MobilityFirst: Architecture Summary & Project Status
EAB Meeting - 30 April 2012

Contact: D. Raychaudhuri
WINLAB, Rutgers University
Technology Centre of NJ
671 Route 1, North Brunswick,
NJ 08902, USA
ray@winlab.rutgers.edu
MobilityFirst Summary: Architecture Features

- **Named devices, content, and context**
  - Human-readable name

- **Strong authentication, privacy**
  - 11001101011100100…0011
  - Public Key Based Global Identifier (GUID)

- **Heterogeneous Wireless Access**
- **Routers with Integrated Storage & Computing**
- **In-network content cache**

- **End-Point mobility with multi-homing**

- **Storage-aware Intra-domain routing**

- **Network Mobility & Disconnected Mode**

- **Service API with unicast, multi-homing, mcast, anycast, content query, etc.**

- **Connectionless Packet Switched Network with hybrid name/address routing**

- **Edge-aware Inter-domain routing**

- **Hop-by-hop file transport**

- **Ad-hoc p2p mode**
MobilityFirst Summary: Protocol Stack

**Control Plane**
- GNRS
- MF Routing Control Protocol

**Data Plane**
- Link Layer 1 (802.11)
- Link Layer 2 (LTE)
- Link Layer 3 (Ethernet)
- Link Layer 4 (SONET)
- Link Layer 5 (etc.)

**Optional Compute Layer Plug-In A**

- App 1
- App 2
- App 3
- App 4

**Socket API**

- E2E TP1
- E2E TP2
- E2E TP3
- E2E TP4

**GUID Service Layer**

- GSTAR Routing
- MF Inter-Domain

**IP**

- Narrow Waist
MobilityFirst Summary: How MF Works -

(1) At the Device End-Points

Service API capabilities:
- `send (GUID, options, data)`
  Options = anycast, mcast, time, ...
- `get (content_GUID, options)`
  Options = nearest, all, ...

GUID lookup from directory

Send (GUID = 11011..011, SID=01, data)

Send (GUID = 11011..011, SID=01, NA99, NA32, data)

Packet sent out by host

GUID <-> NA lookup

GUID = 11011..011

Represents network object with 2 devices

GUID assigned

Register “John Smith22’s devices” with NCS

Name Certification Services (NCS)

GUID lookup from directory

Network object with 2 devices

DATA

Packet sent out by host

MobilityFirst Network (Data Plane)

GNRS

NA99

NA32

GNRS update (after link-layer association)
MobilityFirst Summary: How MF Works - (2) At Router, AP or BS

Example of Functions at Branching Router for a Multicast Packet to be delivered to NA99 and NA32

**GUID–based forwarding**
(slow path)

<table>
<thead>
<tr>
<th>GUID</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001..11</td>
<td>NA99,32</td>
</tr>
</tbody>
</table>

Look up GUID-NA table when:
- no NAs in pkt header
- encapsulated GUID
- delivery failure or expired NA entry

**NA Routing Table** – stored physically at router

<table>
<thead>
<tr>
<th>Dest NA</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA99</td>
<td>NA11</td>
</tr>
<tr>
<td>NA62</td>
<td>NA11</td>
</tr>
<tr>
<td>NA32</td>
<td>NA51</td>
</tr>
</tbody>
</table>

Look up NA-next hop table when:
- pkt header includes NAs
- valid NA to next hop entry

**Network Address Based Forwarding**
(fast path)

Store when:
- Poor short-term path quality
- Delivery failure, no NA entry
- GNRS query failure
- Content cache decision
- etc.

**Router Storage**

Store when:
- Poor short-term path quality
- Delivery failure, no NA entry
- GNRS query failure
- Content cache decision
- etc.
MobilityFirst Summary: How MF Works - (3) End-to-End: Multicast Example

Multicast service example

Send data file to “John Smith22’s devices”, SID= 21 (mcast)
MobilityFirst Summary: How MF Works - (4) End-to-End: Dual Homing Example

Multihoming service example

Send data file to “John Smith22’s laptop”, SID= 129 (multihoming – all interfaces)
MobilityFirst Summary: How MF Works - (5) End-to-End: Disconnection Example

Store-and-forward mobility service example

Delivery failure at NA99 due to device mobility
Router stores & periodically checks GNRS binding
Deliver to new network NA75 when GNRS updates

Send data file to “John Smith22’s laptop”, SID= 11 (unicast, mobile delivery)
MobilityFirst Summary: How MF Works –
(6) End-to-End: Enhanced Service Example

Enhanced service example – content delivery with in-network storage
MobilityFirst Summary: The Technology Solution

**Global Name Resolution Service (GNRS)**

- Hybrid GUID/NA Global Routing (Edge-aware, mobile, Late binding, etc.)
- Storage-Aware & DTN Routing (GSTAR) in Edge Networks

**Name-Based Services** (mobility, mcast, content, context, M2M)

- Optional Compute Layer Plug-Ins (cache, privacy, etc.)

**Optional Compute Layer**

- Hop-by-Hop Transport (w/bypass option)

**Flexible name-based network service layer**

- Name Certification Service (NCS)

**Meta-level Network Services**

**Core Transport Services**

*Pure connectionless packet switching with in-network storage*
Protocol Design: Direct Hash GNRS

- Fast GNRS implementation based on DHT between routers
  - GNRS entries (GUID <-> NA) stored at Router Addr = hash(GUID)
  - Results in distributed in-network directory with fast access (~100 ms)

### Protocol Design: Direct Hash GNRS

**Fast GNRS implementation based on DHT between routers**

- GNRS entries (GUID <-> NA) stored at Router Addr = hash(GUID)
- Results in distributed in-network directory with fast access (~100 ms)

### Global Prefix Table

<table>
<thead>
<tr>
<th>Prefix</th>
<th>AS #</th>
<th>Next-hop address</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/8</td>
<td>1</td>
<td>8.8.8.8</td>
</tr>
<tr>
<td>67.10/16</td>
<td>55</td>
<td>67.10.1.1</td>
</tr>
<tr>
<td>44/8</td>
<td>101</td>
<td>43.32.1.1</td>
</tr>
</tbody>
</table>

### Internet Scale Simulation Results

Using DIMES database

![Diagram showing Internet Scale Simulation Results Using DIMES database](image-url)
Protocol Design: Storage-Aware Routing (GSTAR)

- Storage aware (CNF, generalized DTN) routing exploits in-network storage to deal with varying link quality and disconnection.
- Routing algorithm adapts seamlessly from switching (good path) to store-and-forward (poor link BW/short disconnection) to DTN (longer disconnections).
- Storage has benefits for wired networks as well.

