Executive Summary:

The MobilityFirst project started in Sept 2010 as a collaborative, multi-institutional research initiative under the NSF FIA (Future Internet Architecture) program. The goal of the project was to design and validate a clean-slate mobility-centric and trustworthy architecture for the future Internet. The scope of research included specification of the proposed new network architecture, detailed design and verification of key protocol components, analysis of economic and policy aspects, evaluation of network security and privacy, system-level prototyping and validation of the network as a whole, and "real-world" testbed deployments for evaluation by application developers and end-users.

This is the cumulative second year annual report for the FIA Next-Phase (NP) project which started in May 2014. This second phase (FIA-NP) of the MobilityFirst project is aimed at making the transition from research stage architecture and prototyping to real-world services and systems, taking into account the above mentioned trends and lessons learned. The project plan has two major components: (1) Architectural refinements to MobilityFirst and further development of the core technology in terms of specification, design and prototyping. (2) Experimental trials of the MobilityFirst architecture which aim to validate the technology in realistic “network environment” scenarios including cellular/mobile, content and emergency response. The architectural refinements agenda of the NP project focuses on several key design aspects including application of MF to cellular/mobile networks, generalization of the global name service (GNS), design of advanced content and context services, and integration of mobile cloud services. There is also a related thread of effort on technology platforms such as SDN or optical along with associated incremental deployment strategies.

The MobilityFirst FIA-NP project made progress on a number of topics during this reporting period (year 2 of the grant). Several architectural refinements were designed and evaluated including improvements to the global name resolution service (GNRS), edge-aware inter-domain routing (EIR), virtual network (VN) support, mobility services (including multihoming and vehicular), multicast services, congestion control and transport protocol. Several new research thrusts were also pursued during this period, including cellular-Internet convergence, content delivery, context aware services and Internet-of-Things (IoT), mobile cloud services, technology platforms (SDN, hardware routers and optical) and related deployment strategies. A work package related to economics and public policy implications of the proposed architecture was also carried out, with particular focus on next-generation cellular systems. In addition to these research and design activities, the project team focused significant efforts on prototype development, testbed evaluation and trial deployment. The goal of these prototyping activities is to further validate the proposed architecture in realistic settings and to spur adoption by the community. During this reporting period, extensive prototyping and proof-of-concept validations of the MF architecture were

---

1 Effective 1/2015, Prof. Kurose stepped down from his role as a co-PI on this project to take up a full-time position at the NSF.
continued on the ORBIT and GENI testbeds as well as on the Amazon cloud. Several new experimental evaluations were completed including edge-aware inter-domain routing, enhanced GNS designs (on both ORBIT and Amazon EC2), content networking, virtual network (VN) and IoT. In terms of technology platforms, MobilityFirst has been implemented on both Click software router and SDN (software defined network) platforms in order to provide multiple options for deployment. Research on high-speed GUID-based packet forwarding architectures has also been carried out. Finally, the project team has started work on three “network environment (NE)” trials in order to evaluate mobility, content and emergency services respectively. These NE trials involve mobility services at an ISP (5Nines) in Madison, WI, a content services (CDN) network for a worldwide satellite operator, SES, and the CASA weather emergency context notification system in TX. For the mobility and content services trials, progress has been made towards development of necessary software components and laboratory validation of the field trial systems. For the CASA context notification service, MobilityFirst services have been deployed and are being evaluated for different notification requirements.

The project team has also been active with education and outreach activities with highlights including a summer internship programs for undergraduates in both 2014 and 2015. The REU summer program has been particularly useful in exposing undergraduate students to the future Internet vision and has led to prototyping and demos of a number of novel mobility-oriented applications running on MobilityFirst. The project team has also been active with industry and community outreach, giving invited talks and presenting numerous papers at networking and wireless conferences. Several research collaborations have also been initiated with industry partners on topics such as information centric network approaches to IoT, mobile cloud services and next-generation cellular architecture.

Further details on the project’s progress are provided in the following sections of this report:

1. Project Background and Scope

2. Architectural Refinements and Design Projects
   1. Architectural Enhancements & Global Name Service (GNS)
   2. DMap+, 2nd Generation Global Name Resolution Service (GNRS)
   3. Inter-Domain Routing in MF
      i. Edge-Aware Inter-Domain Routing (EIR)
      ii. Logically Centralized Inter-Domain Routing
   4. Multicast Routing
   5. Virtual Networks in MobilityFirst
   6. Service API and Transport Layer
   7. Congestion Control in MF Networks
   8. Economic Models and Public Policy

3. FIA-NP Research Thrusts
   1. Cellular –Internet Convergence
      i. Next-Gen Cellular Architecture
      ii. Multi-homing in HetNets
      iii. Multi-Network Access in Cellular
   2. Mobile Content Delivery
   3. Mobile Cloud Services
      i. Dynamic Cloud Migration
      ii. Mobile Acceleration with Cloud Replication
   4. Context-Aware Services and IoT
      i. Supporting Context in MF
      ii. Context-Aware IoT Middleware
      iii. Comparison of MF and NDN for IoT
      iv. IoT Use Case and MF/IoT Prototype
      v. Security for MF IoT Architecture
   5. Technology Platforms
      i. Optical, SDN Cut-Through Switching
II. High-Speed Forwarding Engine

   1. Router and Host Stack Software for MF
   2. Experimental Validation of MF on GENI
   3. Network Environment (NE) Trials
   4. Evaluation Methodologies:
      I. GeoTopo Future Internet Topology Generator
      II. Mobility Measurement and Modeling

5. Education and Outreach Activities
   1. Technical community & Industry Outreach
   2. Educational Outreach and the 2014 REU summer internship programs

Note that this report provides a consolidated summary of the overall project activity and results along with some details of the work done at Rutgers, the lead institution. More information about projects done at the other sites can be found in companion progress reports from collaborating institutions – UMass-Amherst for aspects of architecture, global name resolution, content/context services and mobile cloud; U Wisconsin for network management; U Michigan for aspects of mobile cloud and router implementation; Duke for computing layer & security; MIT for economic and policy aspects, and U Nebraska for integration with optical networks.
1. Introduction and Status Summary

1.1 Project Background & Scope: The MobilityFirst project was started in 2010 on the premise that the Internet is fast approaching an inflexion point with wireless/mobile devices poised to overtake wired PCs as the primary end user device. In the 4 years since the first FIA project was started, the trend towards mobility has accelerated and has already resulted in a deep restructuring of the computing industry as it transitions from personal computers to smartphones as the primary application platform. Since the iPhone was first introduced in 2007, smartphone penetration and worldwide usage continue to grow at an exponential rate – the Cisco VNI Mobile Forecast, 2013 predicts that traffic from smartphones alone will account for 67 percent of the total mobile traffic accounting for about 7.5 Exabytes/month in 2017, a factor of ~10x relative to 2013 0. The Cisco report (“The Zetabye Era”, 2013) also forecasts that “by 2016, wired devices will account for only 39% of IP traffic, while WiFi and mobile devices will account for 61 percent”. Growth in mobile devices is not limited to smartphones - tablet computers are another fast growing category, while new machine-to-machine (M2M) devices have just started to enter the mainstream and are expected to approach 1.5 billion in 2017 out of the anticipated total of 10 billion mobile devices.

![Fig. 1. Major design features of MobilityFirst architecture](image)

MobilityFirst (MF) is a future Internet architecture centered around mobility and trustworthiness as driving design goals [2,3]. The key design features of the MF architecture including named objects, global name resolution, in-network storage & computing, hybrid name/address routing, hop-by-hop transport, etc. are outlined in Figure 1 above. A fundamental design decision in MF that helps achieve these two synergistic goals is a clean separation of names and network addresses and logically centralized but physically distributed global name service (GNS) to rapidly bind names and addresses in a dynamic manner [4,5]. In addition to seamless mobility, this clean separation also builds in security as it allows names and addresses to be defined as globally unique identifiers (GUIDs) that are verifiably derived from public keys and are robust to hijacking and spoofing. The GUID-based communication assisted by the GNS forms the “narrow waist” of the MF architecture and is sufficiently flexible to extend to a variety of endpoint principals, e.g., interfaces, devices, services, content, users, context-aware descriptors, etc. and communication primitives such as content retrieval, multicast, context-aware delivery, anycast, asynchronous delivery, etc. Storage-aware routing in MF [6] capitalizes on technology trends, namely, falling prices of storage relative to bandwidth and the disparity between bandwidth at the mobile edge and an optical core, in order to enable in-network storage for disconnected environments and content caching. To simplify network management, MF supports a management plane that improves visibility into network
behavior for operators as well as end-users and enables more effective control of network resources. Finally, in recognition of the fact that it is not possible for any architecture to design for unforeseen usage scenarios of the future, MF supports a compute layer that enables rapid and incremental deployment of new network services without burdening legacy traffic with additional overhead [7].

In Phase 1 of the project (2010-13), we have designed and validated the MF protocol stack via simulation, emulation and proof-of-concept prototyping. The basic feasibility of each key protocol component and of the protocol stack as a whole has been validated through various individual experiments and via long-running experimental deployments on GENI. In starting the next phase of FIA with the benefit of three years hindsight, we see that our original vision of mobility everywhere remains completely relevant – if anything, the pace of change in the computer industry driven by mobility is so rapid that it calls for a compressed time-line for innovations in mobile networking to make their way into standards and products in the next 3-5 years rather than the 8-10 years originally envisaged. We also see a need to refine the project’s vision and research strategy to respond to (and take advantage) of important developments in the world of network technology and services in the past few years. Some of the noteworthy trends since the project started in 2010 are:

- To meet exponential increases in mobile data usage, 4G/LTE technology is being deployed rapidly by cellular operators, often in conjunction with WiFi or “small cells” to increase capacity. At the same time, the industry has initiated work on “5G” cellular with the goal of providing increased capacity, bit-rate and service flexibility.
- Content continued to be the dominant contributor to traffic on the Internet – video alone accounted for about 400 petabytes/month in 2012 (about 50% of the total traffic) in 2012. This has led to a growing interest in “information centric networks” (ICN) with work starting on technical standards within the IETF and other bodies.
- Machine-to-machine (M2M) applications such as building security and health care monitoring have finally started to take off and are expected to become one of the fastest growing mobile services, with the Cisco report estimating 500 petabytes/month in 2017.
- Cloud computing services (enabled by OpenStack software tools, Amazon EC2, etc.) took on an increasingly important role in meeting the needs of Internet applications.
- Software defined network (SDN) technology with network virtualization entered the mainstream in 2012, providing a useful platform for realizing and evolving new protocols.

In addition to observing the above changes to the outside world of services and technology, we also learned some lessons from the first phase of research on MF. One important architectural insight is the criticality of a logically centralized global name service in enhancing mobility, security, and rich network-layer functionality. We had originally viewed the global name service as primarily a distributed resolution infrastructure (similar in spirit to a next-generation “DNS”) to enhance mobility, but have since come to realize that, architecturally, a fast, logically centralized GNS can transform how network-layer functionality is implemented. For example, at the beginning of the first phase, we had identified the need for context-aware communication (e.g., “send an emergency notification to vehicles on I-90”) but were faced with the challenge of reconciling expressive and abstract destination descriptors with verifiability and line-speed processing at routers. Our experience showed that a logically centralized global name service holds the key to balancing performance, expressiveness, security, and usability of context-aware mobile communication. Another major finding from the ongoing project is that in-network storage and associated hop-by-hop transport can offer significant efficiency, robustness and service flexibility advantages over today’s end-to-end IP networks, particularly at the wireless edge where traffic growth continues to outpace deployed capacity. Finally, our early attempts at releasing MF technology in experimental networks reinforced the importance of evolutionary deployment strategies as a prerequisite for the eventual success of any clean-slate project.

This second phase (FIA-NP) of the MobilityFirst project is aimed at making the transition from research stage architecture and prototyping to real-world services and systems, taking into account the above mentioned trends and lessons learned. The project plan has two major components: (1) Architectural refinements to MobilityFirst and further development of the core technology in terms of specification, design and prototyping. (2) Experimental trials of the MobilityFirst architecture which aim to validate the
technology in realistic “network environment” scenarios including cellular/mobile, content and emergency response. The architectural refinements agenda of the NP project focuses on several key design aspects including application of MF to cellular/mobile networks, generalization of the global name service (GNS), design of advanced content and context services, and integration of mobile cloud services. There is also a related thread of effort on technology platforms such as SDN or optical along with associated incremental deployment strategies.

The FIA-NP project’s “network environments” (NE) agenda aims to further validate the proposed MF architecture via alpha-trial experimental deployments in real-world application scenarios. Achievable cellular system capacity/performance gains from MF are being validated experimentally through a “hetnet” 4G/WiFi mobile network environment to be deployed in partnership with a wireless ISP (5Nines) in Madison, Wisconsin. The importance of content motivated a second and very different trial deployment of MF as a private content network which validates the caching and named content retrieval capabilities of the architecture. It is noted here that the original intention was to deploy the content service network across several PBS stations in PA, but the plan had to be modified in mid-2015 due major changes in personnel at the lead PBS station (WHYY) making it difficult to get the necessary access and staff support for the trial. A second early adopter (SES, a worldwide satellite operator) for the MF based content delivery network was identified in late 2015 and work on a CDN service trial was started in Dec 2015 and is progressing towards a complete evaluation during 2016. We are also conducting a third network environment trial of a sensor-based storm warning service with the goal of evaluating context-aware message delivery services enabled by MF for use in fast growing Internet-of-Things (IoT)
scenarios. This trial will leverage the CASA emergency notification system developed under an NSF ERC program, and involves an important public service application and real-world end users of the service.

As shown in Figure 2, the project is organized into five vertical research thrusts: cellular-Internet convergence, mobile content delivery, mobile cloud services, Internet-of-Things (IoT), Networking Technologies and three horizontal integrative thrusts: architecture & protocol refinements, network environment (NE) trials, and real-world adoption (tech transfer, pre-standards activity, incremental deployment etc.). Anticipated sub-topics under each of these thrust areas are also shown in the figure. Further details on progress during this reporting period (the second year of the project) are given in the sections that follow.

References for Sec 1.1:

1.2 Project Status Summary: This section provides a summary of the FIA-NP MobilityFirst project status at the end of this reporting period (i.e. April 2016). Further details on each project area will be given in Secs. 2, 3 and 4 that follow.

**MF Protocol Architecture:** The MobilityFirst architecture has remained stable with consensus among the collaborating institutions on the functional design and interfaces between different components of the protocol stack. The MF protocol stack is based on the concept of delivering information to and from network-attached objects which are denoted by a cryptographically secure identifier called the globally unique identifier (“GUID”). The “narrow waist” of the MF stack is the GUID service layer supported by a logically centralized global name service (“GNS”) which enables creation of flexible services such as mobility, multicast, multi-homing and delay-tolerant delivery with in-network storage. The key architectural insight which has emerged from the project is the recognition that a logically centralized GNS can significantly enhance mobility, security, and network layer functionality. The reader is referred to our overview paper in ACM SIGCOMM CCR 2014 for further details on the architecture.

**Architectural Evaluation:** In the past year, we initiated a new line of research on architectural evaluation, an endeavor that seeks to evaluate aspects of MobilityFirst not just within the context of MobilityFirst but in comparison to other design choices made by other future Internet architecture proposals. In recent work, we identified one such crosscutting issue, namely, location independence that is prized by a number of different next generation Internet architecture proposals. These findings further reinforce the importance of a global name service for any Internet architecture, not just MobilityFirst, in
order to enable a location independent communication abstraction. These findings appeared in a paper [1] presented last year at ACM SIGCOMM 2014. A related result on evaluation reported at ICNP 2015 [2] is the development of the so-called “GeoTopo” Internet scale model which incorporates recent changes in topology caused by trends such as national backbones or increased edge connectivity.

**Global Name Service (GNS):** The GNS continues to be the most critical piece of the MobilityFirst architecture. The FIA-NP project has pursued two parallel tracks on GNS design and development – the first is an in-network realization (called DMap [3]) of the GNS via DHT-based peering between routers, while the second is an overlay implementation (called Auspice) which uses both DHT and demand-aware replication to realize low lookup latency at a global scale. Both versions of the GNS have been implemented and validated via simulation and experiment – the results show that it is indeed feasible to deploy the GNS at global scale and that 90th percentile latencies of the order of 10 ms or less are achievable with a combination of DHT and local caching. The team is continuing work on both functional and performance enhancements to the GNS. The Auspice GNS developed by UMass has been deployed on Amazon EC2 and is available for trial use by researchers or early adopters. An evaluation of Auspice with comparison to best-of-breed commercial managed DNS providers was reported in a paper presented at ACM SIGCOMM 2014 [1].

**Inter-domain Routing:** A second key component of the MF architecture is storage-aware routing, both intra- and inter-domain. A novel storage-aware intra-domain protocol called GSTAR [4] which provides significant performance gains in wireless/mobile access network scenarios was developed in the first phase of the project and has been used in a variety of experimental evaluations in both the FIA and FIA-NP projects. In the past year, we have revisited the key components of a proposed edge-aware inter-domain routing (EIR) protocol, using telescopic flooding of routing updates to address scalability and late-name-to-address binding to address mobility. We have performed in-depth analysis of EIR running on Internet-scale topologies from CAIDA and prototype evaluation on medium scale topologies on the ORBIT testbed, as detailed in our technical report [5]. In addition, we have explored new design choices such as GUID based inter-domain tunnel setup and maintenance through the GNS for optical core scenarios, with a paper to appear in Proc. ICC 2016 [6].

Recently, we also initiated research on whether a logically centralized directory like the GNS could actively assist and enhance inter-domain routing, turning the conventional organization of routing and the name service on its head. Our very preliminary results suggest that a logically centralized approach to inter-domain routing can indeed significantly enhance availability and reduce long convergence delays in the “bad” cases in BGP while adding little overhead in the “good” cases. Conceptually, this approach generalizes what SDNs bring to enterprise networks to an Internet wide control plane and further strengthens the vision of a GNS-driven Internet architecture.

**Security and Byzantine Robustness:** Baseline security models and threat evaluations were developed for MF during the first phase of the project [7,8]. Significant gains in security are derived from the public key GUID header associated with every MF packet, providing a basis for strong authentication. Further, the MF trust model does not require a single root of trust making it possible to run multiple name certification services for different applications, industries or regions without the need for a centralized ICANN like service for address assignment.

An important top level design goal of MobilityFirst is robustness to the presence of malicious network or endpoint nodes. To this end, we are conducting research towards ensuring strong Byzantine robustness in both the GNS as well as the inter-domain routing protocol. Ensuring a Byzantine robustness property naturally requires us to also defend against distributed denial of service attacks. We previously developed an inter-domain routing protocol with provable correctness properties in the presence of malicious ASes, with a paper appearing in ICDCS 2013 [9]. To secure the GNS itself against Byzantine node behavior, we need to extend its two-tier reconfigurable consensus engine to be Byzantine fault tolerant. We have initiated work on a new and vastly improved two-tier reconfigurable consensus engine, GigaPaxos, in the Auspice GNS. Making it Byzantine secure is an important ongoing next step.

**Virtual Networks and Cloud Services:** A recent result in the FIA-NP project is the design of a virtual networking (VN) framework built directly on the MF protocol stack. The MF protocol stack provides native
support of virtual networks because the named-object (GUID) architecture makes it possible to define virtual networks and store the corresponding topology directly in the GNS. Routers support multiple virtual network policies simply by indexing their routing table to the virtual network identifier (VNID). This allows us to operate VN’s on the MF architecture without the need for any additional overlay protocols. The VN design was applied to the mobile edge cloud scenario where the goal is to connect mobile devices to the “best” or “nearest” cloud server using a service similar to anycasting, but with more general routing parameters. A concept called Application Specific Routing (ASR) has been developed for this purpose, where routing decisions are based on both network metrics such as delay and application-specific metrics such as cloud workload/latency. A first prototype of VNs on MobilityFirst has been developed and demonstrated at the GEC22 conference. The VN approach developed is also being used in a collaborative Japan-US (JUNO) project [10] as the foundation for dynamic migration of cloud services applied to real-time CPS environments.

MF Use Cases – 5G/Cellular, Content, IoT & Cloud: The FIA-NP project includes work packages aimed at understanding the application of MF networks to several emerging application scenarios, including cellular/5G, content, IoT, vehicular and cloud. Each such work package involves analyzing the application scenario for key requirements and then meeting those needs with MF components. For example, for the 5G/cellular mobile scenario, MF provides a number of benefits including native support for authentication, mobility, multi-homing and multicast along with advanced features such as in-network storage and late-binding delayed delivery for disconnected scenarios. Simulation results for the 5G/cellular use case showing significant performance gains were reported in [11] and the next step is to evaluate some of these mobility services in a real-world trial as discussed in Sec 4. Similarly, several studies on content delivery over MF have been carried out in earlier phases of the project [12] showing performance improvements from named content and in-network caching. For IoT, we have been working with industry partners to develop an architectural reference model [13], and have also carried out performance studies which evaluate critical metrics such as control overhead and delay. The mobile edge cloud services use case is also under active study now, and is being addressed with techniques such as virtual networks and application-aware routing.

Technology Platforms, SDN and Optical: The ongoing FIA-NP project has a work package on mapping of the architecture on to current and future network technology platforms. Candidate platforms include Click software routers on general-purpose machines, customized FPGA/ASIC hardware routers, software defined network (SDN) switches and controllers, and high-speed optical switches. The MF protocol stack has been ported to all these environments during the course of the FIA-NP project, resulting in a better understanding of the cost/performance trade-offs associated with different platforms. Click software router and SDN platforms have both been shown to achieve ~Gbps line speed forwarding with appropriate optimizations to the code [14]. Hardware acceleration has been shown to provide further speed gains using techniques such as Bloom filters for flat identifier lookups [15]. Finally, recent work on optical switching for core network routers has shown the value of GUID identifiers and the GNS for dynamic setup and maintenance of cut-through tunnels [6].

