Using ICN Slicing Framework to Build an IoT Edge Network

Asit Chakraborti[†], Syed Obaid Amin[†], Aytac Azgin[†], Satyajayant Misra[‡], and Ravishankar Ravindran[†]
[†] Huawei Research Center, Santa Clara, CA, USA.
[‡] Department of Computer Science, New Mexico State University, NM, USA.
[†]{asit.chakraborti, obaid.amin, aytac.azgin, ravi.ravindran}@huawei.com,[‡]misra@cs.nmsu.edu

ABSTRACT

We demonstrate 5G network slicing as a unique deployment opportunity for information centric networking (ICN), by using a generic service orchestration framework that operates on commodity compute, storage, and bandwidth resource pools to realize ICN service slices. In this demo, we specifically propose a service slice for the IoT Edge network. ICN has often been considered pertinent for IoT use due to its benefits like simpler stacks on resource constrained devices, in-network caching, and in-built data provenance. We use a lightweight ICN stack on IoT devices connected with sensors and actuators to build a network, where clients can set realistic policies using their legacy hand-held devices. We employ name based authentication protocols between the service end-points and IoT devices to allow secure onboarding. The IoT slice co-exists with other service slices that cater to different classes of applications (*e.g.*, bandwidth intensive applications, such as video conferencing) allowing resource management flexibility. Our design creates orchestrated service Edge functions to which the clients connect, and these can in turn utilize in-network stateless functions to perform tasks, such as decision making and analytics using the available compute resources efficiently.

CCS CONCEPTS

• **Networks** → *Network architectures*; *Network protocols*;

KEYWORDS

ICN, Network Slicing, IoT Edge

1 INTRODUCTION

The purpose of the IoT Edge is to interact with the real world, gathering environmental data and controlling smart systems. The IoT Edge computing paradigm is proposed to perform data processing at the Edge network instead of using a centralized cloud for that purpose. The main benefit of such a system is the improved scalability and reliability, and the reduced latency from the IoT devices to the distributed Edge intelligence. Processing/filtering of data at the Edge can also reduce bandwidth usage on the backhaul links. In addition, an Edge-based service intelligence can continue to perform critical functions in the event of central component or network failure.

The IoT network also provides an interesting deployment opportunity for ICN as it is more amenable to greenfield deployment compared to other networking contexts such as CDNs. The ICN-based end-point stack can be significantly lightweight with no persistent connections to maintain, easy to achieve name based discovery and self-configuration, and efficient many-to-many communication enabled by default. Additionally, data provenance is an intrinsic part of the protocol, and data caching at less constrained nodes improves reliability and power efficiency. The suitability of an ICN-based IoT Edge has been studied to some extent, but more work is needed to validate the idea using actual IoT devices and practical scenarios in non-trivial configurations. One of the important challenges in this area is the scalable and secure onboarding of IoT devices, for which [1] provides a solution, but it has not been demonstrated with real IoT devices.

Our goal in this demo is to show the applicability of ICN-based IoT Edge that allows secure device onboarding and message exchanges using off-the-shelf resource-constrained devices, while showing its coexistence with other real-time services sharing network resources while meeting individual service requirements. Our virtualized Edge router platform employs service entities providing services to remote clients, one of them being the middleware function to support device onboarding. Use of virtual slices to house these components allows the service to scale in terms of number of supported users and devices. Another system goal is to experiment with multiple ways of introducing computing power at the Edge, the default being the orchestrated service component, but generalized computing tasks can be performed in-network, for instance using Named Function Networking (NFN) [2].

2 IOT NETWORK DESIGN

Network Slicing

Network slicing (NS) offers support for diverse network services in 5G ranging from high bandwidth services to ones requiring very low end-to-end latency and also to those that need to scale out to very large number of devices and applications [3]. Slicing offers logical and/or physical separation of the service, control and data planes among different services based on virtualized or physical resources to guarantee isolation and to satisfy service layer agreements (SLAs) which include QoS, reliability, availability, and security. NS uses softwarization at all levels to allow dynamic spawning and elastic scaling of service subnetworks based on their individual demands. It further leverages software defined networking (SDN) and network function virtualization (NFV) to program system resources among multiple services that span multiple domains. In short, NS allows new network functions and novel network architectures like ICN to coexist with IP based services.

System architecture

Figure 1 shows the proposed demo architecture which carries multiple ICN service slices and implements an overlaid ICN deployment at the Edge infrastructure with constrained devices communicating over native 802.15.4. Here we assume an operator or an enterprise that manages a set of ICN Service Routers (ISRs) that are strategically placed at the network Edge. These Edge nodes are capable of hosting ICN service functions, and in our case they host components to assist the IoT service and also a video conferencing service [4]. The components are housed in their isolated slices that span multiple ISRs.