---

**Diagram:**
- Initial Routing Path
- Re-routed path
- Storage Router
- Low BW cellular link
- High BW WiFi link
- Mobile Device trajectory
- Temporary Storage at Router
- Long-Term Path Quality (LTQ)
- Short-Term Path Quality (STQ)
- Region with STQ >> LTQ
- Opportunistic forwarding
- Sample CNF routing result
- PDU

---

**Sample CNF routing result**
- FTP over OLSR
- CNF Protocol stack
- CNF TP over OLSR
- Number of bytes vs. File transfer delay
- File Transfer Delay vs. Number of bytes
Protocol Design: MobilityFirst Interdomain Routing

- Approach under consideration is to enhance BGP-like protocols with summary node/link info (“Vnode graph”)
  - Summary knowledge of access net properties (Mbps, % avail, etc.), ingress/egress points and alternate paths exchanged between networks/AS’s
  - Network topology information for identifying multiple paths, storage points, etc.
- Inspired by “Vnode” concept in “Pathlet” routing (Godfrey, 2008)
- Support for multicast, anycast, multihoming and multipath
Protocol Design: **Content Delivery in MobilityFirst**

- Content delivery handled efficiently by proposed MF architecture
  - “Content objects” identified by unique GUID
  - Multiple instances of content file identified by GNRS via GUID to NA mapping
  - Routing protocol used for “reverse anycast” to nearest content object

- Approach differs from NDN/CCN, where content attributes are carried in packet headers

- MF uses content GUID naming service & GNRS to keep things general and avoid interpreting content semantics inside network

---

**Diagram Description**

- **Content Store #1** (owner)
  - GUID=13247..99

- **Content Store #2**
  - GUID=13247..99

- **User mobility**

- **Content cache at mobile Operator’s network – NA99**
  - GUID=13247..99

- **GNRS query**
  - Returns list: NA99,31,22,43

- **Data fetch from NA99**
  - Get (content_GUID)

- **Data fetch from NA43**
  - Get (content_GUID)

- **Data fetch from NA29**

- **NA22**
  - Content Store #1

- **NA31**
  - GUID=13247..99

- **NA43**
  - GUID=13247..99

- **NA99**
  - Content cache at mobile Operator’s network – NA99

---

*WINLAB*
Protocol Design: Context Aware Delivery

- Context-aware network services supported by MF architecture
  - Dynamic mapping of structured context or content label by global name service
  - Context attributes include location, time, person/group, network state
  - Context naming service provides multicast GUID – mapped to NA by GNRS
  - Similar to mechanism used to handle named content

Context = geo-coordinates & first_responder

Send (context, data)
Project Status: Yearly Goals/Outcomes

- **Year 1** - *architecture white paper, protocol specs, component-level prototypes* (NRS, GDTN routing, Hop TP, PKI security, context services, computing layer, etc.) and early *lab demo* of the network

- **Year 2** – detailed validations of key components, network evaluation results, and a *multi-site proof-of-concept network prototype*

- **Year 3** - updated protocol design based on evaluation feedback, and medium-scale service deployment using GENI infrastructure.

- The project will conclude with a *comprehensive validation and evaluation of usability and performance* using both controlled experiments and application trials with *real-world end-users.*
Project Status: Year 2 goals as reported at Feb 2011 IAB

Next steps for project:

- Continue architecture convergence process across entire team & develop a baseline MF protocol spec 1.0
- Validate global name resolution service (GNRS) at scale
- Design & prototype hybrid GUID/address routing network
- Design & prototype storage/DTN access routing algorithms
- Simulation and emulation studies of hybrid routing performance
- Security/privacy baseline & initial design for key protocol components
- Design & prototype context support service using GUID framework
- Design & prototype basic set of network management services
- Demo integrated GNRS + storage routing in lab prototype (~7/11)
- Demo GNRS + storage routing + context services in outdoor ORBIT/GENI testbed (~9/11)
- Demo basic MobilityFirst protocol stack with example context services in multi-site GENI demo (~11/11)
Project Status: Progress Highlights as of 4/12 (1)

- Group consensus on MobilityFirst protocol architecture
- Baseline MF protocol design completed and spec doc 0.9 posted – complete ver 1.0 spec release 6/12
- Key protocol components going through design, evaluation and redesign process
  - GUID service layer with support for mcast, mhoming, mpath, ..
  - Global name resolution service (GNRS)
  - Storage-aware intra-domain routing (GSTAR)
  - Edge-aware inter-domain routing with hybrid GUID/NA, late binding..
  - Content and context/M2M services
  - Compute layer for enhanced services
- Spiral development of proof-of-concept MF prototype
  - MF router framework using Click platform; Android mobile protocol stack
  - GNRS and GSTAR protocols
  - Content and context service implementations
  - ORBIT and GENI experiments; first MF demo at Nov 2011 GEC
Project Status: Progress Highlights as of 4/12 (2)

- Initiated project on MF integration with optical networks/OpenFlow
- Ongoing research and design work on security/privacy
  - Core security architecture and protocols
  - Privacy considerations and design options
  - DDOS resistance and robustness
- Economic models and policy analysis
  - Cellular-internet convergence scenarios
  - Industry structure, privacy/censorship issues, network neutrality, etc.
  - Participation in recent FIA meeting (April 19,20) on business models
Project Status: Project goals for 2012

Next steps for project:

- Present MF arch to broader networking community; papers, workshops, talks, demos; software source release in 2012
- Further interaction with policy, legal and economics communities
- Complete architectural work, filling in details of routing algorithms, GNRS tuning, security protocols, name certification service, etc.
- Complete implementation and start at-scale validation of key protocol components
  - GNRS, NCS
  - Hybrid GUID/NA inter-domain routing; integrate with intra-domain
  - Content & context/M2M services
  - Computing layer services
- GENI prototype initially for validation & demos; later phased release of services to opt-in campus users (Android phone, etc.)
- Application of MF components (e.g. GNRS & mobility/content services) to existing systems such as cellular/WiFi access networks or to replace overlay services for M2M, V2V, etc.
- Start project on MobilityFirst router performance aspects – software optimization, hardware acceleration, OpenFlow/optical etc.
Resources

- **Project website**: [http://mobilityfirst.winlab.rutgers.edu](http://mobilityfirst.winlab.rutgers.edu)
- **GENI website**: [www.geni.net](http://www.geni.net)
- **ORBIT website**: [www.orbit-lab.org](http://www.orbit-lab.org)
MobilityFirst: Naming and Routing Architecture

Arun Venkataramani
University of Massachusetts Amherst
arun@cs.umass.edu

Rutgers EAB Meeting
Apr 30, 2012
GNRS: Scalable name resolution service

- Network location: GUID $\rightarrow$ [NA$_1$, NA$_2$, ... ]

- Routing policies:
  - Incoming route preferences: [GUID, NA$_i$] $\rightarrow$ ipref$_i$
  - Outgoing route preferences: [GUID, NA$_i$] $\rightarrow$ opref$_i$
  - Indirection: GUID $\rightarrow$ GUID1

- Access control policies:
  - [GUID, “NA”] $\rightarrow$ blacklist | whitelist

- Temporal policies
  - Preferences and access control policies may be specified as time-dependent functions
GNRS: Service replica placement engine

- GNRS performance metrics
  - Read/write latency
  - Hotspot mitigation
  - Update bandwidth subject to consistency constraints

- Demand-locality-aware placement engine potentially extensible to other cloud services
  - Client VMs in cloud hosting service providers
  - Object update service
Hierarchical interdomain routing

- Interdomain routing consists of two-level hierarchy
  - Core or globally visible networks
  - Terminal networks obtain service from core networks
  - GUID → [NA₁, NA₂, ...] where NA₁ → [Xᵢ, Tᵢ], a core and terminal network pair
  - Terminal networks analogous to prefixes but devoid of structure

- Hierarchy enables scalability and isolation at the cost of increased packet header size
Routing affinity groups

- GUIDs can be aggregated via indirection to affinity GUIDs or AGUIDs
- AGUIDs reduce GNRS update overhead by identifying a set of GUIDs
- AGUIDs can cleanly support
  - Multicast: set of destination GUIDs
  - Content directories: set of content GUIDs for retrieval
  - Group GUIDs that capture correlated device mobility
- Can potentially reduce GNRS update overhead by the size of the affinity group
Global Name Resolution Services Through Direct Mapping (DMAP)

Presented by: Yanyong Zhang
WINLAB, Rutgers University
Technology Centre of NJ
671 Route 1, North Brunswick,
NJ 08902, USA
yyzhang@winlab.rutgers.edu
Name-Address Separation Through GNRS

