ICN Research Group
Internet-Draft
Intended status: Informational
Expires: February 29, 2016

Y. Zhang
D. Raychadhuri
WINLAB, Rutgers University
L. Grieco
Politecnico di Bari (DEI)
R. Ravindran
G. Wang
Huawei Technologies

August 28, 2015

ICN based Architecture for IoT
draft-zhang-icn-iot-architecture-00

Abstract

Internet of Things (IoT) promises to connect billions of objects to Internet. After deploying many stand-alone IoT systems in different domains, the current trend is to develop a unified de-fragmented IoT platform so that objects can be made accessible to applications across organizations and domains. Towards this goal, quite a few proposals have been made to build a unified IoT platform as an overlay on top of today's Internet. Such overlay solutions, however, are inadequate to address the important challenges posed by a unified IoT system, especially in terms of mobility, scalability, and communication reliability, due to the inherent inefficiencies of the current Internet. To address this problem, we propose to build a unified IoT platform based on the Information Centric Network (ICN) architecture, which we call ICN-IoT [20]. ICN-IoT leverages the salient features of ICN, and thus provides seamless device-to-device (D2D) communication, mobility support, scalability, and efficient content and service delivery leveraging in-network computing, caching and storage.

This draft begins by motivating the need for an unified ICN-IoT platform to connect heterogenous IoT systems. We then propose an ICN-IoT system architecture and middleware components which includes device/network-service discovery, naming service, IoT service discovery, contextual processing, pub/sub management to support efficient naming, data discovery, data processing and data distribution.

This Internet-Draft will expire on February 29, 2016.

Table of Contents

1. ICN-Centric Unified IoT Platform

In recent years, the current Internet has become inefficient in supporting rapidly emerging Internet use cases, e.g., mobility, content retrieval, IoT, contextual communication, etc. As a result,

Information Centric Networking has been proposed as a future Internet design to address these inefficiencies. ICN has the following main features: (1) it identifies a network object (including a mobile device, a content, a service, or a context) by its name instead of its IP address, (2) a hybrid name/address routing, and (3) a delay-tolerant transport. These features make it easy to realize many in-network functions, such as mobility support, multicasting, content caching, cloud/edge computing and security.

Considering these salient features of ICN, we propose to build a unified IoT platform using ICN, in which the overlay IoT services are only needed for administrative purposes, while the publishing, discovery, and delivery of the IoT data/services is directly implemented within the ICN network. Figure 1 shows the proposed ICN-centric IoT approach, which is centered around the ICN network instead of today's Internet.

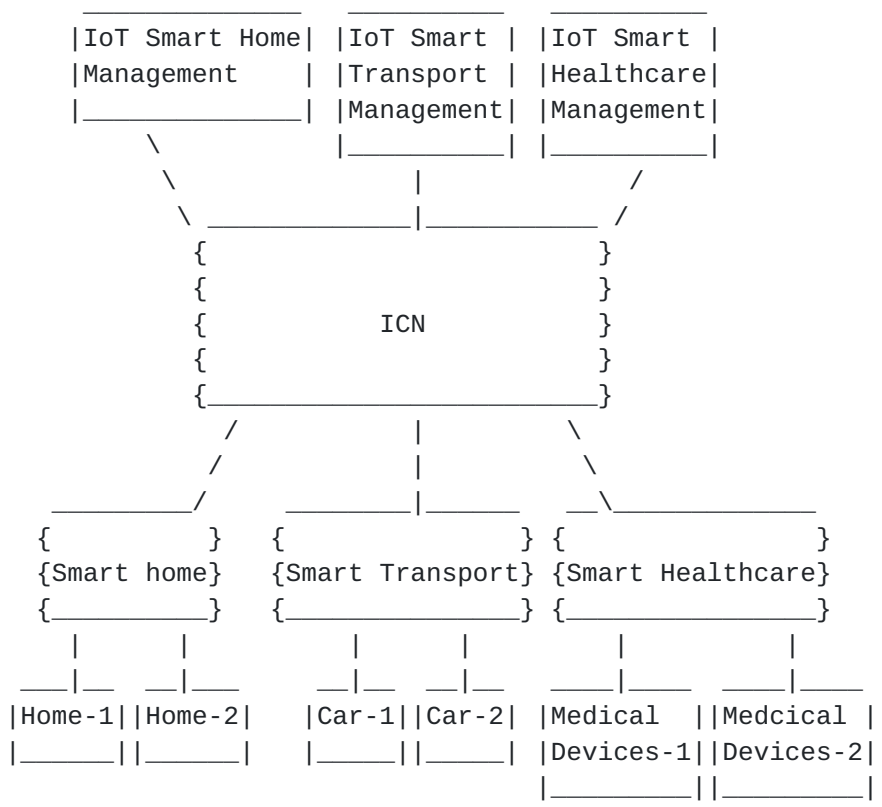

Figure 1: The proposed ICN-centric IoT unified platform.

