# GSTAR: Generalized Storage-Aware Routing for MobilityFirst in the Future Mobile Internet[*]

Samuel C. Nelson, Gautam Bhanage, Dipankar Raychaudhuri
WINLAB, Rutgers University
671 US1 South, North Brunswick, NJ 08902
{snelson, gautamb, ray}@winlab.rutgers.edu

## ABSTRACT

The Internet is at a historic inflection point where mobile, wireless devices are becoming so dominant that core architectural changes are necessary to efficiently support them. This paper presents the high-level concepts and design decisions used to realize the key routing component of the *MobilityFirst* architecture, which is a clean-slate project being conducted as part of the NSF Future Internet Architecture program. In particular, we describe GSTAR, a mobility-centric generalized storage-aware routing approach based on the following key design principles: separation of names from addresses, late binding of routable addresses, in-network storage, and conditional routing decision space. The GSTAR protocol described is based on hop-by-hop forwarding of large protocol data units (PDUs) between routers with storage. The packet header incorporates both name and address information enabling routers to execute a hybrid forwarding algorithm that uses topological addresses when available and refers back to names (i.e. global identifiers) to deal with dynamically changing points of attachment and disconnection. At a local level, GSTAR utilizes both fine-grain path quality information and DTN-style connectivity information to deal with the many challenges found in mobile environments.

## 1. INTRODUCTION

The recent proliferation of wireless, mobile devices highlights the need for flexible, efficient and robust support of mobility services in the future Internet [1]. As a result, there has been renewed interest in “clean-slate” proposals which try to fundamentally address the paradigm shift towards mobile communication. Research programs exploring clean-slate architecture include NSF future Internet architecture (FIA) [2], Future Internet Design (FIND) [3] and GENI [4] in the US, FP7 Future Networks [5] and FIRE [6] in Europe and NGN (New Generation Networks) [7] in Japan. While FIA, FIND and FP7 focuses on re-thinking of the basic design principles of the Internet, GENI and FIRE aim to provide the large-scale experimental networking infrastructure necessary for validation of new protocols. The *MobilityFirst* project, funded by the NSF FIA program, is one such effort currently being carried out by a team of 8 US universities led by Rutgers University. *MobilityFirst* recognizes the coming predominance of mobile networking [8] and aims to directly address the challenges of this paradigm shift.

*MobilityFirst* is designed around the principle that mobile devices, and their associated applications, must be treated as first-class Internet citizens. There are many challenges associated with integrating wireless, mobile communication as a core element of the Internet architecture, including mobility, varying levels of connectivity, multiple network attachment points per device, and a desire for flexible, group-based routing paradigms. Current Internet protocols, such as TCP/IP, are limited in their support for these challenges as they were built using a connection oriented model. Instead, *MobilityFirst* takes advantage of Moore’s Law improvements in processing and storage, shifting some intelligence into the network, decreasing the emphasis on end-to-end setup.

In this paper, we present a high-level, architectural view of the *MobilityFirst* routing component, an integral part of the *MobilityFirst* architecture. The design on this component follows from four core principles: separation of names from addresses, late binding of routable addresses, in-network storage, and a conditional routing decision space. Utilizing these principles, we present an efficient and flexible approach to both global- and local-scale routing that is capable of handling the aforementioned challenges. This includes utilizing high-level network services together with low-level routing protocols to move data closer to the destination network and, once in the destination network, combining storage-aware routing and DTN techniques to traverse the final hops.

The main contributions of this work are three-fold:

1. An exploration of why, fundamentally, current Internet design fails to handle the challenges brought about by mobile devices, which results in the extraction of the guiding principles in our design.

2. A global-scale routing approach that works on names and addresses by utilizing both low-level routing protocols and higher-level network services.

3. A local-scale routing approach, including intelligent buffer management, that utilizes in-network storage to dynamically adjust to varying link-quality and disconnection.