- Globally unique name (GUID) for network attached objects
  - device, content, context, AS name, sensor, and so on
  - Multiple domain-specific naming services
- Global Name Resolution Service for GUID → NA mappings
- Where to store these mappings?
GNRS Design Objective

- A distributed service hosted on ~10,000 gateway routers
  - Fast updates ~50-100 ms to support dynamic mobility
  - Service can scale to ~10B names
Direct Mapping (DMAP)

- **(+)**
  - Strictly 1-overlay-hop lookup
  - No extra routing requirement (e.g. utilize current BGP)

- **(-)**
  - IP “hole” issues
  - Limited locality
Fixing IP Holes for IPv4

- Fixing IP Holes:
  - If hash of GUID falls in the IP hole, rehash that IP \( m \) times to get out of the hole
  - Lookup follows the same process to find GUID

Map of IP (/8) address space (white = unassigned addresses)

Value at \( m=10 \) is 0.0009
Fixing IP Holes for General Network Addressing Schemes

- In a general network addressing scheme, we can have more holes than used segments (e.g., IPv6)
- Used address segments are hashed into N buckets
  - a two-level index: (bucket ID: segment ID)
- Mapping GUID to NA
  - $H_1(GUID) \rightarrow$ bucket ID
  - $H_2(GUID) \rightarrow$ segment ID within a bucket
Mapping Replication

- Every mapping is replicated at K random locations
- Lookups can choose closest among K mappings. Much reduced lookup latencies
Capturing Locality

- Spatial locality: GUIDs will be more often accessed by local nodes (within the same AS)
- Solution: Keep a local replica of the mapping
  - A lookup can involve simultaneous local lookup and global lookup
  - LNRS and GNRS
Inconsistent GNRS Entries

1. GUID Publishing

2. GNRS lookup

3. GNRS Reply: H

<table>
<thead>
<tr>
<th>GUID</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>H</td>
</tr>
</tbody>
</table>

H'
Inconsistent GNRS Entries

1. GUID Publishing
2. GNRS lookup
3. GNRS Reply: H
4. connect
5. miss
6. Keep checking GNRS until H’
Inconsistent GNRS Entries

1. GUID Publishing

2. GNRS lookup

3. GNRS Reply: H

4. connect

5. miss

6. Keep checking GNRS until H’

7. connect

Reply: H’

GUID NA

C H’
Simulation Results – Query Latencies

![Graph showing query latencies for different values of K]

- For K=5, the 95th percentile is at 86 ms.
- For K=1, the 95th percentile is at 173 ms.

The graph plots the fraction of queries against the round-trip query response time (ms) on a log scale.
Simulation Results – Load Distribution
Open Issues (From Oct 2011)

- GNRS performance with tomorrow’s Internet model
- Sophisticated locality considerations
- Caching GNRS mappings
- Security and privacy

- Large scale prototyping and validation
Tomorrow’s Internet

- A Jellyfish model
  - captures each AS’s distance to the core

- Tomorrow’s Internet
  - More and larger ASs
  - More direct paths between ASs and the core
Caching GNRS Entries

<table>
<thead>
<tr>
<th>GUID</th>
<th>AS#</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

AS 1

<table>
<thead>
<tr>
<th>GUID</th>
<th>AS#</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

AS 101

<table>
<thead>
<tr>
<th>GUID</th>
<th>AS#</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

AS 5

<table>
<thead>
<tr>
<th>GUID</th>
<th>AS#</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

AS 200

<table>
<thead>
<tr>
<th>GUID</th>
<th>AS#</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>
Large Scale Evaluation

- Emulation of GNRS on the Orbit Testbed
  - In memory Berkeley DB on each node
  - Topology according to the Jellyfish model
  - Each Orbit node representing multiple ASs
- PlanetLab or GENI
Edge-aware Inter-domain Routing

EAB meeting, 30 April 2012
Tam Vu
WINLAB, Rutgers University
Inter-domain Routing Goals

* Provide network level support for:
  * Multi-homing, multi-path, multi-network operations
  * Disconnection and edge-awareness routing
  * Flexible network formation at edge & core
  * Mobility of devices and networks

Expose more internal states of ASs
High level mechanisms

- Exposing more AS internal information by abstracting network entities to be aggregation nodes (aNode) and connectivity to be virtual link (vLink)
  - Distribute NSP (Network state packets) with information about internal network state
- Telescoping update for scalability
  - Controlling the NSP update rate as a function of distance-to-originating network
- Late name-to-address binding
  - Router has capability of rebinding GUID->address for packet in transit
Example of dual-homing route

Global Naming Resolution Service (GNRS) (E.g., L2 => {NA1:aNode32, NA2:aNode89, ...})

Edge-aware enabled routing example
...are abstract entities inside an AS that is visible to external network to aggregately expose internal information.

- Two types:
  - Transit and Stub \textit{aNodes}
- Propagate \textit{vLink}
  - Availability
  - Absolute Bandwidth (relative value)
  - Bandwidth variation
Control message propagation

- aNode broadcasts **aNode State Packet (aSP)**
  - *its internal properties*
  - *link states to its neighboring aNodes*

- AS updates their neighbor ASs with **Network State Packet (NSP)** - *in telescopic manner*
  - *internal topology of the AS and*
  - *point of attachment of that AS to other ASs.*

- Telescoping state packets: Frequency of link state update depends on:
  - *Hop-count-from-original-announcer*
  - *Route export policies (local policies and BGP-like policies)*
**State packet structure**

* aSP

<table>
<thead>
<tr>
<th>Msg Type</th>
<th>Source_aNode_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Last hop link (Edge) properties - <B,A,V>

Neighboring info:

- Neighboring_aNode_#1, link properties
- Neighboring_aNode_#2, link properties
- ...........

* nSP

<table>
<thead>
<tr>
<th>Msg Type</th>
<th>AS_Num:Source_aNode_ID</th>
<th>Hop-To-Src</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Internal topologies:

- aNode_#1 – link <B,V,A> - aNode_#2
- aNode_#2 – link <B,V,A> - aNode_#3
- ...

Neighboring info:

- Neighboring_aNode_#1, link properties
- Neighboring_aNode_#2, link properties
- ...........
**Data Plane**

- **GNRS Client**: Obtains list of NA#:aNode# if possible
- **Forwarding Table**: Decides who to send to next based on service ID and other factors
Example of dual-homing route

Global Naming Resolution Service (GNRS) (E.g., L2=> {NA1:aNode32,NA2:aNode89,....})

Edge-aware enabled routing example
What to evaluate

- Dissemination information overhead
  - On orbit and simulation

- Routing table size growth
  - Number of aNode & vPath

- Latency in different use cases
  - Multicast, Multi-Path, Multi-homing, Multi-network

- Qualitative evaluation on
  - Route policy expressiveness
  - Asymptotic dissemination overhead
Questions and Discussion
Safari: ScAlable, Flexible, and Accountable Routing Infrastructure

MobilityFirst Project EAB Meeting
April 30, 2012
Z. Morley Mao, M. Reiter
Issues of the Current Internet Infrastructure

• Scalability: The routing table growth is one of the most critical problems.
• Flexibility: BGP performs single-path selection
• Security: BGP is vulnerable to a variety of routing attacks
• Mobility: Tight coupling exists between addressing and network location in the current Internet
Safari’s Goal

Scalability: Safari’s routing table size should be orders of magnitude smaller than the existing BGP’s routing table.

Flexibility: Safari should have multiple routing entries for a destination and provide the flexibility to apply routing policies.

Security: In Safari, every network device can verify the identity of others to prevent security issues such as source spoofing, routing manipulation, and denial of service.

