Demo Abstract: Motion-Triggered Surveillance Camera using MF-IoT

Jiachen Chen
WINLAB, Rutgers University
North Brunswick, New Jersey 80901
jiachen@winlab.rutgers.edu

Sugang Li
WINLAB, Rutgers University
North Brunswick, New Jersey 80901
sugangli@winlab.rutgers.edu

Yanyong Zhang
WINLAB, Rutgers University
North Brunswick, New Jersey 80901
yyzhang@winlab.rutgers.edu

Dipankar Raychaudhuri
WINLAB, Rutgers University
North Brunswick, New Jersey 80901
ray@winlab.rutgers.edu

ABSTRACT

IoT network requires global reachability, mobility support, richer communication patterns and resource efficiency. To address these needs, MF-IoT has been proposed as an extension of MobilityFirst which allows applications even in the low-end IoT devices to use the aforementioned network capabilities. This demo uses surveillance camera as an example to show the feasibility and efficiency of MF-IoT design. It also demonstrates the flexibility of using service-oriented GUIDs that are supported in MF-IoT.

CCS CONCEPTS

•**Networks** →**Naming and addressing;** *Network layer protocols;*
•**Computer systems organization** →*Sensor networks;*

KEYWORDS

IoT, MobilityFirst, Service-oriented Communication, MF-IoT

ACM Reference format:
Jiachen Chen, Sugang Li, Yanyong Zhang, and Dipankar Raychaudhuri. 2017. Demo Abstract: Motion-Triggered Surveillance Camera using MF-IoT. In *Proceedings of IoTDI '17, Pittsburgh, PA, USA, April 18 - 21 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3054977.3057291

1 INTRODUCTION

The advent of new Internet of Things (IoT) devices has posed challenges to the underlying network design. We envision that the new network for IoT devices should support: *1*) global reach-ability – the devices need to be identified and located from any place in the network, *2*) mobility support – the devices need to have seamless connection even in presence of device mobility, *3*) richer communication patterns – the devices need communication patterns like query/response, pub/sub, anycast, *etc.*, and *4*) resource efficiency – a large proportion of IoT devices are severely constrained in energy, computation, and/or network capacity.

IoTDI '17, Pittsburgh, PA, USA

978-1-4503-4966-6/17/04…$15.00
DOI: http://dx.doi.org/10.1145/3054977.3057291

MobilityFirst (MF [1]) is a clean-slate network architecture that has the potential to satisfy the requirements of the new IoT network. Each network object – a device, an application, or a piece of content – is identified by a persistent 20-byte flat Globally Unique Identifier (GUID), which provides global reach-ability. The mapping between the GUID and the location of a network object is stored in a logically centralized Global Name Resolution Service (GNRS), which is similar to the Domain Name Service (DNS) in function. Yet, unlike DNS which is an application-layer service, GNRS is accessed by the network entities (routers). The network can perform late-binding (GNRS re-lookup) on a delivery failure when a node moves. MF also supports different kinds of communication patterns including multicast (pub/sub), anycast (query/response), *etc.* While MF is well suited for the communication requirements of the IoT network, some challenges still remain, including large GUID size, costly GNRS lookup, and heavy link-state routing.

To address these challenges, MF-IoT [2] is proposed as an extension of MF. Since IoT devices usually use different lower-layer protocols, MF-IoT groups them into different domains. Gateways serve as bridges between IoT domains and the core network (running MF). To meet the constraints in the network, MF-IoT adopts the concept of Locally Unique Identifier (LUID) which is only 2 bytes in length. Each GUID has a deterministic mapping to a LUID in each domain for a period of time. This mapping is managed by the gateways and cached in the devices. However, the LUID is agnostic to the application layer since the applications in MF-IoT (both in the core network and in the IoT domains) still use GUID to address each other. The translations between GUID and LUID that occur in the gateways and the operating system of IoT devices are transparent to the applications.

To further satisfy the requirements of IoT devices, MF-IoT encourages the service-oriented communication. Each service can have a GUID. A service provider listens to a GUID and the consumers can query or send data to that GUID. Multiple providers of a same service can listen to a single GUID and use the anycast/multicast function provided by the network. The providers can also shift the duty based on time, policy, or user requirements.

In this demo, we present a motion-triggered surveillance camera application to show the feasibility and efficiency of MF-IoT. The service-oriented communication is also exploited here to demonstrate the flexibility of dynamic change of service providers.

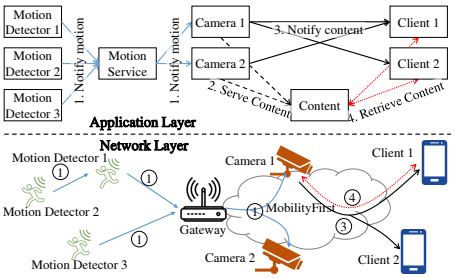

Figure 1: Communication diagram in different views

Figure 2: Screenshot of Client 2

2 SYSTEM ARCHITECTURE

The demo architecture contains different modules including IoT network, IoT applications, gateway, MF network, camera applications and client applications. In Fig. 1, we show both the logical relationship among GUIDs (application layer) and how the packets are routed in the physical network (network layer).

