IoT Devices & Applications
Ultra-Low Power Active RFID (WINLAB, 2010)

- Low-Power “Transmit-Only” Active RFID Technology for IoT applications

### Classic
**TelosB (2004)**

- Antenna
- Radio
- Micro controller
- Battery

### Transmit-Only
**TO-PIP (2013)**
Augmented Reality Tags \textit{(WINLAB, 2014)}

- RFID tag + optical detector for head-mounted AR/VR applications
Wireless Smart Meters (Privacy Study, WINLAB 2012)

- Analyzed existing meters with USRP software radio
- Meters broadcast every 30s
- Significant privacy/security weaknesses as deployed
Pedestrian Safety Services (WINLAB prototype 2015)

- Warning for distracted pedestrians to look up as they walk off from a sidewalk to street
- Profiles surface gradient using shoe mounted inertial sensors
- Uses the ground profiles to detect transitions from sidewalk to street via ramps and curbs

Researchers at Rutgers University made an Android app that senses when a texting pedestrian steps into a street, warning the person to look up. Photo: Geoffrey A. Fowler/The Wall Street Journal
OWL Platform & Pipsqueak Sensor
(WINLAB/InPoint prototype, 2012)

**Owl Platform Software Stack**

- 10+ year ultra-low energy wireless sensor
- Reliable delivery in challenging radio environments
- Multiple sensing capabilities

- Scalable, distributed software architecture
- IoT software stack targeting data streams from wireless sensors
Lab Animal Monitoring Application
(Rutgers deployment, 2014)

- 9-month installation
- Improved staff workflow
- Reduced problem response times
- Supportive CMR management and faculty
- Product requirements developed
OWL User Interface for Building Monitoring Application (WINLAB, 2014)
IoT Requirements
IoT Architectural Requirements

- Future mobile networks should be designed to natively support anticipated IoT communication modes.
IoT Architectural Requirements

- Scalability
  - Scale to billions of devices; low control overhead

- Access Technology Neutral
  - Wireless access (WiFi, 6LoWPAN, LTE, 5G..) as “plug-in”

- Mobility
  - Device and cloud service mobility

- Device Limitations
  - Power/BW limited; sleep modes/disconnected operation

- Performance
  - Low or bounded latency

Reference: IETF Draft, Feb 2016, Zhang et al, ICN Based Architecture for IoT
IoT Architectural Requirements (cont.)

- Naming, Dynamic Name Resolution
  - Device/application centric, persistent during mobility

- Multicast/Gathercast Modes
  - Efficient multicast query and aggregated “gathercast” response

- Contextual Message Delivery
  - IoT devices addressable by context (location, state, etc.)

- Pub/Sub Modes
  - Support for both push and pull modes for IoT data/control

Reference: IETF Draft, Feb 2016, Zhang et al, ICN Based Architecture for IoT
IoT Architectural Requirements (cont.)

- **Security and Privacy**
  - Including support for low-end IoT devices

- **Communication Reliability**
  - Application specific (e.g. Health, Transportation)

- **Storage and Caching**
  - In-network storage for DTN and content cache services

- **Self-Organization & Discovery**
  - Ability to self-organize and discover resources (services/devices) and communicate

- **Ad hoc and Infrastructure Modes**
  - Seamless transitions between the two worlds

Reference: IETF Draft, Feb 2016, Zhang et al, ICN Based Architecture for IoT
IoT Architecture Considerations
Legacy IoT Systems

- **Silo IoT Architecture (Fragmented, Proprietary):**
  - e.g. DF-1, MelsecNet (Mitsubishi Electric), SDS (Honeywell), BACnet, Bluetooth Low Energy, etc
State of the Art in IoT Networks

- Overlay Based Unified IoT Solutions (i.e., OpenIoT, AllJoyn)
- Disadvantages
  - Lack of naming transparency between systems which hinders efficient data/service discovery
  - No general purpose network-layer support for essential services such as pub/sub, multicast, mobility and context-aware delivery
Next-Gen (Mobile) Network has the potential to enable “flat” IoT solution which works across multiple applications, services & access technologies.

- Solutions such as information-centric networks (ICN) have been proposed and are currently under evaluation.
A Typical ICN-IoT System

IoT Server
- Names and the service types that are exposed to the server
- Subscription memberships

IoT Aggregator (e.g. Raspberry Pi, Smart Phone, nest thermostat)
- Hardware information for attached sensors
- Service types of the attached sensors
- Sensor names for ICN-enabled sensors
- Information about sensors attached to peer aggregators

Sensors (e.g., RFID, temperature sensors)
- Sensor hardware information
- Sensor names for ICN-enabled sensors

Reference: IETF Draft, Feb 2016, Zhang et al, ICN Based Architecture for IoT
MobilityFirst “Named-Object” Architecture for 5G/IoT
MobilityFirst Architecture: *Key Features*

- **Named devices, content and context**
  - Human-readable name
  - 11001101011100100…0011 Public Key Based Global Identifier (GUID)

- **End-Point mobility with multi-homing**
- **Strong authentication, privacy**
- **In-network content cache**
  -Hop-by-hop file transport
  - Service API with unicast, multi-homing, mcast, anycast, content query, etc.

- **Routers with Integrated Storage & Computing**
- **Storage-aware Intra-domain routing**
- **Edge-aware Inter-domain routing**

- **Heterogeneous Wireless Access**
- **Connectionless Packet Switched Network with hybrid name/address routing**
- **Network Mobility & Disconnected Mode**
  - Ad-hoc p2p mode

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

WINLAB
MF Architecture: GUIDs & Name Resolution

- Globally unique identifiers (GUID) used to define all network-attached objects

- Key design choice: flat identifier vs. hierarchical semantic identifier….

