decision on the network address of the edge hop and the corresponding outgoing interface. On receiving a chunk, the router does a GNRS lookup, to find the network address of the point of association of the destination GUID . If the destination has currently only one active association, no striping is done. It simply writes the network address in the MobilityFirst header and pushes the chunk out through the appropriate interface. If, however multiple interfaces are present, it decides on the NA, based on the current path quality informations available, as shown in Fig. 3.1. The router has two types of routing informations available. 1) In its forwarding table, it has the overall SETT of all the NAs in the network (calculated using Djikstra) and 2) in its

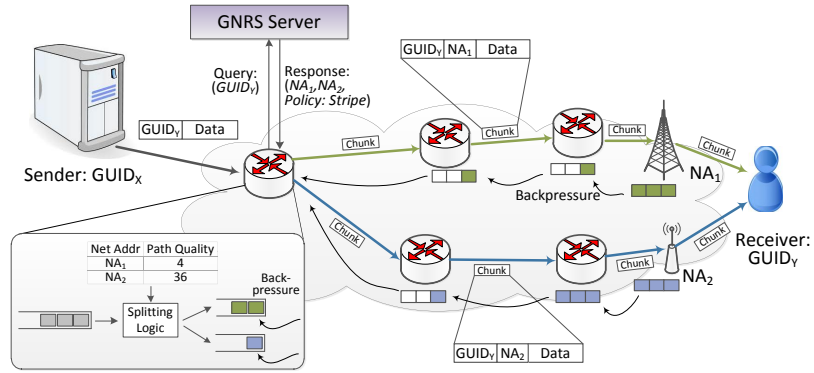


Figure 3.1: Overview of the data-stripping technique using backpressure propagation and path quality metrics at each router

LSA Table, it has the most recently broadcasted flooded link state advertisement (FLSA) from all the other routers in the network. The source router obtains the SETT of the destination GUID from the latest LSA message of each of the NAs, the destination is currently associated with, and adds it to the SETT of the path upto the NA, to have an estimate of the overall SETT of the destination GUID through each of the NAs. We note that, the accuracy of the estimate would depend on the freshness of the LSAs and as such, for a highly mobile client or frequent link fluctuations, the estimate could be quite different from the actual SETT. However, having the exact path quality information is extremely difficult and as we show, next in our evaluation section, routing information of this level of granularity achieves competitive performance against the ‘oracle’ scenario. At the start of the flow, the source router updates the splitting logic with this information and henceforth it gets updated every second in the same manner, or else it timeouts, if the router does not receive any chunk for 5 seconds. From the SETT of each path, the logic calculates the number of chunks, it should send out to that NA in that second, depending on the size of the incoming chunk. Among all the interfaces available, it chooses the best interface, every time it receives a query for the same GUID. We note, that sending to the best interface, may not be the best solution for many practical scenarios, wherein, a bottleneck might arise, if multiple sources try to send data through the same NA. However, this is mitigated by the inherent backpropagation algorithm of HOP, where the next hop refuses to accept chunks above a certain threshold H for each flow.

3.3 Replicated Data Transfer

If the policy is send replicated data through all the interfaces, the source router creates K -copies of the incoming chunk, if the GNRS query returns k -interfaces, writes the appropriate NAs in the header and sends them out through the appropriate interfaces.

4. Evaluation

In this section we present simulation results for the first two modes of multihoming presented in Sec. 3, i.e. Best interface data transfer and Interleaved data transfer. In the first case, we compare the performance of our interface selection and transport mechanisms with a TCP/IP implementation. However, for the second case, since simultaneous transfer over two interfaces, i.e. interleaving of packets is not supported in baseline TCP, we compare the performance of our data-stripping algorithm against an ‘oracle’ solution to quantify how it performs relative to the best-possible scenario.

4.1 NS3 Simulation Setup

We implemented a multi-hop network topology on top of the NS3 platform, using the WiFi and LTE models of the current stable LTE-EPC Network Simulator M5 release [15]. The topology considered has multiple wireless 802.11 Access Points and a single LTE base station, with variable number of hops from the source to the destination. For our micro-level simulations we make the following considerations:

- We consider a small sized environment with end-to-end number of hops varying from 2-5
- The APs are placed along a straight line at random distances, which leads to regions of no connectivity, only LTE available, only 802.11 available, and both available simultaneously.
- Simulation parameters are shown in table 4.1.