The rest of this paper is organized as follows. Section 2 presents a discussion on the challenges brought about by mobile communication, and a set of guiding principles to help. Section 3 presents our efficient and flexible global- and local-scale routing approach that fundamentally addresses

[*]Research supported by NSF Future Internet Architecture (FIA) grant CNS-1040735

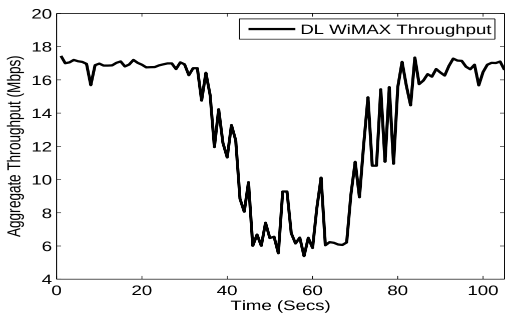

**Figure 1: Fluctuation in downlink throughput performance as seen over a single path traversed by a mobile client associated with the GENI WiMAX basestation.**

the challenges mobile devices bring, with use cases in Section 4. Finally, Section 5 concludes.

# 2. CHALLENGES AND PRINCIPLES

The increase in prominence of the wireless domain [8] has led to many challenges that could be better managed with a clean slate architecture like *MobilityFirst*. Of these, five major challenges include *mobility*, *varying levels of link quality* and *connectivity*, *multiple network attachment points*, and non-traditional types of *routing paradigms*, such as anycast, multicast, and geocast. The current end-to-end approach to Internet communication does not natively address any of these challenges, since wireless devices are not generally considered "first-class" entities. Current Internet protocols have been designed on the assumption that wireless nodes are a "last hop" that are best handled by the very edge of the network. The first step in building the *MobilityFirst* router is understanding the challenges faced by the current Internet architecture, and formulating a set of guiding principles that lead to a resilient router design.

## 2.1 Challenges

We now elaborate on the major routing challenges found in a mobility-centric network.

### C1. Host mobility

Mobile and wireless devices connecting to the Internet have the effect of increasing variance in the physical and topological characteristics of the network. This in turn can result in the performance of the current end-to-end architecture becoming inefficient. Current techniques used to handle mobility, such as MobileIP [9], rely on a "home agent", that the traffic is tunneled through. This is inefficient, as it can force a sub-optimal path towards the destination as well as overload the redirection points. The primary reason MobileIP is necessary with today's architecture is that traffic is bound to specific destination network addresses, since nodes are synonymously *named* by their network address. There has been some initial work from the DTN community on intentional naming, where application data is bound to descriptive names [10]; however, this needs to be taken a further and promoted as a core architectural change.

### C2. Varying Levels of Link Quality

Performance of individual links in wireless networks can see large fluctuations like those seen in Figure 1 from a sample UDP downlink throughput trace of a mobile client associated with a GENI [4] WiMAX basestation. Current Internet protocols, such as TCP, limit the ability to handle mobility-induced fluctuation in connectivity since they are end-to-end in nature. The lack of immediate end-to-end acknowledgements are often wrongfully viewed as congestion, and hence source data rates are unnecessarily throttled. Instead, hop-by-hop transport and the ability to temporarily store data in the network are needed to overcome these issues.

**C3. Varying levels of Connectivity** High mobility can also lead to complete disconnections, where end-to-end protocols such as TCP completely fail as they require a path to be setup before data is sent. A number of solutions have been proposed to overcome this problem, particularly from the *delay-tolerant networking* community. Techniques such as message replication [11, 12] and hop-by-hop transport [13] are utilized to bridge partitions in the network. Unfortunately, there has been no comprehensive solution to bridge varying levels of connectivity, in that DTN protocols are usually not sufficient in highly connected environments and MANET protocols are usually not sufficient in highly disconnected environments. While there has been some work on merging DTN and MANET protocols, they usually consider DTN nodes as specialized entities useful only for extending MANET protocols [14, 15], or consider MANET clusters to be relatively static and simply bridged by DTN nodes [16]. We envision both DTN and MANET capabilities in *all* nodes, allowing them to appropriately choose techniques in a more fluid manner with no reliance on the stability of a local cluster.

### C4. Multi-homing

In addition to high levels of mobility and disconnectivity, mobile devices have multiple radios (e.g., utilizing 3G, WiFi, and BlueTooth) [17], and hence will have multiple network attachment points. There are many advantages to being multi-homed; the device can take advantage of the multiple interfaces to achieve better throughput, increase fault tolerance, and suffer lower latencies during handoff. Currently, devices wishing to utilize multiple interfaces must do so at the application layer, and the device itself must start the connection. Allowing the network to dynamically choose which interface to send to is a more flexible and advantageous approach. One of the fundamental difficulties of this in the current Internet is that application data is bound to IP addresses, and not separate device names. Therefore, the current network cannot independently choose from multiple IP addresses belonging to the same device or entity.