1.1. Strengths of ICN-IoT

Our proposed ICN-IoT delivers IoT services over the ICN network, which aims to satisfy the requirements of the open IoT platform (discussed in "draft-zhang-icn-iot-challenges-01.txt"[20]):

- Naming. In ICN-IoT, we assign a unique name to an IoT object, an IoT service, or even a context. These names are persistent throughout their scopes.

- Scalability. In ICN-IoT, the name resolution is performed at the network layer, distributed within the entire network. Thus, it can achieve high degree of scalability exploiting features like content locality, local computing, and multicasting.

- Resource efficiency. In ICN-IoT, in both the constrained and non-constrained parts of the network, only those data that are subscribed by applications in the specified context will be delivered. Thus, it offers a resource-efficient solution.

- Local traffic Pattern. In ICN-IoT, we can easily cache data or services in the network, hence making more communications within local distances and reducing the overall traffic volume.

- Context-aware communications. ICN-IoT supports contexts at different layers, including device layer, networking layer and application layer. Contexts at the device layer include information such as battery level or location; contexts at the network layer include information such as network address and link quality; contexts at the application layer are usually defined by individual applications. In ICN-IoT, device and network layer contexts are stored within the network, while network elements (i.e., routers) are able to resolve application-layer contexts to lower-layer contexts. As a result of adopting contexts and context-aware communications, communications only occur under certain contexts that are specified by applications, which can significantly reduce the entire network traffic volume.

- Seamless mobility handling. In ICN-IoT, ICN's name resolution layer allows multiple levels of mobility relying on receiver-oriented nature for self-recovery for consumers, to multicasting and late-binding techniques to realize seamless mobility support of producing nodes.

- Data storage. In ICN-IoT, data are stored locally, either by the mobile device or by the gateway nodes or at service points. Also in-network storage/caching [22] also speeds up data delivery.

- Security and privacy. In ICN-IoT, secure binding between application-centric names and content instead of IP addresses to identify devices/data/services, is inherently more secure than the traditional IP paradigm [30].

- Communication reliability. ICN-IoT supports delay tolerant communications [35], which in turn provides reliable communications over unreliable links.

- Ad hoc and infrastructure mode. ICN-IoT supports both applications operating in ad-hoc and infrastructure modes.

- In-network processing. ICN offers in-network processing that supports various network services, such as context resolution, multicast and mobility support.

2. ICN-IoT System Architecture

The ICN-IoT system architecture shown in Figure 2, has the following five main system components:

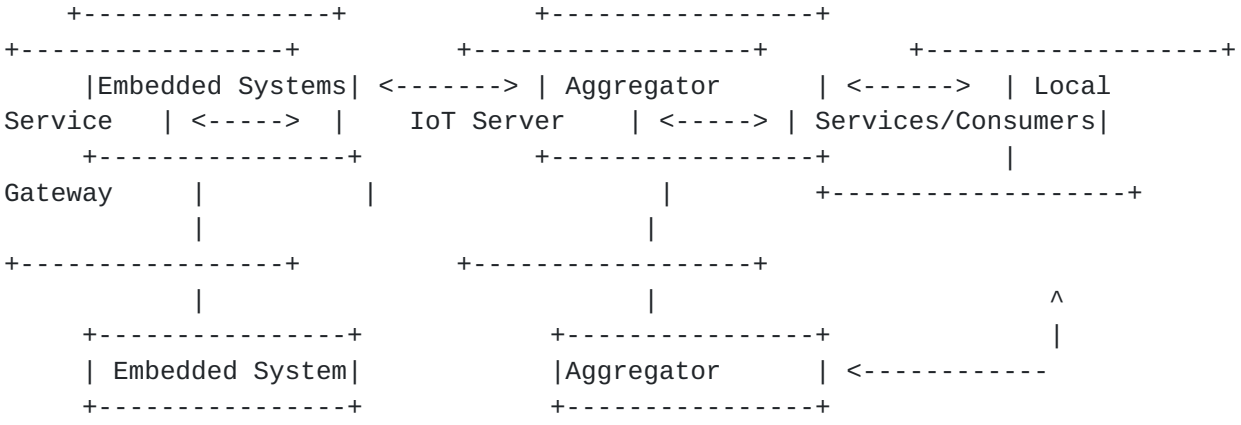

Figure 2: ICN-IoT System Architecture

- Embedded Systems: The embedded sensor has sensing and actuating functions and may also be able to relay data for other sensors to the Aggregator, through wireless or wired links.

