MobilityFirst Prototyping and Evaluation

October 6, 2011
Prototyping and Evaluation: Execution Summary

Phase 1
- Context Addressing Stack
- Global Name Resolution Service (GNRS)
- Storage Aware Routing
- Context-Aware / Late-bind Routing

Phase 2
- Context Addressing
- Global Name Resolution Service (GNRS)
- Storage Aware Routing
- Locator-X Routing (e.g., GUID-based)

Phase 3
- IP Routing (DNS, BGP, IGP)

3-Year Timeline

Prototype
- Standalone Modules

Evaluation
- Simulation and Emulation
- Smaller Scale Testbed

Deployment
- Integrated MF Protocol Stack and Services
- Deployable s/w pkg., box
- Distributed Testbed E.g. ‘Live’ on GENI
MobilityFirst Prototype: Network Architecture

- Edge networks NA-1, NA-2 connected to global core network
- Each of NA-1, NA-2 are contained MF routing domains
- Each WiMAX BSS and WiFi AP is associated with a MF Router
- Node a is **multi-homed** within a network
- Node c is **multi-homed** across 2 networks

![Diagram showing network architecture]

**Ad hoc networks:** Nodes can form ad hoc networks which are named and can attach to existing networks to be globally reachable themselves
Wireless Edge Evaluation: Phases 1, 2 on ORBIT

- Multi-radio indoor and outdoor nodes with WiMAX, WiFi interfaces
Prototyping and Evaluation: Phase 3 on GENI

**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)
MobilityFirst Prototype: Software Router Framework

- Linux-based software router with two-level emulation

- **Portable user-level implementation**
  - Quagga
  - XORP
  - Xing letting
  - Click

- **Forwarding Engine**
  - Commodity Hardware

- **User-Level Control Plane**
  - MF Routing & Mgmt.
  - MF App Services

- **OpenFlow Controller**
- **OpenFlow switch host**
- **NetFPGA**
MobilityFirst Prototype: Click-based Router

- Linux-based implementation with Click modular router as forwarding engine
- Two-level abstraction: fast path as Click elements, slow path as user-level processes (control and support services)

![Diagram of Click Forwarding Engine](image)
MobilityFirst Prototype: Client Architecture

- Linux and Android mobile implementation for clients
- Applications: mobile social networking, content delivery and context-aware messaging
MobilityFirst Prototype: Android/Linux Client Implementation

- Device: HTC Evo, Android 2.3
  - Unbranded and *rooted*
  - Development: SDK, NDK, flash a modified kernel (if required)
  - WiFi, WiMAX interfaces

- Modules in Android’s MF stack
  - MF-socket API - user level library
  - Transport layer
  - Storage aware routing
  - SHIM layer support for multi-homing
  - 1-Hop reliable data transfer

- MF-socket API
  - open, send, send_to, recv, recv_from
  - User policies for resource use and intentional data receipt
MF Services: Network (socket) API

- Open – define identity, network behavior
  - handle$\leftrightarrow$Open (URL, srcGUID, stack-options)
- Send – accommodate DTN delivery
  - Send (handle, message, SrvcFlags)
  - SendTo(GUID, message, SrvcFlags)
- Recv – intentional receipt
  - Recv(handle, recvBuffer)
  - RecvFrom(handle, recvAllowGUIDSet, recvBuffer)
- Get - content retrieval
  - Get(handle, GUID, recvBuffer, SvcFlags)
- Close – remove network state
MobilityFirst Prototype: Status

- Click-based Router v0.1 implemented
  - Routes MF packets over Ethernet or 802.11 MAC frames
  - Hop by hop reliable data transfer (Hop)
  - Storage-aware link-state routing with DTN extensions (GSTAR)
  - Under evaluation in wired and WiFi networks in ORBIT testbed

- Integration of R3, an adaptive message replication protocol from UMass, into Click framework under consideration

- Distributed Name Resolution Server
  - Modular C++ version that can support any defined distribution and resolution strategy under development
  - Java version that uses a fixed participation set under development
  - Initial versions are under evaluation in ORBIT/PlanetLab
MobilityFirst Prototype: Status Contd.

- **Client Stack for Android**
  - API paralleling Berkeley sockets defined
  - Accommodates user intent in stack composition, data reception, and resource-to-performance tradeoffs
  - Protocol stack designed using libpcap for low-level packet handling
  - Routing and hop data transfer modules being ported from Click impl.
  - Sample transport layer implementations under development

- **Client Stack for Linux PC/laptop**
  - Code from Android/Click will be ported
MF Prototype: Next in Implementation/Integration

User-level Processes:
- Inter-Domain
- R3
- GSTAR
- Routing
- Locality-Aware DNS
- Name Resolution
- Compute Services
- Content Cache
- Mgmt.