Network Environment Trials: The FIA-NP project’s “network environments” (NE) agenda aims to further validate the proposed MF architecture via alpha-trial experimental deployments in real-world application scenarios. An NE trial aimed at evaluating use of MF for the mobile data services scenario has been started in cooperation with an ISP in Madison, WI, and the effort is now moving from proof-of-concept experiments to a planned deployment with service trials for opt-in users. The importance of content motivated a second ongoing trial of MF for a wide-area content delivery network (being conducted in cooperation with a global satellite network operator, SES) with the objective of obtaining definitive assessments of network efficiency and end-user services achieved with MF protocol features such as content naming, multicast and caching. We are also in the process of conducting a third network environment trial of a sensor-based storm warning service with the goal of evaluating context-aware message delivery services enabled by MF for use in fast growing Internet-of-Things (IoT) scenarios. This trial leverages the CASA emergency notification system developed under an NSF ERC program [16], and involves an important public service application and real-world end users of the service.
3.10 MF Code and Service Release: Multiple early adopter trials mentioned above also involve hardening of the MF code base and preparing towards ultimate open-source release of the entire MobilityFirst protocol stack for use both by research users and companies interested in evaluating FIA technologies for future products. The project has made good progress towards the goal of community release of MF technologies and software, notably releasing the “Auspice” GNS as a service on Amazon cloud, maintaining a long-running wide-area MF network slice on the GENI testbed, and releasing prototype MF code to several collaborating organizations both academic and industrial. Also, the team is currently working on identifying applications of MF/FIA to government needs such as tactical military networks or emergency response networks.

References for Sec 1.2:
2. Architectural Refinements & Design Projects:
In this section, we report on progress with architectural refinements and design projects being carried out as part of MobilityFirst next-phase. During the past two years, work has been done on enhancements to the global name resolution service (GNRS) which is central to the MF architecture. Two distinct designs, DMap+ and Auspice were developed further at Rutgers and UMass respectively. DMap is an in-network approach involving DHT-based peering between routers, while Auspice is an overlay design with optimal placement of name servers to achieve low delay. Another important architecture/design topic during the 2014-16 period was the edge-aware inter-domain routing (EIR) protocol which provides a number of features useful for mobility and wireless edge scenarios. A scalable multicast protocol and in-network congestion control algorithms were also developed during this period. We also worked on virtual networking over MF in order to provide service isolation and enable cloud services. The Mobilityfirst socket API and transport layer (“MFTP”) options have also been developed further and incorporated into recent prototype code releases. Security protocols have also been developed and evaluated for key architectural components including name resolution and routing. Further details on progress are given in the subsections that follow.

2.1 Architectural Enhancements and Global Name Service (GNS):
Faculty/Senior Personnel: Arun Venkataramani (UMass-Amherst)

This effort is continuing from the previous (2014-15) reporting period. We completed work on the design of the Auspice GNS showing significant improvements over both state-of-the-art research alternatives as well as best-of-breed commercial managed DNS providers. We have also completed several of the goals listed as near-term planned work in previous reports including—a local name service implementation, interoperability with DNS/Internet, a default name certification service—and have initiated work on the other two goals—mobile cloud service support, and distributed context query optimization—as described further below. An important recent milestone was the acceptance of the Auspice paper "A Global Name Service for a Highly Mobile Internetwork" to the ACM SIGCOMM 2014 conference where we presented the work in August 2014.

Since mid-2014, we have worked towards a public release of the Auspice GNS. Much of this work is somewhat thankless and involves careful software engineering and project management activity, but is critical to the success of the MobilityFirst effort. Towards this end, PI Venkataramani has been working "in the trenches" with software engineers and graduate students in the UMass team to reimplement the distributed core or the "reconfiguration engine" in Auspice with a cleaner and significantly more scalable implementation (that can actually scale to billions of name records as opposed to just hundreds of thousands with the previous prototype). A public release of the Auspice GNS to the public was completed in the second half of 2015.

In previous reporting periods, we initiated a new thread of research on quantitatively evaluating how different network architectures including but not limited to MobilityFirst enable location-independence, i.e., an abstraction of communication using fixed endpoint names without concern for their (possibly changing) locations. Despite a staggering number of Internet architectures at least in part sharing the goal of location-independence, there has been almost no work quantitatively comparing different network architectures using uniform cross-architectural metrics, so this effort is important both for the MobilityFirst project as well as the broader research community interested in network architecture.

We found that despite a large body of prior work on location independence, we know of only three "puristic" approaches for that goal, namely, (1) indirection routing, (2) name resolution, (3) name-based routing. Our work has been the first to quantitatively compare these different canonical approaches using a common set of metrics such as routing update cost, path stretch, and forwarding table size. To this end, we also developed NomadLog, an Android app that we released on the Google PlayStore and that has been downloaded by hundreds of users over the course of the last year, in order to measure the extent of device mobility on the Internet today. Our key findings are that: (1) Mobility is the norm, e.g., over 20% of mobile devices make well over 10 network address transitions per day; (2) Name-based routing incurs a
prohibitively high update cost for handling device mobility; (3) Name-based routing incurs a much lower cost for handling mobility of content that happens to be highly aggregateable and moves infrequently today. Taken together, these results suggest that name-based routing approaches (such as Named Data Networking) will need to also rely on either something like MobilityFirst's GNS (or on the more inefficient but simpler indirection based schemes as in cellular networks today) to handle device mobility and thereby serve as a general-purpose replacement for the TCP/IP Internet. The results of this research were presented in the paper titled "Towards a Quantitative Comparison of Location-Independent Network Architectures" at ACM SIGCOMM 2014.

We have also been continuing work on understanding the "fungibility" of different network architectural costs, specifically router update cost, forwarding traffic, forwarding table size, and dynamic forwarding state. For example, it is possible to have multi-port forwarding strategies (e.g., as advocated by NDN and unlike the Internet's best-port forwarding approach) wherein the update cost can be much smaller than the pessimistic numbers in the SIGCOMM'14 paper above but this reduction in update cost comes at the cost of significantly inflated forwarding traffic cost. The SIGCOMM'14 paper did not analyze forwarding traffic cost, so this line of inquiry adds a new dimension to our work on architectural evaluation. The new model including forwarding cost also allows us to incorporate the benefits in path stretch (and by consequence end-to-end user-perceived latency) as a result of on-path caching strategies as advocated by both MobilityFirst and NDN and also adopted in limited settings (in the form of transparent caching by telco CDNs) in the current Internet.

Please refer to the UMass report for further details.

2.2 DMap+: 2nd Generation In-Network Global Name Resolution Service (GNRS):
Faculty/Senior Personnel: D. Raychaudhuri, Yanyong Zhang, Roy Yates, Yi Hu (Post-Doctoral Associate)
Graduate Students: Suja Srinivasan

**Background:** The GNRS provides a service to map a network object (e.g., a user, a device, a content, etc) to its current network address/locator. Users send queries to the GNRS with the name of the object and GNRS replies to the user with the current network address of the object. An enhanced in-network GNRS design called DMap+ was developed and evaluated via large scale simulations and prototyping during this reporting period. DMap+ achieves low lookup latency through a hierarchical organization which exploits common spatial locality patterns in lookup queries while retaining the simplicity and scalability of the replicated random DHT approach. Server workload balance is achieved by deploying K replicas for each identifier-to-locator mapping supplemented by small caches deployed along lookup query routing paths. The K replicas provide balanced workloads for normal network entity queries while the caches alleviate replica server congestion caused by hotspot network entities' queries, whose query popularities are orders of magnitude higher than normal. Specific design issues such as server churn behaviors, cache cost and correctness, and expected server capacity have been addressed. Evaluation results were obtained from a large-scale event driven network simulator of the Internet with over 22,000 ASs under real world geographic and demographic information. The results show that geolocality aware replica placement reduces the 95th percentile query latency from roughly 100ms to less than 40ms and that the maximum server workload deviation is reduced more than fifty-fold compared to the best prior in-network GNRS scheme DMap [1].

**Technical Approach:**
**Cache Structure and Algorithm:** In DMap+, a small-sized LRU (Least Recently Used) cache is deployed at each name server. In addition to the K replica host servers, a lookup query may also be served by a cache along the route to a replica host server. The purpose of our cache is not to improve the overall query hit rate but only to serve queries for the small collection of hotspot GUIDs. Thus we set the cache size to be small enough (e.g., to cache less than 0.05% of total GUIDs) to reduce overhead. We applied two methods to ensure the correctness of a cache entry: First, a predefined time-to-live (TTL) is applied to each cache entry to define its valid period, which provides a baseline consistency for all cached entries. Second, a go-through probability p is associated with each cache entry, which means that for a cache hit,
the query has probability $p$ to go through to the next hop. As a result, a popular GUID’s cache entry will be frequently updated to ensure its correctness, which is the main target of our cache design. Figure 1 shows an example of the go-through probabilistic cache approach in DMap+.

Member AS Availability Management: In DMap+, the correctness of a GNRS operation is ensured by GUID-to-SID mappings, which requires GNRS enabled routers (i.e., SIDs) have a consistent view of global SID availability. However, the dynamic participation of ASs in the GNRS may result in inconsistent SID availability if the changes have are not synchronized between SIDs. Through exploiting the following three features of GNRS, we implemented a two level consistency to manage SID availability. First, all name servers are virtually organized as an overlay by their SIDs. The nearest numeric distance mapping of GUID-to-SID follows the object locating algorithm in an overlay. To ensure fast failure recovery, DMap+ implements sequential consistency among neighboring SIDs so that a GNRS operation will reach the correct replica host when the request reaches a neighbor of the failed SID. Second, the churn rate of SIDs is the frequency of a large PoP up/down status change, which should be practically infrequent (e.g., predict it to be in the same order as large IP prefix re-announcement or withdrawal, at most dozens per hour [2]). On the other hand, the lookup rate issued by a SID is in proportion to the population in its coverage area, which is on the order of tens of millions (or more) per hour. Assuming GUIDs are uniformly distributed among SIDs, and the number of SIDs is in tens of thousands, this suggests that the communication rate between a pair of SIDs is in thousands per hour. Taking advantage of the higher communication rate between SIDs and the strong consistency between neighbor SIDs, DMap+ updates of SID availability can be piggybacked on the GNRS operation replies for global synchronization. Third, as each SID should be certified by an NCS, an admission control can be implemented to exclude volatile ASs from participating in GNRS for a member AS’s participation contribution should outweigh its failure recovery overhead.

Simulation Results: We compare the workload balance in three scenarios – no cache scheme, deploying cache only at the server who issues a query (i.e., source cache) and deploying cache along the route from the source issuing a query to the selected replica host (i.e., route cache). Figure 2 shows the normalized GNRS server workload distribution when the cache size is 0.05% of total GUIDs. The normalized workload is the ratio of each server’s workload divided by the mean of all servers’ workloads. We choose the route cache as a default settings as it makes 99:99% name servers’ workloads are within 7 times of the mean.

![Fig. 2. Workload Balance – Cache Scheme Comparisons](image_url)

![Fig. 3. Cache Error Rate – impacts of Update Frequency and Go Through Probability](image_url)
synchronization. This suggests we can avoid server overloading with proper capacity planning. We measure the error rate as the total number of incorrect cache replies to lookup queries over the total number of issued queries. Figure 3 shows the min, 25th percentile, median, 75th percentile and max error rates by varying the update rate (defined as the average update number over the average query number) from 0:2 to 1. The default go-through probability is applied when a GUID is newly cached. After each run each server adjusts the go-through probability for every GUID. When the update frequency is 0:2 the median successful query rate is over 99:7%. Even with the update frequency equal to 1, the median successful query rate is over 96:8%. Our low cache error rate can be further corrected by “late binding”, where the last hop router re-query GNRS to get refreshed location when it discovers current location is invalid. The overall low cache error rate results from small cache size (i.e., 0:05% of total GUIDs), as the GUIDs that can be kept in the LRU-based cache with such small size is the ones with extremely high popularity, which usually have below average update frequencies given their query number is orders of magnitude higher than that of average GUIDs. Increasing the default go-through probability lowers the error rate to a very limited extent, thus, we set the default go-through probability to be zero.

Three further improvements to the DMap+ design have been made during the most recent reporting period (2015-16) as summarized below:

First, we performed lookup performance comparisons of GMap[1] with Auspice [2] via simulation. Figure 4 shows the query latency with workload consisting of 20% local queries, 20% regional queries and 60% global queries. The results show that GMap and Auspice both improve the lookup latency compared with DMap [3] as the 95 percentile lookup completed by 100 ms improved to within 20 ms. Such performance gain is due to that both GMap and Auspice use geo-aware replica placement to exploit query locality. The difference between the two enhanced GNRS schemes is that GMap uses generalized three-tier hierarchical replica placement in contrast to Auspice uses per-GUID optimized replica placement scheme. Similar results are shown in Figure 5 varying the query workload.

![Figure 4. 20%Local, 20%Regional, 60%Global](image1)

![Figure 5 60%Local, 20%Regional, 20%Global](image2)

Second, we prototyped GMap design and is conducting the deployment in the Orbit test bed. Two major enhancements are implemented in GMap design compared to the existing DMap design. (1) Mapping GUID records to replica host GNRS servers is performed through direct hash the GUID in a record to GUIDs of GNRS servers. (2) The GUID queries and GUID insertions are performed with geo-location awareness.

Third, we extended GUID to NA mapping to generic GUID to network object mapping with <type, values> pairs defined network objects as shown in Figure 6. The generic mapping records provide applications with the flexibility to define desired structures of the mapping object. As a result, users can query GNRS servers with application specific types. And various advance network services can be provided leveraging in-network GNRS.
Figure 6. Generic GUID mapping records

Two example services are currently under study. (1) GNRS-assisted Multicast. We use the GNRS to maintain network based multicast trees and provide fast in-network lookups to resolve branching points for multicast data delivery [4]. A gateway router creates a NA based multicast tree and inserts it into the GNRS. An example is shown in Figure 7. Each branching point in the tree is associated with a GUID, and each GNRS entry is a mapping of a branching point GUID with the GUIDs of the downlink branching points. At each branching point the router performs a GNRS lookup for the corresponding GUID with query type set as 'Multicast' to get the set of downlink destinations. Using this approach of recursive GNRS lookups, a scalable push multicast service is achieved. Work currently in progress is the prototype implementation on Click Modular router, and evaluation of the GNRS based multicast on the ORBIT test bed.

Figure 7. GNRS-assisted multicast data delivery

(2) GNRS assisted late-binding. When a packet is in transit to a mobile GUID, intermediate routers can do a GNRS lookup to get the updated NA of the GUID and deliver the packet more efficiently. Besides this kind of late-binding, GNRS naturally observes each GUID’s mobility pattern. Having a mapping object that stores the previous locations (NA) of a GUID can help us predict the GUID’s probable location in some future time and hence help the network deliver packets to it faster. Work in progress involves building a model to use previous location information from the GNRS to predict future GUID locations and study the degree of improvement in the network.

A technical report [1] and submitted paper [4] on the DMap+ design has been written during this reporting period.

REFERENCES for Sec 2.2
2.3 InterDomain Routing in MobilityFirst:

2.3.1 Edge-Aware Inter-Domain Routing (EIR):
Faculty/Senior Personnel: D. Raychaudhuri
Graduate Students: Shravan Sriram, Shreyasee Mukherjee and Tam Vu (now at U Colorado, Denver)

Background: This project is aimed at design and evaluation of an edge-aware inter-domain routing protocol, called EIR. EIR satisfies the basic inter-domain routing protocol requirements of scalability, robustness, and support for flexible routing policies while also enabling edge-aware routing decisions necessary for mobile/wireless scenarios.

Technical Approach and Protocol Specification: The main idea of EIR is to abstract network entities and their associated properties into aggregated nodes, called aNode, and connectivities to neighbors as virtual links, called vLink. Every network announces (i) its internal states and (ii) its neighbors along with (iii) all states it received from neighbors to its 1-hop neighbors. A network state packet (nSP) with aNode and vLink information is flooded by each network to its neighbors. Each autonomous system makes a local decision on how much aNode and vLink information to provide, and can choose to represent the entire network as a single aNode without any information about internal network structure. The overall architecture is highlighted in Fig. 1.

EIR uses link-state routing throughout the whole Internet in conjunction with telescopic route dissemination mechanism. Specifically, route update messages consisting of both internal and external properties of a network are periodically disseminated by ASes in the form of network state packets (nSPs). Fig. 2 highlights the update format with aggregated state of links expressed in the form of a ⟨Bandwidth, Variability, Availability, Latency, LinkType⟩ tuple. Telescoping route updates are used to control the frequency of relaying nSPs as a function of distance to the originating network domain. Telescopic updates keep route dissemination overhead under control and yet still provide routing states that could help routers along the forwarding path to make “smarter” decision. As a side effect of telescopic route update dissemination, network states that a network observed from a faraway network could be obsolete which could result in routing failure. To address this, EIR utilizes the “late binding” capability intrinsic to the MF architecture. Late name-to-address bindings serve as a fallback mechanism that enables routers to actively react to link/network changes and mobility at the edge of the network. In particular, EIR makes use of MobilityFirst’s fast in-network global naming resolution service, GNRS [1] for late binding when needed.

The EIR protocol also has the provision for a border router initiated intra-domain path setup. In this procedure, the border routers compute paths based on bandwidth, link latency or any other local policy and inject forwarding table entries into internal routers along these paths using route-injection messages. Each of these paths is assigned a unique label. Incoming data at an ingress border router at each AS is marked as transit and appended with the corresponding label of the path, it is intended to transit through, an example scenario of which is shown in Fig. 3. For further details on EIR, refer to our technical report in [2].
Results:

Mobility Evaluations:
One of the key aspects of EIR is its support for mobility, both for individual devices as well as networks as a whole. In order to evaluate such scenarios, we design a realistic inter-domain topology and a probabilistic mobility transition matrix which is briefly described below. We start with a Caida dataset from 2012 [3] that provides router level topologies of ~22,000 ASes, and parse the dataset based on cities. Specifically we focus on San Francisco, which has a point of presence of about 326 ASes. We consider a cooperative scheme where ASes agree to share coverage and connectivity among their customers, i.e. an User X can decide to switch from one network provider to another when moving, provided the latter provides a better coverage in the region. Given the router level topology, a corresponding aNode based topology is developed for each of the participating ASes based on geographical proximity. A mobility model for a typical user, which captures transition probabilities between these aNodes was developed, based on [4], taking into accounts both local and global mobility mechanisms. Using these, we looked at the path stretch that is incurred for such an user with and without late binding, using a Click based prototype on the Orbit testbed. Fig. 4 shows the improvement in path-stretch when packets are late binded along the way at an aNode with a high degree (denoted as the junction aNode). Future evaluations plan to look at different late binding techniques, so as to minimize path-stretch and improve latency of data delivery across a broad range of mobility scenarios. In addition, we are also developing experiments to analyze the benefits of telescopic flooding and late binding for edge network mobility.
Global routing overhead:
In order to analyze the tradeoff between telescopic dampening and routing overhead of nsPs for an Internet scale topology, we considered a current AS level topology available at Caida [3], consisting of 47,445 ASes and 200,812 inter-AS links. Using the above data set, we simulated the generation and propagation of nsPs across the network. Fig. 6 shows the global routing overhead vs. settling time for different values of the parameters of the constant-exponential-constant telescopic function ($y = A\exp(x - \alpha) \exp(\beta x) \exp(-\alpha' x)$). Each colored curve is for a fixed $A$ and $\alpha$, as shown in the legend, with $\beta \in \{3 \rightarrow 8\}$, $\beta \neq \alpha$. As seen from the figure, there are subsets of values ($\alpha = 2, \beta = 5$ and $A \in \{6, 8, 10, 12\}$) that have low overhead as well as low settling time that can be used for setting the telescopic function parameters. Notice, that the worst-case network overhead is about 100 Gbps, assuming 1000 byte nSPs. This is a negligible fraction of the total Internet traffic of $\sim 182$ Tbps as of 2014 [5].

Routing table size:
In order to investigate the scalability in terms of routing table entries, we look at a July, 2012 Caida dataset that provides the intra-domain topology of ~22K ASes. As explained earlier, EIR allows a flexible aggregation scheme, where each AS can independently decide on the amount of information to disclose about internal topology. For the evaluation, we consider all ASes to aggregate uniformly based on the fraction of aggregation internal to an AS, varying from 0 to 1 (0 = no aggregation, 1 = full aggregation). As shown in Fig. 6, the blue curve indicates the inter-domain table size in terms of the number of entries at each border router with varying levels of aggregation. The red curve in Fig. 6 shows the average BGP table size as reported daily by CIDR [6] for the month of July in 2012. As seen from the plot, even though EIR maintains a global view of the network, aNode table sizes are comparable to the current BGP tables, for moderate levels of aggregation.

A technical report on EIR has been completed [2] and a full-length journal paper is being prepared for submission.

REFERENCES for Sec 2.3.1
2.3.2 Logically Centralized Inter-Domain Routing:
Faculty/Senior Personnel: Arun Venkataramani (UMass-Amherst)

The UMass group is working on a novel, logically centralized approach to interdomain routing. This effort is consistent with the spirit of a "cloud-controlled" GNS-based Internet architecture that drives MobilityFirst. However, although we have claimed in the past that a logically centralized GNS can transform traditional network layer functions, we have only demonstrated limited examples of these (e.g., context-based communication) that can also (albeit with more effort) be implemented as an overlay service on top of the current TCP/IP Internet. We now seek to actually conquer the holy grail, namely, to rely on the GNS for interdomain routing—the highest-level network layer function—instead of the current MobilityFirst (and the TCP/IP Internet) model that assumes that interdomain routing is already available and the GNS essentially runs as an overlay on top of that interdomain routing.

