Exploiting ICN for Realizing Service-Oriented Communication in IoT

Information-Centric Networking shifts the focus from location to the (identity of) information. This paradigm can naturally be adapted to IoT communication since service can also be identified as a type of information. To demonstrate the potential of ICN in IoT communication, the authors adopt and modify a particular example of ICN called MobilityFirst, and show that the target architecture can satisfy the requirements posed by IoT communication.

Jiachen Chen, Sugang Li, Haoyang Yu, Yanyong Zhang, Dipankar Raychaudhuri, Ravishankar Ravindran, Hongju Gao, Lijun Dong, Guoqiang Wang, and Hang Liu

Jiachen Chen, Sugang Li, Haoyang Yu, Yanyong Zhang, and Dipankar Raychaudhuri are with Rutgers University.

Ravishankar Ravindran, Lijun Dong, and G. Wang are with Huawei Research Center.

Hongju Gao is with China Agricultural University.

Hang Liu is with The Catholic University of America.

Digital Object Identifier: 10.1109/MCOM.2016.1600405CM

COMMUNICATIONS STANDARDS

Abstract

The rapid growth in Internet of Things (IoT) deployment has posed unprecedented challenges to the underlying network design. We envision that tomorrow's global-scale IoT systems will focus on service-oriented data sharing and processing rather than point-to-point data collection. Requirements such as global reachability, mobility, communication diversity, and security will also develop naturally along with the change in the communication patterns. However, existing (IP-based) networks focus only on locations and point-to-point channels. The mismatch between the dynamic requirements and the functionalities provided by the network renders IoT communication inefficient and inconvenient.

Information-centric networking (ICN) shifts the focus from location to the (identity of) information. This paradigm can naturally be adapted to IoT communication since service can also be identified as a type of information. To demonstrate the potential of ICN in IoT communication, this article adopts and modifies a particular example of ICN called MobilityFirst, and shows that the target architecture can satisfy the requirements posed by IoT communication. Similar adaptations can also be used by other ICN architectures such as NDN and XIA.

Introduction

The use of IoT networks has evolved from simple data collection to service-oriented data sharing and processing. The data providers and consumers focus more on what is provided rather than who is using or providing the data. For example, an e-health app only needs to get the step count of the user, without worrying whether the count is provided by a Fitbit or a smartphone (via the built-in accelerometer). At the same time, the step count provider does not have to worry if the data is used by a personal e-health app or a social network app. Figure 1 depicts the service relationship between several example applications. The relationship among services will become more sophisticated when the IoT devices have more functionalities and the function partitioning among services becomes more fine-grained.

Such a service-centered IoT system demands an underlying network architecture that can reliably deliver data to the consumer (be it service or App) in the case of mobility and other dynamic scenarios. However, the existing (IP-based) networks [1, 2] only focus on the endpoints that are involved in data communication, and are hence inherently ill-suited to support a service-centered architecture. For example, with the current network, when a user intends to switch the step count service from a Fitbit to a smartphone, all of the consumer's apps have to be affected since they need to know the new address of the data provider. Such a switch may cause a large amount of wasteful traffic in the network and more importantly, disrupt the service at the user side; on the day the user forgets to wear their Fitbit, it will be annoying since they need to switch the pairing device to the smartphone for all the related apps, which might reside on the smartphone, on a server, or even in the cloud.

The information-centric networking (ICN) paradigm, as proposed in MobilityFirst [3], NDN [4], and XIA [5], treats information as a first-class citizen. Each entity, no matter if it is a device, an application, or even a piece of content, can have a persistent routable identity in the network. The provider and consumer can dynamically bind services and communicate based on the identity of information rather than the identity of communicating parties. In ICN, the user's step count service would have a unique identity in the network. In order to query this service, the consumer (e.g., the e-Health app) can set the destination to be the step count service's ID, and the network would route the query toward the provider (e.g., Fitbit or smartphone) that is currently providing the service. When switching the service provider, the user only needs to notify the network once about the new provider without affecting any of the apps. The separation between the provider and consumer has the potential to greatly reduce the amount of management and messaging overhead required in future IoT systems.

To realize such a service-oriented IoT architecture is a challenging task, as such an architecture imposes the following requirements on the underlying networking layer [6]:

- Global reachability: Services need to reach each other no matter where they are located.
- Mobility: IoT devices (e.g., wearables) tend to move during a communication session, and sometimes, even if the devices are not moving, the service can migrate from one device to another.
- Communication diversity: Other than the basic query/response type of communication, notification (pushing or multicast) has become another popular communication model since more complicated logic can now be implemented on IoT devices, and they will only generate network traffic when the predefined conditions are detected. It is desired that the network can provide native

support for these communication paradigms in order to reduce the network load and the complexity in the application;
- Security: When the network has the capability to deliver content among services, data integrity and privacy become major concerns since IoT devices tend to generate sensitive data. We also note that IoT devices exhibit great heterogeneity in their capabilities. Many IoT devices have limited computation (≤50MHz), memory (≤50kB), storage (≤300kB), and/or transmission capability (MTU 128B), which will make it even more challenging to satisfy the above requirements.

