NDNoT: A Framework for Named Data Network of Things

Zhiyi Zhang, Edward Lu, Yanbiao Li, Lixia Zhang
UCLA
zhiyi,yanbiao,lixia@cs.ucla.edu, edwardzlu98@gmail.com

Tianyuan Yu
Sichuan University
royu9710@outlook.com

Davide Pesavento, Junxiao Shi, Lotfi Benmohamed
NIST
davide.pesavento,junxiao.shi,lotfi.benmohamed@nist.gov

ABSTRACT

The Named Data Networking (NDN) architecture provides simple solutions to the communication needs of Internet of Things (IoT) in terms of ease-of-use, security, and content delivery. To utilize the desirable properties of NDN architecture in IoT scenarios, we are working to provide an integrated framework, dubbed NDNoT, to support IoT over NDN. NDNoT provides solutions to auto configuration, service discovery, data-centric security, content delivery, and other needs of IoT application developers. Utilizing NDN naming conventions, NDNoT aims to create an open environment where IoT applications and different services can easily cooperate and work together. This poster introduces the basic components of our framework and explains how these components function together.

CCS CONCEPTS

• **Networks** → **Naming and addressing**; *Network protocol design*;

KEYWORDS

NDN, Internet of Things

ACM Reference format:
Zhiyi Zhang, Edward Lu, Yanbiao Li, Lixia Zhang, Tianyuan Yu, and Davide Pesavento, Junxiao Shi, Lotfi Benmohamed. 2018. NDNoT: A Framework for Named Data Network of Things. In *Proceedings of 5th ACM Conference on Information-Centric Networking, Boston, MA, USA, September 21–23, 2018 (ICN '18),* 2 pages.
DOI: 10.1145/3267955.3269019

1 INTRODUCTION

We argue that the Named Data Networking (NDN) [5] architecture provides simple solutions to the communication needs of the Internet of Things (IoT) [3], for the following 5 reasons: (i) NDN builds the data-centric security into the network layer by securing data directly in a local network system instead of relying on secured sessions and trusted cloud servers. (ii) Naming conventions provide an open environment for applications and services to cooperate and function together. (iii) By naming data, NDN enables host multihoming and seamlessly utilizes all communication interfaces (e.g., Bluetooth, BLE, Wi-Fi, 802.15.4). (iv) NDN natively supports content multicast and in-network caching. (v) NDN provides a simple way of developing applications: developers focus on the data itself without worrying about DNS or IP configurations.

In this poster, we introduce NDNoT, an IoT framework running over the NDN architecture. NDNoT uses semantic names as the centerpiece of the system: (consumer) applications use names to fetch named and secured content produced by other (producer) applications. Compared to the existing IP-based IoT frameworks, NDNoT gets rid of the mapping (e.g., DNS, mDNS) between application-readable service identifiers (e,g., URI, service names) and network identifiers (e.g., IPv6 addresses).

NDNoT is designed to work on a variety of hardware platforms: it can run on the RIOT [2] operating system and on Arduino-compatible [1] hardware. We have been experimenting NDNoT with Expressif ESP32 boards and Atmel SAM R21 Xpro boards.

2 NDNOT FRAMEWORK: A TOP-DOWN VIEW

The framework of NDNoT is shown in Figure 1. NDNoT devices are also able to communicate with Android phones and Linux or macOS devices that are running the NDN protocol stack. In an IoT scenario, an Android phone or a Linux/macOS device can play the role of domain controller.

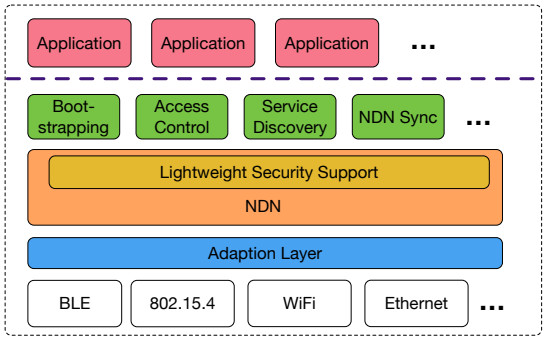

Figure 1: NDNoT Framework

In this section, we provide a top-down view of the framework and explain how each module works.

2.1 Bootstrapping

The bootstrapping module provides applications an automatic way of bootstrapping, which includes both *security* bootstrapping and *network* bootstrapping. This module is strictly required for new added devices to (i) communicate with other IoT devices, and (ii) build a trust relationship with the local IoT system and sign/verify NDN Data packets.