This effort presents both new challenges and opportunities. The main opportunity is that a running interdomain routing (e.g., BGP) inside a datacenter can significantly improve convergence delays (on the order of minutes today), policy flexibility (unlike valley-free policies that are key to interdomain routing stability today), interdomain routing security (largely absent in BGP as deployed today), and long-term evolvability. If we can show that interdomain routing can be logically centralized, it would be the most transformative extension of the software-defined networking (SDN) vision from intradomain routing today to interdomain routing. The main research challenge of course is a chicken-and-egg problem. BGP essentially is a distributed database and so is the GNS, but how can routers rely on the GNS to help with interdomain forwarding behavior when reaching the GNS itself and communication between geodistributed GNS nodes in turn relies on interdomain routing? We believe this chicken-and-egg problem can be addressed using a combination of static routing and a much simpler "GNS routing" protocol that only ensures bidirectional connectivity between each domain's head controller and the GNS (i.e., any GNS node). This simpler protocol will not be plagued by long timers (like MRAI and RFD) or convergence delays induced by complex policy dependencies. We are currently working on making this vision a reality and developing a first-cut prototype of a software-defined Internetwork that can be completely bootstrapped using a cloud-based GNS. Refer to the UMass annual report for further details.

2.4 Multicasting in MobilityFirst:
Faculty/Senior Personnel: D. Raychaudhuri, Jiachen Chen (Post-Doctoral Associate)
Graduate Students: Shreyasee Mukherjee, Suja Srinivasan and Francesco Bronzino

Background: Internet applications like video streaming, online gaming and social networks, e.g. Twitter, often require dissemination of the same piece of information to multiple consumers at the same time. Using appropriate multicast routing solutions would help by improving network efficiency, while reducing the complexity and cost of deploying such applications. We propose a solution based on named objects and a dynamic name-resolution service for mapping names to routable network entities.

Technical Approach:
(a) Multicast Tree Management: We exploit names (or GUIDs) to design a Named Object Multicast (NOMA) solution that relies on the globally distributed, logically centralized name resolution service (or the GNRS [1]) for multicast tree management. Specifically, multicast management consists of two core operations: membership of destination nodes and building and management of multicast trees, both of which can be streamlined using the GNRS. First, a unique name (GMng in Fig. 1) is assigned to perform the task of node membership; all entities interested in receiving data from the multicast flow, can request to join by inserting their own unique name into the corresponding mapping in the table. This information is then exploited close to the source by a multicast service manager, which builds an efficient tree based on the available resources and the size of the required tree. Recursive mappings are then used to express the tree graph: by assigning to each branching router a name that exclusively identifies it in the context of the given multicast flow, we recursively follow the tree structure. For example, in Fig. 1, the root of this tree is identified by the multicast flow unique name mapping to the first branching router (GMulti → Gr11); this router then maps to its children in the tree (Gr11 → {Gr21, Gr22}); this continues until the leaves of the tree are reached, where we identify the leaves as the destination nodes. As time progresses and destinations join or leave the multicast group, the service manager can rebuild the tree information.
contained in the GNRS to trigger the required update.

Fig. 1: Multicast architecture overview    Fig. 2: Device mobility handling in NOMA

(b) Multicast Data Forwarding: Once the multicast tree is established, data forwarding can exploit the information contained in the GNRS to efficiently flow through edges between the nodes of the tree. In order to do so, we exploit address encapsulation, where two pieces of information are carried in data packets at the same time: Internally (i.e. second field in the green packets in Fig. 1), the encapsulated information carries the source and destination of the multicast flow, providing valuable information usable by all nodes along the path to easily identify data streams. Externally, routing information to perform hop-by-hop forwarding from one branching node to the next is placed. At each branching node participating in the multicast, forwarding decisions are performed by querying the GNRS to obtain information on how many next hops it has to forward to, generating required duplicates and replacing the external routing information with the new hop source and destination; this process is exemplified in the figure, where node Gr21 generates 2 duplicates for its two children, replacing headers accordingly. Intermediate nodes along the path forward encapsulated packets based on normal unicast rules. This reduces complexity of multicast packet processing to only a subset of nodes of the tree. To reduce the need of continuously involving the GNRS in the forwarding procedure, mappings can be cached at each hop, avoiding traffic and computational overhead. The tradeoff for this approach comes at the cost of slower reaction times to tree change events. More details on how to handle tree restructuring and end points mobility is provided in the following section.

(c) Handling Mobility: NOMA elegantly handles mobility by separating names from addresses and maintaining a name-based tree in the GNRS. At any point of the tree, failure in delivery to a downstream node results in temporary storage of data packets (MF routers are storage-capable [2]) and re-querying the GNRS for an up-to-date downstream node name (GUID) to its address (NA) mapping. In order to ensure that end-hosts do not lose packets while moving, NOMA supports encapsulated ‘repair’ packets to be sent to the moving client. This again is enabled by the GNRS that maintains the up-to-date location (end-host GUID to NA mapping) as it moves. As shown in Fig. 2, when an end-host, D1, moves from NA14 to NA11, which is not part of the multicast distribution tree, the tree does not change immediately. However, failure to deliver at the edge causes the gateway router at NA14 to query the GNRS for up-to-date mapping of D1. Following association at NA11, the gateway at NA14 can encapsulate the pending data and send it as unicast repair to NA11 as shown. In contrast to multicasting, the repair procedure is transparent to an end-host or application and does not require explicit re-joining from the client side.

Results:
We compare pull-based multicast of NOMA with IP based inter-domain multicast, namely, PIM-SM standard coupled with MSDP [3,4]. Through the results we highlight two key benefits of using NOMA, namely, 1) lower control overhead for maintaining a multicast group, and 2) better handling of mobility for data forwarding.
The advantage of using unicast routes to build the tree is that no multicast specific control overhead needs to be exchanged across networks. This is crucial for inter-domain settings where flooding periodic multicast tree update messages is not tractable. In Fig. 3 we plot the multicast specific messages exchanged for setting up a tree and forwarding packets for increasing graph sizes, with the topology being an Erdos-Renyi random graph, and 50% of the nodes being randomly chosen to have destination clients part of the multicast group. For NOMA this includes 1) the GNRS insert messages from each of the destination networks for joining a particular multicast group, 2) the GNRS insert from the gateway at the source domain to insert the generated multicast tree, and, 3) GNRS query and responses during data forwarding at the branching nodes. In comparison, for PIM-SM+MSDP the overhead numbers comprise of, 1) the flooding of Source-Active (SA) messages from the source domain throughout the network, and, 2) the Join messages from the domains which have destinations nodes interested in receiving packet from that particular source. As seen from the plot, maintaining a multicast tree in the GNRS has higher overhead for smaller sized graphs (for example, for a 20 node topology, shown in the zoomed in section of Fig. 3), but it scales elegantly with size. Using PIM-SM+MSDP, on the other hand, becomes intractable as the number of nodes increases. With more than 40 thousand ASes in the Internet today, if every domain was multicast enabled, the cost becomes too high to maintain a distributed tree.

In order to quantify NOMA’s mobility handling benefits, we performed detailed packet level simulations in network-simulator (ns-3) on a 20 domain random topology with randomly placed mobile and static clients, for both NOMA and an IP multicast implementation of PIM-SM + MSDP. Mobility not only affects the instantaneous throughput at a client, it also leads to loss of packets during the interval of disconnection, re-association of the client, re-joining and re-structuring of the multicast tree. Additionally, in a practical setting, for IP multicast, the mobile client will spend a significant amount of time for new IP address allocation through DHCP, which has not been taken into account for this evaluation. This packet loss and reduction in overall throughput is highlighted in Fig. 4 where we plot the aggregate throughput at a mobile client for increasing rates of mobility, that moves randomly with exponential random mean mobility interval of 50, 20 and 10 seconds. As seen from the plot, aggregate throughput for NOMA does not change with mobility, primarily due to native features of MobilityFirst such as hop-by-hop reliable delivery and storage-capable routers to handle temporary disconnections. In comparison, IP multicast throughput significantly worsens with increasing mobility speeds.

Ongoing work on NOMA includes testing a Click-based prototype on the ORBIT testbed. A full-length paper is currently under submission and a technical report is being prepared.

REFERENCES for Sec 2.4
[3] Dino Farinacci, C Liu, S Deering, D Estrin, M Handley, Van Jacobson, L Wei, Puneet Sharma, David
2.5 Virtual Networks in MobilityFirst:
Faculty/Senior Personnel: D. Raychaudhuri and Yanyong Zhang
Graduate Students: Aishwarya Babu and Francesco Bronzino

Background and concepts: Although the current Internet IP has proven to be malleable in accommodating new applications or overlay services, we often see how, in order to achieve their goals, network applications have to rely on the usage of specific work arounds, such as use of middle-boxes, aimed at overruling network decisions towards meeting the application requirements. In order to fulfill the goal of merging network logic with specific application needs, different techniques have been presented over the years. Most recently two solutions have gained traction as the enabling technology at the base of these solutions: Software Defined Networks and Virtual Networks. The first is particularly attractive as it gives the possibility of centrally controlling network decisions given the complete view of the network resources; while this is certainly powerful, scalability issues are still to be solved. In the MobilityFirst project we exploit the second technique to achieve what is called Application Specific Routing (ASR). ASR is a routing mechanism wherein routing decisions are taken based on multiple parameters combining the network layer's metrics along with specific metrics defined by the application. To illustrate the concept of ASR, let us consider a replicated cloud service. In a regular cloud service scenario when a client requests a service it is directed to the cloud site which is closest to it. In a situation where this specific site is overloaded the client's request may either be dropped or redirected to another cloud site slightly further away. With application specific routing we could avoid the drop of the request or redirection by taking a more informed routing decision early on by combining the routing distance along with the compute load at the cloud site. This gives the service providers the flexibility to incorporate parameters which allow for utilizing the intelligence about Layer 4 - 7 [1].

Our study of this topic so far confirms that MobilityFirst is well suited for support of Virtual Networking and Application Specific Routing. The key features of MobilityFirst which are the name-based services enabled by the GNRS (Global Name Resolution Service), along with routing support for anycast delivery, and the ability to introduce extended computing logic into the routing fabric (i.e. the compute layer). Let us consider the earlier example to understand how these components could simplify the design of the Virtual Network. In the example, a client can request the service identified by a service GUID and the request would be routed to the cloud site which is the best option based on the routing as well as the application specific metric. With service-anycast, the client solely needs to specify the GUID of the service and will be routed to the best (in terms of a combined metric) destination site. While building the virtual network for a certain application we assign a GUID for a virtual network and a distinct GUID for each virtual node in that network. In order to keep track of these names and correspondences we take advantage of the GNRS. In the GNRS we can maintain the mappings between the virtual network GUID and its component virtual nodes. We can also keep track of the virtual GUID to true GUID mapping of each node. Since MobilityFirst implements a Hop-by-Hop transport mechanism, if at any point during the routing of a request the application or the routing state changes, it could help take a more informed routing decision even after the request has made it downstream.
**General Design Approach:** In order to take advantage of the inherent characteristics of the MobilityFirst architecture, a Virtual Network design with support of Application Specific Routing was studied and its main components are summarized here:

*Named Network Partition Using Virtual GUIDs:* Taking advantage of the MobilityFirst name-based architecture, we assign each Virtual Network a GUID that uniquely identifies the VN. Every router that is enabled to support such technology can accommodate multiple virtual network instances. The router participating in the network are also assigned unique names that are used to reference them as part of the network partition. The GNRS keeps track of each node’s Virtual GUIDs and the mappings to the true GUIDs of the nodes. It also stores the Virtual network’s GUID mapped to its member virtual node GUIDs.

*Resource Bootstrap Through Central Coordinator:* A central coordinator is used to convey to the nodes participating in the Virtual Network all the required information to bootstrap the network logic using a network API. This includes for example: the virtual network GUID, the virtual topology and the specific ASR metric to apply. It is also responsible for initially updating the GNRS with details of the virtual network name structure. Eventual modifications to the resources allocation are communicated via the same API.

*Integrated Virtual Network Routing Logic:* Two parallel control plane protocols are used for disseminating routing information across the VN routers: 1) a Link State based Protocol is used to exchange information about virtual links (links connecting VN routers); network level measures are queried by the Virtual Networking layer through an API made available by the lower layers and combined to generate the LSP packets. 2) ASR messages containing the end points’ metric values are disseminated in the network to the participating routers using the native multicast feature of the architecture. The collected information is used to periodically compute the routing tables used in the data forwarding logic.

*Data Forwarding and Encapsulation:* Data forwarding happens on a hop-by-hop manner across routers participating in the Virtual Network. When a data chunk reaches one of these routers and a routing decision is taken, the chunk is encapsulated as shown in Figure 2 where the external network header contains information to reach the next VN router. While crossing nodes not participating in the protocol, normal routing decisions are taken using the external network header.

Current design efforts are directed toward incremental deployment scenarios, where an IP based substrate could be employed in conjunction with the name based VN layer on top. Different solutions are under study and will soon be submitted for publication.

**Implementation Status & Future Work:** The designed components have been integrated into the MobilityFirst protocol stack described in Section 4. The Click implementation for the virtual network comprises elements designed to process control and data messages traversing the virtual network. The current status of the implementation can support a small sample of ASR metrics. These use a combination of the node metric provided by the service / application and the link metric which is has been defined as the path length (number of hops) retrieved from the MobilityFirst router’s GSTAR routing protocol.
A first release of the prototype was showcased at the GENI Engineering Conference 22 (GEC22) in a demonstration entitled “Cloud Services Enhancements Through Application Specific Routing in MobilityFirst FIA”. Based on the topology represented in Figure 3, we demonstrated the effectiveness of ASR in deploying replicated cloud services across physical networks. The premise of this experiment was to demonstrate that client requests could be dynamically routed to a service instance based on the reported node metric. A threshold-based metric was applied (represented in Figure 4). This metric combined the distance to the possible destinations as seen in the physical substrate with the current load experienced by the service instances. The demo showcased how client requests were always routed to the service instance whose node metric was less than threshold and took the shortest route based on the specified link metric (number of hops). The VN approach developed here is being used in a collaborative Japan-US (JUNO) project as the foundation for dynamic migration of real-time cloud services.

A detailed performance evaluation study is now ongoing. A first effort is focused on analyzing the Virtual Network design and implementation on the ORBIT testbed. Different traffic policies are applied to understand the VN overhead and its feasibility under real-world networking conditions against current technologies. ASR evaluation has been instead carried relying on a developed application implementing a cloud based Real-time Digital Museum. We refer to Digital Museum as an application in which the server provides an API for querying paintings detailed information based on video captured by clients’ web cameras. Preliminary results are indeed encouraging: figure 5 shows the processing time of 100 requests with and without ASR. While in the second case the server cannot support the incoming traffic after 53 requests starting to drop them, with ASR, the system is able to handle all requests by redistributing traffic to underloaded servers. Further studies on performance on such scenario will be carried on in the near future. A technical report and paper are also being prepared for publication.

References for Sec 2.5:

2.6 Service API and Transport Layer Protocol:
Faculty/Senior Personnel: D. Raychaudhuri, in collaboration with K.K. Ramakrishnan (UC Riverside)
Graduate Students: Kai Su and Francesco Bronzino

Background: The MobilityFirst architecture is a specific realization of the emerging class of Information Centric Networks (ICN) that are designed to support new modes of communication based on names of information objects rather than their network addresses or locators. ICN architectures including MF are characterized by the following distinctive features: (a) use of names to identify sources and sinks of information; (b) storage of information at routers within the network in order to support content caching and disconnection; (c) multicast and anycast as integral network services; and in the MF case (d) hop-by-hop reliability protocols between routers in the network. These properties have significant implications for transport layer protocol design since the current Internet transports (TCP and UDP) were designed for the end-to-end Internet principle which uses address based routing with minimal functionality (i.e. no storage or reliability mechanisms) within the network. This project seeks a design of a transport protocol, MFTP, for an Information-Centric Network architecture, and MobilityFirst in particular.

Requirements for TP layer service for ICN: Several use cases including web access, large file transfer, machine-to-machine and multicast services are considered to identify four basic functions needed to constitute a flexible transport protocol for ICN: (i) fragmentation and resequencing; (ii) lightweight error recovery: end-to-end signaling for error recovery with minimal overhead; (iii) flow and congestion control: required for large-volume transfers, but optional for short transfers; (iv) in-network proxy: routers which provide temporary storage to deal with clients mobility are delegated with transport layer responsibilities.

MFTP design: The design of MFTP is based on the four characteristics of different ICN proposals mentioned above. In particular, MFTP is designed to operate on top of the MobilityFirst networking stack. The key ingredients of the design are:

Segmentation, sequencing, and in-order delivery: As shown in Figure 1, in-order delivery is strictly enforced among the chunks of a single transported file. This suggests transport will buffer out-of-order chunk arrivals while waiting for the missing chunks. On the other hand, only a loose relationship is maintained across multiple files, because each file has a unique name and there is no need for strict ordering of delivery based on the order of the requests.
End-to-end error recovery, flow and congestion control: We seek to have parsimonious end-to-end mechanisms that have minimal overhead (important in mobile wireless environments). The end-to-end error recovery mechanism is built to be flexible to accommodate application and sender needs (including don’t care, NACK, ACK). With a Negative-ACK, i.e. NACK, the transport reduces end-to-end message overhead, and the receiver provides notification only when a chunk is not delivered over a conservatively long period of time (example shown in Figure 2). MFTP relies on the link layer to perform backpressure based congestion control, and use end-to-end feedback only for a window-based flow control for large-volume transfers.

In-network transport proxy: we postulate having routers which provide in-network transport service such that the original source can delegate part of the end-to-end data transfer responsibility. Such in-network transport proxy would have substantial amounts of memory to temporarily hold in-transit chunks when the destination is unreachable. This disruption may be due to: lack of connectivity to a mobile destination node, until connectivity is subsequently re-established; alternatively, in M2M communication, when a sensor node is only powered on intermittently, it may choose to deliver information chunks to the next hop and then power down.

Multicast: In MobilityFirst, a dynamically formed multicast group is identified by a special GUID, which can be recursively mapped into a set of individual GUIDs. As shown in Figure 4, for a small scale multicast, individual destinations can use the reverse path to the source as a unicast channel to deliver NACK, if any, and the source is able to identify the multicast group that the destination belongs and retransmit. In a large scale multicast, such requests can be aggregated and fulfilled by participating transport proxies.

Evaluation results: We have implemented MFTP both in endhost network stack [1] and in MobilityFirst core router. We evaluated MFTP, and for comparison, TCP’s performance in three different service
scenarios: large content delivery over wireless, web content retrieval, and content retrieval with disconnection. MFTP achieves better throughput in large file transfers, and less page download time in web content retrievals (Figure 5). The use of in-network storage and transport service improves content retrieval response times in the presence of client disconnection, compared with TCP, as shown in Figure 6.

**Status:** We have finished an evaluation of the overall design and implementation of the MFTP protocol, and published a paper in ACM ICN 2015. The MFTP design has been incorporated into the main branch of the MF code release.

**REFERENCES for Sec 2.6**


### 2.7 Congestion Control for MobilityFirst

**Faculty/Senior Personnel:** D. Raychaudhuri, in collaboration with K.K. Ramakrishnan (UC Riverside)

**Graduate Student:** Kai Su

**Background:** Previously, we designed a general framework for a clean-slate transport protocol for MobilityFirst, MFTP [1], and investigated in detail how in-network transport services can be used to support mobile content delivery. In this project, we aim to design a congestion control mechanism for MobilityFirst that can be integrated into MFTP. In MobilityFirst, various intelligent functionalities, such as reliability and storage, are placed inside the network to assist with data delivery. Traditional end-to-end, window-based congestion control such as that carried out by TCP becomes unsuitable as it is unable to take advantage of such in-network functionalities. One approach that works well for hop-by-hop reliable networks is using per-flow queueing (see Fig. 1) and backpressure to alleviate congestion [2]. However, it could become impractical in the presence of a large number of flows, which lead to substantial memory consumption and computational complexity. We seek to build a congestion control scheme with per-interface queueing (see Fig. 2 for illustration), thus achieving improved scalability.

**Design:** We consider each router has one or more outgoing network interfaces. Each interface connects the router to a neighboring node through certain communication medium, e.g. a fiber optical link, or a wireless channel. All traffic that need to be channeled through the same next hop will wait in the same outgoing interface FIFO queue. The unit of data that gets processed, i.e. queued, scheduled, etc, is **chunk**. A chunk is a fragmented portion of application data. The traffic sources implement the Token Bucket algorithm to enforce any sending rate.

---

**Fig. 1.** Per-flow queueing: at node r1, each of the two flows has its own queue

**Fig. 2.** Per-interface queueing: at node r1, only a single queue for all the flows
The general approaches considered here are source rate adaptation with explicit congestion notification (ECN) from routers. The overall framework involves the following two major pieces:

**Router operation:** if an incoming chunk sees router outgoing interface queue meets certain congestion criteria, the router computes a suggested rate for the corresponding traffic source, and sends a notification carrying that rate to the corresponding chunk source.

**Source operation:** If a congestion notification is received, the source synchronously reduce the rate if the suggested rate is lower than the current rate. On the other hand, for each W bytes of data sent, the source additively increases its sending rate if no congestion notification has been received.