Click Forwarding Engine:
- Classifier
- Rx Q
- Block Aggregator
- Next-hop Lookup
- Forwarding Table
- To/From Host
- Host Rx Q
- Host Tx Q
- To Next-hop Lookup
- Hold buffer
- Block Segmentor

Forwarding Elements:
- Wired and wireless i/f
- x86 hardware and runtime
- DMap – DiHT
- Locality-Aware DNS
- GSTAR

Services:
- Networking
- Network Management

Android/Linux Runtime
GEC-12 Demo: Overview

- Network: 3 edge networks connected to Protogeni backbone
  - WiFi and WiMAX at each edge
- Deployed MF Components:
  - MF prototype router across backbone and edge networks – includes both routing and name resolution services
  - MF clients including Android phones, Linux PC/laptops, and vehicular nodes
- Applications: mobility and context driven
  - Unicast, multicast, anycast content delivery
  - Sensor-driven M-2-M communication
  - Live/CBR content delivery
- Demonstration Focus:
  - Multi-homing - convergence of WiFi and WiMAX
  - Network-level adaptation to mobility and disconnection
GEC-12 Demo: Possible 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
- Sensor node
- MF Sensor GW

ProtoGENI Backbone

BBN Cambridge

NYU-Poly Brooklyn

WINLAB N. Brunswick
GEC-12 Demo: ProtoGENI Backbone Deployment Proposal

Ideally R1, R2 are ProtoGENI or myPLC nodes at WINLAB

Rutgers Wireless Edge

ProtoGENI Backbone

BBN Wireless Edge
GEC-12 Demo: ProtoGENI Backbone Deployment Proposal

Mostly NLR north path

OF@Atlanta

Mostly I2 south path

PG2@BBN

PG@Stanford

PG@GTech

PG1@BBN

WiFi AP

WiMAX BTS

WiFi AP

WiMAX BTS

PG1@Rutgers

PG2@Rutgers

BBN Wireless Edge

Rutgers Wireless Edge

Mesoscale OF Backbone
GEC-12 Demo: Visualization

Data collection framework with API, monitors, filters and data warehouse
E.g., Orbit Measurement Library (OML)

What’s on?
1. Network statistics
2. Packet and flow tracing
3. Routing events
4. Application events

Network map credits: ProtoGENI’s Flack tool. http://protogeni.net/trac/protogeni
GEC-12 Demo: Deployment Steps

- ProtoGENI control s/w and WiMAX/WiFi infrastructure
  - Accounts at Utah EmuLab and/or BBN
  - Coordinate with Ivan/GPO for WIMAX/WiFi at 3 sites
- Utah EmuLab as starting point for port from Orbit testbed
  - Deploy and evaluate single domain topology of MF-Net
  - Short on wireless resources – no WiFi, WiMAX
- Expand router deployments to ProtoGENI backbone nodes
  - Using VLANs to obtain a multi-domain topology
- Tap into ProtoGENI’s monitoring infrastructure
  - Packet traffic stats, link delays, node load
Client Stack Implementation
Demos During NSF Visit

1. Storage-aware intra-domain routing in MobilityFirst (GSTAR)
   Presenters: Kai Su, Nehal Somani – WINLAB, Rutgers

2. Multi-homing capability with MF client stack on Linux/Android
   Presenters: Chunhui Zhang – UMass Lowell

3. Resolving Host and Content Mobility using Global Name Resolution Service
   Presenters: Feixiong, Tam Vu – WINLAB Rutgers
   Presenter: Hardeep Uppal – UMass Amherst

4. Intentional Data Receipt using Context Resolution in MobilityFirst
   Presenter: John Austen – WINLAB Rutgers

5. Detecting Driver Phone Use Leveraging Car Speakers
   Presenter: Tam Vu – WINLAB, Rutgers

6. WiRover: Network aggregation using multi-homing and multi-path striping
   Presenter: Suman Banerjee – UWisconsin
Demo 1: Storage-Aware Routing

1. Destination node mobility
   - B moves between AP3 and AP4

2. Variable link quality
   - Access link B-AP4 degrades occasionally
   - Data blocks temporarily stored at AP4

3. DTN routing and mobile data ferry
   - Link R2-R3 completely fails, creating partitions
   - Bus-node bridges partitions, moving from within AP1 to AP3 range
Demo 2: Multi-homing

- Multi-homed mobile with varying link quality (WiFi & WiMAX) receives on either interface or a preferred one
- Multi-homed mobile stripes across two interfaces (WiFi & WiMAX)
  - Cumulate access bandwidth
  - Reordering of striped chunks at receiver
1. Content host A publishes content C to GNRS.
2. Content host B publishes content C to GNRS.
3. Content requestor queries the GNRS for content C.
4. Content requestor gets two locations for content C and chooses a closer one, which is host B. And then it retrieves the content.
Demo 4: Sensor and Context Use Case

1. Name assignment and publishing
   - Mapping from human readable (tags) to GUIDs, for both sensors and context
2. Connect to MF network
   - GNRS is updated as sensor and context apps open MF sockets
3. Caller gets GUIDs
   - Driver’s GUID: non restricted GUID_driver, restricted GUID_context (no call while driving)
   - Seat’s GUID: anyone in the car seat?
4. MF routers route the call to right location (context or phone directly)