This article describes how ICN can be leveraged and adapted to support the above-mentioned requirements in service-oriented IoT communications. More particularly, an instance of ICN, i.e., MobilityFirst, is used as an example in the remainder of the article. However, we believe that the design can be easily adopted to other ICN solutions such as NDN and XIA.

MobilityFirst: ICN with Mobility Support

MobilityFirst [3] is proposed as a future Internet architecture with mobility and global accessibility as core design concerns. It has the potential to become the network for service-based IoT communications. To achieve these features, MobilityFirst introduced several components into the network:

Globally Unique IDentifier (GUID): MobilityFirst utilizes persistent GUIDs to name every network object. The separation between the identifier (GUID) and the locator (network address (NA)) provides support for mobility and global accessibility. Moreover, a GUID can be a public key derived from the properties of the object or a human-readable name, hence it allows the objects to be self-certifiable.

Global Name Resolution Service (GNRS): GNRS is a logically centralized service that maintains the mapping from the GUID of an object to its current NA(s). MobilityFirst routers can perform late binding, i.e., querying the GNRS whenever a destination NA could not be resolved in the local scope. This is a network-layer service that is different from DNS, and it provides better support for mobility since the network has the potential to recover a delivery failure locally. For example, in Fig. 2, after the receiver (with GUID G_R) moves from N_R to N'_R, the router at NR would redirect the packet(s) toward the new destination without affecting the sender. Work in [7, 8] proposed distributed solutions for GNRS implementations that can have acceptable scalability and lookup performance in the core network.

Routing: MobilityFirst routes packets based on the NA(s). Work in [9] proposed a basic intra-domain routing solution in MobilityFirst similar to open shortest path first (OSPF). In this solution, each router maintains the global topology and calculates the shortest path to the destination in a distributed manner. For inter-domain routing, BGP-like solutions can be adopted.

Service ID (SID): To support multiple network services such as unicast, multicast, and in-network computing, a (network) service ID is included in the packet header so that each router is capable of making decisions based on its policy.

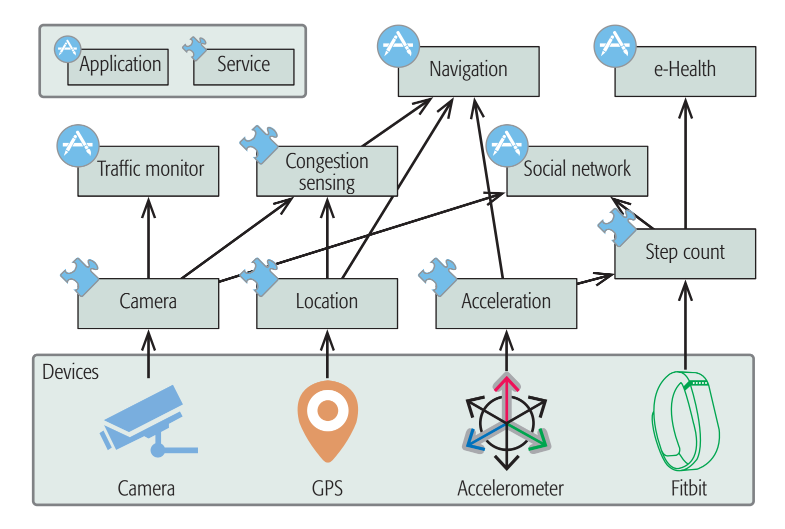

Figure 1. Rich service relationship in a set of real-world applications. Services (e.g., Congestion Sensing) can be both consumers and providers. One service (e.g., Step Count) can also be provided by multiple sensors/services.

Based on these components, MobilityFirst has the potential to satisfy the requirements of IoT such as mobility and communication diversity. However, a number of challenges remain for the deployment of MobilityFirst in IoT systems, such as:

Service-Oriented Communication: Although MobilityFirst can use GUIDs to denote any object in the network, it still needs clarification how the service-oriented communication can be supported with dynamicity and flexibility.

Resource Constrained Devices: To ensure global uniqueness, MobilityFirst uses 20-byte strings as GUIDs, which brings significant overhead in low data-rate networks (e.g., IEEE 802.15.4). Functions such as GNRS lookup, store-and-forward mechanism on each hop, and link-state routing are also too heavy for low-end IoT devices with limited resources.

Security: Self-certifying GUIDs can ensure data integrity and hence prevent in-network cache pollution attacks similar to the attack model described in [10]. However, it is not clear how to prevent privacy leaks [11] and detect malicious data providers in IoT scenarios.

To mitigate the gap between the requirements of IoT and the functionalities provided by MobilityFirst, we propose an adaptation for IoT communication called MF-IoT.

MF-IoT: Leveraging and Adapting MobilityFirst for IoT Communication

In this section, we first describe how we use MobilityFirst to provide basic service-oriented communication. We then take resource heterogeneity, communication diversity, and security into consideration and discuss how to adapt MobilityFirst to satisfy the relevant requirements.