- Aggregator: It interconnects various entities in a local IoT network. Aggregators serve the following functionalities: device discovery, service discovery, and name assignment.

- Local Service Gateway (LSG): A LSG serves the following functionalities: (1) connecting the local IoT system to the rest of the global IoT system, (2) assigning ICN names to local sensors, (3) enforcing data access policies for local IoT devices,

and (4) running context processing services to generate information specified by application-specific contexts (instead of raw data) to the IoT server.

- IoT Server: The IoT server is a centralized server that maintains subscription memberships and provides the lookup service for subscribers. Unlike legacy IoT servers that are involved in the data path from publishers to subscribers -- raising the concern of its interfaces being a bottleneck -- the IoT server in our architecture is only involved in the control path where publishers and subscribers exchange their names and certificates.

- Services/Consumer: These are application instances interacting with the IoT server to fetch or be notified of anything of interest within the scope of the IoT service.

Specifically, the logical topology of the IoT system can be mesh-like, with every sensor attached to one or more aggregators, while every aggregator is attached to a LSG and all the LSGs connected to the IoT server. Thus, each sensor has its aggregators, each of which in turn has its LSG. All sensors share the same IoT server. All the aggregators that are attached to the same LSG are neighbors to each other. Though our middleware supports mesh topology, in the rest of the draft, we will focus on the tree topology for the sake of simplicity.

3. ICN-IoT Middleware Architecture

The proposed ICN-IoT middleware aims to bridge the gap between underlying ICN functions and IoT applications.

The middleware functions are shown in Fig. 3 and it includes six core functions: device and network service discovery, naming service, Context Processing and Storage, IoT service discovery, Pub/Sub management, and security.

In contrast to centralized or overlay-based implementation in the legacy IP-based IoT platform, ICN-IoT architecture pushes a large portion of the functionalities to the network layer, such as naming resolution, mobility management, in-network processing/caching, context processing, which greatly simplifies the middleware design.

