Network-Assisted Multihoming for Emerging Heterogeneous Wireless Access Scenarios

Shreyasee Mukherjee, Akash Baid, Ivan Seskar, Dipankar Raychaudhuri
WINLAB, Rutgers University, {*shreya, baid, ivan, ray*}@*winlab.rutgers.edu*

***Abstract*—This paper presents a technique for enabling multihoming in the emerging heterogeneous ("hetnet") mobile wireless access scenarios, where mobile devices have dual wireless interfaces (such as Wi-Fi and LTE) and can use either or both to achieve significant improvements in performance and service quality. A novel network-assisted technique for multihoming is introduced, enabled by the globally unique identifier (GUID) based routing in the proposed MobilityFirst Future Internet architecture, now under development. In particular, the approach shifts the burden of policy expression and data-striping from end-nodes to in-network nodes, and utilizes named object routing with GUIDs to establish multiple paths to destination mobile devices. The proposed multihoming technique uses hop-by-hop backpressure for data striping at the bifurcation router and includes a robust mechanism to reduce reordering of packets at the receive buffer. We quantify the performance gains using detailed NS3 based simulations and present results from a thorough parametric study to determine the effects of data-rate, delay and hop-count difference between multiple available paths. We also show that when multiple interfaces are available, simultaneous use of both the interfaces is beneficial only under certain conditions depending on the ratio of the data-rate of the interfaces and the size of the flow.**

I. Introduction

While the basic design of the Internet has largely remained the same since its inception, the manner in which devices connect to it has seen a dramatic change - from fixed, wired access to predominantly wireless over Wi-Fi and 3G/4G cellular technologies. This has motivated several recent efforts towards a clean slate re-design of the Internet architecture, in order to better support emerging mobility requirements [1], [2]. As these works have shown, efficient and flexible support of mobile devices requires a fundamental rethinking of the underlying routing and transport mechanisms.

In this paper, we explore the design of a scalable and efficient in-network approach for multihoming based on MobilityFirst [1], a named-object based network architecture. In MobilityFirst, end-points (hosts, devices, content, etc.) are identified by Globally Unique Identifiers (GUIDs), which are dynamically mapped to their current point of network attachment (network address, NA) through a centralized mapping service called the Global Name Resolution Service (GNRS). Our multihoming approach makes use of network-assistance in two important aspects. First, the GNRS is used by multihomed nodes to specify the availability of multiple interfaces and the corresponding preference policies on how to use the interfaces. Second, the task of data-striping is shifted from the end-host stack to in-network routers, which have a better view of the network and can compute alternate paths to the destination end-points. Specifically, the task of determining how much data to send on which path is undertaken inside the network rather than at the source, using an in-network data-striping algorithm. This algorithm uses per-flow queuing and hop-by-hop backpressure to determine link capacities as described in detail in Sec. III.

978-1-4799-4912-0/14/$31.00 © 2014 IEEE

Note that while we describe the network-assisted approach for multihoming using the MobilityFirst architecture, it is applicable to any setting in which the locators and identifiers of network end-points are drawn from separate namespaces, and in which the router functionalities can be enhanced. These are explicit design features of several architecture proposals [2]–[4], and the Locator/ID Separation Protocol (RFC 6830) [5] can be leveraged for applying the principles over legacy IP networks.

The specific contributions of this paper are as follows:

- A flexible, network-assisted approach to support multihomed devices in the Internet is proposed, utilizing a specific in-network data-striping technique, that relies on per-hop backpressure for splitting flows amongst different paths.
- Evaluation results from detailed NS3 simulations are provided, which show additive throughput gains for multihomed devices.
- The effect of network heterogeneity on the striping algorithm is analyzed and an optimized technique to intelligently stripe is introduced, so as to minimize the amount of reordering at the receiver.

II. Network Assisted Multihoming in MobilityFirst

In this section, we walk-through the scenario shown in Fig. 1 to explain the basic features of MobilityFirst [6] and the design principles that enable network-assisted multihoming.