Mobility: First, Safari should clearly separate the responsibilities of routing, addressing, and identification. Second, networks can move as well, e.g., airplane, subway, cruise, etc. The network-level handover should be efficient and reliable.
Safari’s Approach

• Addressing to handle mobility and security
  – provider-rooted hierarchical address
  – self-certifying address

• Source and destination based routing to enhance flexibility and reliability

• Bloom-filter-based routing table to handle scalability and flexibility
Addressing

• Safari requires external naming resolution services
  – naming resolution service (NRS)
  – addressing resolution service (ARS)
**Source- / Destination- Based Routing**

1. The source device sends a request to NARS for the destination device’s address
2. NARS negotiates with the destination device on behalf of the source device
3. The destination device refers NARS to one of its provider based on the destination device’s local routing policy
4. NARS keeps querying until a referred provider does not differentiate which of its upstream provider traffic goes through, or until the core if the source device enforces NARS to get the destination’s full locator

We expect their routing policies should be fairly stable for most network domains. Thus, NARS can perform caching to reduce the querying overhead in steps 2-4 significantly.

5. NARS responds the source devices with the locator authorized by the destination device and corresponding providers
Routing Table

- Peer relationships are propagated, bloom filters are separated accordingly
- Safari stores routing entries in isolated bloom filters
- One bloom-filter entry is a combination of both the domain and the distance
Routing Table (cont.)

The first row represents the destination’s relationship with the neighbor. The first column represents the neighbor’s relationship with the current network.

<table>
<thead>
<tr>
<th>Customer</th>
<th>Provider</th>
<th>Peer</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer</td>
<td>unknown</td>
<td>peer</td>
<td>unknown</td>
</tr>
<tr>
<td>unknown</td>
<td>provider</td>
<td>unknown</td>
<td>unknown</td>
</tr>
<tr>
<td>peer</td>
<td>unknown</td>
<td>unknown</td>
<td>unknown</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N1</th>
<th>N2</th>
</tr>
</thead>
<tbody>
<tr>
<td>provider: T2: (T2,1), ...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>... customer: N4: (N6,2), ...</td>
</tr>
<tr>
<td></td>
<td>... Peer: N2: (N6,2), ...</td>
</tr>
<tr>
<td></td>
<td>... unknown: N5: (N4,3), ...</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>provider: T3: (T3,1), ...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>... customer: N4: (N7,3), ...</td>
</tr>
<tr>
<td></td>
<td>... Peer: N1: (N5,2), ...</td>
</tr>
<tr>
<td></td>
<td>... unknown: N4: (N3,3), ...</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
GNRS Based Content Discovery and Delivery

Yangyong Zhang
WINLAB, Rutgers University
Technology Centre of NJ
671 Route 1, North Brunswick,
NJ 08902, USA
Motivation

- Future Internet: More mobile devices than fixed hosts. Mobile data dominates.

- Majority of Internet usage is about data retrieval and service access. We care about content and is oblivious to location.

- End-to-end TCP/IP stack doesn’t fare well with wireless network.

- Question: how to deal with the new challenge of mobile content delivery?
Objective

- **Content persistence**
  - Content name remains valid as long as possible

- **Content availability**
  - Reliability
  - low-latency

- **Content security**
  - To ensure appropriate content source
  - To ensure content is complete and not be tampered with
  - To ensure confidential transmission
Overall Design Elements

- **Flat content name space**: human readable name → flat globally unique unique identifiers (GUID)

- **GUID to address separation**:  
  - Content GUID -→ Content Location  
  - GNRS: dynamic binding between GUID and locations

- **Reliable content transport**: hop-by-hop storage-aware transport scheme
Content Naming

- **three-level name-to-locator separation**: Human-readable names (HRN), GUID, network address
- **The network offers two services**
  - Name assignment service (NAS): domain-specific service
  - Global Naming Resolution Service (GNRS)

![Diagram](image)

**Fig. 1**: Name-locator separation and mapping services in MobilityFirst
Content Discovery

- **GNRS**: content location directory: in-network 1-hop DHT
- **Two-level Structuring of GNRS**: 
  - GNRS: content GUID $\leftrightarrow$ content’s NA address
  - LNRS: content GUID $\leftrightarrow$ content’s local address
  - content location update resolved at LNRS first, and forwarded to GNRS for global part if needed
  - Local location update only handled at LNRS
  - Local lookup only resolved at LNRS
- **Caching name resolution mappings**
- **Dynamic binding between GUID and address**: 
  - Hybrid GUID and network address routing
  - Requery GNRS to update location of content during delivery
Content Discovery (cont’d)

(a) content handling overview

(b) content GUID resolution procedure

Legend:
- Router/GNRS server
- GUID insert/update
- GUID lookup
- Content retrieval

Request Packet ← Content GUID, src host GUID, service ID

if dst in the same AS then
  1. resolve content host GUID from local GNRS
  2. intradomain routing based on host GUID
else
  1. resolve content network address from local GNRS
  2. interdomain routing based on network address
  3. resolve content host GUID after request reaches the network
  4. intradomain routing based on host GUID
end if
Content Delivery

- Reliable hop-by-hop chunk transfer: content $\rightarrow$ chunks (autonomous data unit)
- Storage-aware routing: exploit storage available at each router to overcome disconnections and intermittent link quality variation
- Dynamic binding
Content Caching

Content caching as a computing layer service

Computing layer provides in-network service besides the routing infrastructure
- Serve traffic inside network: conducting requested computation and storage functions

Many issues
- Where to cache
- What to cache
- Caching overhead
Evaluation Plan

- Orbit Evaluation
- Four Use Cases
  - Use case I: downloading popular contents, e.g., movie or video
  - Use case II: peer-to-peer one-2-many delivery
  - Use case III: content download on the go
  - Use case IV: high density, dynamic content retrieval, e.g., stadium
Questions & Answers
What is Context?
- Environmental state that affects communication

Examples:
Network:
- sender, receiver, connection point, channel state
Spatial-temporal:
- Time and Location
Device:
- Type, energy stored, off/asleep/awake
Social:
- In a meeting, busy, free, neighbors
Context Communication Paradigms

Communicate messages to an identity rather than an address
- covered by GUID layer

Communicate messages to an identity on conditional context:
- receive only during work hours, unless from a specific identity
- send a message to whomever is in a car's passenger seat
- send a message to a phone but not when in a moving car

Send a message to an event
- all people at the Winlab's teatime
- anyone at a talk posted on the calendar
- anyone who is in the building that isn't in a meeting
Context Resolution Service (CRS)

Translation context descriptions into network destinations, identified by GUIDs at any level of a service

Context (CRS) stack  Network Stack  Clients

Resource Discovery

Distributor

Data processor (context, event, analysis)

Readers

Sensors

Resource GUID

CDN GUID

Context GUID

Groups GUID

Things’ GUID

MobilityFirst FIA

M2M Applications

Clients

Middleware

Physical objects

Rutgers

WINLAB
**CRS: Implementation Strategies**

**Context Outside the Network: Heavy Lifting on sending client**
- Client issues a SEND-lookup on a query
- Client talks to any of the Servers from the SEND-lookup GUID list

**Context Inside the Network: Heavy Lifting off of clients**
- Client issues a SEND on a CGUID or query
- CRS computes context and carries out the SEND on the Client's behalf

**Intentional Receipt: Client only gets traffic it wants**
- Server/Sensor can issue a SEND with self-described, untargeted data
- Server/Sensor can also issue a GET-lookup for a data description
  - Server/Sensor can SEND to all returned Client GUIDs
Context: Outside the network

Context now:
- Client queries the server address
- Client talks with particular server
- Server computes context
- Server delivers result
- Client interprets result
- Client sends correct NetOp