4.2 Best Interface Results

Here we compare the performance of the MobilityFirst architecture with the current TCP/IP based Internet access for a mobile node. The simulation topology consists of a single mobile client with 802.11 radio moving along a straight road, with access points deployed along the road at random inter-AP distances d . Along its path, the mobile node experiences regions of good connectivity, poor connectivity, as well as temporary disconnections. A remote data server is assumed to be connected to the access points through the back-end wired network.

Radio Technology	Function	Value
WiFi	MAC	802.11(nonQoS)
	Wireless Interface Mode	Infrastructure
	Propagation Models	Log Distance Loss
		Log Distance Delay
	Rate Control Algorithm	Adaptive ARF
LTE	DL and UL Resource Blocks	6
	Mac Scheduler	Proportional Fair
	DL and UL Tx Power	20dBm

Table 4.1: Simulation Model Details

For the TCP/IP comparison, we do not assume a managed MobileIP implementation since the setup being evaluated here is that of opportunistic connections through open 802.11 APs which are deployed and managed by different entities. As in practise, the moving client in our simulation gets a new IP address via DHCP upon connecting to each AP in the TCP/IP case. Further, in order to make a fair comparison, we assume a ‘smart’ application layer over TCP/IP that resumes transmissions from its point of last connection instead of re-requesting entire transfers. Correspondingly, for the MobilityFirst case, every time the client switches AP, GNRS update/query events are generated which take between 30-170ms as per the evaluation in [13].

The data transmission model emulates a web browsing scenario with intermittent bursts of small transfers from server to the client. The client sequentially requests different sized content packets from the server and we measure the time difference between when each request was made and served, for both MobilityFirst and TCP/IP. The size of the requested content is uniformly distributed between 10KB to 5MB, roughly representing HTTP packet lengths in use today [16]. The speed of the car is kept at a constant 50 miles/hr, while two settings of inter-AP distance d is used: uniformly distributed between 100-300m and same with 300-500m. The cumulative distribution function of the request completion times are shown in Fig. 4.1.

The results show a significant reduction in the transfer times for both values of d - median gains of around 4 seconds or 30% in the [100, 300] case and 5 seconds or 22% in the [300, 500] case. Another way to interpret these results are to look at the percentage of completed requests within a given time-frame. On this scale, there is almost a 2x gain in both cases, for example, when measuring the fraction completed at 5 seconds. This gain can be traced to two key differences between MobilityFirst and TCP/IP. First, when the node disconnects from an AP, the packets destined for it are stored locally at the last AP instead of being dropped, and are sent quickly to the next AP when the node connects there. And second, the amount of ‘useful time’ spent in each AP is increased since the node retains its GUID as it traverses