The MobilityFirst architecture is built upon a new name-based service layer that uses public-key based flat globally unique identifiers (GUIDs). Unlike IP addresses, GUIDs serve as long-lasting, consistent identifiers for each network attached object. In Fig. 1, when "John's laptop" (a MobilityFirst host) connects to the Internet, it is assigned a GUID (a flat 160-bit name) by one of the multiple Name Certification Services (NCSs). After link-level association, the host updates the Global Name Resolution Service (GNRS) with the set of network addresses corresponding to its current points of

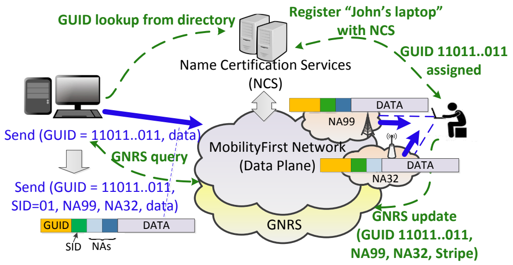

GUID lookup from directory
Register "John's laptop" with NCS
GUID 11011..011 assigned
Name Certification Services (NCS)
Send (GUID = 11011..011, data)
GNRS query
Send (GUID = 11011..011, SID=01, NA99, NA32, data)
GUID
DATA
SID
NAs
MobilityFirst Network (Data Plane)
GNRS
DATA
NA99
DATA
NA32
GNRS update (GUID 11011..011, NA99, NA32, Stripe)

Fig. 1: Example showing message delivery to "John's laptop" that is dual-homed using MobilityFirst

attachment (see [7] for design, implementation and performance details about GNRS). Preference policies (for e.g. stripe through all, only Wi-Fi, replicated data through all, etc.) can also be expressed through this update message, as shown in the figure. When another host wishes to send data to "John's laptop", it obtains the corresponding GUID from the NCS. The GUID is then resolved through a GNRS lookup at the edge router to the set of current NAs, in this case *NA99* and *NA32* and an optional service identifier (SID) corresponding to host-specific preference policies. The packet header actually sent out into the network then consists of a destination GUID, an optional SID and both the network addresses for the network routing protocol to decide on the forwarding path.

MobilityFirst uses the Generalized Storage Aware Routing (GSTAR) protocol for intra-domain routing [8], in conjunction with the Edge-Aware Inter-domain Routing (EIR) protocol at the inter-domain level [9]. The multihoming approach proposed in this paper however only requires the availability of the next hop information given an NA and such, it is equally applicable when a different inter- or intra-domain routing protocol is used, including the current de-factos: BGP and IS-IS. The transport protocol used in MobilityFirst is the Hop protocol [10] which involves hop-by-hop segmentation and retransmission between adjacent routers in the network. In the Hop protocol, instead of packets, routers transfer chunks which are large blocks of contiguous packets to the next node using a reliable transfer protocol on the connecting link. Before sending a chunk to its next hop, a router sends a control message *CSYN*, on receipt of which the next-hop sends a *CACK*, which contains a bitmap of the packets of the chunk that it has correctly received. The receiving node forwards data to the next hop only after it has successfully received the entire chunk. The routers therefore utilize in-network storage to temporarily cache in-transit chunks and reduce re-transmission overhead. Routers also utilize hop-by-hop backpressure through an ack-withholding mechanism for flow control: Every router monitors the difference between *the number of received chunks for a source/destination pair* and *the number of chunks successfully transmitted to its downstream hop* and limits it to a maximum value *H*. Once a router has *H* pending chunks, it stops sending *CACKs* to its upstream neighbor for newer chunks of the same flow. As explained in [10], unlike end-to-end feedback which could be error-prone, this per-flow feedback mechanism is more robust and provides better utilization of resources at no additional overhead.