Aspects:
Heavy lifting on client
- Client needs to understand a lot
- Client has to know what to do based on the server result
- Highly decoupled; can be implemented on nearly any data network

Server
- Needs to manage Sensor data and Context computation

Context Service
- Logically very simple; essentially a database
Context: Inside the network

Context next:
- CRS gets Data
- CRS gets context description
- Client delivers context request
  - or NetOp on context GUID
- CRS computes context
- CRS dispatches correct NetOp

Aspects:
Heavy lifting off of client
- Client does not need to know anything about context
- Client can act on a CGUID without knowing it

Server/Sensor
- Could keep Sensor data local and register as a Sensor itself
- Could specify new operations based on data CRS knows about

Context Service
- Requires a compute layer
- Inherits problems of coherence and consistency
**Context: Intentional receipt**

**Context next++:**
- CRS gets Labeled Data
- Client registers a description
- CRS computes context
- CRS delivers results that match

**Aspects:**

**Client only gets traffic *it wants***
- Client needs to quantify its traffic as context constraints
- Client can use a service to generate these constraints

**Server/Sensor**
- Deliver data with no target address
- Data carries its own description (delivery as a continuation)

**Context Service**
- Computes data self-description against client criteria
- Delivers to clients whose criteria match the data self-description
Example: UbiCab inside network

- UbiCab scenario: “A passenger calls for a nearby cab”
- UbiCab context service (GUID_c) redirects the call based on context – the relative locations of caller and callee
- From outside network (overlay) to inside network (on-router caching)
- Low latency, reduced traffic load on Internet and bottleneck on overlay server
Sensing, Inferring Context

Notion of context varies across different granularities

FIA focussing on a subset
- Location: outdoor, indoor, semantic (e.g., Starbucks)
- User state: driver or passenger, busy or idle
- Predictive: changes in network (WiFi/3G/LTE/Ethernet)

Core approach
- Use mobile phone sensors and network interfaces as “eyes” to observe the user
- Mine the recorded data to infer context
Example:

(1) **Activity: Driver or passenger**
- Phones capture driver signatures
- Enters with right foot first
- Presses gas pedal/brake
- Rotates right to left for seat belt
- Closer to car audio

Acelerometer and gyroscopes adequate for differentiation

(2) **Semantic Location:** Is user in Starbucks vs. Radioshack?
Use phone sensors to sense/fingerprint ambiance

Error zone with existing schemes
PacketCloud: Computing Layer of MobilityFirst

Yang Chen
Networks and Distributed Systems Group (NDS)
Department of Computer Science
Duke University
Basic Idea of PacketCloud

An architectural solution that uses a "cloudlet" co-located with an ISP's routing infrastructure to provide elastic in-network services.
# Service Types

<table>
<thead>
<tr>
<th>GUID ((\text{src, dst}))</th>
<th>Optional NAs</th>
<th>Requested Services</th>
<th>Transparent Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServGUID (= 40) (?)</td>
<td></td>
<td>(Transcoder)</td>
<td>TCP Protocol, DstPort=8080? (\text{Web Cache})</td>
</tr>
<tr>
<td>TP/seq#</td>
<td></td>
<td></td>
<td>DstGUID=20? (\text{IDS})</td>
</tr>
<tr>
<td>PDU (\text{(Protocol Data Unit)})</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Requested Services**: actively requested by end users (mapping rule: a specified service GUID)

**Transparent Services**: dealing with legacy traffic (mapping rule: selected matching fields in packet headers)
Use Cases

For **ISPs**: reducing operational cost (transparent services)
- ✓ IDS: remedy malicious traffic
- ✓ WAN optimizer: reduce redundant traffic
- ✓ Web cache: save upstream transit traffic

For **end users**: in-network add-ons (requested services)
- ✓ Onion routing: anonymous communications
- ✓ Computation offloading from mobile clients: save limited battery

For **third-party application providers**: leasing available cloudlet resource (transparent/requested services)
- ✓ Distributed game proxies: help latency-sensitive games
- ✓ Application-defined IDS, web cache…
An ISP's view

- An ISP deploys multiple PacketCloud sites at its PoPs
- Every site is a service center, hosting multiple services
- A logically-centralized domain controller manages the control plane of all PacketCloud sites of the ISP
A PacketCloud Site

- Forwarding and Service-Mapping Engine (FSME)
  - A MobilityFirst router enabling flow-based demultiplexing
- General-purpose processing Cloudlet
- Site controller

![Diagram of PacketCloud Site](image)
## Service Table of a PacketCloud site

<table>
<thead>
<tr>
<th>Priority</th>
<th>Service</th>
<th>Mapping Rule</th>
<th>Capacity</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IDS</td>
<td>dstNA = NA99/NA32</td>
<td>large</td>
<td>[t1,t2]</td>
</tr>
<tr>
<td>2</td>
<td>Transcoder</td>
<td>Transcoder's ServGUID</td>
<td>large*2</td>
<td>[t1,t3]</td>
</tr>
<tr>
<td>3</td>
<td>Web Cache</td>
<td>tcp, dstPort=8080</td>
<td>tiny</td>
<td>[t2,t3]</td>
</tr>
</tbody>
</table>
Economic model

Operational Cost

Direct charging (End Users)

ISP

Resource Leasing (third-party providers)

Hardware Cost

Cost Saving (Infrastructure, Transit billing)
Implementation

Current progress (IP-based prototype on Deterlab)

- FSME (Open vSwitch + NOX)
- Cloudlet (OpenStack + Open vSwitch)
- One use case: transparent web cache

Next steps:

- Improve the prototype using more CPU-intensive, memory-intensive, and I/O-intensive services
- More use cases and larger scale emulation
- Integrate PacketCloud into the MobilityFirst prototype
Overall goals

- Separate the management plane from data plane
  - Logical virtualization of resources

- Provide visibility into network properties
  - Tie it in with authentication in access

- Include client participation in network management
  - Leverage crowd-sourcing in understanding network performance
Overall goals

- Separate the management plane from data plane
  - Logical virtualization of resources

- Provide visibility into network properties
  - Tie it in with authentication in access

- Include client participation in network management
  - Leverage crowd-sourcing in understanding network performance
Network visibility

- Each entity declares a set of resources that it measures and retains accessible externally
  - Use XML style semantics to declare it
  - Object type (e.g., Cache content), Object value (e.g., stored content in the cache), Timestamps, Certificate of Authenticity, and Users allowed to get/set

- A get, set, and alarm interface
  - Get object @ GUID, requires credentials (or a chain of credentials)
  - Set object @ GUID with value, requires credentials
  - Set alarm, etc.

- Who is allowed access to different objects is a policy decision to be made by administrators
Example in Wireless Enterprise

Can we provide an overall score to the network?
Airshark and WiFiNet

[IMC 2011 and NSDI 2012]

Can we provide an overall score to the network?