### C5. Context-Aware Routing Paradigms

Context sensitivity of applications necessitates flexible routing paradigms like anycast, multicast, and geocast. Group-based examples include the ability to contact *any* emergency responder in a disaster zone as opposed to a specific one, or the ability to push content to *every* content subscriber. Geocast is also important to help satisfy queries such as "send this message to all taxis in New Brunswick", where an optimal strategy may be to geographically route a copy of the message to New Brunswick, and then distribute copies to all taxis in the area. These types of paradigms are very useful for content-driven applications, which will be promi-

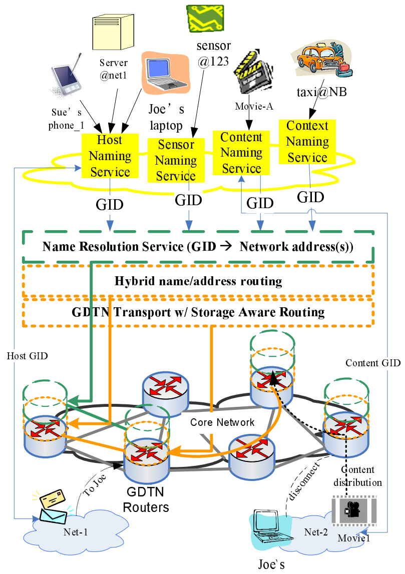

Figure 2: MobilityFirst Protocol Stack

nent in the future Internet. Work on content centric networking (CCN) [18] aims to deliver content efficiently by incorporating content-aware routing techniques at the core of the network. The current Internet is based on traditional point-to-point communication, limiting the ability of applications to specify group-based or context-based destinations. While approaches such as IP Multicast attempt to provide some flexibility, these are inappropriate for quickly changing topologies found in MANET and DTN environments. While not directly discussed in this work, network-layer support for such context-aware services is being built into *MobilityFirst*.

## 2.2 Principles

We now present a set of guiding principles that help address the aforementioned design challenges.

### Separation of Naming and Addressing

As described in the pictorial illustration in Figure 2, our proposed *MobilityFirst* architecture has three levels of identification. At the highest layer, destinations can be specified as human-readable, context strings such as "Joe's laptop", or "taxi in New Brunswick"". These are essentially a set of (name, value) pairs that are handled and resolved at services running on top of the network. Second, destinations can be specified using a long-term, globally unique name (GUID) from a flat naming space that correspond to an entity or context in the network. The *GUID* of an entity does not depend on the network attachment point, and so traffic destined for it does not have to be immediately bound to a specific network. Third, destinations can be specified as a low-level network address, such as an IP address.

Such a structured, hierarchical addressing solution allows the routing design to address host mobility (C1), disconnections (C3), and multi-homing issues (C4) discussed previously in a much better way than that supported by current networks. These multiple layers of identification are built into the *MobilityFirst* architecture, and are heavily utilized in the routing component.

### Late Binding

The separation of naming from addressing opens the door for in-network late binding. In the MobilityFirst architecture, routers along a path will have the option of querying the global name resolution service to bind addresses late. This allows for macro-mobility (C1), link quality fluctuation (C2) and disconnections (C3), multi-homing (C4), and context-aware, group-based routing (C5). As an example, in a case with rapid host mobility, the network can use late or repeated binding to resolve the GUID to a network address at different points along the route, to determine if the destination has changed. In the multi-homed case, the network could resolve independent GUIDs to more than one network addresses thereby providing the routing layer with the flexibility of reaching the host over more than one network. Late binding will also improve delivery efficiency of multicast mechanisms (described in C5) by aggregating traffic whenever possible.

### In-network Storage Utilization

Taking advantage of rapidly falling memory costs, MobiltyFirst utilizes the concept of *storage aware routing (STAR)*, which gives routers the option to temporarily store data as a *network-layer routing decision*. Allowing routers the option to store also implies that transport be done in a hop-by-hop fashion. Essentially, large blocks can be reliably transferred at the link layer, which in turn would be responsible for congestion control, significantly decreasing the role of the end-to-end transport protocol. We envision future routers to have at least three levels of storage: (1) a transit buffer of around 100MB in size, (2) a holding space around 10GB in size that can be utilized by the network for temporarily storing data, and (3) a cache space around 1TB in size that can store long-term content[1].