III. In-Network Data Striping

The GNRS lookup at the first access router binds a chunk to a particular set of network addresses. At any router along the path, if all the NAs have a common next hop, it forwards the chunk downstream till a bifurcation point is reached. At the bifurcation point, the router decides the path(s) on which to send the chunk, based on the user-defined policy (which is expressed through the SID field in the chunk header, similar to the ToS field in an IP header). This provides much flexibility in the way the flows get established, since the bifurcation point can be any intermediate router and can even change during the course of a flow due to end-host mobility. If the user policy indicates that all the interfaces should be used simultaneously for increased throughput, the bifurcation router utilizes a robust backpressure based data-striping algorithm that does not require explicit information on the relative path qualities, and uses locally available state to make the striping-ratio decisions, as described next.

A. The Backpressure Mechanism

Referring to the example shown in Fig. 2, it is observed that dual-homed delivery involves actions by the in-network router at which the paths bifurcate towards the two different wireless interfaces. In this example, when r_3 decides to stripe and send chunks to both r_4 and r_5, it needs to know the actual end-to-end data rate from r_4 and r_5 to the destination, i.e. $GUID_Y$. Our algorithm is designed assuming no accurate end-to-end path quality is available at r_3, and it works as follows: r_3 starts pushing out chunks as it receives, to both r_4 and r_5, which in turn transfer them downstream. Every router monitors a per-flow count of the number of pending chunks, and exploits an ack-withholding mechanism. If a router receives at a rate higher than the rate it could push chunks out, and has *H* pending chunks for that flow, it refuses to accept chunks, until it can send one more chunk out downstream. This essentially throttles the flow from the striping router to the end-client across each of the available paths to the rate of the bottleneck of that path, as shown in Fig. 2. This considerably simplifies the striping algorithm, as r_3 does not require any end-to-end path quality information and instead aims to best utilize the network resources, by pushing chunks out if any/all the next hops accept.

It is important to note that, if *H* is set too low, it might lead to a scenario where a router could be waiting idle for its upstream to send a chunk before it can forward it to its downstream. In contrast, the larger the value of *H*, greater the time required for the flow to fill up the pipe from the striping router to the end-client, which in turn could adversely affect the amount of reordering required. In Sec. IV, we study this trade off and present simulation results justifying the value of *H* chosen for the simulations.

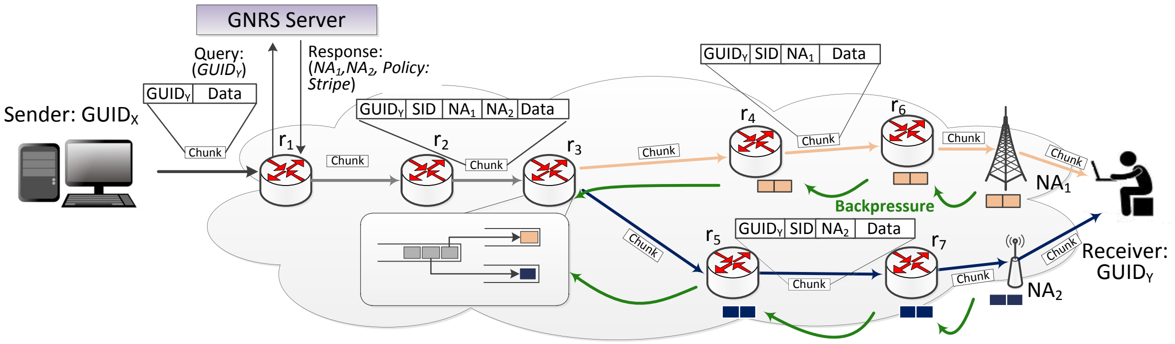
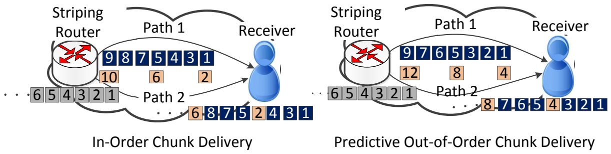