**Algorithm for computing suggested rate:** The algorithm to be developed need to simultaneously accomplish several objectives: the sources are driven towards their fair share of bandwidth; meanwhile, queue length should be constrained to reduce queueing delay. Consider an arbitrary traffic source, and an arbitrary router queue along the source’s end-to-end path. Let \( Q \) be the current queue occupancy, and \( Q_{\text{cap}} \) be the queue capacity. The router generates a congestion notification if the observed queue length is greater than \( Q_{\text{thresh}}, \) a threshold. We use \( s \) to denote the suggested rate, and it is calculated by:

\[
 s = u - f((Q - Q_{\text{thresh}}) / Q_{\text{cap}}) \times u
\]

where \( u \) is an estimate of fair share for this flow. Currently we use a quadratic function for \( f() \). The idea here is to base the suggested rate on the fair share (link capacity divided evenly across all concurrent flows), and reduce the rate further, if queue length is beyond the desired queue occupancy.

**Sample results:** We built a chunk-level simulator for evaluating the proposed congestion control mechanism. Links and nodes are modeled. All operations performed on the data, such as transmitting and receiving, are done on chunks, not packets. We implement both per-flow and per-interface queueing mechanisms, and their scheduling policies: Round Robin and FCFS, respectively. We also implement a module to perform congestion notification, and call it the Explicit Congestion Notification (ECN) module. Our simulation evaluates how ECN can help improve network performance for per-interface queueing, and how close the performance of per-interface queueing with ECN is to that of per-flow queueing. Here we show sample results obtained for the Abilene network topology.

**Mean link utilization:** Fig. 3 shows the mean link utilization, for offered load values from 0.35MB/s to 9MB/s. We can see per interface queueing without ECN performs rather poorly. However, with ECN, per-interface queueing achieves more than 2x magnitude of throughput improvement. Also, its curve flattens out at around 0.93, which is only 6% lower than that of per-flow queueing.
Throughput by (source, sink) pairs: From Fig. 4, we can see that PerFlow’s lower 50% throughput values spread over a wide range. This is because PerFlow reacts to all the congested queues along the path and consequently, the flow with a longer path is more likely to suffer from low throughput. However, PerIfWithECN’s lower 50% throughput values converged in two clusters, as a result of only reacting to the most congested bottleneck along the path. This allows it to improve the minimum throughput and achieve better fairness.

References for Sec 2.7:

2.8. Economic Models and Public Policy
Faculty: William Lehr (MIT), in collaboration with Roy Yates (Rutgers)

Project Objectives
This work package describes the economics and policy research being undertaken as part of the MobilityFirst-Next Phase (MF-NP) Future Internet Architecture (FIA) research project (NSF #1345256). The goal of the economics and policy research package is to help ensure that MF remains consistent with the design principles of Commercializability (economic viability in a realistic industry value chain) and Regulability (consistency with core policy goals) that were articulated in the original proposal, as well as to undertake specific focused research projects related to ensuring that the MF architecture gives rise to technologies, business models, and policies that can live well in the real world.

Several key objectives of the economics and policy work package include:
(1) Promote development of incentive compatible and practical measurement/metrics platforms that can support the socio-economic impact analysis needed to evaluate the MF and enable its real-time management. Increasingly, FIA like MF are embedding functionality to enable much more granular and dynamic resource management (a prerequisite for supporting richer models of mobility). These capabilities enable and call forth a collective need to measure, monitor, and govern the Internet. The measurement/metrics platforms and capabilities will be important strategic components for the Internet ecosystem.
(2) Undertake research to better understand the economic and policy implications of MF’s expanded conceptualization of “mobility” and what this means for policy and industry structure. This includes wired-wireless convergence (as exemplified by the integration of the WiFi and 3G/4G/LTE ecosystems), the growth of cloud computing infrastructure, and the rise of the Internet of Things (IoT).
(3) Undertake research to understand and promote expanded options for shared spectrum access management. The success of MF will depend in part on and will help enable and promote expanded and richer models for wireless access of all sorts, which in turn requires a shift in the spectrum management paradigm to increased reliance on dynamically shared spectrum.
(4) Analyze and interpret ongoing policy debates and emergent challenges for MF deployment and commercialization. This includes evaluating implications of FCC’s Network Neutrality or spectrum management reform policies for the MF ecosystem, as well as emerging issues such as policy for Internet of Things.

Progress Reported
During this reporting year (2015-16), the MF team continued work on the development of prototype components of the architecture and the deployment of those in a series of demonstration usage cases to investigate different aspects of the MF. This included deploying prototype capabilities for doing context-based delivery of messaging services for public safety in Texas; a media file sharing and collaboration platform for a satellite network operator on CDN; and an alpha-test deployment of advanced mobility services for 5Nines, an ISP based in Madison Wisconsin.
Concurrently, work proceeded on efforts to develop appropriate metrics and measurement infrastructures for evaluating the MF architecture and to better understand its potential implications for Internet ecosystem economics and public policy issues; on enabling key complementary developments such as enhanced capabilities for sharing RF spectrum resources that the success of MF both anticipates and enables; and on disseminating lessons learned within the networking research community, as well as to industry and policymakers.

As the MF team economist, Dr. Lehr participated in regular team meetings related to the continued refinement of the MF architectural design and future enhancements, and in preparing for and participating in NSF FIA workshops. This included the planning of the usage case demonstration projects noted above. Dr. Lehr’s contribution focused on understanding business and policy-related challenges and opportunities associated with customer adoption and commercial deployment of the MF architecture that these usage cases might afford. Unfortunately, the projects are still not sufficiently well advanced to allow significant progress to be made regarding empirical research related to the lessons these demonstration projects might provide for the economic/policy work package of the MF project. Purely technical deployment considerations have dominated the work thus far on the demonstration projects.

Work continued with respect to the other work-items noted in the April 2015 report, as noted below.

**Promote development of measurement/metrics infrastructure:**

As noted in the April 2015 report, MF and the other FIA will provide enhanced capabilities for distributing decision-making control of network resources on a finer-grained, real-time basis. This is a direct byproduct of the richer mobility support that is intrinsic to the MF architecture. Managing and coordinating the real-time resource management capabilities enabled by MF will require enhanced support for performance measurement and metrics, embedded in the fabric of the network and accessible to the multiple stakeholders. The decentralization of decision-making control over economic resources will induce strategic behavior among industry stakeholders that need to be anticipated and addressed by policies. In short, evolving expectations of future Internet architectures, traffic growth, and service quality will impact how the network actually evolves.

To address this concern, Dr. Lehr continued work on developing appropriate measurement and metrics infrastructure for the evolving Internet ecosystem. This included continuing to work with MIT colleagues Steve Bauer and David Clark and collaborators KC Claffy and others at CAIDA on joint MIT-CAIDA efforts to develop edge-based measurement strategies.

This work produced two research papers that Dr. Lehr co-authored, and that his contributions to which were partially supported by the MF research project. The first paper by Bauer, Lehr, and Hung (2015) focused on the challenges associated with measuring performance and developing performance-related regulatory policies (e.g., consumer protection and universal service) as the capabilities of Internet access services increase dramatically. This is highlighted by the emergence of super-fast broadband services with Gbps data rates. Wider- adoption of such services will greatly expand options for network-based mobility and the richer sorts of mobility anticipated by MF (and well beyond the simple model of allowing

---

2 For example, better native network-layer support in the MF architecture for multi-homing, DTN, and rich naming capabilities for resources, content, and services (context-based), expands capabilities for organizing economic control (and residual claimant rights) over those same resources, which expands options for decentralizing network management decision-making. In short, end-user are empowered (because they can select their ISP, application, etc. on a more fine-grained and dynamic basis).

3 For example, because of network externalities and irreversible (sunk) investments that can result in path dependent lock-in and can make collective expectations self-fulfilling. (That is, everyone believing that a particular evolutionary path for the industry and policy is the one that will be chosen can result in that outcome being selected.)
mobile phone calls to be made from anywhere). Such infrastructure will open up new opportunities for deploying and managing local network infrastructure, including community clouds.

The second paper by Lehr, Kenneally, and Bauer (2015) focused on understanding the implications of Disclosure and Transparency policies for managing the flow of market information about network performance. As service options and markets become more complex, and stakeholders dispute the risks from potential abuses of market power or coordination failures (e.g., to coordinate investment and management of network resources), calls for increased “transparency” and “data sharing” are common. To often, however, inadequate attention is paid to the strategic implications of policies designed to enhance transparency or induce expanded access to data – both of which are underlying goals of our measurement/metrics work. This paper examined those complexities and options in the context of the FCC’s most recent Network Neutrality order and more generally in the context as industry stakeholders seek to manage the flow of information in the evolving Internet ecosystem.

**Policy/economic implications of enhanced mobility support**

During 2015/2016, Dr. Lehr continued his outreach to industry to better understand how mobility is changing the Internet and impacting the industry value chain. This included continued participation in the activities of the MIT Communications Futures Program (http://cfp.mit.edu) and in directing the bi-weekly calls of the Mobile Broadband Working Group (MBWG). The MBWG has been engaged in discussions of emerging architectures for IoT, including the emergence of new crypto-currencies like BitCoin, as well as the implications of evolving cloud infrastructure for ISPs. These latter discussions focused on expanded softwarization of operator networks as evidenced by growth of SDN and NFV, and the requirements to address 5G challenges in the future. These developments have important implications for the future MF environment and the cloud and edge-based capabilities that will need to be supported.

During 2015, Dr. Lehr’s paper on reliability issues in the Internet Cloud was finally published as part of a book edited by Professor Yoo on Cloud Policy issues that emerged from earlier FIA-related work.

Within the project, Dr. Lehr contributed to discussions of policy concerns and trends (e.g., evolution of Network Neutrality and Interconnection regulations, privacy policies, and increasing focus on security issues) to inform the progress of technical designs; and through participation in industry and policy conferences, helped disseminate information about MF and the implications that its expanded capabilities have for policy and markets.

**Enabling dynamic shared spectrum models**

As noted before, MF enables and will benefit from expanded dynamic access to spectrum resources. Spectrum is just another network resource that MF makes it feasible to manage on a much more decentralized, fine-grained, and dynamic basis. However, MF technology alone is insufficient to realize this potential. Significant work is also required to reform spectrum management practices if MF’s potential is to be realized.

In the area of research on shared spectrum management models, Dr. Lehr has continued to engage with and support the efforts of the Senior Steering Group for the Wireless Spectrum Research and Development (WSRD) effort and continued to organize and direct the activities of the MIT Communications Futures Program Spectrum Policy working group (SpecWG), which has been focusing on the challenges confronting policymakers seeking to promote new sharing models in multiple forums and bands (e.g., at 3.5Ghz, 5Ghz, and in the 600MHz incentive auctions).

During 2015, Dr. Lehr authored or co-authored two papers related to spectrum management that were partially supported by, and informed by his participation in the MF work. These included Weiss, Lehr, Aker, and Gomez (2015) which set forth recommendations for how to address the governance of the Spectrum Access System (SAS) that is emerging first in the context of the 3.5GHz band, but provides a model for spectrum management more generally. That paper frames the discussion with the property rights law & economics literature that is also relevant for the management of other distributed but shared resource systems – or, to put it another way, an MF-fueled future Internet.
Dr. Lehr also expanded and further developed his work on refining spectrum rights regimes, resulting in publication of Lehr (2015). This is an active line of work that is continuing and with support from an NSF EARS project that Dr. Lehr is working on with co-PI Jung-Min Park from Virginia Tech. The focus of this work is on how spectrum rights frameworks (including enforcement) may better balance technical and economic incentive considerations in support of more efficient use of scarce spectrum resources.

References for Sec 2.8


3. FIA-NP Research Thrusts
This section provides further details on the “vertical” research thrusts identified in Sec. 1, including cellular-Internet convergence, content delivery, context-aware services and IoT, mobile cloud and technology platforms (including software defined networks, optical and custom hardware designs).

3.1 Cellular Internet Convergence:
Faculty: D. Raychaudhuri, R. Yates (WINLAB, Rutgers University), in collaboration with Suman Bannerjee (U Wisconsin)
Graduate Students: Shreyasee Mukherjee and Parishad Karimi (WINLAB, Rutgers University)

Background: As Internet-connected mobile devices will soon outnumber fixed PCs, a convergence of business models and technical standards associated with mobile networks and the Internet may be expected over the next decade. This convergence process has already started, with cellular standards embracing the concept of "flat" IP-based networks without centralized gateways. With the emergence of the "5G" cellular roadmap [1-3] there are new opportunities to set industry standards with better alignment between the 3GPP based mobile networks and the Internet. In our view, the next logical step in this direction is unified mobile Internet architecture with native support for basic services such as authentication, dynamic association, multi-homing, handover, inter-network roaming, and disconnection tolerance. In this aspect of the MobilityFirst project, we focus on the protocol design and related techniques needed to enable such a unified mobile Internet architecture.

3.1.1 Next-Gen Cellular Architecture: As shown in Figure 1, in the integrated "mobile Internet" architecture, it will be possible to "plug in" multiple wireless access technologies such as 4G, 5G or WiFi without requiring gateways. Such a uniform protocol solution across wired and wireless network technologies will eventually lead to convergence of cellular and Internet standards, in view of the fact that both industries are serving the same mobile end-users. Beyond mobile data, any new protocol architecture should also support the requirements of emerging machine-to-machine (M2M) communications between embedded sensors, vehicular networks, and Internet-of-Things devices, which are expected to grow significantly over the next decade to an estimated 1.5 billion devices by 2017 [4].
We note that a unified mobile Internet architecture is useful to both cellular network operators seeking to improve performance, as well as to more general Internet service providers (ISPs) aiming to introduce mobility services across heterogeneous access networks. For example, an ISP that currently offers standard Internet access service could expand the offering to include seamless mobility across multiple wireless networks such as WiFi hot-spots using standard network elements (router, basestation, access point) without the need for a specialized control framework. This type of heterogeneous wireless access service is sometimes referred to as “open wireless networks” \[5\] in which loosely coupled access networks use a common protocol to support basic mobility needs such as authentication, handover and inter-network roaming. Cellular providers incorporating WiFi hot-spots and 3G/4G small cells to supplement their existing macro-cellular deployments could also use the same flat future IP protocol to provide mobility services across these heterogeneous networks without the need of any specialized network equipment, as shown in Figure 1.

With the emergence of “5G” activities worldwide, we believe that there is a window of opportunity for clean slate designs for the mobile core network. The 5G mobile area has been identified as an important use case, and the project team has increased outreach efforts in that area including a number of talks given (for example at the IEEE 5G Workshop in Princeton held in 2015 and the NGMN Verticals Meeting held in San Jose in 2016). In addition, in Dec 2015, WINLAB organized a one-day technical workshop on 5G Research Challenges with participation of representatives from a number of major equipment vendors and mobile service operators.

**5G Requirements:** We analyze specific wireless access and mobility service requirements and identify the corresponding architectural considerations for their support.

1. **Host and network mobility:** The need for supporting mobility arises when an individual node or a group of nodes, for example a bus/train/plane network, moves and reconnects to the Internet. Previous studies on opportunistic WiFi through vehicular nodes have shown that mobile nodes suffer frequent disconnections. In addition, nodes change their IP addresses every time they associate with a new access point \[6\]. A cellular network provider performs handover between its basestations transparent to the user, enabling them to hold on to their static IP address assigned by the network provider. However data is routed through a gateway which reroutes it to the current basestation the client is connected to. In this regard, Mobile IP tries to achieve the same with the use of fixed mobility anchors \[7\]. Thus, a fundamental requirement for mobility support is to separate host identity and locators through the use of a permanent name (e.g. GUID in MF).

2. **Varying Wireless Link Quality and Disconnection:** Achievable bit rates in both WiFi and 4G systems can show large variations within a fraction of a second. Temporary disconnections due to mobility and/or insufficient signal strength are also common. While these variations are usually handled at the PHY and MAC layers, they invalidate some implicit assumptions in the control algorithms used in the Internet. For example, it has been long known that TCP congestion control...
treats wireless link errors as congestion losses and performs poorly in high variation and multi-hop wireless channels [8]. Given the last mile connectivity is increasingly becoming wireless, such link quality variations need to be natively supported at different layers of the Internet architecture.

3. Accessing Multiple Networks: A typical wireless device in an urban area today might see 3-5 cellular networks and 10-20 WiFi access points, but accesses only one of these due to both technical and business model constraints. Current techniques supporting simultaneous use of multiple interfaces rely on enhancements to the underlying end-to-end transport layer (see [9] and references therein). Specifically, these mechanisms require a multihomed end-point to inform the sender about its multiple interfaces prior to the commencement of data-flow, and a data-striping algorithm on the sender stack that adapts the packet rate of each interface.

4. Adhoc Networks: Wireless adhoc networks are important for infrastructure-less vehicle-to-vehicle (V2V) and sensor network scenarios, last-mile connectivity and applications such as photo/video sharing, local social networking, and multi-player gaming. One view of Internet design is that adhoc networks are just a type of edge network; as long as they are connected to the Internet via a boundary IP router, the protocols used within the adhoc network can be ignored. However, the ubiquity of non-specialized devices requiring support for adhoc networking (e.g. phones, tablets, laptops, vehicular infotainment systems, etc.) forms a strong argument for an integrated design that avoids boundary translation solution.

3.1.2 Multihoming in HetNets: MobilityFirst is based on the idea of separating “human-readable names” of end-users and their routable addresses, with the mapping of flat GUIDs to its corresponding network attachment points (NA) maintained in a distributed fashion at the global name resolution service (GNRS). Any router within the network can update the GNRS with new mappings and query for up-to-date GUID to NA translation. Consider the example scenario shown in Figure 2: When “John’s laptop” connects to the Internet, it is assigned a GUID by the name certification services (NCS). 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. The GUID assigned to the host remains constant for the lifetime of the device. As the device moves, its up-to-date network location’s mapping changes in the GNRS.

After link-level association, the dual-homed “John’s laptop” updates the GNRS with the set of network addresses corresponding to its current points of attachment. Preference policies (for e.g. best path, lowest cost path, striping over both paths, etc.) can also be expressed through this update message, as shown in the figure. When sending data to John, the GUID is resolved through a GNRS lookup to the set of current NAs, in this case NA_{99} and NA_{32} 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. Availability of multiple paths is enabled through link-state routing utilizing GSTAR and EIR. If the user's policy is to stripe data across all the available interfaces, MobilityFirst utilizes a robust hop-by-hop backpressure mechanism to estimate the ratio of data to be sent across each, as explained in detail in [10].

**Evaluation Results:** The simulation topology consists of a single vehicular client with an 802.11 radio moving along a straight roadway, with access points deployed along the road at random inter-AP distances, $d$. We assume the vehicle to be also connected to an LTE basestation, which provides it with a continuous coverage but lower achievable data rate. $d$ is uniformly distributed between 300-500 m to simulate frequent disconnections through WiFi. The mobile client downloads a large file from the server, while moving at a speed of 10 meters/sec (~22 mph) and we measure the raw aggregate throughput that could be achieved in such a scenario. Since baseline TCP does not support striping of data across multiple interfaces simultaneously, we focus on the advantage of using multiple interfaces in comparison to a single interface in MobilityFirst.

As shown in Figure 3, the in-network data-striping (a detailed description of the bandwidth estimation and striping algorithm, is given in [10]) fully utilizes the WiFi interface whenever it becomes available, achieving significant performance gains over LTE alone.

Note that the above multi-homing approach in MF has also been validated experimentally using ORBIT and GENI resources. This service capability will also be used in the mobility service trials planned next for the 5Nines network in Madison, WI.

### 3.1.3 Multi-Network Access:

MobilityFirst natively supports the routing and naming requirements for enabling reliable, network-assisted multi-network access. In MobilityFirst, the GUID-based routing along with the Global Name Resolution Service (GNRS) provides the underlying network functionalities for simultaneous access
across multiple interfaces on the end-hosts. The host can inject an optional service identifier (SID) in the GNRS update, which expresses the host’s policy preferences, for example stripe across all available cellular links. After the router detects multiple network addresses in a packet header, and based on the SID, it performs the routing algorithm to find the next hop. The router which finds different next hops for the network addresses, will be the designated router in charge of scheduling different subflows on different paths. In our previous study, hop-by-hop backpressure mechanism was utilized to determine the ratio of data forwarded towards each interface in heterogeneous networks.

Due to wide variation in bit-rate and diverse coverage across geographic areas, multi-interface connectivity in cellular networks provides significant gains in peak throughput as well as availability. The subflow scheduling based on cross-layer feedback from access cellular networks was investigated for this case. The bifurcation router will request wireless link quality from either the associated base station or an aggregated controller (in case of SDN cellular network architecture) and then schedule the sub-flows towards multiple interfaces accordingly, as explained in detail in [1]. An overview of the design is shown in Fig.1.

**Figure 1. Overview of System Design for Multi-Network Access in Cellular Networks**

**Evaluation Results:**
In order to evaluate the performance of network-assisted multi-homing (NAMH) (i)baseline and (ii)trace-driven simulations in ns-3 in conjunction with LENA LTE module have been conducted. Moreover, NAMH has been comprehensively compared with MPTCP using Direct Code Execution (DCE) in ns-3. (i)In the baseline scenario, one multi-homed UE is moving across two cellular networks on a random trajectory. This scenario is depicted in Fig.2, along with the aggregated throughput for infinite backlog file transfer over best-interface, NAMH and MPTCP.

**Figure 2. Baseline Scenario performance evaluation**
(ii) In the trace-driven simulation, we collected spatially correlated signal strength measurements of two commercial network providers over the span of 10 days along the route shown in Fig.3. These traces, which demonstrate the spatially correlated channel conditions for two carriers are then replayed in ns-3. (The details of data collection and mapping to ns-3 operating range can be found in [1]). In order to investigate the performance gains of NAMH, the download performance of MPTCP, single TCP for each carrier, best-interface and NAMH for all the traces is compared. A total of 10 series of measurements (total of 217 minutes of measurements) of each carrier have been fed into the LTE physical layer module modeling the retrieval of a large file from a server. The NAMH average throughput outperforms MPTCP by 30-80%, while MPTCP and best interface method perform relatively similar (MPTCP 1.15 times better) and better than single TCP connections (MPTCP 1.4 times better). In terms of RTT, NAMH exhibits 12-80% smaller values compared with MPTCP (on average 2.3 times reduction). Single TCP connections show the largest values for RTT (only 5-9% larger than MPTCP). The average throughput vs. average RTT for the large file download scenario across each of the traces is shown in Fig.3. Overall, the results show that multi-network access enabled by MF’s named-object architecture can effectively combine multiple cellular services into a more reliable, higher-speed aggregate channel.