Goodness of the link

Links

L1
L2
L3
L4
L5
L6
Airshark and WiFiNet
[IMC 2011 and NSDI 2012]

Uses network visibility to provide performance analysis

Goodness of the link

Impact: 0.7
Impact: 0.3
Impact: 0.1
Impact: 0.2
weak signal
Approach

- Aggregate PHY measurements from multiple devices to a controller
- Controller implements machine learning techniques to perform cause analysis
- Proper authentication of who is allowed to access data is needed
Client-assisted management

- Collect measurements from diverse clients
- Use these measurements to understand network properties

Challenges
- How much data to collect?
- How can we trust this data?
- Can we aggregate this data?
WiScape Architecture [IMC’2011]

WiScape stands for the Wireless landScape
Madison, WI – 155 sq.km. (more than 2 yrs of data)
Persistent dominance

Data between Chicago to Madison
Management-data separation

- Management plane should not depend on availability of data plane

- Separate routing computation for the management plane
  - Routing for connectivity and availability alone and not efficiency
  - Can view this as backup path(s) for management functions
  - Only authorized management traffic allowed to use these routing entries
  - Required that the second routing method be independent of the data plane routing method

- Power of two random choices
Management-data separation

- Possible routing solutions for management plane
  - For core networks, use some static routing methods, e.g., a hypercube on GUIDs
  - For DTN components, use some geographic routing techniques, e.g., GPSR

- Possible to leverage additional capabilities
  - A cellular card in each router that is independently addressable
  - Does not matter if the path is low bandwidth and high latency

- Hence, each node has two NAs---a data path NA and a management path NA

- Management traffic can be sent on the data path, but not vice versa
Security and Privacy Issues in MobilityFirst Architecture

Janne Lindqvist
WINLAB, Rutgers University

MobilityFirst EAB, April 30, 2012
Security and Privacy in MF

- In MobilityFirst, we are looking at security and privacy together because they really cannot be separated from each other
  - Introducing a security mechanism can have implications for user privacy
  - Introducing a user privacy mechanism can have implications to security
- For example, without rigorous design, using public keys as identifiers in protocols can potentially identify users better than e.g. IP addresses today
GNRS – a vital point in MF architecture

Name assignment & certification services (...can incorporate various kinds of trust including CA, group membership, reputation, etc)

GUID = public Key

GUID <-> NA binding

Secure InterNetwork Routing Protocol

Secure Data Path Protocol

Figure source: Wade Trappe
Issues in Store-Aware Routing

- **Buffer to store content in transit**
  - We are waiting for the whole file to arrive, use that time wisely...
  - Scan for malware/signatures

- **Hold to store content when router decides not to forward due to disconnection (e.g. DoS), poor path metric, contamination, congestion, etc**

- **Cache for in-network storage, along with redundancy allow for fail-safe mechanisms**

Figure source: Wade Trappe
Security & Privacy for MF packets

- Off the record messaging on the network layer
  - Authentication
  - Encryption
  - Deniability
  - Perfect forward secrecy

- We can build non-repudation and e-commerce applications on top of off-the-record network layer
  - The other way round does not work without additional complexity (e.g. overlays)

- Disposable identifiers [e.g. Gruteser’03, Lindqvist’05, Lindqvist’08] that do not have geographic or semantic mapping
Open Issues

- Analysis on impact of disposable identifiers in MF
  - Today, routing scales because you can request only as many disposable identifiers (IP address) as have been provisioned to the network
  - In MF, you could have arbitrary number of disposable identifiers

- What should the (clean-slate) security & privacy user experience be?
  - e.g. today’s browser security infrastructure is broken
  - User experience and security design introduced later
  - We could design the user experience clean-slate as well
Thank you
Network-Wide Management of IDS and Other Middleboxes

Michael K. Reiter
Scaling intrusion detection & prevention

Traffic increase → More packets to process
User applications changing → More functions/modules
Attacks evolving → Larger set of “rules” to apply
Traditional approaches for scaling NIDS/NIPS

1. Build clusters
2. Better algorithms
   ...

Single-vantage-point solutions designed for *perimeter* deployments (gateway between internal/external network)
Single-vantage view is restrictive

Capacity of each installation = 50 units

Overloaded!

Many NIDS/NIPS infrastructures are multi-node deployments e.g., ISPs, large enterprises, datacenters
Alternative: Network-wide scaling

Treat NIDS/NIPS infrastructure as one system
- Leverage on-path opportunities for distributing work
- Complements single-vantage scaling
Key requirements

1. Resource Constraints

2. Placement Affinity

3. Network-wide objectives

Systematic designs for network-wide NIDS/NIPS deployment
Current Directions (1)
[w/ Heorhiadi & Sekar]

• Initial work [CoNEXT 2010] assumed ...
  – Monitoring responsibilities could be distributed only on-path
    • What about forwarding traffic to a datacenter?
  – Analysis is self-contained
    • Topologically constrains certain types of analysis

• Goal: a more general framework for exploiting traffic replication and result aggregation
Network Evolution today: Middleboxes!

<table>
<thead>
<tr>
<th>Type of appliance</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firewalls</td>
<td>166</td>
</tr>
<tr>
<td>NIDS</td>
<td>127</td>
</tr>
<tr>
<td>Media gateways</td>
<td>110</td>
</tr>
<tr>
<td>Load balancers</td>
<td>67</td>
</tr>
<tr>
<td>Proxies</td>
<td>66</td>
</tr>
<tr>
<td>VPN gateways</td>
<td>45</td>
</tr>
<tr>
<td>WAN Optimizers</td>
<td>44</td>
</tr>
<tr>
<td>Voice gateways</td>
<td>11</td>
</tr>
<tr>
<td><strong>Total Middleboxes</strong></td>
<td><strong>636</strong></td>
</tr>
<tr>
<td><strong>Total routers</strong></td>
<td>~<strong>900</strong></td>
</tr>
</tbody>
</table>

Data from a large enterprise: >80K users across tens of sites

Just network security
$10 billion
Current Directions (2)
[w/ Sekar, Egi, Ratnasamy, & Shi]

Goal: Consolidation at two levels corresponding to two sources of inefficiency

1. Consolidate Platform

2. Consolidate Management

Network-wide Controller
Mobility First Future Internet Architecture (FIA) 
External Advisory Board (EAB) Meeting

Economic models and Business Structure

William Lehr
Massachusetts Institute of Technology
wlehr@mit.edu

Rutgers, WINLAB
April 30, 2012
Mobility First: Economics and Policy

(1) Mobility First Policy Challenges & Opportunities: white paper
- Privacy
- Network Neutrality
- Wireless & Spectrum Management
- Reliability
- ??

(2) Modern Mobile Networks
- MF & LTE:

(3) Ancillary Activities
- Cloud Reliability: Issues of FIA and Reliability
- Broadband Internet as the New PSTN
- Economics of Mobility and Mobile Services
Mobility First: Value Chain => Competition

- **MF facilitates mobile (wireless) – Internet convergence**
  - Expands competition between fixed/mobile Internet
  - Enhances value of mobility generally, wireless especially
  - Reduces cost for fixed to support mobile services

- **MF empowers end-users**
  - End-user deployed, “carrier-free” infrastructure
  - Intensifies equipment/service provider competition
  - Support for ad hoc/mesh networking

- **MF enables new wireless**
  - Support for DSA
  - Lower entry-barriers for heterogeneous wireless (Beyond LTE)
  - Internet shifted closer to wireless edge
Mobility First: Censorship?

- **What is censorship?** Government limiting access to select content
  - What about pornography? Illegally accessed copyright material?
  - Violation of free speech? Legitimacy of government censorship?

- **Aspects of MF that might make easier**
  - Granular support for policy (context-based routing)
  - Support for strong authentication (enforcement sanctions)

- **Aspects of MF that might make harder**
  - Support for competition
  - GUID: flat address/routing and encryption, anonymous GUIDs
  - Multihoming support
  - Support for distributed resources, no single point failure
Mobility First: Spectrum management

- **RF spectrum is the scarce resource in wireless**
  - Future must be shared spectrum: across users/uses, bands (waveforms), infrastructures
  - Dense antenna architectures (with dense fiber)
    - “Last-mile” antenna+fiber a public utility?
    - How to support network competition on shared antenna+fiber plant?