Fig. 2: Overview of the data-striping technique using backpressure propagation



In-Order Chunk Delivery

Predictive Out-of-Order Chunk Delivery

Fig. 3: Comaparison of in-order vs. out-of-order chunk delivery

B. Modification to Reduce Reordering

The transport protocol algorithm to push out data as long as the next hop accepts, is geared towards increasing utilization of link capacities in terms of raw throughput. However, if the link bandwidths and delays are not comparable, this could lead to a large number of out-of-order chunks at the receiver, as demonstrated later in Sec. IV. In order to limit reordering, we propose the following modification to our baseline algorithm. The striping router monitors the number of chunks requested by each interface for a flow to estimate the data rate ratios of the two paths, and sends chunks from the back of the queue across the slower interface. The ratio of chunks sent on the two paths is set in a way so as to minimize reordering requirements at the client. Fig. 3 shows an example where the striping router estimates path 1 to be three times as fast as path 2, and sends chunks from the back of the queue through the slower path. Notice that instead of sending the first chunk in line, it sends every 3^{rd} chunk from the front of the queue across path 2, as it estimates path 2 to be three times as slow as path 1. In contrast to similar algorithms that work on prior knowledge of link qualities [11], in our algorithm the in-network routers transmit data hop-by-hop and as such have no prior estimate of the links. Our scheme starts striping with equal weightage, and switches to out-of-order once the observed outgoing rates of the interfaces start to differ.

IV. Evaluation

We evaluate the performance of the proposed in-network approach to multihoming explained in Secs. II and III through detailed NS3 simulations in which we model MobilityFirst's naming, routing, and transport mechanisms, and use NS3's Wi-Fi and LTE modules [12] to realistically capture the properties of the respective wireless interfaces. Table I shows the values of the key parameters used. Next we explain the setup and results from three different simulation scenarios.

Radio	Parameter	Value
Wi-Fi	MAC	802.11a(nonQoS)
	Propagation Models	Log Distance Loss
	Rate Control Algorithm	Adaptive ARF
LTE	DL and UL Resource Blocks	15
	MAC Scheduler	Proportional Fair
	Tx Power	DL-30 dBm, UL-10 dBm

TABLE I: Simulation Model Details

A. Opportunistic Wi-Fi through vehicular node

First, we study the raw throughput gains that can be achieved by vehicular nodes when they opportunistically use Wi-Fi hotspots while already being connected to an LTE network. In the simulation, a multihomed mobile client moves along a straight road at varying speeds, with access points deployed along the road at random inter-AP distances d. The values for d are chosen from a uniform distribution between 300-500 meters, and thus, the mobile node experiences varying connectivity, as well as temporary disconnections through Wi-Fi. The LTE connection is simulated to have a stable coverage but with lower achievable data rate, as is prevalent in typical vehicular scenarios. The number of hops from the server to the client is kept at 4, however, the link delay from the core to the edge is set to 10ms to have a realistic network model. Fig. 4 shows the aggregate throughput at the client, when it receives a continuous stream of data for the entire simulation, along with the transfer completion times when the client requests a single file of random size between 60-80 MBytes. Three sets of experiments were performed: single-homed over Wi-Fi, single-homed over LTE, and dual-homed with simultaneous transfer over both the interfaces. Each set was averaged over 10 runs. As the figures indicate, the in-network data-striping algorithm fully utilizes the Wi-Fi interface whenever it becomes available. This is indicated by the multihoming throughput being close to the sum of the throughputs achievable through individual interfaces.

B. Effect of link parameters on reordering

Our next set of simulations study the effect of reordering on the application layer buffer requirements at the client.