![Figure 3. Mobility trajectory along which RSSI measurements were made, Average throughput vs average RTT for all traces](image)

REFERENCES for Sec 3.1:
Publications:

3.2. Mobile Content Caching and Prefetching in MobilityFirst Networks
Faculty: Yanyong Zhang, K. K. Ramakrishnan (UCR), and Dipankar Raychaudhuri
Graduate Students: Feixiong Zhang

Project goals: The MobilityFirst architecture uses identifiers, instead of network addresses, at the network layer, and a name resolution service (GNRS) [1] for dynamic binding of identifiers to network addresses. As a result, as mobile clients move through network attachment points (e.g., WiFi access points (APs)), its associations with APs are observed by the network. Moreover, we envision that these APs, when equipped with storage, can be utilized as a distributed content caching system. In this project, we aim to utilize the AP's storage such that a mobile client downloads chunks of content from APs it connects to over time, thus improving mobile client's content download performance. We realize this by leveraging aggregated network-level prediction and prefetching as well as popularity-based caching at the network-edge.

Technical approach: Our goal is to improve mobile content delivery performance by anticipating user requests and predicting their mobility trajectories. To effectively achieve this goal, we propose a framework in which an AP’s storage is separated into a cache buffer and a prefetch buffer, with the former caching popular content chunks while the latter buffers chunks that are likely to be accessed by the individual mobile client. That is, while most popular content chunks at an AP are cached at the cache buffer, the immediate (and thus most “urgently” required) content chunks expected to be needed in the near future are prefetched and buffered at the AP's prefetch buffer.

Prefetching content chunks for mobile devices requires predicting the next network that the device is moving to, and at what time. We develop network (access point) level mobility models based upon aggregated mobility information collected by the network, which better captures time-varying mobility patterns than personalized mobility models.

MobilityFirst is ideally suited to develop such a network-level mobility model. Since each mobile device updates its latest network address with GNRS when entering a new access network, GNRS naturally observes each device's mobility pattern. Based upon the aggregated mobility patterns from all the devices that pass the network, we build the mobility model for each network – we model user movement as a second-order Markov chain with fallback to a first-order model [2], and build such network transition probability table (NPT) at each AP. Given the previous AP of a mobile client, the NPT at current AP returns the probable future AP and corresponding transition probability. Each AP also builds a residence time history table (RHT) by extracting statistics from recent residence times observed by the AP. An AP then estimates a particular device's residence time based upon the residence times of past clients. Figure 1 shows the overall framework of our scheme.
Results: We compare our scheme with three other algorithms and report the performance with different values for the Zipf exponent parameter in Figure 2. From the results, we have the following observations. First, with reasonable values for the Zipf exponent, prefetching is more effective than caching. As the Zipf exponent becomes larger, the content access popularity distribution becomes more dominant and thus, caching becomes much more important than prefetching. Then the difference between an LRU cache and popularity-based cache becomes less important. As a result, all four strategies converge when we have large Zipf exponent values. Second, our strategy always fares better than the other three in achieving a greater percentage of local hits at the edge.

References for Sec 3.2
[1]. T. Vu et al. DMap: A Shared Hosting Scheme for Dynamic Identifier to Locator Mappings in the Global Internet, *In IEEE ICDCS, 2012*

Publications
3.3 Enabling Mobile Cloud Services with MobilityFirst

3.3.1 Dynamic Cloud Migration in MF
Faculty: Prashant Shenoy (UMass), K. K. Ramakrishnan (UCR), Yanyong Zhang (Rutgers) and Dipankar Raychaudhuri (Rutgers)
Students: Tian Guo (UMass), Anusha Sheelavant (Rutgers), Wuyang Zhang (Rutgers)

Background: In the context of MobilityFirst, our Mobile Cloud work focuses on the problem of determining how to incorporate new functionality into the cloud platform to serve the needs of mobile users. Our work over the past year focused on designing a geo-distributed cloud platform to handle the aggregate workload dynamics exhibited by a geographically diverse base of mobile users—specifically the spatial and temporal dynamics exhibited by such users.

Technical Approach: To address this problem, we developed a system called GeoScale that implements geo-elasticity into a distributed cloud platform – see Figure 1. Geoelasticity enables resources allocated to a cloud service to be scaled across regions depending on the locations of mobile users and the workload they generate for the cloud service. GeoScale implements a geographic workload clustering algorithm that clusters the incoming workload of a cloud service based on the locations of mobile users to determine the aggregate traffic from different regions. We are currently integrating this functionality using MobilityFirst's GNS service to track user locations and provide location-specific workload statistics. Given these workload statistics, GeoScale combines model-driven proactive provisioning with agile reactive provisioning to handle long- and short-term geographic and temporal workload variations. Our approach is based on a queuing-theoretic model-driven algorithm that determines the cloud server capacity to be dynamically provisioned in each cloud location; virtualization mechanisms are used to migrate and replicate application state across cloud sites to enable the provisioning of cloud service replicas. We implemented a prototype of GeoScale and evaluated it on Amazon's distributed EC2 cloud. Our results show that GeoScale yields up to a 47% reduction in the 95th percentile response times for representative web applications, when compared to local elasticity mechanisms. Further, GeoScale's pre-copying optimizations enabled new capacity to be provisioned in tens of seconds to handle sudden workload changes.
As part of our planned work for year 2, we plan to further integrate GeoScale with MobilityFirst to demonstrate how GeoScale and MobilityFirst work together in concert to handle mobile workload dynamics. We also plan to focus on designing additional cloud primitives to handle dynamics exhibited by individual users, rather than those of the aggregate user base. Specifically, we plan to study cloud mechanisms to handle nomadic users from the perspective of a distributed edge cloud. The overall system model that we plan to develop is outlined in Figure 2 which shows how virtual networks with MobilityFirst service anycast features developed at WINLAB can be combined with the cloud migration techniques developed by Prof. Shenoy's group at UMass.

As shown in the figure, the system uses a virtual MF network as its foundation in order to support seamless global mobility (involving functions such as fast name resolution, authentication and path rerouting). In addition, the virtual network design we have in mind supports the concept of “service anycast with application layer routing (ASR)” which makes it possible for cloud service requests to be routed to a nearby server/cluster with appropriate performance and load parameters. The virtual network is also responsible for resource management and isolation that ensure good quality of experience for users on the move. Each virtual router in the network can be programmed to run an application-specific routing (ASR) protocol which takes into account both network level and application level routing metrics – for example, an ASR metric for cloud service would include both network-level path delay as well as server capacity/load factor in order to be able to route to the “best” service rather than the nearest one.

Work on this project is ongoing and a working prototype system is anticipated by the end of year 2 of the FIA NP project. Collaboration with Prof. K.K. Ramakrishnan’s group at UC Riverside on aspects of edge cloud will also be initiated during year 2 of the project.

Note: An alternative cloud migration approach called “GigaPaxos” has been developed by Prof. Arun Venkataramani’s group at UMass. GigaPaxos is a new, ultra-lightweight implementation of geo-distributed, reconfigurable Paxos that can be used by general third-party object lookup/update services for reconfiguring their geo-distributed placement. A key innovation in GigaPaxos is that it can map each object to its own, unique paxos replica group and easily scales to billions of paxos instances per commodity machine (unlike a comparable placement engine in the first implementation of Auspice that only scaled to millions of paxos instances). Please refer to the UMass report for further details.

Publications:
3.3.2 Accelerating Mobile Applications through Cloud Replication
Faculty: Z. Morley Mao (University of Michigan, Ann Arbor)

**Background:** Mobile devices have less computational power and poorer Internet connections than other computers. Computation offload, in which some portions of an application are migrated to a server, has been proposed as one way to remedy this deficiency. Yet, partition-based offload is challenging because it requires applications to accurately predict whether mobile or remote computation will be faster, and it requires that the computation be large enough to overcome the cost of shipping state to and from the server. Further, offload does not currently benefit network-intensive applications.

This project considers the design of Tango, a new method for using a remote cloud server to accelerate mobile applications. Tango replicates the application and executes it on both the client and the server. Since either the client or the server execution may be faster during different phases of the application, Tango allows either replica to lead the execution. Tango attempts to reduce user-perceived application latency by predicting which replica will be faster and allowing it to lead execution and display output, leveraging the better network and computation resources of the server when the application can benefit from it. It uses techniques inspired by deterministic replay to keep the two replicas in sync, and it uses flip-flop replication to allow leadership to float between replicas. Tango currently works for several unmodified Android applications. It provides speedups of up to 3x for compute-intensive applications and up to 2.6x for network-intensive application. The next step with this project is to integrate with name-based service primitives offered by MF and evaluate achievable performance improvements.

Please refer to the UMichigan annual report for further details.

3.4 Context Service and Internet-of-Things (IoT)
Faculty/Senior Personnel: Yanyong Zhang, Rich Martin, Wade Trappe, Jiachen Chen (Post-Doctoral Associate)
Graduate Students: Sugang Li, Xirou Liu

**Background:** Efficient support of Internet-of-Things scenarios in the future pervasive computing environment is one of the key design goals of MobilityFirst. A project on IoT as a use case for the MobilityFirst protocol stack has been initiated with the goals of developing an architectural reference model and validating it through a prototype system that takes advantage of the new service APIs offered by the MF protocol. Support for IoT is closely related to the architectural design goal of incorporating context-awareness into MF’s GUID-based networking services.

**3.4.1. Supporting Context in MF:** Context awareness is a vast subject that has received much attention over the years. In the communication space, context means that communication partners can change in relation to their environmental context. We found that the question of how communication is impacted by context ultimately means changing the communication source and destination endpoints depending on the environmental properties. While simple on the surface, there is no welldefined and accepted definition of meaningful environmental state. Thus, we first reduced the context space to a few specific definitions. We then demonstrated how using those definitions communication changes as these environmental properties change. We used MobilityFirst structures, in particular the Name->GUID and GUID->Network Address mappings to achieve the dynamic bindings needed for context aware communications.
Figure 1: Example Context Aware Network using Mobilityfirst

From a communication standpoint, we identified the following environmental context properties. The first are network properties which include the source address, destination address, network attachment point, device ID, and 1-hop neighbors. The second set of context state centers around physical properties, including location, time, speed, direction, available energy, and device capabilities. A primary result of our work is that we found that MobilityFirst's ability to quickly change dynamic bindings is critical for context aware communications. Figure 1 shows a typical example of the kind of dynamic bindings needed for context aware communications. In the example, location and time are used as context state to dynamically change the bindings for Tom's communication endpoint. Depending on the time and location, an context aware entity, in this case Tom's location tag, will update the name service and GNRS mappings for the GUID needed to reach Tom.

A second result we found is that MobilityFirst's ability to quickly form anycast and multicast groups also critical for context aware communications. Two examples that illustrate this result are in the vehicular networking space. The first communication paradigm maps well to anycast; for example, locating a vehicle of a specific type and location. For example, one might want to send a message of all ambulances within 10 miles of an intersection. This relies on dynamically changing the GNRS to support anycast over a set of GUIDs, where the set is defined by the context of device type and location. A second example shows how context aware communication leverages MobilityFirst's dynamic multicast capabilities. For example, a communication may go to all vehicles in a given area. In this case, we found we could leverage the GUID->GUID mappings to build sets of communication endpoints matching the context state of device type and locations.

### 3.4.2. Context-Aware IoT Middleware using MobilityFirst:

**Background:** There is a recognized need for unified IoT platforms where objects can be made accessible to applications across organizations and domains. However, most available solutions are based on client-server overlays on today's Internet. These solutions inherit the inefficiencies of the current Internet—especially in terms of mobility, scalability, and communication reliability. To address this problem, we propose to build the unified IoT platform leveraging the key features of Information-Centric Network (ICN) [1] architectures, which we refer to as ICN-IoT. Specifically, we have designed a context-aware Mobility
First IoT middleware to support seamless device configuration, naming, context processing and publish/subscribe.

Technical Approach: Over the years, many stand-alone IoT systems have been deployed in various domains. These systems usually adopt vertical silo architectures and support a small set of pre-designated applications. A recent trend, however, is to move away from this approach, towards a unified IoT platform in which the existing silo IoT systems, as well as new systems are rapidly deployed that will make their data and services accessible to general Internet application. In such unified platform, physical resources can be accessed over Internet and shared across many applications.

This project is aimed at defining context-aware ICN IoT middleware (illustrated in Figure 1), in which overlay network services are only needed for administrative purpose, while the device discovery, device naming, context processing and resource publish/subscribe are directly implemented within the ICN (in this case, MobilityFirst) network.

![Figure 1: ICN IoT middleware architecture](image)

An ICN-IoT architecture based on MF has been developed with the following key features:

**Device Discovery**
Device discovery is the process when a new device especially a low-end IoT device needs to be discovered by the system. In MF, the process is the preparation stage for the next naming process since the new device broadcasts its manufacture ID (serial number) to next hop which later will be used to generate its unique GUID name.

**Device Naming Service**
To enable the reachability of thousands of devices in the MF network, a scalable and efficient naming scheme is critical for the IoT system. Specifically, Local Service Gateway is required to generate a name locally or retrieve a name from naming authority (ex. Handle system [2]/NCRS [3]). Due to the resource constraint of low end device, an IoT aggregator is in charge of aggregating the data and maintains the name for the attached devices.

**Context Data Processing**
Sensor fusion and contextual information are more interesting for most of the subscribers compared with individual sensor reading, and MF provides a decentralized architecture, hence the context data processing module can be implemented at each local IoT site to handle certain low level context processing without administrative involvement.

**Pub/Sub Management**

In ICN context-aware middleware, Pub/Sub management is a centralized service located at the IoT server. Similar to overlay IoT platforms, it provides a unified environment where publisher and consumer can exchange information. The feature differentiates itself from other state of art overlay IoT is that publisher and subscriber only need to exchange the ICN name (e.g. MF GUIDs) instead of the data itself. Once they retrieve service or resource names from the IoT server, data transportation happens within the ICN network. For the next step, we will investigate and evaluate the performance and control overhead of different naming schemes (hierarchical as in NDN, and flat as in MF) on device discovery and Pub/Sub case.

### 3.4.3 Comparison of MobilityFirst and NDN for IoT:

After completing the IoT system design and prototype outline above, we carried out a comparison with NDN to get an understanding of the performance and overhead trade-offs [1]. In this project, we considered the performance a unified ICN platform, in which publishing, discovery, and delivery of the IoT data/services are directly implemented within the ICN network. The resulting network architecture is called ICN-IoT. Specifically, our goal was to compare two different ICN architectures – MobilityFirst and NDN – referring to them as MF-IoT and NDN-IoT respectively. For the evaluation, we considered two realistic IoT applications scenarios: a smart building scenario and a smart campus bus scenario, with the former representing stationary IoT devices while the latter involves mobile IoT devices and content.

![Fig. 3. Delay Performance for ICN-BMS scenario](image)

We have conducted detailed evaluations of an ICN-BMS (Building Management System) and ICN-BUS (School Bus System) systems using NS-3. We first look at the average data reporting delay, which is the average delay between a sink and the server, in Figure 3. In the simulations, each sink aggregates 10 sensors, with each sensor reporting data every 0.5 seconds. We observe that the average reporting delay is mostly influenced by the number of hops between the sink and the ICN-BMS server. Since MF router implements link state control messages and acknowledgement for reliable delivery due to which extra delay is introduced. Hence we observe, the average delay of MF is slightly higher than that of NDN.

We also evaluated the control overhead in the case of producer mobility. In NDN, a mobile producer has to be dealt with some form of flooding to ensure the Interest delivery; while in MF employees late-binding, hence the network only needs to issue a GNS query at the last hop(this query may be issued multiple times until the new location is obtained). MF incurs significantly lower control overhead than NDN, as shown in Figure 4, even when NDN smart flooding techniques are used.
3.4.4 Internet-of-Things Use Case and MF/IoT Prototyping:
The rapid growth in Internet of Things (IoT) deployment has posed unprecedented challenges to the underlying network design. We envision tomorrow’s global-scale IoT systems should focus on service-oriented data sharing and processing rather than point-to-point data collection. The data providers and consumers focus more on what is provided rather than who is using or providing the data. For example, an e-health app only needs to get the step count of the user, without worrying if the count is provided by a Fitbit or a

![Diagram of service relationships in real-world applications](image)

**Figure 4:** Rich service relationship in a set of real-world applications. Services (e.g., Congestion Sensing) can be both consumers and providers. One service (e.g., Step Count) can also be provided by multiple sensors/services.
smartphone (via the built-in accelerometer). At the same time, the step count provider does not have to worry if the data is used by a personal e-health app or a social network app. Figure 10 depicts the service relationship between several example applications. The relationship among services will become more sophisticated when the IoT devices have more functionalities and the function partitioning among services becomes more fine-grained. Requirements like global reachability, mobility, communication diversity and security would also come naturally along with the change in the communication pattern, although the fact of resource heterogeneity in IoT did not change much—many of the IoT devices have limited computation (<50MHz), memory (<50kB), storage (<300kB) and/or transmission capability (MTU<128B), which will make it even more challenging to satisfy these requirements. However, existing (IP-based) networks focus only on locations and point-to-point channels. The mismatch between the dynamic requirements and the functionalities provided by the network renders the IoT communication inefficient and inconvenient.

Information-Centric Networking (ICN) shifts the focus from location to the information itself. This paradigm can be naturally adapted to IoT communication since service is also a type of information. This project tries to study how ICN can be leveraged and adapted to support the above-mentioned requirements in service-oriented IoT communications. More particularly, MobilityFirst (1) is used as an example. We design and implement MF-IoT, which adopts and modifies MobilityFirst, a particular example of ICN. Via real implementation and event-based simulation, we will show the target architecture can satisfy the requirements posed by IoT communication with high efficiency and low energy consumption. We also believe that such design can also be adopted in other ICN solutions like NDN (2) and XIA (3).

**Network Architecture Design:**
The architecture mainly focuses on service-oriented communication which is provided by GUID in MobilityFirst. With some adaptations, the architecture can further support requirements like resource heterogeneity, communication diversity and privacy & trust.

**Service-based Communication**
Since MobilityFirst uses GUID as the unified routing label in the network (similar to IP address in current Internet), it is straightforward to use GUID to represent IoT services. We can grant each service (e.g., Alice’s step count) a GUID (e.g., $G_{ASC}$). The sensors/program/node (Fitbit) which is providing the service GUID with its own NA (e.g., $G_{ASC} \leftrightarrow NA_{Fitbit}$). To send a message/request to the current instance of a service, the data consumer only needs to address the instance with the service GUID (e.g., Alice’s e-health App can send a request with $src = G_{ASC}^{e\text{-}health}, dst = G_{ASC}$).

Unlike IP, MobilityFirst separates the identity (GUID) from the location (NA). The benefit of the design is, when an object moves from one location to another, its GUID does not have to change. In IoT scenario, since our design treats each service as an object (with a GUID), the migration of service instance can be viewed as the service moving from one place to another. Although the location of the service changes, the GUID (which is used by the consumers) can be kept same. In Figure 11, when Alice’s cellphone is responsible for her step count service ($G_{ASC}$), it registers $G_{ASC}$ with its current location $NA_{cell}$, all the queries to $G_{ASC}$ will be routed to the cellphone automatically. This transition is totally transparent to the consumers like e-health and social.

**Resource Heterogeneity**
It is common in IoT that devices with heterogeneous resources would communicate with each other. E.g., a motion detector using IEEE 802.15.4 with small memory and limited power might need to notify a
cellphone which communicates over WiFi and cellular. It is quite infeasible to run full-fledged MobilityFirst on the motion detector due to the length of MobilityFirst packet header (>100B), GNRS lookup and link-state routing. Therefore, to support the functionalities of MobilityFirst across the devices in IoT, we have to lighten the protocol when applying it to resource constraint devices. In MF-IoT, we made the following changes for resource constraint devices:

- **Resource constraint domains**: Domains consist of low-end devices which are using the same link-layer protocol should be created to enable direct communication among the nodes. In these domains, packet header should be compressed for efficient data dissemination. Gateways that translates between different link-layer protocols and (de-)compress the packet headers.

- **GUID vs. LUID**: GUID is the key to global uniqueness. However, in each low-end domain, it is an overkill. To compress the packet header, we use Local Unique Identifiers (LUID’s) which has a one-to-one mapping to the GUID’s used in the domain. In each domain, LUID only consumes 2B and the whole packet header is shortened to 10B. LUID’s can be recycled based on LRU or TTL.

- **Forwarding**: The mapping between GUID’s and LUID’s are managed by the gateway of each domain. When a constraint device joins a domain, it registers its own (listening) GUID’s with the gateway and obtain corresponding LUID’s. To send a message, a constraint device would request for the LUID mapping of the destination no matter if the destination is on a normal device in the core network, a constraint device in the same domain or a constraint device in another domain. If the destination is in the same domain, packets will be forwarded directly, otherwise, the packets will be forwarded to the gateway, translated to a MobilityFirst packet and then sent out. On receiving a MobilityFirst packet, the gateway will create a new mapping for the source GUID if it does not exist in the mapping table. Then the packet will be compressed and forwarded to the receiver based on the destination LUID (mapped from destination GUID).