### Conditional Routing Behavior

Due to the highly local and rapidly changing connection qualities between wireless nodes, it is important that the network have greater intelligence then it currently does. One way to do this is to provide it with a greater array of routing options. To accommodate for the wide range of network connectivity levels (discussed elaborately in the challenges described in $C3$), we propose that the routing layer should be able to make per-hop decisions of either (1) storing the content, (2) forwarding to the next hop, (3) caching and replicating the content for future use, or (4) marking the content for opportunistic delivery in DTN scenarios. Such an approach to routing differs significantly from the conventional approach in the current architecture of always forwarding frames and helps the network cope with issues arising out of varying levels of connectivity.

# 3. MOBILITYFIRST ROUTING LAYER

The previous section illustrates how a set of principles are necessary to address the challenges associated with handling mobility. In this section, we utilize these principles to show how both global and local scale routing can be efficiently done in the *MobilityFirst* network. To achieve a scalable, mobility-centric solution, our system utilizes both *names*

[1] The actual sizes of the memory stores would need to be scaled depending on transit block sizes and the general network load.

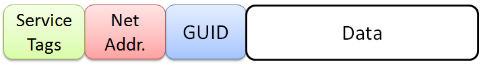

**Figure 3: PDU structure (data size can be large)**

(i.e., GUIDs) and *addresses* cooperatively. Since we envision a purely packet-switching architecture, the header containing name and address information may be larger than current IPv4 packet headers. To help amortize this overhead and take advantage of plentiful storage, we envision the data portion of the protocol data unit at the routing layer to be large. Furthermore, these large PDUs will be reliably transmitted by the network on a hop-by-hop basis.

There are three primary headers for PDUs at the routing layer: (1) GUID information, (2) address information, and (3) service tags. Since application data is bound to GUIDs and not network-level addresses, the destination GUID acts as the most authoritative piece of routing information and must always be present in the PDU. Since the *MobilityFirst* architecture allows dynamic, in-network GUID-to-address queries via the GNRS, a second header storing at least the current destination address is also necessary. Finally, a third header includes service tags, which indicate specific characteristics of the data itself such as whether it is real-time traffic or not. Figure 3 presents a high-level illustration of *MobilityFirst* routing PDUs.

With the previously mentioned principles in mind, we now discuss how GUIDs and network addresses are used in combination to route data through *MobilityFirst* networks. To illustrate the fact that our approach can be implemented as a "clean-slate" architecture or integrated with current IP, IP addresses are used as the low-level network addresses in this section.

## 3.1 Routing on a Global Scale

The MobilityFirst routing design philosophy is to provide enough information and resources to individual routers for them to make intelligent, hop-by-hop decisions. To this end, both high-level GUIDs and low-level IP addresses are exposed to routers for the decision making process. Routers obtain information from two sources: (1) network services and (2) the inter-domain routing protocol. A prominent example of a network service is the aforementioned global name resolution service (GNRS). This service responds to GUID-based queries with a set of IP addresses currently bound to that GUID. In the case that there are no IP mappings (e.g., the node is disconnected from the network), it will return a set of networks and the historical probability of that node attaching to one of those networks in the future. Another example service could include a location service, where the result would be the current geographical location of a GUID. For the time being, we assume BGP is used for the inter-domain routing protocol.

It is important to emphasis, however, that GUID is still the focal point of the MobilityFirst architecture and hence has is the most authoritative piece of routing information in the PDU header. The destination IP address(es) may change in-network via a GNRS re-lookup, but the GUID stays the same. This is consistent with binding application data to GUIDs, not IP addresses.

## 3.2 Routing on a Local Scale

Once the PDU has been delivered to the destination network, local *storage-aware routing* techniques can be used to deliver to the final host.

### Route selection and routing decisions

MobilityFirst uses a two pronged approach for intra-domain routing that is capable of quickly responding to link quality changes for nearby nodes as well as remaining robust in the face of disconnectivity and partitioning. At a high level, individual routers maintain two types of topology information, one useful for responding to fine-grained changes to links and nodes *within* the router's current partition, and one useful for responding to course-grain changes to connection probability for all nodes in the network.