We have implemented the MF-IoT protocol on Atmel SAM R21 XPro with RIOT OS. Each board is used as a micro-controller and network adapter for a PIR sensor, or as a relay. The MF-IoT is implemented as a network layer module in RIOT similar to 6LowPAN. The module listens to the normal MF send packets from the application layer, performs GUID to LUID mapping and send MF-IoT packets over 802.15.4. On receiving MF-IoT packets, the module would lookup the local routing table and forward it to neighbor(s). The module would also map the LUID to GUID and forward a MF packet to the applications listening to the GUID.

The gateway is implemented as an IoT board connected to a PC via USB. The gateway application on the board reads packets from MF-IoT module and prints the binaries to the debug console. The PC reads the debug output via the USB and reconstructs MF packets accordingly. The MF packets would be sent to the core network and reach the destination(s) listening to the dst GUID. The gateway also manages the GUID-LUID mapping in the domain.

The application running on motion detector node listens to the motion sensor via UART. During the period that motion is detected, the applications send MF messages periodically to a *GUID* representing Motion Service (step 1 "Notify Motion"). The MF-IoT module performs the GUID to LUID translation and forwards the packets based on the routing. The packets will be relayed on the intermediate nodes and reach the gateway eventually. They will be translated back to MF packets on the gateway and be forwarded to the application(s) listening on this service (cameras). With the motion service GUID, the cameras do not need to change their reconfigurations when a new motion detector is added to the system.

On receiving the notification from the motion detectors, the cameras would start recording the video until no notification is received for a timeout period. The cameras would combine a certain number of frames into a chunk and get a content GUID from Name Certification Service (NCS) for each chunk. After saving the chunk, the camera would notify GNRS that it is serving the chunk GUID (step 2 "Serve content" in Fig. 1). The camera would then send a notification to the GUID of the camera service. The content GUID is placed in the payload of the notification. Whoever is interested in (or has the right to receive) the camera data would listen to the camera service GUID and get the corresponding notifications (step 3 "Notify content" in Fig. 1). Note that we allow different cameras to create different services to enforce the policy and/or satisfy user interests. In the figure, Client 1 is interested in both cameras while Client 2 is only allowed to see Camera 2. Therefore, Client 1 listens to GUIDs of both cameras and Client 2 only listens to GUID of Camera 2.

When a client receives the notifications from the camera service(s), he would look into the payload and get the GUID of the content. He can then query the MF network with the content GUID whenever he wants to watch the captured video (step 4 "Retrieve content" in Fig. 1). Similar to other Information-Centric Networks (ICNs), MF would route the request to the nearest content provider or even get the content from the cache in the network. The screenshot of Client 2 is shown in Fig. 2.

3 DEMO SCENARIOS

We would use the normal surveillance camera function to show the feasibility and efficiency of the design. During the demo, we would also dynamically adjust the network to demonstrate the flexibility of using service-oriented communication.

Scenario 1 *Normal Surveillance Camera Function:* We use 3 motion detectors and several relays to form IoT domain (Fig. 1). Due to the space limit in the demo site, we will pre-configure a virtual topology so that the IoT nodes would only accept the packets from neighbors. The core MF network comprises 2 MF routers and several end-hosts (cameras and clients). We would show that when any of the motion sensors detect the motion, messages will be sent (via relays) to the gateway using MF-IoT over 802.15.4. A sniffer node would be placed to show how the packets and sent among the IoT devices. On the gateway, we would capture both the MF and MF-IoT packets to explain how the translation is performed. The cameras would get the messages via MF multicast and generate contents with the images they capture. Via the log on the cameras, we would show the messages the cameras receive, the new content GUIDs they create and the requests form for the contents. On the client side, we would allow the clients see the video immediately when the video is generated (real-time mode). The clients can also choose to play the video recorded earlier (playback mode).

Scenario 2 *Dynamic Configuration Adjustment:* In the demo, we use motion service and camera service GUIDs. When we add a new motion detector or use another camera to replace the existing one, no configuration would be needed. We will demonstrate how the network enables the automatic adjustment.

REFERENCES

[1] D. Raychaudhuri *et al.*, "MobilityFirst: A Robust and Trustworthy Mobility-Centric Architecture for The Future Internet," *SIGMOBILE*, vol. 16, no. 3, pp. 2–13, 2012.

[2] S. Li *et al.*, "MF-IoT: A MobilityFirst-Based Internet of Things Architecture with Global Reach-ability and Communication Diversity," in *IoTDI*, 2016.