- **Routing**: Routing in the domain should be light-weight also. We do not limit the choice of routing in each domain. Therefore, energy-efficient routings that are used in Wireless Sensor Networks (WSN) like RPL (4), AODV (5) can be adopted. Centralized (SDN-like) routing can also be used to further reduce the workload on the constraint devices and provide flexibility for all kinds of communications in IoT.

**Communication Diversity**

The target design can support communication patterns like query/response, notification, multicast and anycast. The “observe” mode (6) can also be achieved in MF-IoT with multicast enhancement. MF-IoT further leverages the broadcast media in multicast when available. Once a device has multiple next-hops for a packet, the device can broadcast the packet just once so as to reduce the network traffic and energy consumption.

**Privacy and Trust**

Since ICN is used as the network for IoT, the security and privacy can now shift the focus from securing a channel (like the ones in IP) to securing the contents (or services). With such security paradigm, the network can send a same (encrypted) piece of message to multiple consumers via multicast (for efficiency) and only the legal ones can see the content. The signature of the content can ensure the data integrity and the key that is used to sign the data can also be used to validate if a provider is has the right to serve the (content/service) identity. Figure 12 shows how MF-IoT preserves privacy...
and trust for Alice’s step count service.

MF-IoT can prevent malicious data provider based on chain of trust (7). Each device (Alice’s cellphone and Fitbit) that can serve as a provider for Alice’s step count service would get a key ($K_{A,cell}$ and $K_{A,sc/AF}$) signed by the key of the service ($K_{A,sc}$). Each provider would use its key to sign the payload. The receiver and/or the network can validate the eligibility of the provider by checking if the key used for the signature is actually signed by $K_{A,sc}$. The signature can be used to ensure the data integrity. Attribute-Based Encryption (ABE (8)) can be used to preserve privacy in MF-IoT. Each service has its own attribute in ABE (e.g., $Attr_{A,sc}$). All the messages sent to Alice’s step count service would be encrypted by this attribute. Only the eligible receivers will get the key with attribute $Attr_{A,sc}$ and therefore they can decrypt the message.

We recognize that the adopted mechanism has relative high requirements on the computation, storage and transmission, especially for the constraint devices. To reduce the computation and transmission, the communication parties can either use the proposed mechanism to exchange symmetric keys and use the symmetric keys to secure the communication; or set up intermediate nodes (e.g., firewalls) to validate the messages for IoT devices.

**Prototyping:** We implemented MF-IoT on a low-end sensor board (Atmel SAM R21-Pro). This board uses an ARM micro-controller (48MHz frequency), with 32kB RAM and 256 kB flash. The communication interface it uses is IEEE 802.15.4. As an extension, we link a motion detector (Parallax PIR sensor Rev B) on the sensor board as event trigger (for notification and multicast communication). We run RIOT on this board since it has native C/C++ support and light-weight but well-designed thread-based implementation. Similar to 6LowPAN (9), we build MF-IoT directly over link-layer (802.15.4) in order to minimize the overhead. We use a “MF-IoT” thread to listen to packets from both the applications (from other threads) and the network (from the driver). With proper neighbor, routing and (temporal) GUID table size, MF-IoT only consumes less than 10kB of RAM. We can leave around 16kB RAM for the other applications. With a preliminary evaluation on the prototype, we can see that the forwarding efficiency is acceptable (within 1ms) and the boards can make use of the broadcast media effectively.

The gateway is implemented on a normal Linux box. To enable the communication between the gateway and the sensor boards, we link a sensor board on the gateway via debug port as a network card. Tools are developed to capture 802.15.4 packets from the board and send 802.15.4 packets via the board. With the gateway implemented, we are able to perform seamless translation between the constraint devices and the normal devices. Trials like motion detector sending a notification, service mobility (within- and cross-domain) are tested.

**Evaluation:** To evaluate MF-IoT in large scale, we simulated a set of sensors forming an $N \times N$ grid ($N \in [2,20]$) in our event-driven simulation. Heavy workload (including both multicast and unicast traffic) is placed on the grid to test the performance. The result shows that MF-IoT can outperform NDN thanks to the push-based mechanism. It can also outperform 6LowPAN (with RPL) since it enables direct node-to-node communication. With the use of broadcast media, MF-IoT can further reduce the network traffic and end-to-end latency. We also performed behavior study on constraint to non-constraint and constraint-to-constraint communication when the nodes are moving. The result shows that MF-IoT can provide seamless mobility support for all kinds of nodes. For the detail of the evaluation, please refer to (10)

**Publications:** The basic design of this work is published in IEEE IoTDI 2016 (10) as an invited paper. The paper describes the basic design of MF-IoT with preliminary simulation and behavior study results. A magazine paper that highlights the benefit of service-oriented communication is under review in IEEE Communications Magazine (11).
References for Sec 3.1.4:

3.4.5 Security of MobilityFirst IoT Architecture
Faculty: Wade Trappe and Yanyong Zhang
Graduate student: Xiruo Liu

Background: Though the IoT has a promising future, many current IoT designs do not seamlessly support applications. Historically, a large amount of stand-alone IoT systems were proprietary implementations that ran across the Internet. These fragmented solutions were typically integrated vertically and characterized as "silos" solutions. This "silos" nature conflicts with the open spirit of the Internet and introduces problems of inter-operability and service-level interaction, which limits the benefits of IoT systems and could impede large scale IoT deployment.

MobilityFirst IoT Architecture: As a robust and trustworthy mobility-centric architecture with abundant in-network services for the future Internet, MobilityFirst can address many challenges that today's IoT is facing, such as scalability, mobility, content retrieval, inter-operability, security and privacy, etc. The architecture of a unified IoT platform based on MobilityFirst network is shown in Figure 1.
There are four basic building blocks comprising the MobilityFirst IoT platform: (1) things: a wide variety of devices, such as sensors, actuators and tags, that use embedded techniques to sense, communicate and/or interact with the external environments. (2) MobilityFirst network: MobilityFirst provides the connectivity for different distributed IoT devices and applications. Due to the seamless mobility support associated with MobilityFirst, more dynamic IoT applications and systems can be developed than with the traditional silo approach. (3) IoT middleware: the middleware integrates a basic middleware system with the flat name scheme in MobilityFirst architecture. The functional components of the middleware have been divided into three layers— aggregator, solver and world model. (4) applications: end users who consume the IoT data and may engage in feedback to the external environment through actuators.

Security approaches: Our goal is to provide clean and secure data to various applications/services in the upper layer and make their development/management easy. Thus, as illustrated in Figure 2, we integrate security/privacy mechanisms into the network, the IoT middleware and the lower layer “things”, but not the application layer. Specially, we introduce a middleware to the architecture to handle the data processing and distribution in the data plane as well as system management in the control plane. This approach allows a clean separation of the IoT systems and the underlying network so that the network is only responsible for transmitting the IoT data just as other network traffics. Therefore, integrating IoT platform into the MobilityFirst architecture has a lightweight touch and is easy to deploy.

(1) In-network Security: Rich in-network services of MobilityFirst enable powerful functionalities to protect against a wide variety of attacks. The Name Certificate & Resolution Service (NCRS) serves the role of a certificate registration center that associates human-readable names to certificates. This enables various security methods, such as encryption and authentication, to secure the data transmission above the IoT devices layer. On the other hand, another core service, GNRS, can protect against attacks involving location forgery, such as false registration attack and device misplacement attacks by examining the gateway address. Also, the access control enforced at the GNRS protects the privacy, which is one of the major concerns for the proliferation of IoT systems.

(2) IoT middleware security: The middleware has a three-layer structure as shown in Figure 3. The bottom layer is the aggregator, which supports sensor abstraction to hide the specifics of sensor hardware and presents a single interface to query and subscribe sensor data. The solver layer bridges the aggregator and the world model. It contains a network service solver, a data integrity solver and several function solvers depending on the application requirements. The world model sits in the upper layer and is essentially a database, where all the data must associate with an object. It allows the system to construct complex semantic knowledge.

Three types of solvers in the IoT middleware work together to complete specific tasks and provide service/data to related applications: (1) network service solver serves as a surrogate for in-network...
services provided by MobilityFirst network. Typical operations include assigning and managing GUIDs for devices and data; (2) the data integrity solver performs data validation and cleanses to provide reliable data for other solvers or applications directly; (3) function solvers are application-oriented analysis/processing modules as they perform the operations that are closely related to applications. Since it is impossible to envision every possible application, we allow the removal or addition of function solvers when necessary, which makes the system extensible and flexible.

The security mechanisms at the IoT middleware are enforced by the network service solver and the data integrity solver. A large quantity of low end devices, such as RFID tags and small sensors, have very limited computational capability and strict resource constraints in terms of storage, power, and bandwidth. Hence they are unable to perform many heavy-duty tasks, including the computation operations of public/private cryptography. As the identifier GUID and the cryptography scheme play an important role in the MobilityFirst architecture, it is necessary to have a surrogate to handle naming for low end IoT devices. To better utilize network services, the network service solver has two main functionalities: (1) assign a GUID for IoT devices or data; (2) manage the GUIDs, e.g. renewing a GUID or revoking a GUID. With the assistance of the network service solver, IoT devices and data can obtain GUIDs (notably for information centric networking), which enables many functions, such as facilitating data retrieval, cryptography algorithms, privacy protections, etc.

Unlike traditional communication networks, whose sole task is to deliver data between network entities, IoT systems are facing new security issues, such as a natural loss of calibration, or a deliberate perturbation of the measurement environment by an attacker, or even directly tampering the sensed data by an adversary on the physical device level. More seriously, after being processed and evaluated, the data may be fed back to the real world through actuators. Therefore, a data integrity solver has been introduced to prevent these measurement corruptions and protect the data integrity. The data integrity solver has three main components: classifiers, an array of classifications algorithms that can detect corrupted data; enforcers, an array of filter enforcement strategies that can clean up the corrupted data; and policies, a set of policies that dynamically choose the appropriate classification/enforcement strategies based on application requirements and platform configurations.

(3) Security mechanisms on the devices: Conventional network security techniques fail at the IoT devices because of resource limitations. Hence, we have to explore alternative security approaches: either reuse existing functions and thereby not introduce additional energy burden, or be very selective in what additional functionality we employ. For example, signal processing can be applied (at the receiver) to authenticate whether a transmission came from the expected transmitter in the expected location without introducing energy overhead. Current work is exploring lower layer mechanisms to support authentication of devices and thereby prevent spoofing and Sybil attacks from affecting the network.

References for Sec. 3.4.5

3.5. Technology Platforms

3.5.1 Optical, SDN and Cut-Through Switching in MobilityFirst:
Faculty/Senior Personnel: Byrav Ramamurthy (UNL), KK. Ramakrishnan (UCR) and D. Raychaudhuri (Rutgers)
Graduate Students: Adrian Lara, Pan Yi (UNL), and Shreyasee Mukherjee, Shravan Sriram (Rutgers)
Background: In previous work, we addressed the challenge of cut-through switching at an intra-domain scale [1, 2]. Our motivation was that when both endpoints of a flow are static, storage and routing delays required by MobilityFirst to guarantee efficient delivery of content to mobile devices can be avoided. In such scenarios, it is possible to transfer the data over a lower layer, thus avoiding the overheads at the MobilityFirst network layer. However, the network must remain mobility-aware, and cut-through tunnels must be created taking the mobility of the endpoints into consideration. Hence, a fine-grained, per-flow characterization is needed to decide when a flow benefits from using such a cut-through tunnel. Using SDN to implement the cut-through mechanisms is beneficial, because it allows the controller to route on a per-flow basis and dynamically react to changes in network and mobility conditions.

The results obtained at intra-domain scale (increased performance and flow aggregation) motivated us to focus on inter-domain cut-through switching as a next step. Specifically, we propose an SDN-based routing framework capable of mobility-aware routing and inter-domain cut-through tunneling. The framework builds on top of EIR (Edge-aware Interdomain Routing), the inter-domain routing protocol for MobilityFirst. Additionally, the framework provides mobility-aware cut-through switching at intra-domain and inter-domain scales. The routing framework of MobilityFirst is edge-aware and capable of flow aggregation and multi-domain provisioning, while enabling efficient and granular, per-flow routing. A key challenge being addressed by MobilityFirst is to support highly mobile endpoints. An important goal for MobilityFirst is to efficiently handle diverse flows, from small ‘mice’ flows to large ‘elephant’ flows, for both mobile and static endpoints.

To show the benefits of the proposed framework, we describe a traffic engineering use case where the controller implements three techniques to route traffic and takes full advantage of the intra-domain and inter-domain label-based tunnels. The first technique consists of detecting elephant flows based on their traffic rate and duration. The second consists of tracking the mobility of destination devices to decide when is it beneficial to route a flow through a cut-through tunnel. The third relies on the sender proactively requesting to have traffic sent through a tunnel by modifying the service type field of the packets. By implementing these techniques at the controller, routing is done on a per-flow basis, thus treating each flow independently based on flow-behavior and destination mobility.

A working prototype of the routing framework was developed and deployed on the GENI testbed. Figure 1 shows three SDN-based domains where each controller can query the GNRS. To experiment with inter-domain cut-through switching tunnels we send data from a source node in AS1 to a destination in AS3. Figure 2 shows the impact of inter-domain cut-through switching on the number of packet_in messages received by an in-transit domain. Our preliminary results show that this reduction is in the order of 75%.

A paper on cut-through bypass using the EIR routing framework with SDN controllers has been completed by Adrian Lara at UNL and accepted for presentation at ICC 2016. This work [3] has been carried out in collaboration with researchers at Rutgers University (Prof. Dipankar Raychaudhuri,
Shreyasee Mukherjee, Shravan Sriram) and UC Riverside (Prof. K.K. Ramakrishnan). An example result from this work showing the reduction of control messaging due to GNRS-assisted tunnels is given below.

We show how multiple domains agreeing on an inter-domain label-based tunnel reduce the number of packet_in messages received by transit controllers. To do so, we randomly send 25 flows between AS1 and AS3. Fig. 3a shows the number of packet_in messages received per second by the controller. When all traffic is forwarded without using an inter-domain cut-through tunnel, the controller receives a total of 128 messages (top curve). However, when inter-domain tunnels are created for some of the flows, the total number of messages is reduced to 33 (bottom curve), for a 75% reduction. These results are specific to this topology and flow demands, but our goal is just to demonstrate how the creation of inter-domain tunnels can reduce the control plane delay.

![Fig. 3: (a): The accumulated number of packet in messages received by the AS2 domain controller with and without inter-domain tunnels; (b): The number of messages needed to setup inter-domain tunnels using LDP or the proposed framework](image)

References for Sec. 3.5.1:

3.5.2 High-Speed and Memory-Efficient Forwarding Engine for Future Internet Architecture

Faculty/Senior Personnel: Z. Morley Mao (University of Michigan, Ann Arbor)
Graduate Students: Mehrdad Moradi

Background: Aiming at providing more secure, robust, and flexible Internet, the networking research community recently has focused on developing new architectures for the next-generation Internet. For example, AIP [1] introduces accountability at IP layer, thus enabling simple solutions to suppress a wide range of attacks. XIA [2] supports evolvable Internet by providing the capability to accommodate potentially unforeseen diverse protocols and services in the future. The efforts of the MobilityFirst project aim at developing efficient and scalable architecture for emerging mobility services [3].

All the above proposals shed light on addressing various issues of today’s Internet. We notice that there are two important features shared among their addressing schemes: separation of addressing from network locations and cryptographic verifiability based on a decentralized certification authority. The separation feature enables improved mobility support and multi-homing. The cryptographic aspect facilitates authentication and authorization of control and data messages. However, on the down side,
both features require addresses to be inherently *long* and thus take up significant memory space due to a lack of hierarchical structure to support aggregation. For instance, in the design of MobilityFirst, each address component can be a few kilobits in size. Not surprisingly, it is expected to have forwarding tables in the order of gigabytes in future Internet architecture designs. Such addressing schemes make the design and implementation of high-speed *border routers* challenging, detailed below.

First, memory provisioning becomes more difficult compared to existing network elements. The future Internet will experience a tremendous surge of the number of addressable end-points. Recent studies have predicted that the number of connecting devices and active address prefixes will jump to 50 billion and 1.3-2.3 million respectively by the end of 2020. Second, power consumption of border routers is expected to increase substantially. Most of high-speed routers and switches utilize a specialized fast memory called Ternary Content Addressable Memory (TCAM) due to its speed and in particular its parallel nature in lookup. TCAM is the most expensive and power hungry component in routers and switches. It requires 2.7 times more transistors per bit [4] and consumes an order of magnitude more power [5] compared with the same size of SRAM. Third, the critical-path fast memory components of high speed.

This project has focused on development of *Caesar*, a high-speed, memory-efficient, and cost-effective forwarding and routing architecture for the future Internet border routers. At a high-level, Caesar leverages Bloom Filter, a probabilistic and compact data structure, to group and compress addresses into scalable filters. Our design focuses on improving performance, memory footprint, energy usage, and scalability of routers deployed at future Internet domain borders. Caesar benefits from two logical data structures: routing information base (RIB) and forwarding information base (FIB). The RIB maintains all paths to destinations ADs; the FIB is used to match ingress packets to outgoing links. Similar to modern hardware routers, Caesar implements the RIB and FIB using slow and fast memories respectively.

Caesar has a novel FIB design as illustrated in Figure 1, which consists of two *forwarding paths* or *pipelines*. Each pipeline performs a different series of actions on the input packet, but they both run in parallel. The vast majority of packets go through the *primary path* that leverages our scalable and flexible filters constructed in TCAM. The *backup path* is built from the fast memory and handles uncommon cases where the primary path is not reliable due to false positives in the filters thus *rarely* is less efficient when it accesses the RIB. In other words, the primary path ensures the common-case high-speed forwarding while the backup path guarantees correctness.

Caesar minimally extends the RIB to support routing updates and keep filters of the primary path highly utilized in such events; It also optimizes the computational overhead of hash functions to remove a potential processing bottleneck. Our design provides a practical solution that can be implemented by existing hardware (*e.g.*, SDN switches) with guaranteed performance and can be easily replicated to support specific future forwarding schemes such as GUID routing in MobilityFirst. A paper on the high-speed router architecture has been accepted for presentation at ANCS2015 [6]. Future work will consider evaluation of the proposed hardware architecture for a specific MF use case leading to estimate on achievable high-speed router performance.

![Fig. 1. Overview of Caesar Architecture](image-url)
References for Sec 3.5.2

4. Proof-of-Concept Prototyping, NE Trials and Evaluation Models
This section describes progress on research projects aimed at the design, evaluation and prototype validation of key individual components of the MobilityFirst architecture. During this reporting period (year 2 of the project), the team has focused on the following: (1) upgrade and maintenance of MobilityFirst router software code releases; (2) experimental validation of key protocol refinements (such as global name service improvements, new edge-aware inter-domain routing, multicast, transport layer protocol, etc.); (3) continued deployment of MF on a long-term GENI “slice”, with the experimental network used to run a sequence of demonstrations intended to highlight and validate key MF capabilities; (4) preparations for network environment (NE) trials involving setup of equipment and connectivity at multiple trials sites including the 5Nines network in Madison, WI and the SES satellite CDN trial; and (5) further development of evaluation methodologies including design of a future Internet topology model which incorporates recent trends towards increasing numbers of edge networks, country backbones, etc.

4.1 Router and Host Stack Software Prototypes:
Faculty/Senior Personnel: D. Raychaudhuri, Ivan Seskar, Kiran Nagaraja (now at Ericsson) in collaboration with A. Venkataramani (UMass) & Suman Bannerjee (U Wisconsin)
Graduate Students: Francesco Bronzino

Click Software Router Implementation: Elements that implement GSTAR, EIR, Hop data transport, data-plane storage, and dynamic resolution interface to the GNS have been integrated into a Click-based prototype. Also, simple implementations of anycast, multicast, and multi-homed delivery have also been completed. The prototype also supports a compute layer plug-in of arbitrary services that may be co-hosted with the router – for example, a DPI security filter or an en-route video-transcoding service. Finally, the router is being extended to support the MobilityFirst Virtual Network (together with Application Specific Routing) and content based operations.

The software router can sustain reasonably high traffic rates for realistic evaluation of protocol characteristics and application behavior. Fig. 1(a) shows forwarding performance of the Click-based router prototype with GNS service access and running GSTAR and Hop protocols. The router runs on an Intel Quad Core i7 2.93 GHz CPU and 3 GB physical memory and is configured as a 1-Gbit one-port (with LAN access) route and achieves peak throughput of at least 450Mbps. Several improvements are being explored including thread assignment and scheduling of Click elements on underlying cores, and also moving the current user-level implementations to the Click kernel module. In addition, performance under a mix of workloads where certain flows require GNS lookups, or temporary router storage, and other more advanced router functions such as scalable multicast are currently under evaluation.
We also benchmarked the MobilityFirst host protocol stack performance under wired and wireless access conditions. Fig. 1(b) shows the throughput under different chunk sizes during stack-to-stack data transfer on Intel i7 K875 processors and 8GB of memory connecting over Intel 54 Mbps WiFi interfaces. While the performance is a slightly lower at the higher end to what we get from using iperf with UDP (21 Mbps on average) and TCP (17.1 Mbps on average), the behavior is consistent with Hop transport protocol where protocol overhead due to reliable transfer are amortized over larger chunk sizes. Under link quality fluctuations and occasional disconnections, end-to-end performance of Hop protocol has been shown to exceed that of end-to-end signaling transport protocols.

Open Flow Controller Implementation: The project also includes a thread of effort aimed at developing and maintaining MF code for use with OpenFlow controllers such as FloodLight and OpenDaylight. A typical implementation structure is shown in Fig. 2. Much of this work was completed in the final year of the previous FIA project, and the results obtained show that routing throughputs of the order of 800 Mbps can be achieved on a commodity OpenFlow platform with NA forwarding, while GUID forwarding can be done at speeds between 200 Mbps and 800 Mbps depending on chunk size. These numbers are expected to improve over time as processor speeds increase with Moore’s law and controller implementations become more efficient. Note that the Network Environment (NE) trials planned for this stage of the project will use OpenFlow switch implementations of MF in order to provide virtualization and backward compatibility with IP.