The **intra-partition** graph is formed by collecting topology messages that are periodically flooded by all nodes in the network. These topology messages contain *time sensitive* information about the link quality for each of the node's 1-hop neighbors. These are transmitted per interface, allowing for in-network multi-homing. Since the messages are flooded, and hence immediately broadcasted and dropped, they will not traverse across partition boundaries. This allows all nodes in the network to have an up-to-date view of the current link qualities within its current partition. In addition to storing current link qualities, all routers maintain a history of link quality information received in the past, which, as will be shown, is useful for routing decisions. If control messages have not been received from a particular node for some period of time, and hence its long term link qualities have become low, a router may assume that node has left the partition and remove it from the graph.

The **DTN** graph is formed by collecting topology messages that are periodically epidemically disseminated by all nodes in the network. Epidemic dissemination, where control messages are carried by intermediate nodes, is a common technique used in delay-tolerant networking, and allows messages to cross partition boundaries [11, 19]. In essence, these messages are not immediately dropped, but rather carried for a long period of time such that if a node moves from one partition to another, it can ferry messages between the two. These topology messages contain *time insensitive* information about connection probabilities between the source node and all other nodes in the network. This graph allows a node in the network to be aware of the general connectivity patterns of all nodes, even those outside of its current partition.

These two graphs can then be used together to help route messages to their destinations. For a given message, a router first checks its intra-partition graph for the destination. If the destination exists, then the router will then choose the best path from multiple ones by considering the short term link qualities for nearby hops and the long term link qualities for hops further away. Given that path, if the short term link quality for the next hop is much greater than the long term link quality for the next hop, then the router should immediately forward the data to take advantage of the abnormally good link. On the other hand, if the short term link quality is abnormally bad, then the router should store the message and re-evaluate later. A pictorial example of

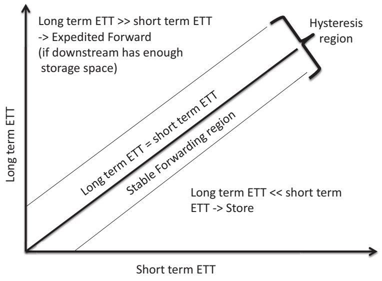

**Figure 4: Intra-Partition Routing Decision**

the routing decision graph, using the estimated transmission time (ETT) [20] as the routing metric, can be seen in Figure 4. We have previously explored this technique with the CNF architecture [21, 22] and are continuing to incorporate additional factors into the routing decision such as message size and available storage.

If the destination is not found in the intra-partition graph, the router tries to make progress along the DTN graph, which contains a general overview of connectivity throughout the network. The router will compute all shortest paths according to that graph, and attempt to forward a replica of the message to all 1-hop neighbors along those paths. In essence, this DTN-style approach attempts to make use of readily available storage to bridge partitions in the network. An example is given in Figure 5, where node S forwards a replica of a message destined for node D to both nodes A and B. Note that this message is not copied to node C, as this would not progress the message. It is important to note that this framework is flexible enough to incorporate many existing DTN routing protocols. For instance, the protocol-specific metric being used to capture connection probability can be used as the MobilityFirst DTN "link-quality" metric that is epidemically disseminated to form the DTN graph. One example is to utilize the *average availability* metric, described in PREP [19], to capture the historical percentage of time two nodes were connected. Furthermore, replication rates can be determined by the DTN protocol in question.

**Buffer prioritization and scheduling**
Since we are anticipating messages to be relatively large, conditions such as link quality and even overall connectivity can change on a per-message basis. Therefore, the message transmission order can greatly impact routing metrics such as average end-to-end delay and throughput.

The general approach taken by the intra-domain routing component is to establish a primary ordering based on information from the intra-partition and DTN graphs. Since the intra-partition graph has time-sensitive weights, messages whose destination is found in the intra-partition graph are given higher priority, in hopes of utilizing good, currently available links. We refer to this as *priority 1*. Blocks whose destination is only found in the DTN graph are given lower priority, or *priority 2*. Finally, all remaining messages are given the lowest priority, or *priority 3* since they will be either stored or dropped.