- MobilityFirst uses a flat public key as the GUID

- NDN uses a hierarchical content descriptor
MobilityFirst Protocol Stack

- **NCS**: Name Certification & Assignment Service
- **GNRS**: Global Name Resolution Service
- **MF Routing Control Protocol**: GSTAR Routing, MF Inter-Domain, Hop-by-Hop Block Transfer
- **GUID Service Layer**: Switching Option
- **Socket API**: Optional Compute Layer Plug-In A
- **Link Layers**:
  - Link Layer 1 (802.11)
  - Link Layer 2 (LTE)
  - Link Layer 3 (Ethernet)
  - Link Layer 4 (SONET)
  - Link Layer 5 (etc.)

- **Applications**: App 1, App 2, App 3, App 4

**Control Plane**

**Data Plane**
IoT Services over Named-Object Network

Sensing Functionality (temperature) -> Sensing Service

Actuating Functionality (air conditioning) -> Actuating Service

Name Assignment & Certification Services

Human Readable Names <-> global identifier

Application

Application Service (subscribe, publish, …) -> Application Service

Dynamic binding of GUID <-> network address

Global Name Resolution Service

Network Routing on GUID (MobilityFirst)
MF Example 1: Service Migration

GUID-Based Network

- Listen to G1
- Notify/Respond

Subscribe to temperature service G1
Query current temperature via G1

Listen to G1
Notify/Respond
MF Example 2: Service Caching (Power Saving)

GUID-Based Network

Listen to G1
Respond

Sink node

Subscribe to temperature service G1
Query current temperature via G1

Wake up periodically
Publish to G1, update sink
MF Example 3: Pub/Sub & Multicast

GUID-Based Network

- Subscribe to G1
- Notify user

- Subscribe to G1
- Analyze data

- Publish temp to G1
  Notify everyone who is interested

- Subscribe to G1
  Control air conditioning

- Publish temp to G1
  Notify everyone who is interested
MF Example 4: Supporting Context

- Example: Send message to “Yellow Cabs in New Brunswick”
  - 1. Messages sent to the end-networks that maintain context mappings
  - 2. At end-network, message is late-bound to destinations

```
GUIDc → {N₁, N₂, ..., Nₖ}

send(GUIDc, NA=N₁)
send(GUIDc, NA=N₂)
send(GUIDc, NA=N₃)

GUIDc @Nₖ → {t₁, t₂, ..., tₖ}
```

Global Name Service

![Diagram showing the process of sending messages to different end-networks and the late-bound destinations.](image-url)
MF Example 5: Sensor Data Aggregation

- Get the average temperature
- Multicast query with G1
- MF compute layer for Data aggregation
- Aggregated response packet with average temp
MF IoT Use Case Summary

- IoT: S1 data is picked up by Mobile GW and sent to MF network for A1 that subscribes.
- Context: T1, first_responders@NJ, is subscribed by P1 …P4; A2 sends a file to T1 reaches P1..P4 via GUID-based MF routing.
- IoT/Context server updates MF-GNRS for mappings: S1→ A1 and T1 → P1…P4.

**GNRS Entries:**

- S1 -> A1
- C1 -> NA1
- P1 -> NA2, P2->NA2
- P3 -> NA2, P4->NA3
- A1 -> NA4
MobilityFirst/IoT: Prototype Components

**WSN Gateway (Android)**
- MF Client Protocol Stack
- WSN Access Point Stack
- WSN to MF GW processing

**IoT/Context (naming) Server**
- Definitions and descriptions
- GUID assignment & GNRS update
- Subscription management

**Applications (PC)**
- Lookup a GUID
- Subscribe to GUID
- Send/Rcv data of GUID

**Downloadable User-level App for Android**

**Gateways**
- GUI/settings
- Lookup Sensor GUIDs & description XML
- MF formatting & delivery
- Raw data processing
- Java wrap up API
- Java USB Driver for Wireless Sensor AP
- Sensor Radio (Proprietary now)
- WiFi or 3G/LTE

**Native MF Client Stack**

**GUI (Sensor/Context on Map)**
- Naming Mod (GUID assignment)
- Service Mod (GUID lookup, Subscription)
- GNRS Module (GUID update)
- GW Module (GUID lookup)
- Sensor / Context Management (add/remove resources)

**Sensor/Actuator App (smartGrid)**
- Java Wrapper
- Browser / HTTPclient
- TCP/IP

**Context App (File transport)**

**GNRS**
- Sensor GUID -> App GUID

**MF Router Stack**
- GNRS
- Caching
- Computing
- MF Router Stack

**Native MF Client Stack**
- Apache Server
- mySQL DB
- TCP/IP

**Native MF Client Stack**

**Web services**

Request GUID corresponding to alert multicast group

Insert multicast group and GUID of devices belonging to it

GUID resolution

Sends alert to multicast GUID

Nearby Users (by geographical distance) in specified context group

Busch Campus Outdoor Wireless Deployment

Police Station

MF Context Service For Alerts

Global Name Resolution Service

MF Router
BMS-Evaluation

1. Based on MF-sim and NDN-sim
2. Based on campus building floor plan
Building Management System (BMS)
BMS-Evaluation – NDN vs. MF

1. Average Data Report Delay On Server

2. Average delay from sink to actuator

3. Total PIT size in the network

4. Goodput at the server

Web Sites for More Information:

- WINLAB:  www.winlab.rutgers.edu
- ORBIT:  www.orbit-lab.org
- MobilityFirst:  http://mobilityfirst.winlab.rutgers.edu