Open Source Software: In an effort to provide openly available tools for researchers to experiment with the prototype of the MobilityFirst architecture, we organized all the developed material over the project
years under a unique prototyping framework [6]. This framework includes the following components: a) all the prototype source code available both as open source software both as ready-to-run Debian packages; b) documentation organized in the form of wiki to support the understanding of the available material in connection to the concepts at the base of the architecture; c) automated tools to deploy and perform experiments in multiple research testbeds, including ORBIT and GENI testbeds; d) a web based tool to be used to track experiments through live monitoring of important events or post-processing statistics experienced during prototype runs.

Current work is focusing on technology transfer prospects; development has been directed to support the widest possible array of scenarios by supporting two different environments: a native MobilityFirst network deployed on top of an experimental testbed (i.e. GENI) and an overlay based network, built on top of the current IP Internet, using industry level platforms (e.g. Amazon EC2). The first testbed builds on our experience running MF on GENI, while the second testbed is a logical progression of the currently deployed Auspice GNS on Amazon cloud. A process of hardening the code toward a second open source release adding new functionalities is scheduled for the second half of 2016. In addition, we plan to provide improved automated configuration tools for ease of use by independent research groups. We also plan to release experimental measurement tools for monitoring and evaluation, using extensions to the ORBIT measurement library (OML) currently used in our experimental GENI deployment.

Note: the UMass group led by Prof. Venkataramani has independently developed similar MF components including a mobility service socket API for clients and an SDN controller for routing. Please refer to the UMass report for further details.

4.2 Experimental Validation of MF Architecture on GENI

Faculty/Senior Personnel: Ivan Seskar, Kiran Nagaraja (now with Ericsson) D. Raychaudhuri (WINLAB, Rutgers University)
Graduate Students: Francesco Bronzino, Kai Su, Shreyasee Mukherjee, Shravan Sriram, Feixiong Zhang

Background: The GENI network has been used extensively for validation and evaluation of the MobilityFirst protocol stack, starting in the first phase of the FIA project, and continuing into FIA-NP. GENI provides the necessary scale and geographic distribution necessary to test key design features such as name resolution and inter-domain routing. GENI also makes it possible to transition from technical experiments to service trials by bringing in opt-in users in different campus networks across the country. Several GENI sites, including the site at Rutgers, have wireless deployments which are equipped to support real-world mobility experiments over a variety of access technologies including WiFi, WiMax and LTE (to be released shortly).
A long-running deployment of the MF network was set up on a GENI slice (virtual network) starting in 2013 and is still being used for FIA-NP evaluations. Currently, we have deployed MF routers/APs at seven GENI sites over the country as shown in Figure 1. The routers, naming servers, and applications run on Xen VMs (total 14, 2 VMs per site), provisioned on InstaGENI racks, each with 1 GB memory and one 2.09 GHz processor core. Each MF router is configured with 1 or 2 interfaces depending on their role as core router or as an access/edge router, respectively. All routers have a core-facing interface connected to a layer-2 network that connects all seven sites. This was setup using a multi-point VLAN feature provided by Internet2's Advanced Layer-2 Service (AL2S). Routers at three sites (viz. Wisconsin, Rutgers, NYU) are configured with a second interface connecting to the local wireless network (WiMAX). Mobile wireless or emulated clients connect to MF network through this interface. Routers are configured with 500 MB of hold buffer space to support the storage-aware routing protocol, and they each have access to a co-located Global Name Server (GNS) service instance, often on the same node. The GNS service is the DMap instantiation of our global name resolution service and it runs at all seven sites using a replication factor of k=3 for distributing GUID-to-NA mappings. The following major MobilityFirst demonstrations were given during 2013-14 using the long-running MF slice setup on GENI:

**Contextual Messaging Using Name-Based Networking Concepts in MobilityFirst – GEC 18, NYU Polytechnic, Brooklyn, NY, October 2013.**

A contextual messaging application -- Drop It -- was developed using name-based networking abstractions provided by MobilityFirst, which allows users to drop messages at particular locations, and to pick up messages left by others at the same location. MobilityFirst allows locations (contexts, in general) to be assigned unique names (a GUID – globally unique ID) which help identify them for network operations such as send, recv or get (for named content retrieval). Locations in physical space can be defined (or fenced) by a set of GPS coordinates, for example, and a persistent GUID can be assigned to them by a well-known service. Next, by maintaining meaningful address mappings for a location GUID in the GNRS, endpoints can send and receive messages to/from this context. For instance, a mapping of location GUID to the set of all phones that dropped messages at that particular location can enable a pure peer-to-peer realization of the contextual messaging service, where the ‘pick-up’ can implemented as an efficient multicast request to each of the phones by using MobilityFirst’s get API. It is also possible to realize alternate approaches to pure p2p, where in-network message caches could enable a more robust operation when phones go offline.
The demonstration was run across five of the seven sites within the long-running MobilityFirst network deployment (shown in Figure 2.a). The two edge sites at NYU Poly and Rutgers WINLAB, hosted both WiFi and GENI WiMAX access networks that were connected back to the GENI core. Ten Android phones (some with dual WiFi/WiMAX interfaces), each running the MobilityFirst protocol stack and the "Drop It" application (shown in Figure 2b) were carried around by volunteers (except two which were static at Rutgers and remotely accessible) who performed message drop and pick-up operations at the several preset locations on the demo floor. Each location was marked with a QR-code tag that encoded the location's GUID and was directly scanned by the app to retrieve the context GUID. We used QR codes to identify locations primarily due to the difficulties of using GPS indoors at the demo floor.

In-Network Rate Adaptation using Compute Extensions in the MobilityFirst FIA – GEC 20, June 22-24, 2014, UC Davis (collaboration with Xiaowei Yang and Yang Chen, Duke University)

Traffic from mobile wireless networks has been growing at a fast pace in recent years and poses significant challenges to service providers in scale and efficiency of data delivery. In the MobilityFirst project, we are looking at using a pluggable in-network compute-layer services that we think could help improve mobile end-user experience by either off-loading end-user computation, or by enabling en-route service-adaptation through context-awareness (e.g., knowing contemporary access bandwidth). Streaming video, in particular, is a perfect example of

![Figure 2a: GEC-18 deployment at five rack sites on the GENI wide-area and edge testbeds at Rutgers and NYU Poly (shown expanded).](image1)

![Figure 2b: The GUI for the DropIt application showing the message drop and pickup screens.](image2)

![Fig. 3: Overview of in-network rate adaptation service using the compute layer in MobilityFirst](image3)
a service that could benefit from our approach. At the current state, most used protocols (e.g. DASH) rely on the ability of a client to estimate the available bandwidth, a task arguably very difficult, especially under wireless and mobile environments. By having an in-network service that would dynamically adapt the encoded bitrate of delivered content according to available bandwidth at the access link would provide a system to move the adaptation logic where that information is more easily accessible. We proposed a network-assisted solution (overview in Figure 3), by leveraging MF’s compute-layer extensions, for in-network rate-adaptation of ongoing video-streams implemented close to the edge where there’s better visibility of load and access link quality.

To evaluate our solution, we used our long-running GENI deployment described above to run the in-network rate-adaptation service for a DASH video streaming application. In order to do so, we modified the VLC DASH plugin to use the MF network API. The DASH-enabled content server was run at the Wisconsin site, and the client ran at the Rutgers WiMAX network. The rate-adaptation service runs within a cloudlet co-located with the Rutgers edge router and was instantiated using the PacketCloud framework. Figure 4(a) shows the overhead in introducing in-network processing for a video stream under 2 different rate-adaptations. While the overhead can be reduced by making right hosting choices for the compute layer, it can also be traded off against the edge bandwidth conditions to dynamically decide benefit of rate-adaptation. In our scheme, the client’s access link is monitored using a routing layer service at the access router and the rate-adaptation is dynamically invoked if the access bandwidth drops below the server encoded rate. For the demonstrations, the drops in access bandwidth were emulated by adjusting bandwidth reports, to simulate link quality variation from client mobility. Figure 4(b) shows the reduction in client traffic with transcoding.

Context-aware services demo (Aditya Yadav, Arun Venkataramani, Misha Badov, Roman Lutz, Jim Kurose, Mike Zink, UMass)

MobilityFirst’s context-based communication generalizes traditional name-or address-based communication to arbitrary primitives. This primitive allows developers to bind a socket to a context descriptor such as msocket.bind([lat, long, radius]).

This capability also allows, for example, emergency notifications to be customized for different target populations, e.g., one for emergency personnel, one for lay citizens, and possibly a different one for children and senior citizens. We have conducted an early demo of
context-based targeted communication in front of the North Texas Council of Governments (NTCOG) in Arlington, TX with participants from the National Weather Service (NWS). We are currently conducting further research and development to make general-purpose context-based communication practical at Internet scales (see Figure 5).

Edge-Aware Inter-Domain Routing -- GEC 22, March 23-25, 2015, Washington DC (Shravan Sriram, Shreyasee Mukherjee and Francesco Bronzino, Rutgers)

Using 5 (Rutgers, NYSERNET, Wisconsin, Urbana-Champaign and Utah) out of the 7 sites available for long-term experimentation on GENI, a demonstration of the protocol designed for Inter-Domain Routing was shown at the GENI Engineering Conference 22 (GEC22) – see Figure 6. The core idea that drove the realization of the demo was to highlight how the novel protocol would make it possible for network operators to expose topology and edge network quality to better support advanced mobility services such as multi-homing or late binding. Moreover it provided an example of how the Edge-Aware Inter-Domain Routing Protocol also enables different types of traffic to be routed differently based on local transit policies.


While context based services were already introduced in previous demonstrations (see points 2 and 3), a new extended demo was presented at the GENI Engineering Conference 22 (GEC22) – see Fig. 7. Two main innovative factors were highlighted as part of the demo: deploying contextual services in MobilityFirst (i.e. geocasting of alert messages) through the use of in-network context services and provide an example of how these services could be tightly integrated together with other innovative technologies (e.g. self-reporting vehicles, aerial vehicles as first responders) to provide novel safety instruments to be integrated as part of the safety agencies daily used tools. This demonstration also showcased an example of how very heterogeneous devices (mostly mobile devices) could easily communicate by exploiting name based communications primitives that characterize the MobilityFirst architecture. The ecosystem of these devices included: connected vehicles, video streaming drones, fixed servers and a set of Android smartphones (also distributed among the conference attendees to be live tested).
5. **MF Virtual Network with Cloud CPS Application, GEC 22 and WINLAB Research Review, June 2015 (Francesco Bronzino and Ivan Seskar, Rutgers, in collaboration with K. Nakauichi, NICT, Japan)**

A demonstration of the virtual networks running natively on MF was shown to industry attendees at the Spring 2015 research review and also at GEC22. The prototype demonstrates multiple VNs running on the same MF substrate, and also shows how ASR (application specific routing) can be used to perform anycast routing from a mobile device to nearby edge clouds using the service addressability feature in MF. This demo is being prepared for a broader US/Japan audience at the 2016 NSF JUNO PI meeting.

**List of MF Demos:**


[11] Feixiong Zhang, Francesco Bronzino, Chao Han, Shreyasee Mukherjee, Akash Baid, Kiran Nagaraja, Ivan Seskar and Dipankar Raychaudhuri, "Content Services in MobilityFirst Future Internet Architecture (FIA)", Poster and Demonstration at the 19th GENI Engineering Conference (GEC-19), March 2014

[12] Francesco Bronzino, Chao Han, Yang Chen, Kiran Nagaraja, Xiaowei Yang, Ivan Seskar and Dipankar Raychaudhuri, “In-Network Compute Layer in MobilityFirst Future Internet Architecture (FIA)”, Poster and Demonstration at the 20th GENI Engineering Conference (GEC-20), July 2014

4.3 Network Environment (NE) Trials

Faculty & Senior Personnel: D. Raychaudhuri, Ivan Seskar, Jiachen Chen, Kiran Nagaraja (now at Ericsson), D. Raychaudhuri, Suman Bannerjee (U. Wisconsin), A. Venkataramani (UMass)

Graduate Students: Francesco Bronzino, Feixiong Zhang, Shashikanth Penugonde, Xiaoyan Hu

Project Goals: The next-phase MobilityFirst project has a major focus on validation and use of the protocol stack in realistic network environments (NEs). The three selected environments for trial deployments of MF are: NE1: Mobile Data Services; NE2: Content Delivery Network; NE3: Public Service Emergency Notification. NE1 of course represents a sweet spot of the MF architecture, and applies to both cellular and ISP-based wireless access networks discussed earlier. NE2 is a very different content network chosen as a counterpoint to NE1 to demonstrate the validity of the architecture for infrastructure media services closer to the wired core network. NE3 is in the emerging category of sensor/IoT/M2M networks which benefit from the key context-aware service feature of MF. Further details about deployment and evaluation status for the NEs are given below:

During the first year of the project, we have worked on setting the stage for the NE trials in terms of overall system design, MF related equipment to be installed, strategies for non-disruptive deployment of MF, client platforms to be used, applications to be supported, etc. There are some non-trivial logistical and practical challenges in setting up the experimental trials, such as deployment of MF routers at customer sites or points-of-presence without disrupting any of their existing services/operations. There is also a need to set up high bandwidth connectivity between our project home locations at Rutgers, Wisconsin and UMass and the remote trial sites in order to connect them to other MF networks and to enable remote network management and monitoring. Our plan is to complete most of the preparations and laboratory validations by mid-2016 and then start conducting specific experiments and user trials during the second half of 2016. The current status of each of the three planned NE trials is further summarized below:

**NE1: Mobile Data Services**: This network environment is intended for experimental evaluation of cellular mobile and heterogeneous networking scenarios. The goal is to validate MF mobility support features from the different perspectives of ISP or cellular network operator. One of the network environments to be trialed (in partnership with a broadband wireless ISP, 5Nines, in Madison, WI) involves deployment of a loosely coupled 4G (WiMAX)/WiFi hetnet infrastructure and then using MF protocol capabilities to provide a seamless mobility experience to end-users. The plan calls for a
service trial involving ~50-100 student/faculty end-users with mobile devices, and will also include a local
cloud computing infrastructure for hosting mobility related services – see Figure 1. The 5Nines public ISP
network is co-located in Madison, WI with the GENI wireless deployment operated by the U Wisconsin
group, and will include connectivity to users and applications accessible through the GENI infrastructure.

During this reporting year of the project (2015-16) project, the team in UW-Madison has implemented a
functional version of the compute layer that is a key component of the MF architecture and one of the
components needed for the 5Nines trial. The compute layer allows distributed computational functions to
be placed inside the network allowing for improved performance and efficiency.

As an example of the compute layer, the PI team implemented a transcoding service at different locations
across an ongoing deployment and trials in and around Madison, WI. These trials are being conducted
across UW-Madison campus network and our partner ISP’s network (5Nines).
This system is currently live and allows us to efficiently broadcast multiple TV channels in the Madison
area. The live streams are available from: [http://m.datn.wisc.edu](http://m.datn.wisc.edu)
This website is live to all users in the Madison area only (due to restrictions from the content providers). It
essentially takes live TV channels and allows to adapt their video bitrate dynamically to channel
conditions for wireless-based viewing. We are currently supporting 20+ TV channels. CNN, Fox, C-SPAN,
and even NSF channels are among them. We are getting lots of local area users who want to watch this
channel on their mobile device. This is a great example of MobilityFirst's compute layer functionality
running on the wireless edge, is deployed and with hundreds of users

There has been some delay in deploying the more complete MF trial due to various logistical factors and
additional time needed to develop the service software. We have made progress with key components
needed for the 5Nines deployment as demonstrated by the emergency responder service scenario over
MobilityFirst which was shown at the GEC-22/US Ignite conference held in Washington D.C. March 2015.
Traffic from the 5Nines network has been tunneled to MF routers in GENI, making it possible to offer
specific MF services such as mobility support (via dynamic GNRS resolution of names), context-aware
message delivery, in-network transcoding, multi-homing across WiFi and cellular and delay-tolerant
network (DTN) operation. Mobile cloud services hosted within the 5Nines local network is also one of the
items to be evaluated in the trial. The Wisconsin team is working with 5Nines personnel on arrangements
for co-locating MF routers at their sites with layer-2 Ethernet VLAN connectivity across their network. The
MF router to be deployed is the SDN version (see Sec 4.1) which makes it possible to set up a virtual
network slice with MF and another VN for backward compatible network monitoring and measurement
functions using conventional IP. This router configuration (which is similar to that used in NE2 described
below) has been set up and tested in the lab at Rutgers and will deployed at specific 5Nines locations at
the next stage of the project. Prof. Bannerjee’s group at Wisconsin also has considerable experience with
mobility experiments using a variety of platforms including smartphones and laptops with multi-homing
capabilities.

Experiments planned for the trial include:
- Validation of basic mobility support via GNRS in MF
- Evaluation of multi-homing performance gains with WiFi and WiMax
- Study of mobile content delivery using MF in-network caching and prediction
- Mobile cloud services such as the compute layer transcoding service developed at UW

The plan is to integrate these capabilities into ~50-100 user devices (~10-20 at the first stage), make
them available to student/faculty volunteers, and then run realistic services over the 5Nines network
which exercise the above mentioned capabilities of MF. The experiments will start in 2H2016 and we
expect to report early experimental results in the Fall.

**NE2: Content delivery network:** The second content services trial was originally planned for deployment
on the Kinber/PennREN network which provides connectivity to a number of PBS stations including
WHYY. The plan for this trial had to be modified due to key personnel changes at WHYY which made it
difficult for us to get the necessary equipment access and manpower support at the site. We were able to
identify SES (a worldwide satellite operator) as a second early adopter for information-centric content services. In particular, SES agreed to trial the MF GUID based content naming technology to realize a satellite-based content delivery network with in-network content caching at edge network routers. This deployment is being realized as an overlay implementation of MF at several sites with connectivity over IP tunnels. The project started in Dec 2015 and has progressed to the stage of a first demonstration system, with full functionality expected by the end of 2016. Further details about this trial project are given next.

**Background:** Content delivery is one of the core functions of today’s Internet. To better optimize user experience (with lower latency, higher throughput), existing solutions tend to place popular contents (caches) closer to the clients. Content-Delivery Networks (CDN) are a particular example of such of solutions. Companies like Akamai, AT&T, Verizon, etc., provide explicit CDN services. The contents placed in CDNs can also range from smaller popular web pages, images to even videos with several GB in size. Video-on-Demand (VoD) is a typical use case of CDNs. To better support VoD, content providers usually place popular contents to the caches in the neighborhoods. Instead of querying data from the servers which are far away, the clients can get the contents directly from the cache in the neighborhood.

In CDNs, especially the ones for VoD, each piece of content usually has to be sent to multiple destinations based on the policy of the provider. To enable efficient content delivery, companies like AT&T build their backbone network and transmit data using IP multicast. We observe that satellites provide an efficient mechanism for this kind of content delivery because of the intrinsic multicast capability.

The trial with SES aims to build such a Satellite CDN (S-CDN) framework using MobilityFirst. Figure 10 shows a simplified scenario of S-CDN. When the provider wants to send a piece of content (Movie A) to a set of edge caches, either it is because there are queries, or because it predicts Movie A will be popular, it can send the content to the satellite. Then, the satellite can forward the data to all the edge caches via broadcast media. The edge caches can choose if it will cache the movie based on its own policy.

MobilityFirst is a suitable network architecture for such communication since it shifts the focus from location (IP) to information. In MobilityFirst, Globally Unique Identifiers (GUIDs) are used to represent objects including nodes, applications and even contents. Thanks to the awareness of contents in the network, the client requests will be anycast to a best source (server or cache) without complicated translation associated with use of DNS for CDN data retrieval, or application-layer solutions like P2P. MF provides both push and pull mechanisms for content caching and retrieval in such a CDN. Further details on the progress of this trial system are given below.
**S-CDN Architectural Design:**

**Overall Architecture**

Figure 14 shows the overall architecture of S-CDN design. The black boxes are the existing components and the red boxes are the entities or modules that have to be added. They include: server logic that can choose what to cache and when to send them, central station that chooses either to send a content via satellite or terrestrial, a protocol that satisfies satellite transmission, cache at the edge, name certification service and object resolution service.

**Unifying Push and Pull Services**

The CDN provider would try to send out data in two circumstances – user query and its own prediction (push). However, in the network, it does not bring much benefit if we try to distinguish these two scenarios. Therefore, in our design, the CDN provider will use a single API (push) when he decides to send data no matter in which situation.

The push mechanism is important for ensuring QoS. In normal CDN case where the contents are only queried, the first client who queries the data (even if it is popular) will suffer high latency and low throughput. This is undesired in VoD case. Therefore, we enable the push mechanism so that the provider can place the popular contents (e.g., the episodes that are going to show tomorrow) in the cache even before the first client queries.

**Multicast-aware Routing Logic**

To make S-CDN efficient, two choices have to be made: 1) which destinations to send to, and 2) how to send it (satellite or terrestrial). We believe that these two choices should be made by two parties separately. The CDN provider has the information about content, so that it can predict the popularity and the locality of the content. Based on the policies and the video features, it can decide which edge caches need to receive the content (or even make the caches collaborate). However, the network has more information about the topology and the link state (via exchanging routing information). It will be more efficient if we leave the real deliver job to the network.

Therefore, in our project, the S-CDN provider would predefine some multicast groups based on the policy. While sending each content, the provider would address the destination with either the GUID of edge cache or the multicast GUID. The multicast-aware routing logic (network) will take over the deliver job from that on. At the central controller, the routing would decide if it will forward a content to satellite based the cost for both mechanisms for the same multicast group. If it decides to use terrestrial links, path will be built through the multicast routing logic.

