

User-Driven Smart Home Control System Based on Named Data Networking

Manxin Huang, Ru Li*, Jun Fan, Xin Zhang

College of Computer Science

Inner Mongolia University

Hohhot, China

cshmx@mail.imu.edu.cn, {csliru, csfj, cszhangxin}@imu.edu.cn

Abstract—Named data networking (NDN) is one of the future Internet architecture, which can well support smart home. The existing NDN-based smart home system has realized the device interconnection, but they lack of the flexibility of user control. In this paper, we have designed a user-driven smart home control system based on NDN exemplified by lighting and music player control. The system provides an interactive experience in a user-driven mode.

Keywords—named data networking; smart home; internet of things; application; NFD-Android

I. INTRODUCTION

Named Data Networking (NDN) [1] is one of the future Internet architecture. NDN is content-centric, in order to accommodating the over-growth of data in the network nowadays. NDN was introduced in 2010. It satisfies the data acquisition and sharing from the network architecture itself. NDN has begun to take shape, including the running testbed containing 39 nodes across four continents. With the development of NDN's application-driven architecture, the research requires the improvement of details, especially the development of driven and application in the actual environment, to verify and shape the architecture [2].

The advantages of using NDN for Internet of Things (IoT) have already been discussed and confirmed. NDN's native support for mobility, automatic configuration management, scalability, huge volume of short data exchange and in-network caching is proving to be highly advantageous for IoT ecosystems [3]. As a hotspot of IoT in recent years, smart home has a big impact on the transformation and promotion of people's daily life style. The main feature of smart home environment is that it contains a large number of smart ends. Smart ends are usually composed of sensors, actuators and electrical controls. The large volume of data transmission and classification produced by the sensor, and the continuous expansion of the meaning of end resources are both the prominent problem that current network cannot be alleviated. NDN has excellent natural support in terms of device scalability, data transmission and data cache.

Owing to the Advantages of NDN for Smart Home, many researchers have proposed the concepts of smart home architecture based on NDN. NDN-BMS [4] described a data-centric building management system architecture. NDOMUS [5] that contains management face, control face and data face,

implemented the NDNIoT architecture design recommendation, which they had previously proposed in [6] for the combination of NDN and IoT. Ahmed S H, et al. [7] proposed to use NDN to connect home servers, sensors, executors and private clouds to build smart home. There are also some researches conducted specifically for network protocol and network security in smart home. NDN security architecture of instrumented environment in building automation systems [8] and NDNIoT smart home network protocol [9]. Based on the developing of key techniques, the smart home system of NDN structure has appeared. De Silva U, et al. [10] designed and implemented a set of home smart lighting system. Taking office environment as the main focus of research, it realized the NDN communication between home routers, sensors and node controllers. The above scenarios emphasize the interaction between objects, but family environment emphasize more on the interaction between users and devices. As the enjoyers of smart homes, users' needs should be considered and satisfied in a timely manner. When the user wants to change the mode of the end device, it needs an interaction process between the user and the end device. User-driven NDN smart home control scheme has been proposed in previous studies. eDomus [11] allowed users remotely control home network via Facebook. Facebook is a third-party social network service platform, but users are concerned about their security. Accordingly, our system considers designing and developing mobile application to achieve the interaction with users.

In this paper, we design and implement user-driven smart home control system based on NDN. Users' need is the starting point of our system consideration. Lighting and music player are used as examples. Smartphone application collects users' demand instructions and transmits it to the smart control device through NDN. Smart control device completes the instruction operation. As a data transmission network, NDN provides support for the interaction between users and devices in the home environment.

The rest of the paper is organized as follows. Section II provides an overview of NDN and NFD-Android. In Section III, we demonstrate the design and implementation of the entire system architecture, especially in NDN design. The test results are reported in Section IV. We conclude this paper and future work in Section V.

This paper is supported by the National Natural Science Foundation of China (Project No. 61363079).

* Corresponding Author

II. NAMED DATA NETWORKING

A. NDN

NDN relies on Forwarding Information Base (FIB), Content Store (CS) and Pending Interest Table (PIT) data structures to cache and forward. Intermediate nodes in the network provide data caching. The above design ended the end-to-end communication. Caches in the network increase the reusability of data and improve the efficiency of content distribution.

Content as the first type of entity in the network, NDN uses the name to request data. Name design benefits data distinction, data expansion and data classification.

- Data distinction. Name can easily distinguish between different types of data, which implements a mapping from name to data content. For example, /user/Alice is a username, meanwhile /light/on is an operation instruction. They are easy to distinguish.
- Data expansion. With the development of external devices and the expansion of computer functions, the data has not only been limited to its original meaning. Such as /light/on, an operation instruction also becomes a data.
- Data classification. With huge volume of data, data classification is also a tricky issue. Taking an ordinary temperature control sensor as an example, it gives feedback on the temperature data regularly. Cumulative data classification is a problem that must be faced. If the two data are both generated by the same thermometer, their name prefix will be same, such as /thermometer01/201802101 and /thermometer01/201802102. This is equivalent to adding logo to the data directly, which is convenient for data classification.