ICN '18, Boston, MA, USA

DOI: 10.1145/3267955.3269019

Based on a pre-shared secret (e.g., by scanning a QR code), the security bootstrapping enables an IoT device/application to (i) install the certificate of the controller as a trust anchor and (ii) obtain a certified name linked to a certificate signed by the controller. The network bootstrapping step automatically configures the NDN forwarder with the node's available network interfaces (e.g., Bluetooth, BLE, Wi-Fi, IEEE 802.15.4) and learns the name prefix of the connected IoT system. The NDNoT framework also allows applications to specify which underlying link-layer protocol to use for different communication scenarios.

2.2 Service Discovery

The service discovery module helps applications find available services in the local IoT network and also to advertise their own services to other nodes. Running over NDN, the service discovery is implemented by prefix discovery and prefix registration. Unlike the service discovery solutions in TCP/IP networks which are based on a (distributed) database query, NDNoT utilizes NDN's naming convention and the use of application names at the *network layer* to facilitate the discovery/advertisement process.

2.3 NDN Sync

NDNoT re-implements DSSN [4], a sensing data synchronization protocol, in a lightweight manner for the IoT network. This enables an application to easily synchronize the content with other devices in the IoT system. We are also extending the sync protocol to support a lightweight publish-subscribe message pattern.

2.4 Securing IoT Networking

Underneath the integrated modules that provides different functionalities to application developers, NDNoT provides a lightweight implementation of NDN communication and NDN security support. Different from the NDN protocol stack implementation for operating systems like Linux, the NDNoT implementation aims to be as efficient as possible, taking into account the power and capacity constraints of small IoT devices.

2.4.1 Schematizing Trust. Instead of trusting a piece of data or a node by verifying the signature against some commercial CA's certificate, NDNoT supports schematized trust where applications can define their trust policies with names, so that only those Data packets whose names have proper relations with the signing key names will be accepted. Schematized trust enables applications to achieve fine-grained data authentication, thereby improving the overall security of the IoT system.

2.4.2 Localized Access Control. IoT devices like smart home cameras and door locks require high data confidentiality and strict access control, which are supported by NDNoT's access control module. Instead of storing the access keys in remote cloud servers, NDNoT provides localized access control, allowing the local user (e.g., home owner) to have complete control of the data produced and consumed by the IoT system. By naming both digital keys and data, NDNoT automates the key distribution process of the access control scheme, thus minimizing manual configuration.

2.5 Adaptation Layer

The adaptation layer abstracts different link-layer protocols and wraps the NDN Interest and Data packets into link-layer frames. This layer maintains a table indicating which network interface an Interest packet should be forwarded to based on its name. An application can select different interfaces for different packets simply by tagging NDN packets instead of learning how underlying protocols work and invoking their APIs. Regarding the implementation, the adaptation layer works as a separate multiplex/demultiplex process and communicates with NDN applications using Inter-Process Communication (IPC) or other equivalent mechanism.

3 AN APPLICATION SCENARIO

In a classic smart home application scenario, the home owner uses his Android phone or a Linux/macOS laptop as the controller to manage the IoT system. With NDNoT, each IoT device (e.g., camera, temperature sensor, etc.) trusts the controller and is able to verify signatures generated by other IoT devices, so that the commands or content with fake or untrusted signatures will never be accepted. Notably, the trust anchor (i.e., the controller certificate) is stored locally on the controller device instead of a remote cloud server. All devices register their name prefixes for provided services and a device is able to discover the available services under other prefixes by fetching the metadata. A device records the discovered name prefixes with the corresponding network interfaces (e.g., BLE, 802.15.4) where they are fetched, so that the application simply fetches named content or issues commands without worrying about which network interface to use. When data privacy is needed, the home owner can configure the access rights for each device/service in the system so that only authorized devices/services can access (i.e., decrypt) the private data.

4 CURRENT STATUS AND FUTURE WORK

We are currently working on the implementation of NDNoT for boards that work with RIOT and Arduino. We are adding support for different link-layer network protocols on low-powered devices with limited processing capabilities and, at the same time, optimize the memory and energy efficiency by specializing the implementation for different hardware platforms and link protocols. We plan to have a live demo of smart home scenarios before October and to release the package by the end of the year.

ACKNOWLEDGMENTS

This work is partially supported by the US National Science Foundation under award CNS-1719403.

REFERENCES

[1] Arduino. Arduino home website. https://www.arduino.cc/. Accessed: 2018-07-29.
[2] Emmanuel Baccelli et al. Riot: an open source operating system for low-end embedded devices in the iot. *IEEE Internet of Things Journal*, 2018.
[3] Wentao Shang et al. Named Data Networking of Things. In *Internet-of-Things Design and Implementation (IoTDI), 2016 IEEE First International Conference on*, pages 117–128. IEEE, 2016.
[4] Xin Xu, Haitao Zhang, Tianxiang Li, and Lixia Zhang. Achieving resilient data availability in wireless sensor networks. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2018.
[5] Lixia Zhang et al. Named Data Networking. *ACM Computer Communication Review*, July 2014.