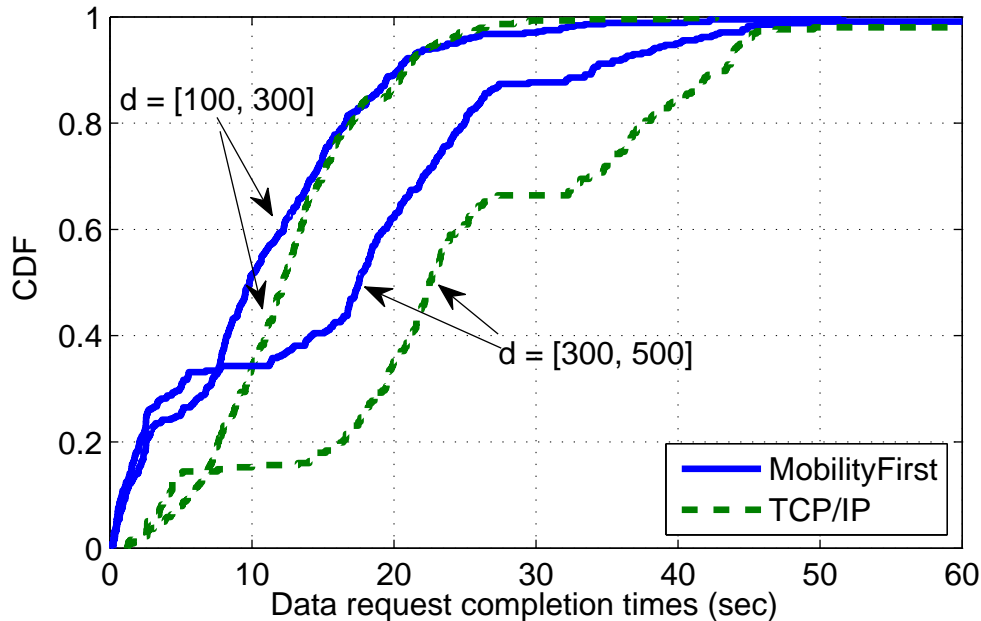


Figure 4.1: Cumulative distribution of the data request completion times in the web-browsing emulation scenario

multiple APs.

4.3 Interleaved Data Transfer Results

For the purpose of this simulation, we design a hypothetical ‘oracle’ scheme, where all the network entities have complete knowledge of the instantaneous link qualities of all the links of the entire network, and as such would achieve, the maximum throughput possible for an application that interleaves data over multiple interfaces. We next evaluate how much of this maximum throughput can MobilityFirst achieve in a practical scenario. The simulation topology is shown in Fig. ???. The simulation topology is clearly simplistic, however, it serves the aim of our simulation effort through the presence of a wired core and Wi-Fi and LTE edge links. In the simulation, a client, equipped with an 802.11 and an LTE radio, moves at a constant velocity of 5 meters per second and while already being connected through LTE, comes in the range of an access point, moves out of range, and comes in the range of another access point. It requests a large file from the back-end server at the start of the simulation and the file transfer continues till the end. We compare the aggregate throughput at the client, when it uses only Wi-Fi, only LTE and both Wi-Fi and LTE against out ‘oracle’ application Fig. 4.3 shows the total data received by the client as a function of simulation time. Next, we vary the periodicity of the flooded link state advertisements(FLSA’s) and measure how