NDN has two types of communication packet formats, interest package and data package. The content that requester needed is the power source. Needed content is transformed into interest packages and send into network. If an intermediate node which has content cache receives the interest package, it will return the data packet. The data packet reverses back to the requester along the path which the interest package arrived. It is worth mentioning that interest packages are lightweight.

B. NFD-Android

NDN Forwarding Daemon-Android (NFD-Android) [2] is a network forwarding daemon running on the Android system. NFD-Android implements a naming data network protocol as an overlay network over IP protocol [12]. NFD-Android provides NDN network transport support for applications based on Android platform.

NFD-Android forwards interest packets and data packets abstract lower-level network transport mechanisms into NDN faces, maintains basic data structures like CS, PIT and FIB, and implements the packet processing [13]. Applications can use faces as an interface for NFD-Android to transmit and forward

packages. NFD-Android assists applications in transmitting data, which makes applications can focus on their own design and implementation.

III. DESIGN AND IMPLEMENTATION

This section we elaborates the design structure of the system from four angles: system architecture design, NDN design, user input converts to NDN name in user end, and device end determination in device end.

A. System Architecture Design

The design of is user-driven smart home control system based on NDN follows the hierarchical structure of IoT. From bottom to top, it is divided into three levels: the perception layer, network layer and application layer. The system architecture design shows in Fig. 1. The layered system is divided into three parts: user end, NDN application design and device end.

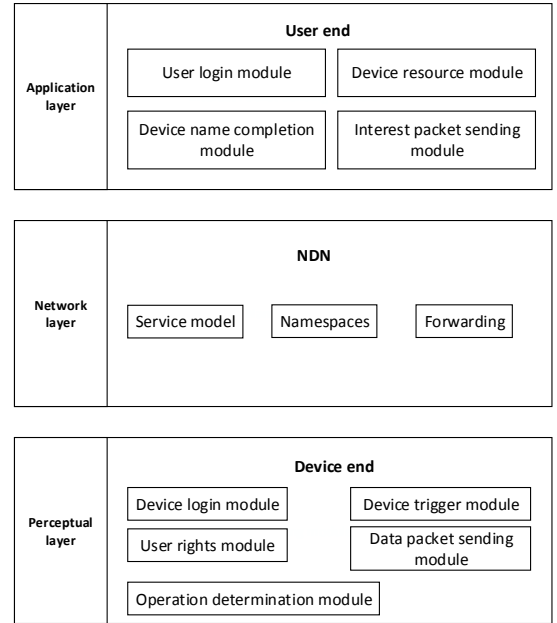


Fig. 1. System architecture design

The user end is to implement the interaction between the system and the user. The user end receives the user's input and converts the user input into a control interest packet to the network layer. According to the functions that the user end needs to implement, the system divides the user end into four modules: user login module, device name completion module, device resource module, and interest packet sending module. The user login module and the device name completion module provide the NDN name component to the interest packet sending module, which is to implement the conversion of the user input to the NDN name. The user login module confirms the identity of the currently operating user and provides an operation user for the NDN name. The device name completion module provides a complete device name and subsystem type for the NDN name based on the device name entered by the user. The device resource module displays operable devices and provides functionality that is shared with other members of

the family. The interest packet sending module is responsible for generating a control interest packet and sending the interest packet.

The NDN application is designed at the network layer to implement network transmission of the system, that is, information transmission between the user and the device. During the design of NDN applications, the system considers the service model, forwarding strategy and namespace, which implements the forwarding addressing of NDN names in the network, as well as the interaction between the user and the device. Instead of establishing an end-to-end connection, NDN uses data names to forward data. A namespace is a specification of a data name, which defines the constituent components and the meaning of the data name. Forwarding considers the forwarding mode of the router. The combination of name design and forwarding will directly affect the performance of network forwarding. The service model defines the interaction between the client and the device.

The device end acts as a controller to control the device. The device receives the control interest packet and parses the control instruction to control the device. According to the functions that the device end needs to implement, the system divides the device into five modules: device login module, user rights module, operation determination module, device trigger module, and data packet sending module. The device login module determines the device name, which uniquely identifies a device on the network. The control interest package is based on this device name to find the operating device. The user rights module records the user who can perform operations on the current device, and it provides access control authority discrimination. The operation determination module determines whether the control instruction brought by the current control interest package can be executed. When the operation determination module confirms that the operation can be performed, the device trigger module implements the operation of the device. The data packet sending module is responsible for generating the operation result data packet and sending the data packet out.

B. NDN design

1) Service model

The service model describes the interaction between the user end and the device end. In NDN, it is no longer end-to-end, nor is it one packet type. Based on the description to the data, the ends become the consumer and the producer, and the package type is divided into the interest packet and the data packet. Consumers send interest packages and producers send data packages. The system needs to determine the identity of the end and the type of package used.

Interest packets and data packets have their own advantages and disadvantages. Interest packets are lightweight, but cannot carry data. Although data packets can carry data, unless there is an interest packet request corresponding to the content name, the data packet cannot actively sent nor move autonomously between nodes.