Service-Oriented Communication

Since MobilityFirst uses GUID as the unified routing label in the network (similar to an IP address in the current Internet), it is straightforward to use

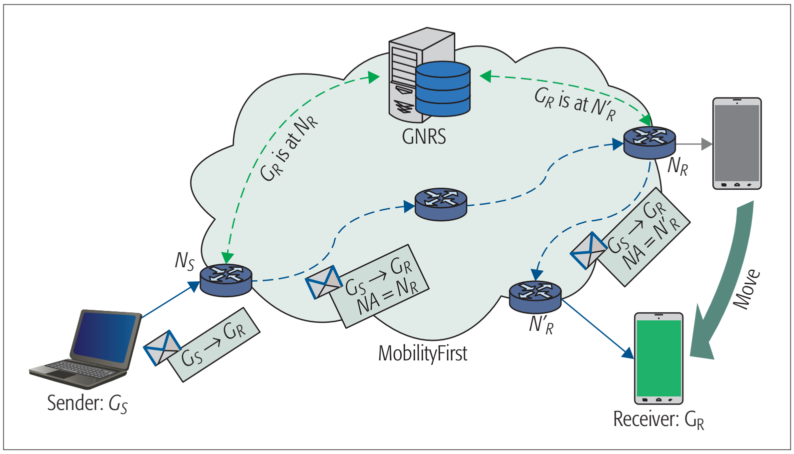

Figure 2. Late binding in MobilityFirst. Last hop (RS) performs GNRS lookup for the message after receiver (GR) moves away. The sender can simply "send and forget."

GUID to represent IoT services. We can grant each service (e.g., Alice's step count) a GUID (e.g., $G_{A.sc}$). The sensors/program/node (Fitbit) that is providing the service registers the service GUID together with its own NA (e.g., $G_{A.sc} \mapsto NA_{A.Fitbit}$). To send a message/request to the current instance of a service, the data consumer only needs to address the instance with the service GUID (e.g., Alice's e-health App can send a request with $src = G_{A.eHealth}$ and $dst = G_{A.sc}$).

Unlike IP, MobilityFirst separates the identity (GUID) from the location (NA). The benefit of this design is that, when an object moves from one location to another, its GUID does not change. In the IoT scenario, since our design treats each service as an object (with a GUID), the migration of a service instance can be viewed as the service moving from one place to another. Although the location (NA) of the service changes, the GUID (which is used by the consumers) can be kept the same. In Fig. 3, when Alice's smartphone is responsible for her step count service ($G_{A.sc}$), it registers $G_{A.sc}$ with its current location $NA_{A.phone}$, and therefore all the queries to $G_{A.sc}$ will be routed to the smartphone automatically. The transition to a new provider at a new NA is totally transparent to the consumers, e.g., e-health and social media apps.

With the GNRS lookup as a network-layer function, the new mapping can take effect immediately, even if the access routers of the consumers (R_s and R_e) are not updated or even if there are still packets in transit. The packets with old mapping ($G_{A.sc} \mapsto$ NAA:Fitbit) will be forwarded to the access router of Alice's Fitbit (RF). Since the Fitbit has detached itself from $G_{A.sc}$, there is no directly-linked device at the RF that is serving the GUID. In this situation, R_F would send a GNRS lookup to find the new location of $G_{A.sc}$. When the GNRS responds with the new mapping, the RF would forward the packet toward $NA_{A.phone}$. After a period of time, the mapping states on R_s and R_e will be updated and triangular routing will be avoided. We can see that the whole process is also transparent to the data consumers, so that they can simply transmit and assume that the network can take care of the packet. Such a communication model can be particularly useful to IoT devices with power constraints since they do not have to keep alive and wait for an ACK signal from the recipient node.

Dealing with Resource Heterogeneity

It is common in IoT that devices with heterogeneous resources communicate with each other, e.g., a motion detector with small memory and limited power might need to notify (using IEEE 802.15.4) a smartphone that communicates over WiFi and cellular networks with other services. It is quite infeasible to run full-fledged MobilityFirst on the motion detector due to the length of the MobilityFirst packet header (>100B), the costly GNRS lookup procedure, and the need for dynamic link-state routing. Therefore, to support the functionalities of MobilityFirst across all devices in IoT, we need to lighten the protocol when applying it to resource constrained devices. In MF-IoT, we made the following changes for resource constrained devices:[1]

Resource Constrained Domains: Resource constrained devices tend to use energy-efficient link-layer protocols such as IEEE 802.15.4 (LRPAN) or Bluetooth Low Energy (BLE). These protocols usually have a much smaller maximum transmission unit (MTU) compared to Ethernet and WiFi. Therefore, the MF-IoT network layer packets (headers) need to be compressed to fit into the MTU.

To allow efficient communication among constrained devices, we need to group these devices into constrained domains. In each domain, the same link-layer protocol and the lightweight MobilityFirst network-layer protocol can be used so that the devices can send/receive messages directly. To enable global reachability, we need to ensure that the devices in different domains and those in the core network can communicate with each other. Similar to IP network designs, gateways should be used at the border of each domain to translate between the lightweight and the normal MobilityFirst protocols. However, different from IP network address translation (NAT), the gateways in MF-IoT do not modify the semantics (source, destination, etc.) of the packets. Instead, they perform translation between MobilityFirst packets and their equivalent lightweight MobilityFirst packets. We ensure that the translation is transparent to programs (applications, middleware), and even to transport protocols using MF-IoT. They only see the MobilityFirst packets and therefore can address each other via GUID. Global reachability is therefore retained.