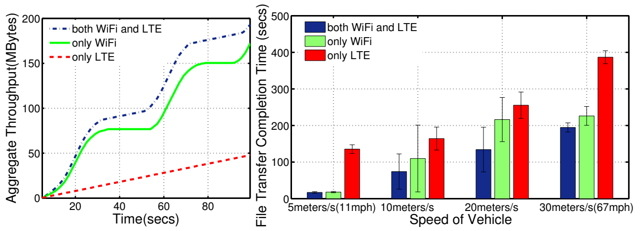


Fig. 4: Aggregate throughput and file transfer completion times for a multihomed mobile client with a Wi-Fi and an LTE interface

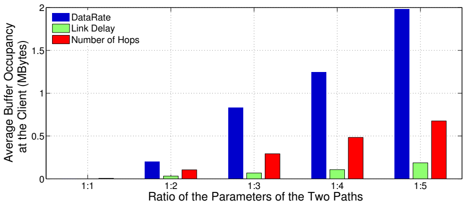


Fig. 5: Effect of link bandwidth, delay and number of hops on the reorder buffer requirements at the client

We study three key parameters which can affect reordering, namely, edge link data rate, link latency and number of hops from the striping router to the end host. The basic topology remains the same as before, but this time the client is equipped with two Wi-Fi interfaces and is kept static. For the baseline scenario, the number of hops from the striping router to the end host is kept at 2, the end-to-end link delay is kept at 10ms and the Wi-Fi edge data-rate for each of the interfaces is set to 36Mbps. Each parameter is then changed, such that they are in ratios of *1:1*, *1:2* to *1:5*, keeping the others constant. The buffer size at the application is measured, from which the average buffer occupancy is calculated in each case. As seen in Fig. 5, the dominant factor in reordering is the disparity in the data rate of the edge links. Note that increase in the number of hops from the striping router to end-host increases the delay for the ack-withholding to set in and therefore increases the reordering. However, even with a hop ratio of *1:5*, the reordering requirements are minimal. Fig. 6 shows the improvements in terms of reduction of buffer requirements, when we employ out-of-order chunk delivery, as proposed in Sec. III-B. This optimization does not affect the reordering due to number of hops, however, the scheme brings down the reordering due to both link delay and data rate disparities. Note that this would not in any way reduce the raw throughput at the client, as the striping router only sends chunks out-of-order, if available, and does not wait idle for chunks to arrive at the back of the queue if it already has pending chunks to send.

C. Effect of backpropagation threshold on performance

Next we investigate the effect of the back-propagation threshold (referred to as H in Sec. II) on the performance of the striping algorithm, with the client having two Wi-Fi interfaces of physical data rates 54Mbps and 6Mbps respectively, requesting a 200MB file from the server (other parameters remaining the same as before). In Fig. 7, we see that the file transfer completion time is highest for H=*1*. This is because when H is set to 1, a router starts accepting a chunk, only when it has been able to successfully transfer the previous chunk downstream and therefore waits idle until it receives the next chunk from upstream. Higher the value of H, larger is the time required for completion as the striping router keeps sending chunks equally across both the interfaces (instead of the ideal rate of $6 : 54$) for a longer time, until the ack-withholding starts, as shown in Fig. 7. At the same time, the lower the value of H, lesser would be the reordering at the application, as the striping router would be able interpret the data rates faster, with the minimum being at H=*1*. For the chosen chunk size and delay bandwidth products of the upstream and downstream links at each router, the value of 2 appears to be ideal. For a different network configuration, a suitable value of H could be chosen based on similar analysis.