Priority 1 messages can be further prioritized based on a utility function $u$ that captures how beneficial it would be to transmit the message over the next-hop link in that link's *current form*. We define $u$, for a given destination $d$ as

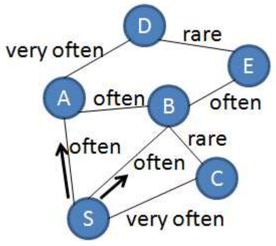

**Figure 5: DTN-graph Routing**

$$u(d) = \frac{stlq(nextHop)}{ltlq(nextHop)}$$

where $stlq(nextHop)$ and $ltlq(nextHop)$ are the short term and long term link qualities (higher is better) of the next hop to reach the destination $d$. Note that the next hop comes from the routing computation over the intra-partition graph, utilizing both short and long term link qualities, as previously described. Intuitively, $u$ captures how much better the current next-hop link quality is compared to its historical average. Values greater than 1 indicate better than average qualities, and values less than 1 indicate worse than average qualities. Priority 1 messages with higher values of $u$ are given higher priority than priority 1 messages with lower values of $u$.

Intra-prioritization of priority 2 messages can be very flexible. In the event that an existing DTN routing protocol is being utilized for the DTN graph computation, it may be useful for the buffer prioritization method used by that protocol to be used for ordering the priority 2 messages. One simple prioritization method can be to assign higher priorities to messages whose total path to the destination, along the DTN graph, is of low weight.

## 4. USE CASES

To illustrate how these two pieces of information are utilized by the routing protocol, consider the topology in Figure 6. We now present three scenarios using this topology:

**Multi-homing Example**: Assume S has a message destined for D. S first transmits the message to its default border router in AS1. This router then queries the GNRS services, and obtains two IP addresses, IP1 and IP2, indicating that the host D is multi-homed. Consulting its BGP table, it realizes that AS3 is common along the two paths: IP1 – AS2:AS3:AS4 and IP2 – AS2:AS3:AS5. It therefore *postpones* the decision to choose which IP address the destination will be by tunneling to AS3. Practically, this can be accomplished by indicating to a router in AS3, via the message header, to perform a GNRS re-resolution on the GUID. AS3, following the instructions, re-queries the GNRS, obtaining up-to-data link quality metrics associated with the wireless attachments for IP1 and IP2. Using this, it decides to send to IP1, and hence transmits the data through AS4. At this point, the local protocol utilizing CNF and DTN techniques will progress the data towards the destination.

**Disconnection Example**: Assume a similar case as before, except when AS1 makes the query to the GNRS, the

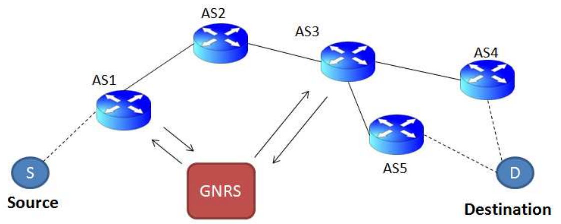

**Figure 6: GUID+IP Routing Use Case**

GNRS indicates that D is disconnected and returns a set of networks (e.g., ASes) that D frequently connects to, which includes AS4 and AS5. AS1 decides again to tunnel through AS3. AS3, obtaining the same information from the GNRS, *replicates* the message to both AS4 and AS5 using D's GUID as the only destination. Now, if D connects to either AS4 or AS5, it will immediately receive a copy of the data via the local routing protocol.

**Content Multicasting Example**: Assume S generates a context-rich multicast message (e.g., "push this software update") destined for nodes in both AS4 and AS5. Furthermore, assume a service (represented by the GNRS in this example) maps context to GUID and, in turn, to a set of IP addresses. A similar scenario as before occurs, where the message would be unicasted to AS3 and, realizing a split is necessary, AS3 would send copies of the message to AS4 and AS5. At this point, the local routing protocols can use their knowledge of the network graph to multicast the message.

In addition to these qualitative use case examples, we have implemented and experimented with components of the routing architecture, namely the local routing techniques. These components routinely obtain significantly better performance than currently Internet protocols such as TCP and OLSR. Due to space constraints, we omit these intermediate simulation and prototyping results; however, we plan to continue to comprehensively evaluate our architecture as a whole via ns2 simulation and prototyping on ORBIT and GENI testbeds, and will present our results in future work.