GUID vs. LUID: The 20-byte GUID is a key element in ensuring global uniqueness in MobilityFirst network. However, it is also the cause of the large packet header. Now that we have divided the constrained devices into different domains, much shorter identities can be used as substitutions for the GUIDs used within the domain. As long as the short identities (referred to as local unique identities, or LUIDs) have a one-to-one mapping to the GUIDs, the gateway can perform proper translation between the lightweight protocol and MobilityFirst. Similar substitutions can be performed on fields such as SID, version, etc. To ensure the local uniqueness over long timespans, revocation of GUID–LUID mapping needs to be enabled. Each domain can have its own revocation policy,

[1] For the design detail of MF-IoT, please refer to [12].

based either on usage (e.g., LRU) or on timeout (e.g., TTL). We place the management of these mappings on the gateways since they usually have larger storage, more computation resources, and a larger power supply.

Table 1 depicts the packet format in the lightweight protocol. To minimize the packet size, we used fixed-length packet headers instead of the type-length-value (TLV) format. Fields such as version (`VER`), packet type (`PKT_TYP`), and SID (`SVC_TYP`) are shortened to four bits and the source (`SRC_LUID`) and destination (`DST_LUID`) only consume two bytes each. The total size of the lightweight header is reduced to just 10 bytes and can thus allow up to a 116-byte payload even with IEEE 802.15.4.

Forwarding: When a constrained device joins a domain, it registers its GUIDs (each service listens to a GUID) at the gateway, and the MF-IoT module in the device keeps the LUID mapping for each GUID.

When an external application sends a message to a certain GUID (G), it sends the MF-IoT module a full sized MobilityFirst packet (the translation is transparent to the applications). The MF-IoT module would request G's LUID from the gateway. The gateway will create an entry (GUID = G, LUID = L) if there is no entry of G in its mapping table. The generation of L for G can be random or hash-based with reuse-avoidance. Note that at this stage, the gateway does not need to perform a GNRS lookup so that it can acknowledge the request immediately. On getting the response, the MF-IoT module can compress the packet and send it to the next hop based on the local routing.

Upon arrival of a MF-IoT packet (either from a normal node or from another gateway), a gateway would look up its mapping table and obtain the GUIDs for both source and destination and then forward the packet using MobilityFirst logic. At this point, it might need to look up the GNRS for the destination's NA if it is unknown. On the other hand, when the gateway receives a MobilityFirst packet whose destination GUID (G_d) is in its domain, the gateway would create a LUID (L_s) for the source GUID (G_s) then send a lightweight packet consisting of L_s and L_d. This entry is created so that the destination device can send a message to the sender.

Figure 4 depicts three scenarios where a constrained node (n_1) wants to send a message to a node (n_2) in the same domain, an infrastructure node (n_3), or a constrained node (n_4) in another domain. In Fig. 4, traffic in constrained domains is represented by green lines and MobilityFirst traffic is represented by blue lines. Note that we use dotted lines here to denote that the traffic is not necessarily direct traffic between the two nodes, because there might be relay nodes between them. We next describe the protocol exchange according to the labels in the figure.

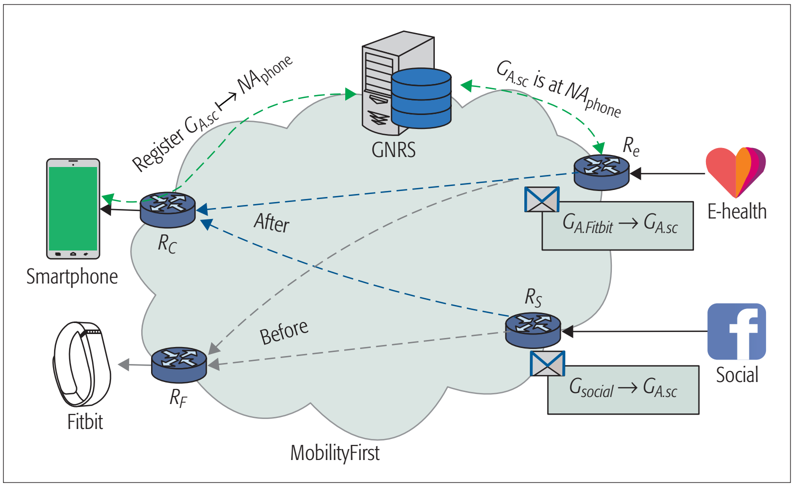
Register $G_{A:sc} \longmapsto NA_{phone}$
GNRS
$G_{A:sc}$ is at NA_{phone}
R_e
E-health
$G_{A:Fitbit} \rightarrow G_{A:sc}$
Smartphone
R_C
After
R_S
Social
$G_{social} \rightarrow G_{A:sc}$
Before
Fitbit
R_F
MobilityFirst

Figure 3. Service migration in MF-IoT (Alice registers her smartphone ($NA_{A:phone}$) as the provider of her step count service ($G_{A:sc}$) when she forgets to wear her Fitbit.