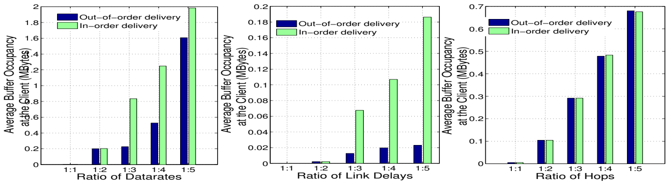


Fig. 6: Reduction in buffer requirements with out-of-order delivery of chunks

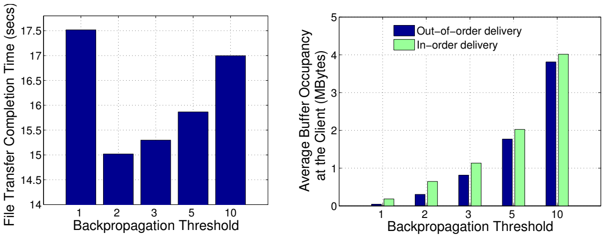


Fig. 7: Effect of variation of H on throughput and reordering at the application

D. Use of both vs. best interface

Finally, we examine the parameter regime (i.e. bit-rate ratio, packet size, etc.) for which striping across multiple interfaces is beneficial. Fig. 8 summarizes the comparison of sending through the best vs. sending through both interfaces for different file download sizes and different data rate ratios of the edge Wi-Fi links. The z-axis plots the ratio of file transfer completion times of sending through the best interface to using both the available Wi-Fi interfaces at the client. The gray plane cutting across the plot is at z=*1*. Ratios higher than 1 indicate striping is advantageous as it results in lower completion times. From the plot, we can draw two interesting observations: Firstly, if the request is too small, the application actually suffers a performance degradation due to striping and secondly, the benefit of multihoming decreases as the disparity in data rates of the two interfaces increases. This motivates the use of a soft threshold for the flow size and the data rate disparities below/above which the routers would decide to use the best interface.

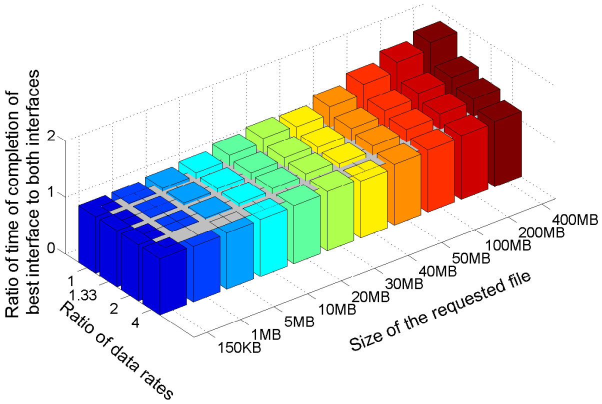


Fig. 8: Comparison of using the best vs. both interfaces for different data flow sizes and ratios of data rates of the interfaces

V. Related Work

The in-network approach towards multihoming described in this paper is complementary to a rich literature supporting multihomed devices in the Internet. Past proposals in this regard can be categorized based on the protocol layer at which the support for multihoming resides. Reference [13] proposes a network layer proxy mechanism for multihoming, but requires fine-grained feedback of link quality information. Schemes proposed in [14], [15] enable vertical handoffs between 3GPP, WiMax, however, they do not support striping of data over multiple cellular interfaces. Moving upwards in the protocol stack, most recent efforts have gone towards a transport layer approach to multihoming support [16]–[18]. While these end-to-end transport-layer proposals have started seeing some early deployments, they offer limited flexibility in the manner in which the multiple interfaces can be used. Asymmetric paths provide further challenges to end-to-end TCP based multihoming implementations as shown in [19].

In this regard, Multipath TCP [20] allows multipath aware applications to express policy preferences and aggregate bandwidths over multiple redundant paths. Authors in [21] perform extensive measurement based studies on MPTCP for dual-homed devices (with Wi-Fi and 3G/LTE). The performance gains and the tradeoffs of flow size reported are similar to the results presented in this work. However, the key distinction that we would like to make is not performance based but implementation based. MobilityFirst provides a cleaner network layer solution, with hop by hop reliable data delivery. Firstly, the network has improved visibility compared to end-hosts which allows it to make better decisions. Secondly, pushing the intelligence down to the routers allows the end-hosts to run a variety of applications on top. Intelligent devices could express their policy to use multiple redundant paths, similar to MPTCP, whereas legacy devices could let the network decide on their behalf and yet achieve comparable performance benefits.