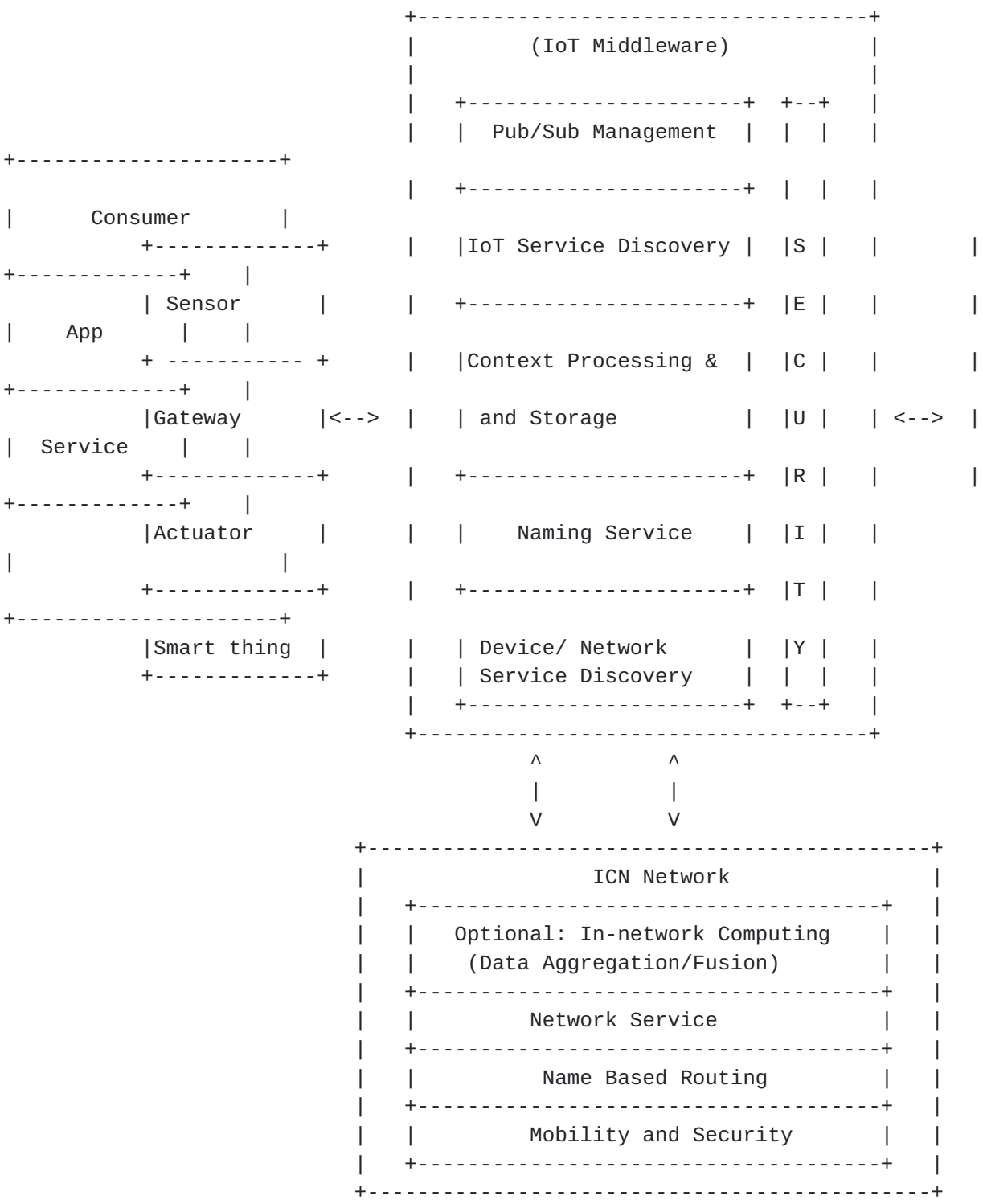

Figure 3: The ICN-IoT Middleware Functions

4. ICN-IoT Middleware Functions

The ICN-IoT middleware mainly consists of the following functions: device and network service discovery, service discovery, naming service, publish/subscribe management, context processing, and security. For each of these functions we highlight what the function achieves, advantages an ICN architecture enables in realizing this function, and provide discussion of how the function can be realized considering NDN [24] and MobilityFirst (MF) [23].

Please note that most of these functions are implemented on aggregators, LSGs and the IoT servers, and only very limited functions (mainly for device discovery) are implemented on resource-constrained sensors.

4.1. Device and Network Service Discovery

Device discovery is a key component of any IoT system. The objective of device discovery is to expose new devices to the rest of the IoT system -- every entity should be exposed to its direct upstream device and possibly other devices. Specifically, it includes the following three aspects: (1) a newly-added sensor should be exposed to its aggregator, and possibly to its LSG and the IoT server; (2) a newly-added aggregator is exposed to its LSG, and possibly to its neighbor aggregators; and (3) a newly-added LSG should be exposed to the IoT server. We note that device discovery could be used in other contexts, such as neighboring sensors discovering each other to form routing paths, but in this draft, we use the term to specifically mean discovering new devices for IoT middleware purpose.

During device discovery for newly-added sensors, the sensor passes its device-level information (such as manufacture ID and model number) and application-level information (such as service type and data type) to the upstream devices. If the sensor is to have an ICN name, the name is assigned by the naming service (described in Section 4.3), and recorded by both the LSG and the aggregator (and possibly the IoT server).

ICN enables flexible and context-centric device discovery which is important in IoT ecosystem where heterogeneous IoT systems belonging to different IoT services may co-exist. Contextualization is a result of name-based networking where different IoT services can agree on unique multicast names that can be pre-provisioned in end devices and the network infrastructure using the routing control plane. This also has an advantage of localizing device discovery to regions of network relevant to an ICN service, also enabling certain level of IoT asset security. In contrast IP offers no such natural IoT service mapping; any forced mapping of this manner will entail high configuration cost both in terms of device configuration, and network control and forwarding overhead.