- To initiate the communication with n_2, n_3, and n_4, n_1's forwarding module needs to first get their LUIDs from the gateway. For n_2, GW_1 can respond directly since it has an entry in the translation table. For the other two nodes, GW_1 creates new LUID entries.
- The routing algorithm in the constrained domain forwards the packet based on the destination LUID. Since n_2 is in the same domain, the local routing algorithm would forward the packet to n_2 eventually.
- If the destination LUID (Ln_3 or Ln_4) is not in the same domain, the local routing algorithm forwards the packet to $GW1$, which translates the packet to MobilityFirst packets $\{G_{n1} \rightarrow G_{n3}\}$ or $\{G_{n1} \rightarrow G_{n4}\}$.
- Next GW_1 sends the packets with traditional MobilityFirst logic. In MobilityFirst, the first step is a GNRS lookup for the NA of the destination. If the destination is a constrained node in another domain (n_4), GNRS would reply with the NA of the corresponding gateway (GW_2). For a normal node (n_3), GNRS would respond directly with its NA (not shown in the figure).
- After getting the NA, the packet will be forwarded in the core network and eventually reach n_3 or GW_2. Note that thanks to the late-binding technique in MobilityFirst, the packet would reach the destination even if n_3 or GW_2 has moved and has a new NA. This provides seamless mobility support when nodes move.

<table>
<tr><td colspan="2">0 1 2 3 4 5 6 7</td><td colspan="2">0 1 2 3 4 5 6 7</td><td colspan="2">0 1 2 3 4 5 6 7</td><td colspan="2">0 1 2 3 4 5 6 7</td></tr>
<tr><td>VER</td><td>PKT_TYP</td><td>SVC_TYP</td><td>PROT</td><td>TTL</td><td colspan="3">PKT_LENGTH</td></tr>
<tr><td colspan="4">SRC_LUID</td><td colspan="4">DST_LUID</td></tr>
<tr><td colspan="4">NONCE</td><td colspan="4"></td></tr>
<tr><td colspan="8">PAYLOAD</td></tr>
</table>

Table 1. MF-IoT packet format (4 octets per row).

With the advent of software defined networking, the use of a central controller eases traffic engineering and policy enforcement in the network. This concept can also be used in IoT communications. The sensors can report the link changes to the gateway and get forwarding rules either proactively or on demand.

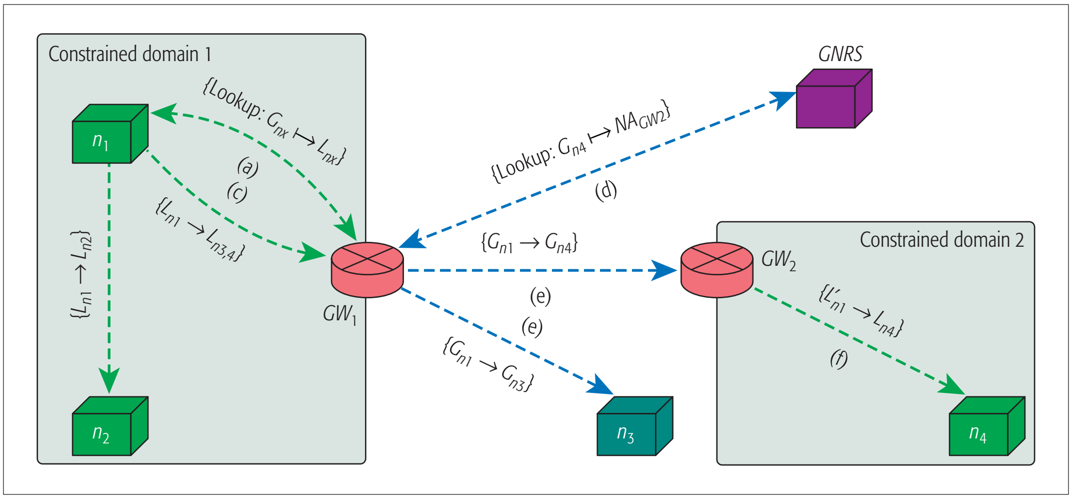
Constrained domain 1
n_1
$\{Lookup: G_{nx} \mapsto L_{nx}\}$
(a)
(c)
$\{L_{n1} \rightarrow L_{n3,4}\}$
$\{L_{n1} \rightarrow L_{n2}\}$
n_2
GW_1
GNRS
$\{Lookup: G_{n4} \mapsto NA_{GW2}\}$
(d)
$\{G_{n1} \rightarrow G_{n4}\}$
(e)
(e)
$\{G_{n1} \rightarrow G_{n3}\}$
n_3
Constrained domain 2
GW_2
$\{L'_{n1} \rightarrow L_{n4}\}$
(f)
n_4

Figure 4. Illustrations for global reachability in MF-IoT. (blue: traffic in the core network, green: traffic within constrained domains).

•When GW_2 receives the packet destined for G_{n4}, it checks the translation table and finds that n_4 belongs to the local domain. It then creates a LUID mapping for G_{n1} (L'_{n1}) and forwards the compressed packet $L'_{n1} \rightarrow L_{n4}$. Note that the LUID of G_{n1} in this domain does not have to be the same as L_{n1} given by GW_1. However, this new LUID does not affect the communication between the applications on n_1 and n_4 since they are communicating with the GUID, whereas the LUID is kept hidden from them.