# 5. CONCLUSIONS AND FUTURE WORK

This paper presented a high-level overview of the routing component for *MobilityFirst*. In particular, four general concepts were proposed to efficiently handle the numerous challenges brought about by mobile devices: separation of names from addresses, late binding of routable addresses, in-network storage, and conditional routing decision space. On a global scale, routers utilize information obtained from dynamically querying the global name service as well as querying low-level routing tables to move data closer to its destination. The use of network services allows challenges such as mobility, disconnection, and multi-homing to be handled in the network itself. At the local level, routers utilize both storage-aware routing and DTN techniques to forward data in an efficient and timely fashion, even across partition boundaries.

The proposed design includes several components, including a storage-aware intra-domain routing protocol that seamlessly meshes with DTN techniques, as well as a large-scale GUID and IP hybrid protocol. We are currently refining and evaluated individual components through simulation and the ORBIT testbed. This evaluation includes tens of wireless nodes, many mobile, with each node having access to WiFi, WiMax, and/or a wired connection to the broader Internet. Once individual components are tested, we plan to bring the system together as a whole and test it with real-world applications and end-users using the GENI testbed [4].

# 6. REFERENCES

[1] S. Paul, R. Yates, D. Raychaudhuri, and J. Kurose. The cache-and-forward network architecture for efficient mobile content delivery services in the future internet. *ITU T Kaleidoscope: Innovations in NGN*, May 2008.

[2] NSF FIA project. `http://www.nets-fia.net`.

[3] Networking technology and systems: Future internet design (FIND), NSF program solicitation, 2007.

[4] Global environment for networking innovations (GENI), NSF program solicitation. `http://www.geni.net`, 2006.

[5] FP7 information and communication technologies: Pervasive and trusted network and service infrastructures, european commission, 2007.

[6] M. Lemke. Position statement: The european FIRE initiative, NSF/OECD workshop on social and economic factors shaping the future of the internet, January 2007.

[7] New generation networks. `http://www2.nict.go.jp/w/w100/index-e`.

[8] Cisco visual networking index: Forecast and methodology, 2009-2014. `http://tinyurl.com/3p7v28`, 2010.

[9] Ip mobility support for ipv4. `http://tools.ietf.org/html/rfc3344`.

[10] M. Chuah, S. Roy, and I. Stoev. Secure descriptive message dissemination in dtns. In *Proceedings of MobiOpp*, 2010.

[11] J. Burgess, Br. Gallagher, D. Jensen, and B. Levine. MaxProp: Routing for vehicle-based disruption-tolerant networks. In *Proc. IEEE INFOCOM*, 2006.

[12] A. Balasubramanian, B. Levine, and A. Venkataramani. DTN routing as a resource allocation problem. In *Proc. ACM SIGCOMM*, 2007.

[13] Ming Li, Devesh Agrawal, Deepak Ganesan, and Arun Venkataramani. Block-switched networks: a new paradigm for wireless transport. In *Proceedings of NSDI*, 2009.

[14] Jörg Ott, Dirk Kutscher, and Christoph Dwertmann. Integrating DTN and MANET routing. In *Proceedings of ACM CHANTS workshop*, 2006.

[15] G. F. Aggradi, F. Esposito, and I. Matta. Supporting predicate routing in dtn over manet. In *Proceedings of ACM CHANTS*, 2008.

[16] J. Whitbeck and V. Conan. HYMAD: Hybrid DTN-MANET routing for dense and highly dynamic wireless networks. *Comput. Commun.*, 33, August 2010.

[17] M. R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely. Energy-delay tradeoffs in smartphone applications. In *Proceedings of MobiSys*, 2010.

[18] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proceedings of ACM CoNEXT*, 2009.

[19] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan. Prioritized epidemic routing for opportunistic networks. In *Proceedings of MobiOpp*, 2007.

[20] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of ACM MobiCom*, 2004.

[21] S. Gopinath, S. Jain, S. Makharia, and D. Raychaudhuri. An experimental study of the cache-and-forward network architecture in multi-hop wireless scenarios. In *Local and Metropolitan Area Networks (LANMAN)*, 2010.

[22] G. Bhanage, A. Chanda, J. Li, and D. Raychaudhuri. Storage aware routing protocol for the mobilityfirst network architecture. In *European Wireless, DTN Session*, 2011.