In the device discovery phase, sensors expose their information, such as its manufacture secure ID and model name, to the upstream aggregators pulling information from sensors. There are two ways of achieving this aim: (1) sensors pushing the information towards the aggregators, and (2) aggregators pulling the information from sensors. In both NDN and MF, the pulling method is used. In NDN, this process is initiated by the configuration service running on

LSG, which periodically broadcasts discovery Interests (using the name /iot/model).The new sensor replies to the discovery interest with its information, and the configuration service then registers the sensor and generates a local ICN name for the sensor. In MF, we can set group-GUID as the destination address, and the configuration service issues a request via multicasting. When receiving such request, the new device replies with the manufacture secure ID, and the configuration service registers the device and generates a local ICN name for it.

The network service discovery for IoT infrastructure service like naming, gateway, or context-processing service is hosted on LSGs or Aggregators. These devices periodically broadcast their services, which will be responded to by sensors that need these services. Please note that only those sensors that need naming service or context processing service will respond. The detailed process is very similar to that involved in the device discovery (which is described above).

4.2. Service Discovery

Service discovery intends to learn IoT services that are hosted by one aggregator by its neighbor aggregators. The requirements include low protocol overhead (including low latency and low control message count), and discovery accuracy.

In today's IoT platforms, sensors, aggregators and LSGs are connected via IP multicast, which involves complicated group management and multicast name to IP translation service. Multicast, however, is greatly simplified in ICN as most ICN architectures have natural support for multicast.

Below, we explain how service discovery is implemented. The key to service discovery is to expose aggregator's services to its neighbor aggregators. How this is implemented differs in NDN and MF. In NDN, the source aggregator broadcasts an interest using the well-known name /area/servicename/certificate, which will eventually reach the destination aggregator. NDN's Interest/Data mechanisms allows only one response for each Interest send while discovery requires to learn multiple entities, hence efficient discovery is realized using exclusion via Selectors in the protocol or as an overlay protocol [29].

After establishing the multicast group, the source aggregator sends a request containing the service name and certificate to the multicast group. The destination aggregator that hosts the service checks the certificate and registers the source Aggregator if there is a matched

service. It replies with an acknowledgment containing certificate to the source aggregator.

For secure service discovery, a secured name needs to assigned to the service host. Especially in MF IoT, secured group GUID is utilized to realize service request multicast, which may be owned by multiple hosts, hence conventional public/private key scheme may not be suitable for this case. As an alternative, group key management protocol (GKMP) [31] can be adopted to resolved the issue above -- A naming service residing at LSG or IoT server (depending on application scope) generates a group public key that used as group GUID for a service, then this group public/private keys pair is assigned to each Aggregator that host this service. The service host Aggregator in the group then listen on this group GUID, and use the group private key to decrypt the incoming discovery message. Finally, we note that this form of secure service discovery is difficult for NDN.

As an example of NDN smart home, a thermostat expresses a request to discover a AC service using well-known name /home/ac/certificate via broadcast channel. In MF case, a multicast group GUID 1234 can be assigned to all home appliance IoT service. The thermostat sends request containing the service name and certificate to 1234. In both cases, the AC hosting this services replies with acknowledgment if all conditions match.

4.3. Naming Service

The objective of the naming service is to assure that either device or service itself is authenticated, attempting to prevent sybil (or spoofing) attack [31] and that the assigned name closely binds to the device (or service). Naming service is specific to MobilityFirst, and it is hard to achieve in the context of NDN. Naming service assigns and authenticates sensor and device names. An effective naming service should be secure, persistent, and able to support a large number of application agnostic names.

Traditional IoT systems use IP addresses as names, which are insecure and non-persistant. IP addresses also have relatively poor scalability, due to its fixed structure. Instead, ICN separates names from locators, and assigns unique and persistent names to each sensor/device, which satisfies the above requirements.

In what follows, we discuss an approach for secure naming service in ICN-IoT. We first consider the case where the embedded system is programmable so that before deployment, the owner can preload identity information (such as secure ID, a pair of public/private key and a certificate) , or has some manufacture ID and a pair of public/

private key (which is certified by the manufacturer). That is, the device is associated with information including device identity, public/private keys (PK_{device}, SK_{device}) and a certificate either from the owner or the manufacturer which certifies the device identity and public/private keys. When such a device is discovered, the aggregator will first verify the device identity (e.g., the device can generate a signature with the private key SK_{device} and present the signature and the certificate to the aggregator so that the aggregator can verify it), and then assign a name to the device as follows: the aggregator will issue a request to LSG together with its device identity and PK_{device}, so that LSG can assign an NDN name and generate a certificate (certifying the binding of NDN name, PK_{device}). To this end, the ICN name and the certificate will be sent back to the aggregator and will be stored locally if the device is resource-restricted. Otherwise, the ICN name and the certificate will be passed to the device.