Routing: Routing should also be simplified in the constrained domains to reduce the memory and power requirements. However, MF-IoT does not restrict the exact routing mechanism adopted in these domains. The application designers can feel free to use any existing routing mechanism or design their own according to the communication pattern they envision. Here, we suggest several mechanisms that could be appropriate:

RPL [13] is widely used in the existing IoT systems. The solution builds a tree among the nodes and usually the gateway is seen as the root. The solution benefits the applications that mainly depend on sensor to gateway and sensor to infrastructure communication paths, since all the traffic has to go through the gateway.

AODV [14] is used by Zigbee as the default routing. It provides on-demand distance vector routing to accelerate the direct sensor-to-sensor communication (they do not need to go to the root of the tree as in RPL). However, to find a path on demand, a request has to be flooded in the whole network, which makes the solution less efficient when the network is large or lossy.

With the advent of software defined networking (SDN), the use of a central controller eases traffic engineering and policy enforcement in the network. This concept can also be used in IoT communications. The sensors can report the link changes to the gateway and get forwarding rules either proactively or on demand. This solution has the potential to reduce the amount of flooding, and supports efficient sensor-to-sensor communication. At the same time, the constrained devices do not have to calculate the routing, thus reducing their storage and power consumption needs.

Communication Diversity

As described earlier, MF-IoT can support direct communication among constrained nodes (both intra-domain and inter-domain) and the communication between constrained and normal nodes. Here, we describe additional communication patterns that are supported in MF-IoT.

Multicast: Since we use service-based GUIDs that are independent of any specific node, every node in the same constrained domain can listen to an identical service GUID. Therefore, multicast can be supported naturally in MF-IoT. Hence, we lump unicast and multicast together and refer to them as a *to-all* service. The forwarding module on the branching point would have more than one entry in the FIB for a LUID if there is more than one receiver. It then replicates the packet and sends a copy to each next hop (either another node or an application on the same node). MF-IoT also takes advantage of any broadcast media that all the wireless nodes are using. When the number of next hop nodes is larger than a (case-dependent) threshold, a node can broadcast the packet instead of replicating and sending the packet multiple times. The next hop nodes will look up their FIB and discard the packet if no matching entry is found. Please note that although it is meant to be sent to multiple receivers, a multicast packet (via broadcast) will consume a similar amount of energy on the nearby nodes compared to a unicast packet, since in the wireless environment, they will anyway receive the packet and discard it if they are not supposed to receive it.

Anycast: In addition to unicast and multicast, MF-IoT also supports anycast. The listeners in anycast work in the same way as in multicast: they would listen to the same GUID and a tree would be formed by the routing protocol either proactively (e.g., OSPF-like) or reactively (e.g., AODV-like). When sending an anycast packet, the sender would place a different `SVC_TYP` value in the packet header and the intermediate nodes would only forward it to one of the next hop nodes based on its policy (e.g., shortest path, or to a node with the longest battery life, etc.).

"Observe" Mode: According to [15], the observe mode is important for IoT applications. In this

mode, the observer registers a specific event at a sensor, and when the event is detected, the sensor notifies the observer. Usually one registration in the observe mode can result in multiple later notifications from a single sensor.

The observe mode can also be supported in MF-IoT, and furthermore, we can provide additional mobility handling and multicast support. The observers (either in the same constrained domain, in the core network, or even in different constrained domains) can listen to the same specific GUID. When an event is triggered, the subject can efficiently send the notification to all the receivers through multicast. With the mobility support and an inherent push model, the solution allows the notifications to be sent in a timely and efficient manner.

Privacy and Trust

Since ICN is used here as the network for IoT, the security and privacy can now shift the focus from securing a channel (as in IP) to securing the contents (or services). With such a security paradigm, the network can send the same (encrypted) piece of message to multiple consumers. The article shows that the target design can support service oriented communication with global reachability, mobility, communication diversity, and security on IoT devices with heterogeneous resource constraints. Similar adaptations can be used by other ICN architectures such as NDN and XIA.

The requirements identified in this article have been proposed as an IRTF ICNRG draft [6]. In future work, we plan to standardize in the IRTF/IETF the service-oriented communication paradigm and mechanisms via multicast (for efficiency), and only the authorized ones can decrypt the content. The signature of the content can ensure the data integrity, and the key that is used to sign the data can also be used to validate whether a provider has the right to serve the (content/service) identity.

Due to the varying security requirements across applications, MF-IoT leaves the security component within the application layer. Each application can choose its desired security model without affecting the others. Thanks to the service-oriented communication model, MF-IoT also makes it possible to secure the communication based on services. Figure 5 shows how MF-IoT preserves privacy and trust for Alice's step count service.

MF-IoT can reject malicious data providers based on the chain of trust [16]. Each device (e.g., Alice's smartphone and Fitbit) that can serve as a provider for Alice's step count service would get a key ($K_{A.sc/A.phone}$ and $K_{A.sc/A.F}$) signed by the key of the service ($K_{A.sc}$). Each provider would use its key to sign the payload. The receiver and/or the network can validate the eligibility of the provider by checking if the key used for the signature is actually signed by $K_{A.sc}$. The signature can also be used to ensure the data integrity.

Attribute-based encryption (ABE [17]) can be used to preserve privacy in MF-IoT. Each service has its own attribute in ABE (e.g., $\text{Attr}_{A.sc}$). All the messages sent to Alice's step count service would be encrypted by this attribute. Only the eligible receivers will get the key with attribute $\text{Attr}_{A.sc}$, and therefore they can decrypt the message.