ICN '18, September 21–23, 2018, Boston, MA, USA

ACM ISBN 978-1-4503-5959-7/18/09.
https://doi.org/10.1145/3267955.3269017

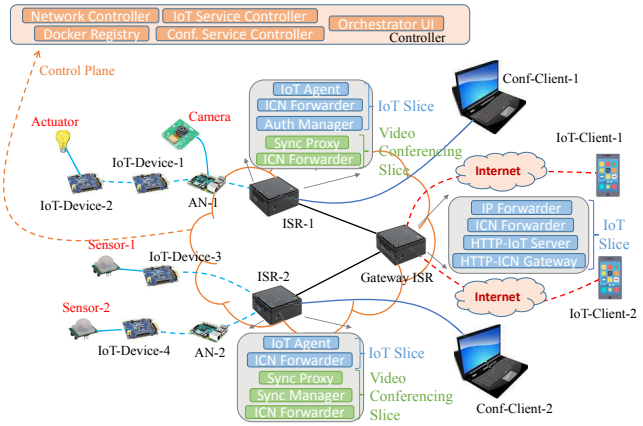

Figure 1: Demo system architecture

Service orchestration

Docker registry is used to provision the service component images, and the swarm mode is used to orchestrate services. A simple JBOSS based UI is used to manage the service slices. The controller framework includes an ONOS based network controller that provisions FIB entries between multiple entities, and service level controllers that maintain critical service information and helps achieve scalability of the system by triggering appropriate actions like provisioning additional components.

ICN-IoT Slice Service Management

The ICN IoT slice gets provisioned using the Orchestration UI, this spawns at least one instance of Authentication Manager (AM), and an instance of IoT Agent on all the regular ISRs. On specially provisioned Gateway ISR it spawns an HTTP-ICN Gateway and an HTTP-IoT Server instance. The AM plays an important role in IoT device onboarding by acting as a trust and verification agent between the IoT devices and the IoT service administration. The IoT Agent is an ICN application that implements IoT policies by managing and multiplexing the information exchanges with the clients as well as the IoT devices. In addition, the IoT agent is also responsible for processing, storing, and publishing IoT data and logs. The HTTP-ICN Gateway translates HTTP API based exchanges to ICN API and the HTTP-IoT Server provides RESTful APIs that the IoT service clients use. IoT service clients are implemented using an Android application to check the IoT device status, create IoT policies, display IoT device data, and control IoT devices.

IoT component details

Our demo features two types of IoT devices, Constrained IoT Nodes (also referred as `Standard Nodes`) and the relatively resource rich `Anchor Nodes`. We use SAM R21 Xplained Pro evaluation kit [1] as the `standard node`, and Raspberry Pi 3 Model B[2] as the `anchor node`. The SAM R21 features 256 KB Flash, 32 KB SRAM, a microcontroller capable of running at 48MHz, and an Ultra Low Power Transceiver for 2.4GHz ISM Band. The `anchor nodes` include Openlabs radio module to communicate with the `standard nodes`. The `standard nodes` run Riot OS whereas the `anchor nodes` run Linux, and CCN-lite [5] is used as the unifying ICN layer spanning all the IoT devices and the IoT slice. Sensors and actuators can be connected to `standard nodes` as well as `anchor nodes`. We connect a camera to the `anchor node` and light actuators and motion sensors to the `standard nodes`.

[1] atsamr21-xpro, https://www.microchip.com.
[2] raspberry-pi-3-model-b, https://www.raspberrypi.org.

IoT device onboarding

We use a three phase onboarding scheme based on the LASeR protocol [1]. In the first phase, the new device sends a broadcast request to ask an already authenticated and onboarded neighbor node for the necessary information to authenticate the network. In the second phase, the new device authenticates itself to the Authentication Manager in the network. The third phase establishes the routing in the network necessary to reach the new node. The core trust mechanism in this protocol depends on pre-shared keys between the Authentication Manager and the IoT devices, and in this simplified implementation the network administrator programs the Authentication Manager with the pre-shared keys for the IoT devices.

Service coexistence

To show the coexistence of multiple service slices, we include video conferencing clients that produce video and consume video produced by other clients using a Sync framework hosted in a network slice [6]. The Sync framework consists of the Sync Manager and Sync Proxy service functions that are provisioned on the regular ISRs to assist the clients to synchronize state with other participating clients.

3 DEMO FEATURES

Our demo shows how ICN based network slices for multiple services (*i.e.*, IoT and video conferencing) can coexist in a unified control plane. Specifically, we show secure IoT device onboarding using real IoT devices that utilizes the IEEE 802.15.4 wireless standard. This framework is then made available to IoT clients using legacy IP based programs, where a network service component at the gateway node performs the necessary translation from ICN to IP. On top of these capabilities, we show IoT policies being enabled by the service, which allows the client to instruct the network to perform operations such as *"record video using specific cameras and turn on specific lights when motion is detected on specific sensors"*.

4 CONCLUSIONS AND FUTURE WORK

Our demo showcases the potential of ICN deployment at the Edge to host an IoT service within the 5G slicing framework. We demonstrate this using resource constrained devices running ICN protocols that are assisted by the network service to provide a satisfactory user experience to legacy clients. Computing infrastructure in the service slice is orchestrated using traditional methods, and we intend to investigate in-network methodologies to explore compute resources in the Edge network that will provide more adaptability and context sensitivity to our solution.

REFERENCES

[1] T. Mick, R. Tourani, and S. Misra. LASeR: Lightweight authentication and secured routing for NDN IoT in smart cities. *IEEE IoT Journal*, 5(2):755–764, May 2017.
[2] M. Sifalakis, et al. An information centric network for computing the distribution of computations. In *ACM ICN*, 2014.
[3] NGMN Alliance. 5G White Paper, Feb 2015.
[4] R. Ravindran, et al. 5G-ICN: Delivering ICN services over 5G using network slicing. *IEEE Communications Magazine*, 55(5):101–107, May 2017.
[5] CCN-lite, https://ccn-lite.net.
[6] A. Chakraborti, et al. ICN based scalable A/V conferencing on VSER platform. In *ACM ICN*, 2015.