For the MF-IoT, assigning a GUID for a device is rather straightforward: after verifying the device identity, the Aggregator inserts the public key PK_{device} and device information to the upper layer component to verify if there is a conflict in the corresponding scope. Specifically, LSG is in charge of local scope and IoT server guarantees the global uniqueness. Finally, the unique public key is used a GUID for the new device. Analogously, service discovery can be secured in a similar way.

In the case where devices are only associated with the secure manufacture ID while without being pre-loaded public/private keys and the certificate, it is critical to assure that devices are authenticated by using other trust model. For example the system can take advantage of the web-of-trust model or the contextually semantic information so that the devices manufactured by the same vendor can authenticate each other. Moreover, in order to comply with the capability of resource-restricted devices, light-weight cryptographic primitive (such as symmetric cryptography) may be used instead of public key cryptography.

Finally, we note that the same naming mechanism can be used to name higher-level IoT devices such as aggregators and LSGs.

4.4. Context Processing and Storage

In order to facilitate context-aware communication and data retrieval, we need to support context processing in the IoT system. The objective of context processing is to expose the sensor's low-level context information to upstream aggregators and LSGs, as well as to resolve the application's high-level context requirements using

lower-level sensor contexts. The context processing service usually runs on both aggregators and LSGs.

Context processing requires the underlying network to be able to support in-network computing at both application and network levels. ICN inherently supports in-networking computing and caching, which thus offers unique advantages compared to traditional IP network where the support for in-network computing and caching is poor.

Application level contexts differ from application to application, and therefore, we need to provide a set of basic mechanisms to support efficient context processing. Firstly, the network needs to define a basic set of contextual attributes for devices (including sensors, aggregators, and LSGs), including device-level attributes (such as location, data type, battery level, etc), network-level attributes (such as ICN names), and service-level attributes (such as max, min, average, etc).

Secondly, we need to have means to expose sensor/aggregator/LSG contextual attributes to the rest of the system, through centralized services such as naming resolution service.

Thirdly, the IoT server needs to allow applications (either producers or consumers) to specify their contextual requirements. Fourthly, the unconstrained part of ICN-IoT needs to be able to map the higher-level application-specific contextual requirements to lower-level device-level and network-level contextual information.

4.5. Publish-Subscribe Management

Data Publish/Subscribe (Pub/Sub) is an important function for ICN-IoT, and is responsible for IoT information resource sharing and management. The objective of pub/sub system is to provide centralized membership management service. Efficient pub/sub management poses two main requirements to the underlying system: high data availability and low network bandwidth consumption.

In conventional IP network, most of the IoT platforms provide a centralized server to aggregate all IoT service and data. While this centralized architecture ensures high availability, it scales poorly and has high bandwidth consumption due to high volume of control/data exchange, and poor support of multicast.

Next we consider two decentralized pub/sub models. The first one is the Rendezvous mode that is commonly used for today's pub/sub servers, and the second one involves Data-Control separation that is unique to ICN networks where the control messages are handled by the centralized IoT server and the data messages are handled by the

underlying ICN network. Compared to the popular Rendezvous mode where both control and data messages both meet at the centralized server, separating data and control messages can greatly improve the scalability of the entire system, which is enabled by the ICN network.

In today's IP network, Rendezvous mode is the classic pub/sub scheme in which data and requests meet at an intermediate node. In this case the role of the IoT server is only required to authenticate the consumers and providing it Rendezvous service ID.

While NDN is a Pull-based architecture without supporting the Pub/Sub mode naturally, COPSS [33] proposes a solution to fix this problem. It integrates a push based multicast feature with the pull based NDN architecture at the network layer by introducing Rendezvous Node(RN). RN is an logical entity that resides in a subset of NDN nodes. The publisher first forwards a Content Descriptor (CD) as a snapshot to the RN. RN maintains a subscription table, and receives the Subscription message from subscriber. The data publisher just sends the content using Publish packet by looking up FIB instead of PIT. If the same content prefix is requested by multiple subscribers, RN will deliver one copy of content downstream, which reduces the bandwidth consumption substantially.