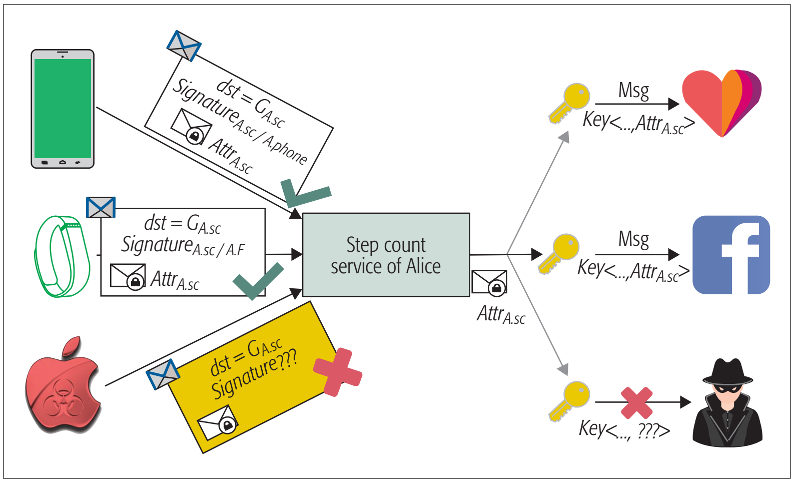

Figure 5. Service-oriented privacy and trust in MF-IoT.

We acknowledge that the above ABE mechanism has relative high requirements for computation, storage, and transmission resources, especially for the constrained devices. To reduce the computation and transmission, the communication parties can alternatively use the proposed ABE mechanism to exchange symmetric keys and use the symmetric keys to secure the communication; or else set up intermediate nodes (e.g., firewalls) to validate the messages for IoT devices.

Conclusion

This article identified the need for service-oriented communication in IoT and the benefits of using ICN to support the communication model. Through leveraging and adapting a particular example of ICN (MobilityFirst), the article shows that the target design can support service oriented communication with global reachability, mobility, communication diversity, and security on IoT devices with heterogeneous resources constraints. Similar adaptations can be used by other ICN architectures such as NDN and XIA. The requirements identified in this article have been proposed as an IRTF ICNRG draft [6]. In future work, we plan to standardize in the IRTF/IETF the service-oriented communication paradigm and mechanisms.

References

[1] G. Montenegro *et al.*, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944 (Proposed Standard), Internet Engineering Task Force, Sep. 2007, updated by RFCs 6282, 6775.

[2] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," RFC 7252 (Proposed Standard), Internet Engineering Task Force, June 2014.

[3] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, "MobilityFirst: A Robust and Trustworthy Mobility-Centric Architecture for the Future Internet," *ACM SIGMOBILE Mobile Computing and Commun. Rev.*, vol. 16, no. 3, pp. 2–13, 2012.

[4] V. Jacobson *et al.*, "Networking Named Content," Proc. 5th Int'l. Conf. Emerging Networking Experiments and Technologies (CoNext), ACM, 2009, pp. 1–12.

[5] A. Anand *et al.*, "XIA: An Architecture for an Evolvable and Trustworthy Internet," *Proc. 10th ACM Wksp. Hot Topics in Networks (HotNets)*, 2011.

[6] Y. Zhang *et al.*, "ICN based Architecture for IoT," IETF Internet Draft draft-zhang-ioticn-architecture-00. IRTF, Tech. Rep., 2013.

[7] T. Vu *et al.*, "DMap: A Shared Hosting Scheme for Dynamic

Identifier to Locator Mappings in the Global Internet," *IEEE 32nd Int'l. Conf. Distributed Computing Systems (ICDCS)*, 2012, pp. 698–707.

[8] A. Sharma *et al.*, "A Global Name Service for A Highly Mobile Internetwork," *Proc. 2014 ACM Conf. SIGCOMM. ACM*, 2014, pp. 247–58.

[9] S. C. Nelson, G. Bhanage, and D. Raychaudhuri, "GSTAR:-Generalized Storage-Aware Routing for MobilityFirst in the Future Mobile Internet," *Proc. 6th Int'l. Wksp. MobiArch., ACM*, 2011, pp. 19–24.

[10] M. Xie, I. Widjaja, and H. Wang, "Enhancing Cache Robustness for Content-Centric Networking," *Proc. IEEE INFOCOM*, 2012.

[11] J. Gubbi *et al*, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, vol. 29, 2013, no. 7, pp. 1645–60.

[12] S. Li *et al*, "MF-IoT: A MobilityFirst-Based Internet of Things Architecture with Global Reach-ability and Communication Diversity," *1st IEEE Int'l. Conf. Internet-of-Things Design and Implementation (IoTDI)*, 2016.

[13] T. Winter *et al*, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550 (Proposed Standard), Internet Engineering Task Force, Mar. 2012.

[14] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561 (Experimental), Internet Engineering Task Force, July 2003.

[15] K. Hartke, "Observing Resources in the Constrained Application Protocol (CoAP)," RFC 7641 (Proposed Standard), Internet Engineering Task Force, Sept. 2015.

[16] "Chain of Trust," https://en.wikipedia.org/wiki/Chain of trust, accessed on 2016.09.05.