- **MF support for Dynamic Spectrum Access (DSA)**
  - Internet to manage spectrum assets (dBase)
  - Virtual LTE base-stations and MF-enabled peer-to-peer mesh support

- **MF challenges (opportunities) for cross-layer optimization**
  - DTN networking and DSA?
  - Network MIMO? Application MIMO? An extension of multi-homing?
Prototyping and Evaluation of MobilityFirst Architecture

Rutgers, The State University of New Jersey

WINLAB

Contact: nkiran (at) winlab (dot) rutgers (dot) edu
Objectives

- Software prototypes of main arch. pieces with end-to-end protocol implementations
  - Consider OpenFlow/SDN, FPGA platforms, and optical components

- Mobility, first
  - Mobile host platforms and multiple wireless access technologies
  - Ad hoc, disconnection, network mobility as common cases

- Meaningful at-scale evaluation
  - Realistic network conditions
  - Access-Core-Access evaluations
  - Target long running deployment and opt-in users
Execution Summary

Phase 1
- Context Addressing Stack
- Content Addressing Stack
- Host/Device Addressing Stack
- Encoding/Certifying Layer
- Global Name Resolution Service (GNRS)
- Storage Aware Routing (e.g., GUID-based)
- Context-Aware / Late-bind Routing

Phase 2
- Context Addressing
- Content Addressing
- Host/Device Addressing
- Encoding/Certifying Layer
- Global Name Resolution Service (GNRS)
- Storage Aware Routing (e.g., GUID-based)
- Locator-X Routing (e.g., GUID-based)
- Context-Aware / Late-bind Routing
- IP Routing (DNS, BGP, IGP)

Phase 3

Prototype
- Standalone Modules
- Integrated MF Protocol Stack and Services
- Deployable s/w pkg., box

Evaluation
- Simulation and Emulation
- Smaller Scale Testbed
- Distributed Testbed
  E.g. ‘Live’ on GENI
Prototype: Click-based Router

User-level Processes

Click
Forwarding Engine

Inter-Domain
R3
GSTAR
Routing
Locality-Aware DNS
DMap – DiHT
Name Resolution
Compute Services
Content Cache Service
Mgmt.

PacketCloud Framework

To/From Host
Host Rx Q
Host Tx Q

Forwarding Table
Next-hop Lookup

Block Aggregator
Service Classifier
Packet Classifier

To Next-hop Lookup

Rx Q

Hold buffer

Block Segmentor

Wired and wireless i/f

x86 hardware and runtime

Early Dev.
Integrate
Prototype: Host Protocol Stack and Network API

- 'Socket' API
  - open
  - send
  - send_to
  - recv
  - recv_from
  - close

- Network API
  - E2E Transport
  - GUID Services
  - Security
  - Routing

- Interface Manager
  - 'Hop' Link Transport
  - WiFi
  - WiMAX

- Context API
  - Context Services
  - Sensors

- Network Layer
  - User policies
  - Early Dev.
  - Integrate

Linux PC/laptop with WiMAX & WiFi

Android device with WiMAX & WiFi

Device: HTC Evo 4G, Android v2.3 (rooted), NDK (C++ dev)
GENI Deployment & Evaluation

**Deployment Goals**
- Large scale, multi-site
- Mobility centric
- Realistic, live

**Mapping onto GENI Infrastructure**
(ProtoGENI nodes, OpenFlow switches, GENI Racks, DieselNET buses, WiMAX/outdoor ORBIT nodes)

Legend
- Internet 2
- National Lambda Rail
- OpenFlow Backbones
- OpenFlow
- WiMAX
- ShadowNet
- MobilityFirst Router & GNRS Servers
- Mobile Hosts
- Static Hosts
GENI Deployment: GEC-12 Demonstration Topology

- WiFi AP
- WiMAX BSS
- MF Router + Name Resolution Server
- Android Client w/ WiMAX + WiFi
- Linux PC/laptop w/ WiMAX + WiFi
- Vehicular node w/ WiMAX

ProtoGENI Backbone

BBN
Cambridge, MA

Mesoscale
WiMAX and OF

WINLAB
N. Brunswick, NJ
GENI Deployment: GEC-12 Demo Physical Topology
GENI Deployment: GEC-12 Demo Application Scenario
Content Delivery to Mobile Hosts

Content Publisher

WiFi AP

GUID=1

GUID=2

GUID=3

GUID=4

GUID=5

GUID=6

GUID=7

WiMAX BTS

 content

NA

DATA

GUID & SID

GUID=101

BBN Wireless Edge

ProtoGENI Backbone

Rutgers Wireless Edge

Content Subscriber

WiFi AP

GUID=201

WiMAX BTS

NLR path using VLANs 3716, 3799 (Clemson)

I2 path using VLANs 3715, 3745(BBN), 3798 (Clemson)

ProtoGENI host running MF Router, GNRS Server
Evaluation Concerns

- Limitations in evaluating global scale services
  - Candidate testbeds are short on both realism and scale
  - GENI good for realism, but limited to 10s of nodes (ProtoGENI nodes)
  - Emulation still best bet for at-scale

- For mobility events, settle for emulation?
  - Opt-in users present real opportunity

- How to make MF services readily available?
  - Avoid requiring users to root their phones
  - Legacy applications
  - Cellular access using IP-tunnels and protocol translation at gateways
BACKUP SLIDES
Router Architecture and Services Abstraction

Messages To/From Services

Other Services (E.g., Compute, Content Caching)

Register Name Res. Callback

Name Resolution

Forwarding Table Updates

Register Topology Callback

Routing

Get/Set Parameter

Register Exception Callback

Management

Forwarding Engine Abstraction

Message Relay

Services Registry

Name Resolver Proxy

Topology Discovery Manager

Events

Callback Manager

Register

Callback

Forwarding Engine

Services Table

Name Resolution Table

Forwarding Table

Configurations and Execution State

Register Service

Name Resolution Request

Forwarding Table Updates

Get/Set Parameter

Exceptions

Interface X

Interface Y

Interface Z
## Protocol Details: Routing Header

<table>
<thead>
<tr>
<th>Service ID</th>
<th>Header Length</th>
<th>Next Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol ID</td>
<td>Hop Count</td>
<td>Payload Offset</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload Length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination GUID (short)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination Network Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source GUID (short)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Network Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extended Service Headers (if any)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Extended Service Header

<table>
<thead>
<tr>
<th>Service ID</th>
<th>Header Length</th>
<th>Next Header</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Custom Service Fields</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Source Routing Extension Header

<table>
<thead>
<tr>
<th>Service ID</th>
<th>Header Length</th>
<th>Next Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>= [0x8001]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Next Destination Offset
- Intermediate destination 1 (e.g., GUID or NA)
- ...
- Intermediate destination N

### PKI Cryptography Extension Header

<table>
<thead>
<tr>
<th>Service ID</th>
<th>Header Length</th>
<th>Next Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>= [0x8000]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Source GUID = [PEM encoded Source Public Key]
# Service Identifiers (SID) for In-Network Services

<table>
<thead>
<tr>
<th>Service</th>
<th>16-bit hex code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicast</td>
<td>0x0000</td>
</tr>
<tr>
<td>Multicast</td>
<td>0x0001</td>
</tr>
<tr>
<td>Anycast</td>
<td>0x0002</td>
</tr>
<tr>
<td>Block Transfer</td>
<td>0x0004</td>
</tr>
<tr>
<td>Stream or Real time</td>
<td>0x0008</td>
</tr>
<tr>
<td>Delay Tolerant</td>
<td>0x0010</td>
</tr>
<tr>
<td>Acknowledge on Store</td>
<td>0x0020</td>
</tr>
<tr>
<td>Acknowledge on Delivery</td>
<td>0x0040</td>
</tr>
<tr>
<td>Content Request</td>
<td>0x0080</td>
</tr>
<tr>
<td>Content Response</td>
<td>0x0100</td>
</tr>
<tr>
<td>Compute Layer Processing</td>
<td>0x0200</td>
</tr>
<tr>
<td>Public-Key Cryptography</td>
<td>0x8000</td>
</tr>
<tr>
<td>Source Routing</td>
<td>0x8001</td>
</tr>
</tbody>
</table>
Demonstration: HotMobile’12
Delivery Services for Multi-Homed Devices with User preference of delivery interface