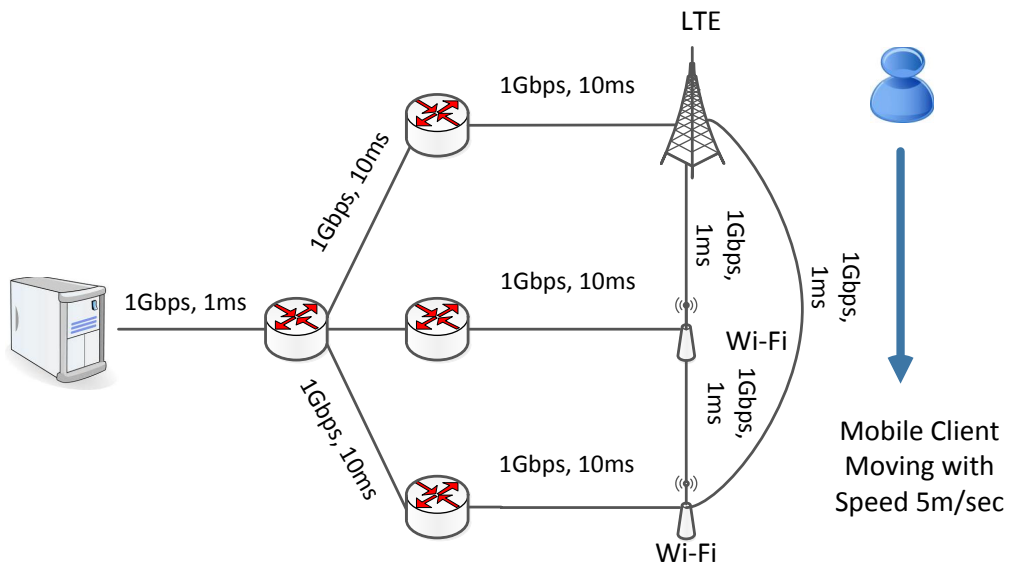


Figure 4.2: Simulation Topology Used for Evaluation of Data Interleaving

that affects the overall multihoming throughput. As explained earlier, at the routing layer, all nodes periodically broadcast link quality information of their current 1-hop neighbors, in the form of FLSA's. These FLSA informations are utilized in estimating the striping ratio at the source. Reducing the periodicity, of such advertisements, would therefore, reduce the freshness of the information at the source and hence affect the throughput at the client. Fig.4.4 shows the aggregate throughput as a fraction of that achievable by the oracle for different values of periodicity. We see that increasing the periodicity up to 5 seconds does not significantly affect the throughput. However, at values of 10 seconds and above, the data-striping algorithm does not get the information on the link qualities in time for efficiently segmenting the flow. This leads to a decrease in the performance compared to the oracle application.