Compared with the Rendezvous mode in which data plane and control plane both reside on the same ICN network layer, we consider an architecture where the control message is handled by the centralized server while data is handled by ICN network layer. Following the naming process mentioned above, the LSG has the ICN name for the local resource which is available for publishing on IoT server. IoT server maintains the subscription membership, and receives subscription requests from subscribers. Since the subscribers has no knowledge about the number of resource providers and their identities in a dynamic scenario, IoT server has to take responsibility of grouping and assigning group name for the resource.

MF takes advantage of Group-GUID to identify a service provided by multiple resources. This Group-GUID will be distributed to the subscriber as well as the publisher. In an example of NDN, it uses the common prefix/home/monitoring/ to identify a group of resource that provides multiple monitoring services such as /home/monitoring/ temperature and /home/monitoring/light. The subscriber retrieves the prefix from the IoT server, and sends Interest toward the resource. In a MF example, GUID-x identifies the "home monitoring" service that combines with "light status" and "temperature". The resource producers, i.e. the host of "temperature" and the host of "light status" are notified that their services belong to GUID-x, then listen on GUID-x. The subscriber sends the request containing GUID-x

through multicasting which ultimately reach the producers at the last common node. Once receiving the request, the resource producer unicasts the data to the subscriber. In addition, if multiple resource consumers subscribe to the same resource, the idea of Group-GUID can be reused to group the consumers to further save bandwidth using multicast.

4.6. Security

This spans across all the middleware functions. Generally speaking, the security objective is to assure that the device that connects to the network should be authenticated, the provided services are authenticated and the data generated (through sensing or actuating) by both devices and services can be authenticated and kept privacy (if needed). To be specific, we consider the approach to secure device discovery, naming service and service discovery, because other services, such as pub/sub management and context processing and storage, can be properly secured according to application-specific demands.

5. Informative References

[1] Cisco System Inc., CISCO., "Cisco visual networking index: Global mobile data traffic forecast update.", 2009-2014.

[2] Mohsen, D. and M. Michael, "Smart home mobile RFID-based Internet-of-Things systems and services.", International Conference on Internet-of-Things systems and services. , 2008.

[3] Zhu, Q., Wang, R., Chen, Q., Liu, Y., and W. Qin, "Iot gateway: Bridgingwireless sensor networks into internet of things.", Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on. IEEE, 2010 , 2010.

[4] Huang, R., "Smart Campus: The Developing Trends of Digital Campus.", Open Education Research 4 (2012): 004 , 2012.

[5] Zhu, Q., Wang, R., Chen, Q., Liu, Y., Qin, W., and W. Qin, "ICN based Architecture for IoT - Requirements and Challenges.", draft-zhang-icn-iot-challenges-00.txt , 2014.

[6] Piro, R., Cianci, I., Grieco, L., Boggia, G., and P. Camarda, "Information Centric Services in Smart Cities.", Elsevier Journal of Systems and Software , 2014.

[7] Grieco, L., Alaya, M., Monteil, T., and K. Drira, "Architecting Information Centric ETSI-M2M systems.", Proc. of IEEE PerCom (to appear as work in progress paper) , 2014.

[8] Dietrich, D., Bruckner, D., Zucker, G., and P. Palensky, "Communication and computation in buildings: A short introduction and overview.", IEEE Transaction of Industrial Electronics , 2010.

[9] Zhang, Y. and R. Yu, "Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid.", Advances in Energy Engineering (ICAEE), 2010 International Conference on. IEEE , 2010.

[10] Zhang, Y., Yu, R., Nekovee, M., Liu, Y., Xie, S., and S. Gjessing, "Cognitive machine-to-machine communications: visions and potentials for the smart grid.", IEEE Network , 2012.

[11] Hong, Z., Liu, B., and D. Wang, "Design and Research of Urban Intelligent Transportation System Based on the Internet of Things.", Internet of Things. Springer Berlin Heidelberg , 2012.

[12] Min, Z., Yu, T., and G. Zhai, "Smart Transport System Based on the Internet of Things.", Applied mechanics and materials , 2011.

[13] Min, Z., Yu, T., Zhai, G., Zhai, G., and G. Zhai, "The internet of things for ambient assisted living.", Information Technology: New Generations (ITNG), 2010 Seventh International Conference on , 2010.