Router bifurcates PDU to NA1 & NA7 (Routing based on client policy)

Data Plane

GNRS Entries with client policy:

<table>
<thead>
<tr>
<th>GUID</th>
<th>NetAddr</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxx</td>
<td>NA1:PA22</td>
<td>3</td>
</tr>
<tr>
<td>xxx</td>
<td>NA7:PA13</td>
<td>1</td>
</tr>
</tbody>
</table>

GUID & SID
Send data file to “Alice’s laptop”

NetAddr= NA1.PA22
NetAddr= NA7.PA13

Alice’s laptop
GUID = xxx

Dual-homed mobile device

Periodic association message to Access Router reporting the binding state and policy; Access Router in turn reports to GNRS
Demonstration: GEC-13
Content delivery to WiFi/WiMAX Multi-homed Mobile Device

GENI Mesoscale

MobilityFirst Router hosted on Protogeni node

Rutgers Wireless Edge

WiMAX coverage
Wi-Fi coverage
Core Optical Layer Integration for MobilityFirst

Byrav Ramamurthy
(with Grad Students: Joyeeta Mukherjee and Lin Liu)
University of Nebraska-Lincoln

with input from

Kiran Nagaraja
WINLAB
Rutgers University

April 2012
Introduction

• Need for hybrid optical/packet core architecture for MF
  - Increasing Big data and QoS-guaranteed services
  - Elephant Flows

• Reality
  - Two types of networks coexist: packet network and circuit network.
  - Optical circuit switch consumes 1/10 volume, 1/10 power and 1/10 cost with the same capacity => Savings $$$
  - OpenFlow and caching network can help circuit creation in MF
  - So far, MF architecture assumes packet traffic between adjacent MF routers => Need to incorporate “bypass” (at L2, optical layers)
MF architecture

- GUID based naming of network objects and groups
  - Handles mobility using a global name-to-address mapping service (GNRS)
- Flexible hybrid name-address routing, that is both edge-aware (at the core) and storage resources aware (at the edge)
- Storage capable edge routers to handle disconnection and intermittent bad links
- Routing control and management plans share topology (alternate routes) and traffic conditions at edge
  - Core reconfiguration can alleviate edge traffic stresses edge via bypass or alternate paths

![Diagram of MF architecture]

- Remote web servers providing content to end-users at edge networks
- Dynamic appearance of mobile hotspots, and network-level mobility. E.g., a football game.
- Alternate access networks for delivery (WiFi, cellular, WiMAX)
- Dynamically reconfigurable optical core segment
- Receives edge condition updates from control/management plane

Network – NA1

Network – NA2

Data plane

Global Name-Address Resolution Service (GNRS)

Edge-aware inter-domain routing

Storage-aware edge routing

Alternate access networks for delivery (WiFi, cellular, WiMAX)
Optical Network

• Optical Network Layers
L2 Circuit Creation (Esnet / I2 ION)

- Uses OSCARS (L2 over MPLS)

![Diagram of L2 Circuit Creation](image-url)
DCN between UNL and MAX

- Dynamic circuits established between UNL and MAX
- Layer 2 connection between Netgear switch and Prairiefire supercomputer at UNL
- US CMS Physics data from UNL transferred to MAX Planetlab machine
Scenario 1: Static L2 VLAN setup

- Packets are sent from the Source to the destination
- A VLAN Tunnel created from MFR1 to MFR3 bypassing MFR2
- The switching or Tunnel is created based on Network parameters
- Network parameters include bandwidth/capacity, cache size and resource (wavelength) availability
Scenario 2: OpenFlow Controller

- Flow Tables are modified by the OpenFlow Controller
- Modification is done based on the Network parameters
- Packets destined for GUID “D” needs to be directly forwarded to MFR3 from MFR1
- MFR1 Flow Table is modified accordingly
Advantages of MF and Optical Layer Integration

- Anticipated traffic loads can be managed with ease by creating optical layer circuits ahead of time
- Circuits can be dynamically created and torn down at the optical layer depending on the network needs
- A direct tunnel created in the optical domain reduces processing overhead at the MF Routers
- Optical integration also allows us to take advantage of OpenFlow (SDN) technologies by allowing the controller to request switched connectivity from the optical domain.
- MobilityFirst architecture can benefit
  - GUID Service Layer (use optical circuits to facilitate content caching)
  - Multi-homing across LTE and WiFi networks (use of optical layer multicast trees)
Research Questions

- Circuits: Multi-domain scenario, Path Computation Element design, Route optimization, Cost modeling, Static/Dynamic circuits
- Trigger: When to setup circuits (trigger-based?) and for how long? How many circuits? Granularities?
- Heterogeneous network scenarios: Optical/OTN/Ethernet/MPLS networks
- End-to-end edge-to-edge circuit setup
- Use of Openflow controllers (and integration with OSCARS software for the backbone networks)
- Prototype validation and ORBIT testbed experiments
- Large-scale deployment (GENI network)
Extra Slides on Experimentation
Experiment

- Worked with two topologies
- The image file used to image the nodes is *mf-proto-rc1_0.ndz*
- The scripts used to run the experiments
  - `testcfg_1-gstar_3node.rb`
  - `testcfg_2-gstar_7node_multiflow.rb`
- A file is transmitted from the Sender to the Receiver
- `> ssh root@node1-1 (Receiver) to check for the file`
Experiment cntd…

- Experiments run on Sandbox 9

```bash
joyeeta@console.sh9:~$ vim testcfg_1-gstar_3node.rb
joyeeta@console.sh9:~$ omf exec testcfg_1-gstar_3node.rb
INFO NodeHandler: init Experiment ID: sh9.orbit-lab.org_2012_04_24_18_13_17
INFO Experiment: load system:exp:stdlib
INFO property.resetDelay: value = 80 (Fixnum)
INFO property.resettries: value = 1 (Fixnum)
INFO Experiment: load testcfg_1-gstar_3node.rb

Adding node: n_1_4 as Sender

Adding node: n_1_3 as Router

Adding node: n_1_1 as Receiver

INFO whenAll: *: 'status[@value='UP']' fires
INFO exp: Request from Experiment Script: Wait for 5s....

Bringing up Router...

Bringing up Receiver...

INFO exp: Request from Experiment Script: Wait for 10s....

Bringing up Sender...

Waiting for transfer to complete...
(check Receiver node data folder: /root/file)
INFO exp: Request from Experiment Script: Wait for 60s....
INFO Experiment: DONE!
INFO NodeHandler: Shutting down experiment, please wait...
INFO run: Experiment sh9.orbit-lab.org_2012_04_24_18_13_17 finished after 1:21
joyeeta@console.sh9:~$ 
```

Future plans

- Perform Experiments on Sandbox 4 to try out the wireless environment
- Implement new topologies
- Implement Scenario 1 with VLAN configuration
- Implement Scenario 2 with OpenFlow capabilities on Sandbox 9
- Once caching and storage is implemented, incorporate them in the current scenarios
Football game scenario

- From Rutgers create a switched 1Gbps circuit to UNL (for example to share football game video)