VI. Conclusion

In this paper, we proposed and evaluated a flexible network-assisted multihoming scheme, utilizing the protocol features associated with the MobilityFirst clean-slate protocol with features of name/locator separation, storage-aware routing and hop-by-hop transport. We described a specific data striping algorithm which allows simultaneous data transfer across multiple interfaces with per-flow based back-pressure link quality estimation. NS3 simulation results for realistic mobility models with heterogeneous Wi-Fi/LTE coverage demonstrated aggregate throughput benefits with effective reduction in reorder-buffer requirements at the application. Finally, our results also identified the parameter regime (in terms of wireless channel bit-rate ratios of the two interfaces) for which dual-homing is beneficial relative to simpler best path routing.

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, 2006.

[3] A. Achour, B. Kervella, and G. Pujolle, “SHIM6-based mobility management for multi-homed terminals in heterogeneous environment,” in *Proc. of IEEE WOCN 2011*.

[4] 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.*, 2007.

[5] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, *The Locator/ID Separation Protocol (LISP)*, IETF Internet Standard, RFC 6830, 2013.

[6] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, “Mobilityfirst: A robust and trustworthy mobility-centric architecture for the Future Internet,” *ACM SIGMOBILE Mobile Computing and Communications Review*, 2012.

[7] T. Vu et al., “DMap: A Shared Hosting Scheme for Dynamic Identifier to Locator Mappings in the Global Internet,” in *Proc of IEEE ICDCS 2012*.

[8] S. C. Nelson, G. Bhanage, and D. Raychaudhuri, “GSTAR: Generalized Storage-Aware Routing for MobilityFirst in the Future Mobile Internet,” in *Proc. of ACM MobiArch 2011*.

[9] T. Vu, A. Baid, H. Nguyen, and D. Raychaudhuri, “EIR: Edge-aware Interdomain Routing Protocol for the Future Mobile Internet,” WINLAB, Tech. Rep. WINLAB-TR-414, June 2013.

[10] M. Li, D. Agarwal, and V. A., “Block-switched Networks: A new paradigm for wireless transport,” in *USENIX Symposium on NSDI 2009*.

[11] Wang,J., Liao,J. and Li,T., “OSIA: Out-of-order Scheduling for In-order Arriving in concurrent multi-path transfer,” *Journal of Network and Computer Applications*, 2011.

[12] G. Piro, N. Baldo, and M. Miozzo, “An LTE module for the NS-3 network simulator,” in *Proc. of SIMUTools 2011*.

[13] Chebrolu,K., Raman,B. and Rao,R., “A network layer approach to enable TCP over multiple interfaces,” *Wireless Networks*, 2005.

[14] W. Song, J.-M. Chung, D. Lee, C. Lim, S. Choi, and T. Yeoum, “Improvements to seamless vertical handover between mobile WiMAX and 3GPP UTRAN through the Evolved Packet Core,” *IEEE Communications Magazine, IEEE*, 2009.

[15] J. Hare, L. Hartung, and S. Banerjee, “Beyond deployments and testbeds: experiences with public usage on vehicular WiFi hotspots,” in *Proc. of the ACM MobiSys 2012*.

[16] J. Iyengar, P. Amer, and R. Stewart, “Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths,” *Networking, IEEE/ACM Transactions on*, 2006.

[17] Hsieh,H. and Sivakumar,R., “pTCP: an end-to-end transport layer protocol for striped connections,” in *Proc. of IEEE ICNP 2002*.

[18] Magalhaes,L. and Kravets,R., “Transport level mechanisms for bandwidth aggregation on mobile hosts,” in *Proc. of IEEE ICNP 2001*.

[19] T. Dreibholz, M. Becke, E. P. Rathgeb, and M. Tuxen, “On the use of concurrent multipath transfer over asymmetric paths,” in *Proc. of IEEE GLOBECOM 2010*.

[20] M. Zhang, J. Lai, A. Krishnamurthy, L. L. Peterson, and R. Y. Wang, “A transport layer approach for improving end-to-end performance and robustness using redundant paths,” in *USENIX Annual Technical Conference, General Track*, 2004.

[21] Y.-C. Chen, Y. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, “A measurement-based study of Multipath TCP performance over wireless networks,” in *ACM IMC*, 2013.