Context-Aware Services

Professor Rich Martin
Joint with Professors Marco Gruteser, Jon Li, Romit Roy, Yanyong Zhang

Rutgers, The State University of New Jersey
http://www.winlab.rutgers.edu/

Contact: Professor Rich Martin
rmartin@cs.rutgers.edu
MobilityFirst Background

Recall 3 abstractions from prior talks:

1. Symbolic name
   Human readable name
2. Global Unique ID (GUID)
   Large # of bits (>128) communication end-point
3. Network Address/locator
   A routable address in the network
Mapping Services

Key services:

1. Name Assignment Service (NAS)
   Symbolic Name->GUID

2. Global Name Resolution Service (GNRS)
   GUID->Address
MobilityFirst Graph Model

Key architectural element:
a vertex is named distinctly from edges (unlike IP)

GUID=vertex
Address = edge set (path)
What is Context, anyway?

Vertex environment = context

Change communication behavior depending on the changes in the environment
Example Contexts

Domains specify context differently:

Social
  In a meeting, busy, free

Activity based
  Walking, sleeping, eating, at dinner, working

Network
  Source, destination, attachment points, Device ID, 1-hop neighbors

Physical properties
  Location, Time, Speed, Direction, Energy, Device capabilities
Impact of Context

Network actions on packets or connections become context dependent:

- Accept or drop
- Redirect/reroute
- Store/buffer
- Split or merge

Optionally notify source of actions
Applications and Themes

Application Areas:

- Vehicular networks
- Sensor networks
- Emergency/disaster
- Developing countries
- Social Mobile networks

Common Themes:

- Context specifies a Group of vertices matching a condition
- Context specifies when a vertex matches a condition
Concrete Proposal

Overload the name and GUID to name a context:

Example names:
rich.martin@anywhere -> GUID1
rich.martin@isfree -> GUID2

External context services reprogram network mappings
(GUID->address, or GUID->GUID)
Extended Network Elements

A super-router does the following based on external context service updates:

Forwarding:

Address -> output port map

Name and Resolving:

Name -> GUID (like DNS)
GUID -> address (like DNS)
GUID -> GUID (like NAT)
Address -> Address (like NAT)

Packet manipulation:

Split/Merge (like *cast)
Store until change in context (DTN)
Scenario: Tom’s phone accepts calls only from VIP when he is in a meeting

GUID1: Cellphone, Attributes: depends on GUID2
GUID2: Location tag

8AM: GUID1 → N2:p2
10AM: GUID1 → N1:p3

VIP’s NRS query (GUID1)

8AM: GUID1 → N2:p2
10AM: GUID1 → N1:p3

Other’s NRS query (GUID1)

8AM: GUID1 → N2:p2
10AM: GUID1 → NULL

GUID2 indicates a context change, which determines how NRS resolve GUID1’s current address
Pitfalls and open issues

Still early in the project

Can’t easily define sub-contexts

More receiver driven than sender driven ->
  • Receiver must publish and update contexts
  • Rely on attached devices to resolve context actions
  • What happens when a vertex “goes dark” (never updates)

Must support local and global context names

The “art” of balancing simplicity, generality, security and privacy
Backup slides

Following slides are for backup
Simplified Graph Model

Network is a graph

IP address specifies the route + the communicating entity
IP address + BGP = current edge set + end vertex

MobilityFirst:
  GUID specifies a vertex
  Address specifies the current edge set
Isn’t DNS the GUID?

No!

Name not carried in all communications

Re-resolution does not happen within network elements

All the problems of deep packet inspection
Example: Vehicular Networks

Node is “Hard braking” is a context

If (I am braking) then
   {send a brake packet to neighbors}
If (I receive a brake packet) then {
   if (source location, speed and director are ahead)
      then {forward the packet to neighbors}
**Example: Mobile telemetry**

**Scenario**: Distribution of blood pressure data of Tom while he is on daily mobility

- **Doctor** gets **GUID1**
- **GUID1** assign
- **GUID1** publish
- **GUID2** assign
- **GUID2** publish
- **GUID2**: certificate, CA, owner. Attributes: type/sensor, function/bloodpressure

**Global Name Resolution Service**

**Name Assignment (Nurse’s choice)**

- **N1**
- **N2**

**NRS query (GUID1)**

- Friday: **GUID1** \(\rightarrow\) **N2**:P2
- Saturday: **GUID1** \(\rightarrow\) **N1**:P3

**Bundle**: \(08:3A \rightarrow N1:p3\)

**NRS update (GUID1/2)**

- **Bundle**: \(08:3A \rightarrow N2:p2\)

**Name Assignment (Tom’s choice)**

Sensor phy addr: **08:3A**

Current technology would only allow sensors being shared at app level
Example: Emergency Services

Contact all ambulances within K kilometers

If ( I am an ambulance) then

{deliver the packet} else {}
always {store data, forward to neighbors when available}
Example: Social Mobile Network

If ( I am in a meeting) then

    {allow members on the family list to contact my device}

else { reject the communication}

How to implement in MobilityFirst:

A social service application on phone changes GUID->GUID mapping based on the location service, time, calendar and address book