[17] "Attribute-Based Encryption," https://en.wikipedia.org/wiki/Attributebased encryption, accessed on 2016.09.05.

Biographies

Jiachen Chen [M] has been a postdoc at WINLAB/ECE, Rutgers University since 2015. Before joining Rutgers, he received his doctor degree from the Institute of Computer Science, University of Göttingen, Germany. His current research includes information-centric networking (ICN), Internet of Things (IoT), cloud computing, and network management.

Sugang Li is a Ph.D. candidate at WINLAB/ECE, Rutgers University, working with Prof. Yanyong Zhang. Before coming to Rutgers, he studied biomedical engineering at Southern Medical University as an undergraduate student from 2007 to 2011. His research interest is focused on mobile computing, machine learning, and Internet of Things (IoT) over information centric networks (ICNs).

Haoyang Yu has been a technology analyst at Goldman Sachs since February, 2016 focused on the cloud platform. He received his master degree from WINLAB/ECE, Rutgers University in January 2016, and a bachelor degree in electronic engineering from the University of Electronic Science and Technology of China in July, 2014. His working areas are databases, middleware services, and trading systems in the financial industry.

Yanyong Zhang [M] is currently a professor in the Electrical and Computer Engineering Department at Rutgers University. She is also a member of the Wireless Information Networks Laboratory (Winlab). Before joining Rutgers at 2002, she earned her Ph.D. at Penn State University. Her current research interests are in future Internet and pervasive computing. She is a member of ACM, and an associate editor for *IEEE Transactions on Mobile Computing and Service Computing*. She has published more than 100 technical papers in various international conferences and journals.

Dipankar Raychaudhuri [F] is a distinguished professor, electrical & computer engineering, and the Director of WINLAB (Wireless Information Network Lab) at Rutgers University. As WINLAB's Director, he is responsible for an internationally recognized industry-university research center specializing in wireless technology. He is also PI for several large U.S. National Science Foundation funded projects including the "ORBIT" wireless testbed and the MobilityFirst future Internet architecture. He has previously held corporate R&D positions including: chief scientist, Iospan Wireless (2000-01); AGM and department head, NEC Laboratories (1993-99); and head, broadband communications, Sarnoff Corp (1990-92). He obtained the B.Tech (Hons) from IIT Kharagpur in 1976 and the M.S. and Ph.D degrees from SUNY, Stony Brook in 1978 and 1979, respectively.

Ravishankar Ravindran is a principal staff researcher at Huawei. He has been conducting advanced telecommunications research for over 16 years. His current area of research focuses on information-centric networking, 5G, Internet of Things, software defined networking, and network function virtualization. His research focus currently is on NSF-funded future Internet proposals such as NDN/CCN, MobilityFirst, and XIA , which is in collaboration with external research groups and academia. Prior to this role, he was part of the CTO office at Nortel, where he was part of the Advanced Technology Group focused on research areas such as control plane routing protocols related to IP/(G)MPLS, scheduling problems in 4G wireless, and end-to-end QoE/QoS engineering for multimedia. Over the course of his research, he has been part of 10 granted patents and more than 30 pending filings in various areas of networking technologies. He has more than 30 technical publications in conferences and journals. Dr. Ravindran received a Ph.D. in systems and computer engineering from Carleton University in Canada.

Hongju Gao is currently an associate professor at the College of Information and Electrical Engineering, China Agricultural University, Beijing, China. She received her master and Ph.D. degrees in electrical engineering from Rutgers, The State University of New Jersey in 2000 and 2007, respectively. Her current research interests focus on agricultural applications of wireless sensor networks and Internet of Things, architecture, and protocols of wireless ad hoc networks, etc.

Lijun Dong has been a senior staff researcher at Futurewei Technologies Inc. since July 2015. She received her Ph.D degree in electrical engineering and a master degree in statistics from Rutgers University in 2010. She has research and innovation experience in the area of Internet of Things/machine-to-machine architecture, services, protocols, applications, and information centric networking.

Guoqiang Wang is a principal researcher at Futurewei Technology, working on information-centric network architecture, protocols, and emerging applications.

Hang Liu joined the Catholic University of America as an associate professor in the Department of Electrical Engineering and Computer Science in 2013. Prior to joining CUA, he had more than 10 years of research experience in the networking industry, and worked in senior research and management positions at several companies, including InterDigital Communications LLC, Thomson (now Technicolor) Corporate Research Lab, and NEC Laboratories America. He also led several industry-university collaborative research projects. He was an adjunct professor at WINLAB, Rutgers University, from 2004 to 2012. Dr. Liu has published more than 80 papers in leading journals and conferences, and received two best paper awards and one best student paper award. He is the inventor/co-inventor of more than 90 granted and pending international patents. He was an active participant in the IEEE 802 wireless standards and 3GPP standards, to which he made many contributions and for which he received an IEEE recognition award. He was the editor of the IEEE 802.11aa standard and the rapporteur of a 3GPP work item. Hang Liu received a Ph.D. degree in electrical engineering from the University of Pennsylvania. His research interests include wireless communications and networking, mmWave communications, cognitive radio networks, Internet of Things, mobile computing, Future Internet architecture, information-centric networking, software-defined networks, content distribution, video streaming, multimedia networking, and network security.