# Demonstrating Named Data Networking Integration into DataTweet IoT Architecture

Soumya Kanti Datta and Christian Bonnet
EURECOM
Sophia Antipolis, France
E-mails: {dattas, bonnet}@eurecom.fr

***Abstract*—Content Centric Networking (CCN) is an important aspect of the Future Internet Architecture. CCN investigates the evolution from current host-centric network architecture of IP to data centric networking architecture. This paper aims at demonstrating the integration of Named Data Networking (NDN) mechanism, an instantiation of CCN, in previously developed DataTweet Internet of Things (IoT) architecture. The main accomplishment is in achieving and demonstrating interworking between the NDN and IoT paradigms. We briefly describe the identified challenges, their solutions and a real prototype showcasing the integration of NDN into the DataTweet IoT Architecture. Early results about performance evaluation and future works are outlined.**

***Index Terms*—Content Centric Networking; Internet of Things; Named Data Networking; Prototype.**

## I. INTRODUCTION

The current IoT ecosystem utilizes the IP network architecture as a communication backbone which gives rise to two main challenges - (i) need of a DNS system and (ii) no native support for mobility especially for the connected cars applications [1]. Moreover, majority of the IoT systems consist of Cloud systems disseminating data to multiple consumer mobile devices. The concept of Content Centric Networking (CCN) investigates the evaluation of current host centric IP into a data centric network architecture [7]. An instantiation of CCN is Named Data Networking (NDN) [9] which provides a set of benefits including in-network data caching, reduced congestion control, data security and simple network configuration. NDN natively supports mobility and does not require DNS mechanism for network operations [4]. NDN mechanism makes data directly addressable and routable. These features make NDN suitable for information dissemination in the IoT systems. We have developed a data centric (DataTweet) IoT architecture that decouples dependency from IoT infrastructure and focuses on data centric services [5]. DataTweet architecture building blocks follow oneM2M standard specifications, allows integration of vehicular resources into IoT and developing horizontal IoT application. While devising a mechanism to integrate NDN (to exploit its stated benefits) into the DataTweet architecture, we identified three main challenges - (i) uniform naming of interests generated by data consumer, (ii) uniform mapping of interest & data and (iii) dissemination of both the raw data and high level intelligence. To address these challenges, we developed a uniform naming scheme and uniform mapping between interest and data. The resulting IoT architecture with NDN components is shown in Fig. 1. This paper aims to present an overall system that is combining the IoT and CCN paradigms. Our main contributions are -

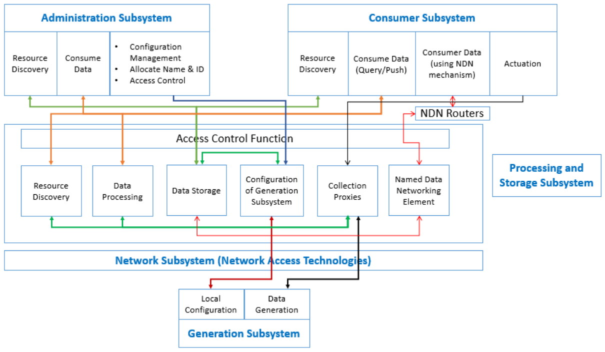


Fig. 1. DataTweet IoT architecture with NDN components.

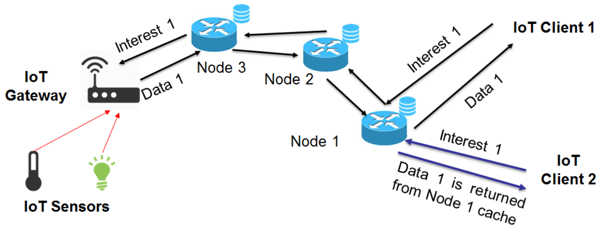
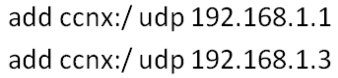


Fig. 2. A generic use case for NDN IoT prototype implementation.

(i) demonstration of an NDN stack and its components' integration into the DataTweet IoT architecture, (ii) dissemination of both raw sensor data and high level intelligence computed on the raw sensor data using NDN and (iii) applicability of the overall system into an use case. The impacts of the paper are - (i) the combined system moves beyond client-server and subscription-notification models and (ii) data caching in network nodes reduces congestion.

## II. NDN IoT PROTOTYPE IMPLEMENTATION

This section describes the prototype implementation. We have chosen the CCNx which provides a complete stack of NDN. Note that each node participating into the NDN paradigm must integrated the stack. To demonstrate the prototype system, we describe an use case shown in Fig. 2. It considers two consumer devices as IoT clients, three NDN routers (Node 1, Node 2 and Node 3) and an IoT gateway which manages two IoT sensors(temperature and light). The IoT client 1 creates and sends its Interest 1 to Node 1. It in turn forwards the interest based on forwarding mechanism. Finally, the interest arrives at the IoT Gateway which contains Data 1 corresponding to Interest 1. The data takes the same route back to IoT Client 1 and is cached in Node 3, Node 2 and Node 1. When IoT client 2 issues the same interest, the data is returned from Node 1 cache and the interest is not propagated to the IoT Gateway. The subsections below describe the software implementation of the building blocks.



```
add ccnx:/ udp 192.168.1.1
add ccnx:/ udp 192.168.1.3
```

Fig. 3. CCNx routing entries at Node 2 configuration file.