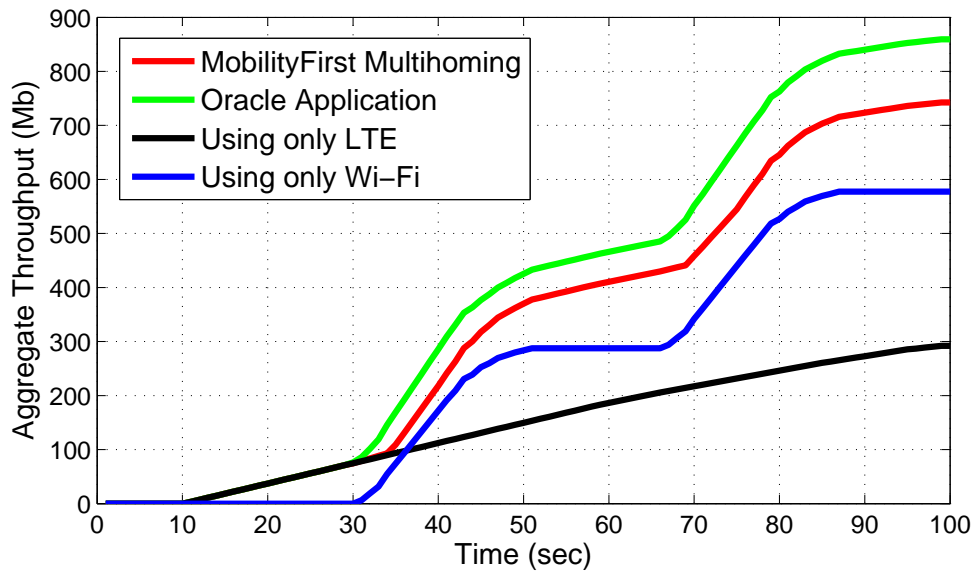


Figure 4.3: Simulation Topology Used for Evaluation of Data Interleaving

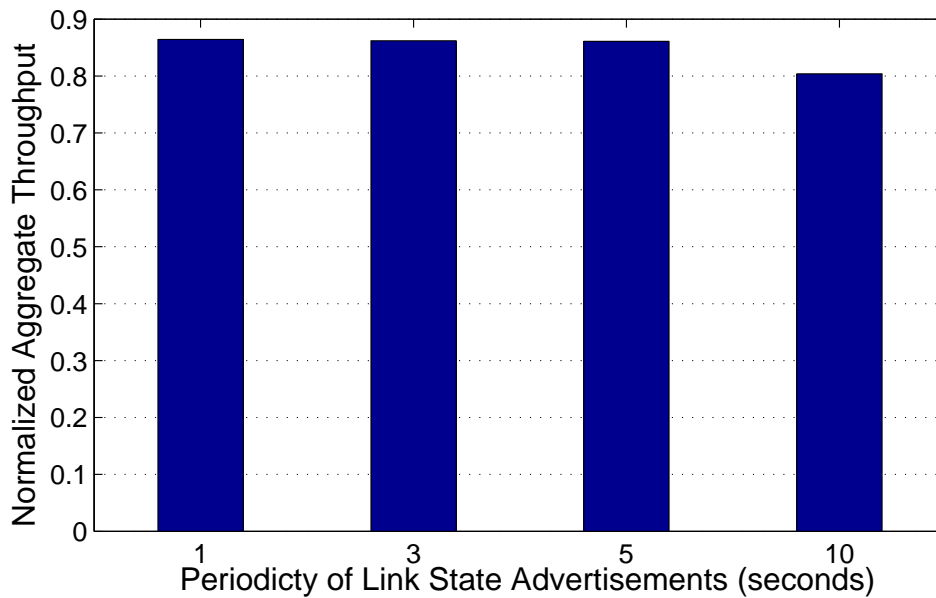


Figure 4.4: Performance of MobilityFirst multihoming algorithm vs. varying periodicity of link state advertisement messages

5. Conclusion

In this paper, we described a MobilityFirst based flexible network-assisted multihoming scheme, utilizing the key advantages that MobilityFirst inherently provides through dynamic GUID-to-NA mapping, hop by hop reliable data transfer and storage aware routing. We described a specific data striping algorithm to allow simultaneous data transfer across multiple interfaces. We present two different simulation results, for multihoming using the best interface and using all interfaces. Results from simulations shows improvements of up to 30% in file transfer times when compared to TCP/IP when connecting through the best interface, and aggregate throughput of more than 90% of the maximum achievable value, when connecting through Wi-Fi and LTE simultaneously.

References

- [1] MobilityFirst Future Internet Architecture Project, <http://mobilityfirst.winlab.rutgers.edu/>.
- [2] R. Moskowitz and P. Nikander, *Host Identity Protocol (HIP) Architecture*, IETF Internet Standard, RFC 4423, May 2006.
- [3] T. You and H. Jung, “Exploiting mobile oriented future internet (mofi) for future cloud datacenter networks,” in *Proceedings of the 7th International Conference on Future Internet Technologies*, ser. CFI '12, 2012.
- [4] M. Py, *Multi Homing Translation Protocol (MHTP)*, IETF Internet Draft, draft-py-multi6-mhttp-01.txt, May 2002.
- [5] J. Iyengar, P. Amer, and R. Stewart, “Concurrent multipath transfer using sctp multihoming over independent end-to-end paths,” *Networking, IEEE/ACM Transactions on*, vol. 14, no. 5, pp. 951–964, 2006.
- [6] H.-Y. Hsieh and R. Sivakumar, “A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts,” in *Proceedings of the 8th annual international conference on Mobile computing and networking*, ser. MobiCom '02, 2002, pp. 83–94.
- [7] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, “A data-oriented (and beyond) network architecture,” *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 181–192, Aug. 2007.
- [8] D. Cheriton and M. Gritter, “Triad: A new next-generation internet architecture,” 2000.
- [9] Cisco White Paper, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016,” Feb. 2012.
- [10] S. C. Nelson, G. Bhanage, and D. Raychaudhuri, “GSTAR: Generalized storage-aware routing for MobilityFirst in the future mobile Internet,” in *Proceedings of MobiArch'11*, 2011, pp. 19–24.
- [11] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri, “MobilityFirst Future Internet Architecture Project,” in *MobilityFirst Project, Proc. ACM AINTEc 2011*.
- [12] N. Somani, A. Chanda, S. C. Nelson, and D. Raychaudhuri, “Storage-Aware Routing for Robust and Efficient Services in the Future Mobile Internet,” in *Proceedings of ICC FutureNet V, 2012*.
- [13] T. Vu et al., “DMap: A Shared Hosting Scheme for Dynamic Identifier to Locator Mappings in the Global Internet,” in *Proceedings of ICDCS '12*, 2012, pp. 698–707.
- [14] M. Li, D. Agarwal, and V. A., “Block-switched networkd: A new paradigm for wireless transport.”
- [15] G. Piro, N. Baldo, and M. Miozzo, “An lte module for the ns-3 network simulator,” in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, ser. SIMUTools '11, 2011, pp. 415–422.
- [16] N. Basher et al., “A comparative analysis of web and peer-to-peer traffic,” in *In Proceedings of the 2008 WWW Conference*, 2008.