**Transport Protocol for Satellite**

MobilityFirst uses hop-by-hop reliable transfer to better satisfy the communication in mobility cases. However, hop-by-hop reliable transfer is difficult to achieve in satellite links mainly for two reasons – large latency and lack of backward transmission. It is difficult for the edge caches to send ACKs (or NACKs) back to the central station. To address this issue, we modified the transport protocol when it is from the satellite. We allow unreliable transfer (no CSYN or CACK) but with checksum for each packet to validate the integrity of the content. On seeing a corrupted packet, the edge cache would send an explicit query via the terrestrial link for just a part in a data chunk.
Edge caching
We placed an extra module to store contents at the edge. The module uses both the RAM and the hard disk on the router to enable large cache size (given the fact that Video caches might consume several TB of storage space). We build an index and a small fast access space in the memory and use the hard disk as the main storage. Hardware like Solid-State Drive (SSD) can be further leveraged to speed up the cache retrieval.

Name Certification Service (NCS) and Object Resolution Service (ORS)
Since MobilityFirst uses 20-byte GUID as the routing label, it is critical for the provider to get the GUIDs while creating the content. It is equally important for the consumers to get the GUID for the content they are going to query. Name Certification Service (NCS) is adopted in this project to grant GUIDs for contents. After getting the GUIDs, the provider would “advertise” the contents to the Object Resolution Service (ORS) pretty much like how they give information to search engines nowadays. Based on the information the provider gives, the ORS can index the metadata and allow the client search the contents that might interest them. The clients would get the GUIDs they want during the interaction with ORS and then send out queries to get the real content.

Prototyping:
The first stage of the project is mainly focusing on the proof-of-concept demonstration. The purpose is to show that MobilityFirst is capable for CDN data delivery and can be easily adapted to satellite communication. The prototyping is divided into 3 phases:
Phase 1: Basic component test in ORBIT lab (February – June, 2016). Test the components in a controlled environment. Satellite is emulated using a PC with controlled delay and error rate.
Phase 2: Component test in GENI (July – October, 2016). Test the efficiency in a geo-distributed environment. VLAN Tags will be used to emulate satellite traffic.
Phase 3: Real satellite test (November –December, 2016). The prototype will be tested on a real satellite with receivers across the country.
An intermediate first phase demo showing the basic functionality of the CDN in a lab setup was completed in March 2016 and demonstrated to SES engineers. Work on the prototype is continuing and the next milestone is a full system emulated demo on the ORBIT testbed.

Publications:
Feixiong Zhang finished his PhD dissertation (1) on content delivery in MobilityFirst. He successfully defended his work on Apr. 25th, 2016.

The project is in collaboration with SES. The project is reported by Yahoo Finance on May 4, 2016 (2).

References for Sec 4.3:

NE3: Public Service Weather Emergency Notification System: This network environment is in the class of sensor networks/Internet-of-Things scenarios which benefit from context-aware MF services. Emergency notification systems also benefit from accurate large-scale multicast and delay tolerant delivery over multiple radio interfaces. The proposed trial aims to integrate MF as the networking substrate for a metro-scale (Dallas/FW) meteorological sensor deployment (CASA) and context-aware message delivery to first-responders and citizens (Figure 4). This trial will leverage the existing CASA
deployment resulting from an NSF ERC and will highlight several service flexibility features of MF including (i) context-aware multicast message delivery on the basis of attributes such as location, mobility, and individual demographics (e.g., age, access, language), (ii) CASA code migration/execution using the GNS to provide location transparency, (iii) advanced wireless edge network to enhance MF-enabled services (e.g., multi-homing, DTN delivery or geo-location). The trial deployment engages Emergency Response managers (in Dallas FW), AT&T Dallas/FW, and end-users. Key performance metrics for this trial include usability from the application provider’s perspective and quantitative metrics such as notification delay, and network efficiency improvements via mechanisms such as multicast and network storage.

The UMass group initiated an activity on conducting a field trial of the MobilityFirst architecture through a deployment of a GNS-based hazardous weather notification system in the CASA radar testbed in the Dallas-Fortworth area. To this end, we have been working with Brenda Philips who co-directs the CASA ERC effort as well as a HazardSEES project on hazardous weather warning systems as well as people from the National Weather Services (NWS).

Before this, we carried out a proof-of-concept demonstration of MobilityFirst's context-aware emergency notification service to the North Texas Council of Governments (NTCOG) in Arlington, TX including participants from the National Weather Service (NWS). This early demo is encouragingly in line with one of the three planned MobilityFirst field trials in the FIA-NP phase, and a lot more research and development is needed to actually accomplish the NP field trial itself. We have since developed a first-cut prototype of the hazardous weather warning app for IOS devices based on the GNS as the back-end and plan to begin beta-testing it shortly with internal users in the April-July storm season. Based on the experience gathered during this period, we will refine both the app, the GNS backend, as well as the middleware called the "alert control system" that provides controls to the disseminator stakeholder (e.g., a municipality or a company marketing the app) to tailor the warnings according to customer preferences. A key distinguishing feature of this MobilityFirst GNS-based weather app compared to many other weather apps on smartphone app stores is its ability to customize warnings according to end-user and disseminator preferences. For examples, end-users can specify sophisticated attributed such as warning radius of hazard, lead time, etc. for when they should be notified and disseminators can tailor different warnings to different groups of people (e.g., emergency personnel, lay citizens, senior citizens, vehicular passengers, and so on). The GNS' support for context-based communication is key to making this feature possible and practical.

Further details can be found in the UMass annual report.

4.4 Evaluation Methodologies:

4.4.1 GeoTopo - A Geographically Aware Topology Model for the Future Internet

Faculty/Senior Personnel: Yi Hu (Post-Doctoral Associate), K.K. Ramakrishnan (UCR) and D. Raychaudhuri

Project Goals: Network topology plays a critical role while designing and evaluating network protocols. When evaluating the impact of increasing traffic, changing economic interests of network providers and users, and their relationships as well as the impact of increased use of mobility, having a tool to create realistic and scalable topologies
is very useful. Existing topology generators do not comprehensively model the evolving characteristics of the current Internet, such as the ‘flattening’ of the Internet as a result of direct peering between edge networks at Internet Exchange Points (IXPs). Existing generators are focused on the properties of the graph and are more suitable for generating the traditional hierarchical structure of the Internet. We have developed a more comprehensive topology generator, which we call ‘GeoTopo’, that can take as parameters many of the realistic characteristics of today’s network topologies, such as the inter-AS connection and AS- peering policies, business relationships, intra-AS topology structures and the influence of the large-scale IXPs and open peering, as well as geographic and demographic information.

GeoTopo is designed to not only fulfill the requirement of mapping real world network demands to a typical realistic topology, also model the topology evolution for projecting envisioned future network demands to an evolved topology. Figure 1 illustrates the three major aspects of the Internet and their evolving directions that GeoTopo aims to model. First, AS geo-coverage and PoP deployment in the topology is modeled, which can represent the network’s geographic expansion. The studies on Internet evolution show that major content distribution networks (CDNs) (e.g., Google, Akamai, Limelight) have expanded to almost every region of the developed world and have a ongoing trend to further extend their geo-coverage globally. In GeoTopo, we categorize AS geo-coverage into four levels: metro, regional, country and global level and indicate geo-locations of PoPs from each AS in the topology. Second, the AS peering policies and IXP deployment are modeled to represent the formation of inter-AS peering links, whose denser fabric at IXPs and the wider deployment of IXPs are an important factor for Internet “flattening”, a significant Internet evolving direction. Besides capturing the IXP deployment, AS peering policies for peering inter-AS link formation, in GeoTopo we also model the AS resilience requirement for forming customer-provider inter-AS connections. Third, GeoTopo aims to generate large-scale topologies that consisting tens of thousands of ASes in the Internet and model the hierarchical roles of an AS like core AS or edge AS and business type of an AS. The AS number from each type or each hierarchy in the topology can represent the Internet growth featured by emerging edge ASes, which is resulted from easier AS formations and more prevalent network infrastructures.

Technical Approach: None of existing topology generators can fully capture the above three engineering aspects of the Internet and its corresponding evolution directions. The majority of topology generators focus on graph properties of the network, such as hierarchical structures or degree distributions. While it is important to reflect the graph theoretic properties of network topologies, the evolution of the Internet over the last two-to-three decades indicates that engineering factors also play a fundamental role in the eventual topology of networks. Lacking of engineering considerations makes this line of generators unable to map real world network demands to the topology or the envisioned demand to the projected topology. Recently, a topology generator iGen [1] models the geolocations of PoPs and intra-domain structures, also, the principles in [2] model router technologies and user demands for intra-AS topology generation, but they only work efficiently on a single AS level, incapable to capture inter- AS relationships and their connections. Agent-based network formation models [3], [4] are developed to generate inter-AS connections with modeling AS economic interests and traffic flows, but they only work on AS level without considering PoP level intra-AS topologies. A critical limitation of agent-based network formation model is that they can only study a limited number of ASes (e.g., 500) which is far insufficient to model the real Internet consisting of more than 40000 ASes and its growth featured by emerging edge networks.
The measured topology we use in this experiment is from RocketFuel data, which contains PoP-level intra-AS and inter-AS topologies of 21 tier-1 ASes, 36 tier-2 ASes, and 12 tier-3 ASes. In total, over 1900 PoPs are included in the RockeFuel topology (RF topology). We focus on modeling the inter-AS topologies and choose IXP peering model in GeoTopo to generate synthetic topologies. We find that in RF topology over 50% of the inter-AS links are between PoPs co-located in a set of cities. Such inter-AS connections can be well captured by IXP topology. We extract the IXP located cities from RF topology and for each AS we extract their peering policies based on their tier information. For the degree based model we profile the inter-AS degree distribution and joint degree distribution and follow the method in [5] to generate synthetic topologies. Due to their lack of modeling inter-AS links between a pair of connected ASes, we choose the pair of nearest PoP from each AS to form the inter-AS link. We repeated the synthetic topology generation experiments for 50 runs to plot the comparison results in Figure 2.

The results show that GeoTopo outperforms the degree based method in modeling the network hierarchy and network delay when compared with actual data from RocketFuel. Specifically, GeoTopo captures the network hierarchy phenomenon that a small number of core links that carry the large volume of network traffic aggregated from the large number of edge links, which is represented by the inter-AS link rank results in Figure 2. As both GeoTopo and RF topologies have less than 0.06% of links with ranks higher than 0.1 and over one third of links of rank less or equal to $10^{-7}$. However, degree based method fails to capture such distinctive features of core and edge links.

A paper on GeoTopo [6] was presented at ICNP held in Nov 2015.

References for Sec. 4.4.1:

4.4.2 Mobility Measurement & Modeling
Faculty/Senior Personnel: Arun Venkataramani & Jim Kurose (UMass-Amherst)

This effort is continuing from the previous reporting periods. Our research in [Yang 2014], has continued our measurement study of user transitioning among networks, using IMAP server logs to track a user’s changing network address. This research differs from previous mobility studies that have developed
models of a single device's changing MAC or IP addresses while physically moving between access points or base stations. During this past year, we have expanded our measurements from a group of 80 users to more than 7,000 users. We have also developed and validated a parsimonious Markov chain model of canonical user transitioning among networks, and studied the clustering of users.

Our measurement study quantitatively characterizes network transitioning in terms of transition rates among networks, network residency time, degree of contemporaneous connection to multiple networks, and more. We find that users spend the majority of their time attached to a small number of access networks, and that a surprisingly large number of users access two networks contemporaneously. We also show that our Markov chain model of a canonical individual user, in spite of its many simplifying assumptions, can accurately predict aggregate transition rates, the degree of contemporaneous multi-homing, and other key network-transitioning performance metrics for an aggregate population.

These measurements provide quantitative insight into the location management and signaling overhead needed by modern and proposed name/address translation and location management protocol, such as MobilityFirst, and the ability to design, dimension and analyze such systems; indeed our measurements from [1] were used in our evaluation of location-independent network architectures in [Zhao 2014]. More generally, we believe that while physical mobility and the design of link-layer and intra-sub-network handoff protocols are relatively well-understood, the behavior, modeling and measurement of users transitioning among networks and the design of protocols for managing that mobility at global scale are much less well-understood. This work is an important step in deepening that understanding.

Our second mobility measurement and modeling activity accomplished under this grant during the past year measured and is characterizing the physical mobility of devices in a large campus network. Our work in [Steshenko 2014] uses network management logs from a campus 802.11 wireless network of nearly 4,500 ARUBA access points (APs) at the University of Massachusetts Amherst, to characterize the mobility of tens of thousands of network users. We extract raw event data from these logs and transform them into per-user, per-session movement trajectories among APs. We are using these logs for evaluating our own mobility management protocols and have shared these logs with several other research groups at UMass. We have developed a Terms of Use agreement with the University of Massachusetts to allow these traces to be used by research groups outside the university. We have completed the first measurement and modeling activity described above and presented a paper describing the research at IEEE INFOCOM 2015.

See UMass annual progress report for further details.

References for Sec 4.4.2:

5. Education & Outreach Activities

Note: Information on activities in this section pertains primarily to the Rutgers site (lead institution). Please refer to individual site reports for further details on education and outreach at collaborating institutions.

5.1 Technical Community & Industry Outreach

MobilityFirst project members continued to speak at conferences and industry events with the goal of disseminating the architectural concepts. A list of external papers and conference talks can be found in Sec. 6. These include keynote or invited talks in other countries including the Tokyo Wireless Technology Summit, March 2014, Japan; USTC and Tsinghua University, China, Oct 2014, and IEEE Comsnets, India, Jan 2015, IEEE 5G Workshop, April 2015, NGMN meeting, March 2016, etc.. We see an increased interest in clean-slate designs for emerging standards such as the 5G mobile core network or NGMN, Next Generation Mobile Network. IETF’s recent focus on ICN and IoT standards is another opportunity to
consider networking features beyond IP. The project team has also engaged with US government/DoD agencies through presentations at forums such as LSN (Large Scale Networking) and the US Army ITA (International Technology Alliance), and US Army CERDEC. Project members have also engaged with industry, for example with Ericsson on applications to next-gen mobile networks, Huawei on information-centric approaches to IoT, and Cisco on edge-cloud mobile services.

WINLAB hosted one-day research workshop for industry participants on the topic of 5G research challenges in Fall 2015, covering many of the architectural issues and mobility/IoT use cases.

5.2 Educational Outreach and the REU Summer Internship Programs
The project team has also been active with education and outreach activities with highlights including summer internship programs for undergraduates in 2014 (supported by an REU supplement). Students worked on a variety of MF related topics including context-aware application development, social media applications and real-time virtual reality type applications running on MF. (1) Thesis research for PhD students and MS students in the ECE and CS programs – topics include intra- and inter-domain routing, global name resolution, context-aware services, mobile content delivery, security and software-defined network implementation. A total of 5 PhD students and 3 MS thesis students are currently working on MobilityFirst related research projects at Rutgers, with two graduating in Spring 2016. (2) Curriculum development for related graduate courses at Rutgers – in particular, the MobilityFirst architecture inspired a clean-slate protocol design and prototyping project in Prof. Raychaudhuri’s ECE544 Communication Networks II class during the period 2012-16. The class project incorporated several novel elements including a standards committee process during which groups of students negotiated on harmonization of a single protocol specification prior to implementation. The Spring 2015 ECE544 class project was aimed at development of a clean-slate information centric protocol for content delivery, while the Spring 2016 class worked on an “n-out-of-m” multicast concept.

In addition, the project hosted a total of ~8 REU students in 2015 under the structure of WINLAB’s summer internship program, and another 3-4 are expected during 2016. The REU program is open to students from Rutgers as well as other universities in the US. Students received training and prototyping experience on a number of topics including concepts for future Internet applications and mobile Android platforms/apps. The REU summer program has been particularly useful in exposing undergraduate students to future Internet application concepts and has led to prototyping and demos of a number of innovative mobility-oriented applications running on MobilityFirst. Corresponding PhD research, curriculum development and undergraduate research activities were carried out at each of the other sites – please refer to individual site reports for further details.
(B). Products:

Papers: (from start of FIA-NP project)
10. Ashish Patro and Suman Banerjee, "Outsourcing Coordination and Management of Home Wireless Access Points through an Open API", in Proceedings of IEEE Infocom 2015
22. A. Lara, B. Ramamurthy, K. Nagaraja, A. Krishnamoorthy, and D. Raychaudhuri, "Using openflow to provide cut-through switching in MobilityFirst," Photonic Network Communications, pp. 1–13, 2014
23. A. Lara, S. Mukherjee, S. Sriram, B. Ramamurthy, K. Ramakrishnan and D. Raychaudhuri, "SDN-based Inter-domain Routing with Cut-through Switching for the MobilityFirst Future Internet Architecture", under submission to ACM SOSR 2015
31. Francesco Bronzino, Chao Han, Yang Chen, Kiran Nagaraja, Xiaowei Yang, Ivan Seskar and Dipankar Raychaudhuri, "In-Network Compute Extensions for Rate-Adaptive Content Delivery in Mobile Networks", In Proceedings of International Workshop on Computer and Networking Experimental Research using Testbeds (CNERT), October 2014
35. S. Mukherjee, A. Baid and D. Raychaudhuri, "Integrating Advanced Mobility Services into the Future Internet Architecture", in Proceedings of IEEE Comsnets 2015 (Invited Paper)


Demos:


[12] Francesco Bronzino, Chao Han, Yang Chen, Kiran Nagaraja, Xiaowei Yang, Ivan Seskar and Dipankar Raychaudhuri, “In-Network Compute Layer in MobilityFirst Future Internet Architecture (FIA)”, Poster and Demonstration at the 20th GENI Engineering Conference (GEC-20), July 2014


MS and PhD Thesis Dissertations (at Rutgers):

Software & Technology Products:
1. DMap and Auspice implementations of MobilityFirst global name resolution service (GNS). MobilityFirst GNS Portal, https://GNS.name/
2. MobilityFirst Intra- and Inter-domain Routing Protocol (GSTAR, Core-Edge and EIR InterDomain routing) Software.
3. MobilityFirst Router supporting GUIDs, GNS lookup, intra/inter-domain routing and in-network storage features (both Click-software based and OpenFlow/SDN based versions), http://mobilityfirst.winlab.rutgers.edu/Prototype.html
(C). Impacts:
The impacts of this project can be summarized in the following categories:

1. **Impact on networking research community:** While it is still premature to evaluate the impact of the proposed MobilityFirst architecture on the networking research community, some of the key ideas (such as the separation of names and addresses, the use of a fast global name resolution service for the Internet, storage-aware routing, and integrated mobility support) are considered useful and have started to influence other groups and projects in the field. Related future Internet architecture research projects have appeared in Europe and Asia, and the MobilityFirst project members have been invited to participate in research forums, workshops and conferences where is subject is being discussed. The MobilityFirst project has also been visible at GENI and US Ignite conferences and other testbed venues, where several proof-of-concept demonstrations of the key protocol concepts have been given. This has led to some research collaborations beyond the current project team, for example with U Tokyo on implementation of MobilityFirst on the FLARE SDN platform, with NICT, Japan on mobile cloud services using MobilityFirst as the foundation, and with USTC China on advanced media services over MobilityFirst.

2. **Impact on industry:** The MobilityFirst project has maintained technical interfaces with a number of corporate research organizations including AT&T, T-Mobile, InterDigital, Ericsson, NTT DoCoMo, Verizon Wireless, Huawei Technologies and Toyota ITC using the WINLAB industry sponsor program as the base. Project results have been reported on a regular basis including the Spring and Fall 2014 industrial advisory board (IAB) meetings held at WINLAB. A full-day research review on Connected Vehicles and Internet-of-Things (IoT) was held Spring 2015 and included participants from major auto manufacturers. A second research review on 5G mobile network was held in Fall 2015 and had a strong attendance of over 100 participants from several major companies. We have also started engaging on aspects of MobilityFirst with global communication equipment vendors, including Cisco for mobile edge cloud, Huawei for IoT, and Ericsson on future mobile network architecture.

3. **Impact on Government/Policy:** The MobilityFirst project has a thread of activity led by an economics/policy expert (Bill Lehr at MIT) aimed at identifying the relationship between proposed Internet architectures and public policy questions related to broadband, spectrum and wireless access. The project has produced policy papers and workshop presentations aimed at the policy community, most notably on the topic of dynamic spectrum access, open wireless networks and LTE.

4. **Impact on Education:** The FIA-NP project continues to train several of graduate students at the PhD/MS level, providing opportunities to develop skills in network technology, protocol software development, system prototyping, mobile application development and so on. A total of about 5 PhD students at Rutgers have graduated from this project in the past 2 years. In addition, the REU program associated with the project has provided research exposure to a number of undergraduate students introducing them to mobile application development and network prototyping. The project has also inspired novel curriculum concepts for graduate networking courses at Rutgers including a one-of-a-kind Computer Networks class project in 2014, 15 and 16 aimed at design and standardization of a simple clean-slate information centric network.

(D). Changes & Problems:
The Network Environment (NE) trial on content services planned for deployment at PBS stations in PA has been changed to a satellite content delivery network trial with SES, a global satellite network operator based in Luxembourg and Princeton, NJ. The PBS trial encountered logistical and staff support problems due to key personnel changes at WHYY Philadelphia, so that we decided to pursue the same kind of trial with another early adopter, SES. The technical features of MobilityFirst to be evaluated in the SES trial
are content addressability and caching, very similar to what was planned with PBS so that there is no impact on the evaluation plan as a whole. There has been some delay in the project timeline due to the change, but we are making good progress with the revised trial with SES and expect to complete the evaluation by the end of 2016.