### A. IoT Gateway and Sensors

The IoT Gateway functionalities support sensor management, discovery and data collection over RESTful web services. A lightweight python based web framework (Flask) is running on the Raspberry Pi 2 Model B. It also has an internal database to store the sensor data as well as mapping of sensor data to consumer interest. In NDN terms, the Gateway is a 'producer'. The sensor data are collected and formatted using JSON key value pairs following the SenML recommendation [6]. Following that, a uniform mapping is created locally between the sensor data and interest to disseminate the data using NDN mechanism. This mapping is taken care by another software module that is a part of the CCNx stack.

### B. NDN Routers

Three Linux computers are configured as NDN routers since the CCNx stack currently supports only Linux. After installation, a configuration file (ccnd.conf) is created on each such routers that contains the IP addresses of the NDN nodes connected to them. For example, the ccnd.conf of Node 2 contains the following shown in Fig. 3. "ccnx:/" and "udp" refer to CCNx mechanism being utilized on top of UDP. The 192.168.1.1 and 192.168.1.3 are the IP addresses of Node 1 and Node 3 which are the immediate neighboring nodes for Node 2. The database to cache data corresponding to the interest and all CCNx security key store are automatically managed by the CCNx stack.

### C. IoT Clients

The IoT client 1 has the IP address of Node 1 in its ccnd.conf file. For uniforming naming we utilized the following URL format - "scheme://host/thingType/thingID". The 'thingType' is used in the URL since that also represents the data generated. For example, "ccnx://datatweet/tempSensor/qazxsw21" stands for an interest generated by the IoT Client 1. It is evident from 'tempSensor' that the data of interest is temperature data. The CCNx reply contains the SenML formatted data in this case. Before the generation of the interest, the client must perform a discovery[2] of available things to learn the type and unique ID. When the same interest is launched from IoT Client 2, the data is obtained from the Node 1 cache demonstrating the NDN nature of the overall system. Dissemination of raw data and high level intelligence is achieved using modifying the mentioned URL format to "scheme://host/thingType/thingID/raw" and "scheme://host/thingType/thingID/derived".

### D. Evaluation Results

We have performed early evaluation of the prototype. The combined size of python scripts and the database managing sensor data are less than 5MB. During the operation of CCNx, the CPU load is 2-6 percent on average. The tests are performed on Raspberry Pi 2 Model B. The CCNx stack itself requires 160 MB of memory but have been easily accommodated into all participating nodes.

### E. Applicability of the NDN and IoT Integration

The NDN stack integration into DataTweet IoT architecture leads to its application into several use case scenarios. For example, sensor data collected by a Smart City Cloud system will compute important events at the city. Such events can be 'showered' to subscribed consumers mobile devices using the NDN. Road and environment conditions (e.g. fog) can be disseminated using NDN to autonomous vehicles making them aware of their surroundings.

## III. CONCLUSIONS

The paper demonstrates the feasibility and interworking of NDN mechanism with a data centric IoT architecture. Our early performance evaluation in terms of memory size and CPU load point out that the prototype can work with standalone embedded devices as well. It has been assumed that each sensor is associated with an URL. But a device with multiple sensors may be addressable by more URLs. The current naming convention needs to be updated to deal with such scenarios. In future work, we aim to compare our NDN-IoT integrated stack with similar other stacks evaluate performance and scalability aspects. We are also extending our work to integrate the CCNx stack on mobile devices [8] and improve the freshness of forwarding strategies and data dynamically [3].

## IV. ACKNOWLEDGMENTS

This work has been partially funded through the French ANR research project Data Tweet (ANR-13-INFR-0008) and the EU H2020 HIGHTS project (636537-H2020).

## REFERENCES

[1] M. Amadeo, C. Campolo, and A. Molinaro. Content-centric networking: Is that a solution for upcoming vehicular networks? In *Proceedings of the Ninth ACM International Workshop on Vehicular Inter-networking, Systems, and Applications*, VANET '12, pages 99–102, New York, NY, USA, 2012. ACM.

[2] A. Bröring, S. K. Datta, and C. Bonnet. A categorization of discovery technologies for the internet of things. In *Proceedings of the 6th International Conference on the Internet of Things*, IoT'16, pages 131–139, New York, NY, USA, 2016. ACM.

[3] J. Cao, D. Pei, Z. Wu, X. Zhang, B. Zhang, L. Wang, and Y. Zhao. Improving the freshness of ndn forwarding states. In *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, pages 189–197, May 2016.

[4] S. K. Datta and C. Bonnet. Integrating named data networking in internet of things architecture. In *2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pages 1–2, May 2016.

[5] S. K. Datta, C. Bonnet, R. P. F. D. Costa, and J. Hrri. Datatweet: An architecture enabling data-centric iot services. In *2016 IEEE Region 10 Symposium (TENSYMP)*, pages 343–348, May 2016.

[6] S. K. Datta, C. Bonnet, and N. Nikaein. An iot gateway centric architecture to provide novel m2m services. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 514–519, March 2014.

[7] T. Kurita, I. Sato, K. Fukuda, and T. Tsuda. An extension of information-centric networking for iot applications. In *2017 International Conference on Computing, Networking and Communications (ICNC)*, pages 237–243, Jan 2017.

[8] X. Piao, L. Huang, K. Yuan, J. Yuan, and K. Lei. The real implementation of ndn forwarding strategy on android smartphone. In *2016 IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pages 1–6, Oct 2016.

[9] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang. Named data networking. *SIGCOMM Comput. Commun. Rev.*, 44(3):66–73, July 2014.