[14] Reijo, S., Abie, H., and M. Sihvonen, "Towards metrics-driven adaptive security management in E-health IoT applications.", Proceedings of the 7th International Conference on Body Area Networks. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) , 2012.

[15] Yan, Y., Qian, Y., Sharif, H., and D. Tipper, "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges.", IEEE Communications Surveys and Tutorials , 2013.

[16] Shafig, M., Ji, L., Liu, A., Pang, J., and J. Wang, "A first look at cellular machine-to-machine traffic: large scale measurement and characterization.", Proceedings of the ACM Sigmetrics , 2012.

[17] The European Telecommunications Standards InstituteS, ETSI., "http://www.etsi.org/.", 1988.

[18] Global Intiative for M2M Standardization, oneM2M., "http://www.onem2m.org/.", 2012.

[19] Constrained RESTful Environments, CoRE., "https://datatracker.ietf.org/wg/core/charter/.", IETF, 2013.

[20] ICN based Architecture for IoT - Requirements and Challenges, ICN-IoT., "https://tools.ietf.org/html/draft-zhang-iot-icn-challenges-01.", IETF/ICNRG 2015.

[21] Ghodsi, A., Shenker, S., Koponen, T., Singla, A., Raghavan, B., and J. Wilcox, "Information-Centric Networking: Seeing the Forest of the Trees.", Hot Topics in Networking , 2011.

[22] Dong, L., Zhang, Y., and D. Raychaudhuri, "Enhance Content Broadcast Efficiency in Routers with Integrated Caching.", Proceedings of the IEEE Symposium on Computers and Communications (ISCC) , 2011.

[23] NSF FIA project, MobilityFirst., "http://www.nets-fia.net/", 2010.

[24] NSF FIA project, NDN., "http://named-data.net/", 2010.

[25] Kim, B., Lee, S., Lee, Y., Hwang, I., and Y. Rhee, "Mobiiscape: Middleware Support for Scalable Mobility Pattern Monitoring of Moving Objects in a Large-Scale City.", Journal of Systems and Software, Elsevier, 2011.

[26] Hui, JW. and DE. Culler, "IP is dead, long live IP for wireless sensor networks.", ACM, SenSys, 2008.

[27] AllSeen Alliance, AllJoyn., "https://allseenalliance.org/developers/learn", 2015.

[28] "https://www.iotivity.org/about", 2015.

[29] Ravindran, R., Biswas, T., Zhang, X., Chakrabort, A., and G. Wang, "Information-centric Networking based Homenet", ACM, Sigcomm, 2013.

[30] Nikander, P., Gurtov, A., and T. Henderson, "Host identity protocol (HIP): Connectivity, mobility, multi-homing, security, and privacy over IPv4 and IPv6 networks", IEEE Communications Surveys and Tutorials, pp: 186-204, 2010.

[31] Newsome, J., Shi, E., Song, DX., and A. Perrig, "The sybil attack in sensor networks: analysis and defenses", IEEE, IPSN, 2004.

[32] Harney, H. and C. Muckenhirn, "Group Key Management Protocol (GKMP) Architecture", RFC Editor 1997.

[33] Jiachen, C., Mayutan, A., Lei, J., Xiaoming, Fu., and KK. Ramakrishnan, "COPSS: An efficient content oriented publish/subscribe system", ACM/IEEE ANCS, 2011.

[34] Marica, A., Campolo, C., and A. Molinaro, "Multi-source data retrieval in IoT via named data networking", ACM ICN Siggcomm, 2014.

[35] Nelson, S., Bhanage, G., and D. Raychaudhuri, ""GSTAR: generalized storage-aware routing for mobilityfirst in the future mobile internet", Proceedings of the sixth international workshop on MobiArch, pp. 19--24, 2011.

Authors' Addresses

Prof.Yanyong Zhang
WINLAB, Rutgers University
671, U.S 1
North Brunswick, NJ 08902
USA

Email: yyzhang@winlab.rutgers.edu

Prof. Dipankar Raychadhuri
WINLAB, Rutgers University
671, U.S 1
North Brunswick, NJ 08902
USA

Email: ray@winlab.rutgers.edu

Prof. Luigi Alfredo Grieco
Politecnico di Bari (DEI)
671, U.S 1
Via Orabona 4, Bari 70125
Italy

Email: alfredo.grieco@poliba.it

Ravi Ravindran
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: ravi.ravindran@huawei.com

Guoqiang Wang
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: gq.wang@huawei.com