The user's control instruction to the device needs to be sent actively, so the system uses the interest packet that can be actively sent to send the user's control instruction. At this point,

the user end becomes a consumer and needs to request data from the device. The device side becomes a producer and generates data back to the consumer. The returned data is the result of an operation that either succeeded or failed.

The data packet returned by the interest packet is one-way active. If only interest packets from the user to the device are designed, the device cannot actively send information. Therefore, the system also designs a service type from the device to the user. Its role is to transmit device status, including device abnormality prompts and data generated by sensors.

Fig. 2 describes the above service process.

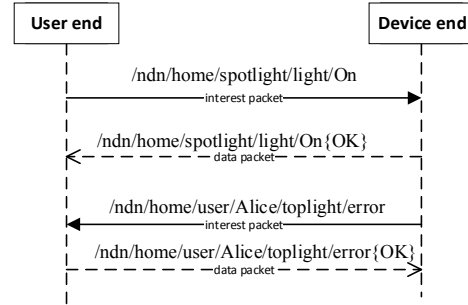


Fig. 2. Service process description diagram

2) Namespace

Interest package cannot carry data. How control operations use interest packet transmissions needs to be designed. Due to the different namespace design of different applications, the decoupling methods for transmitting data in the network are different. Therefore, data name is not transparent to the network, and system can use interest name to express the control operation. How to use interest name to express control operation is the focus of namespace design.

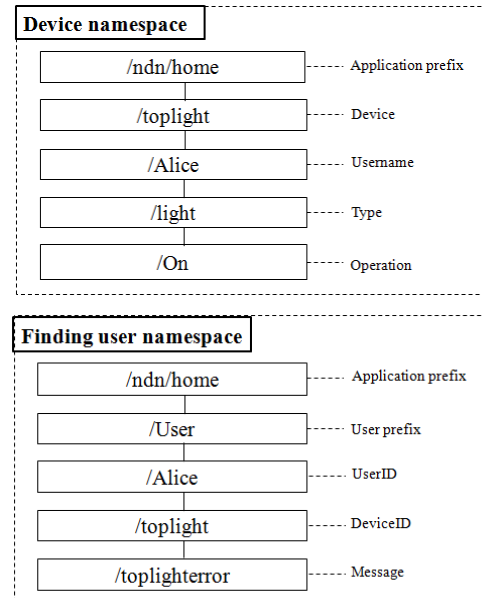


Fig. 3. Namespace structure diagram

The entire system namespace design shows in Fig. 3. The application prefix is /ndn/home. Namespace is divided into two naming subspaces for device control subspace and finding user subspace. The system uses the device control subspace to transmit the control instruction and find the target device node in the network. The system uses finding user subspace to find the target device node in the network.

a) Application prefix

The application prefix is an integral part of the content name. Its specific role is as follows:

- Application label. The application prefix is like the identity tag for interest packet and data packet with the application identity. Different data content distinguishes the specific application to which it belongs by applying the prefix.
- Data aggregation. The data that belongs same application has the same application prefix. When interest packets are transmitted in the network, the name of data is forwarded based on the previous forwarding record. The data belonging to the same application can be easily forwarded and aggregated together because they have the same application prefix.
- Local interface. In NDN, interfaces are uniformly abstracted as Face. Both the application interface and the underlying transport protocol interface are represented by Face. In the local environment the prefix is applied as Face interface to distinguish the local ownership of different interest and data packets.

b) device control subspace

The main function of the device control subspace is to transmit the control information. To control a device, the main contents required include the operator, operation object and operation type. Since the system is a smart home control system that includes multiple functional subsystems such as lighting control and music playback control, it is also necessary to add subsystem type. Therefore, the device control subspace includes four contents: Device, Username, Type and Operation.

- Device: operation object. The name of the operating device in the network.
- Username: operator. Users requesting to operate the device.
- Type: subsystem type.
- Operation: operation type. Operational specific instructions.

c) finding user subspace

The main function of the finding user subspace is to transmit device information to users. Specific device transmission information need include notification users, specific device, and specific information. Since the device control subspace and the finding user subspace are at the same level, they need to make a distinction. The main function of the system is device control. In contrast, although the finding

user subspace is indispensable, it is not the main function of the system. Therefore, the system adds a layer of user prefixes in the finding user subspace, to separates finding user subspace from device control subspace. The finding user subspace includes four contents: User prefix, UserID, DeviceID and Message.

- User prefix: it is to separates finding user subspace from device control subspace.
- UserID: notification users. The name of the user in the network.
- DeviceID: specific device. The specific device that needs to be described.
- Message: specific information.

3) Forwarding

The forwarding process in an IP network is simply a direct transmission after viewing the routing table, which is called "smart routing, clumsy forwarding". NDN uses the interest name matching forwarding strategy, which is called "smart forwarding" [14]. A reasonable name structure matching forwarding strategy can enhance the robustness of packet delivery, which can make efficient use of network resources.

The system uses BestRoute forwarding strategy. BestRoute uses the longest prefix match to select the forwarding exit. According to the forwarding method, if the design of the naming scheme can make the nodes with closer distances have longer same name prefixes, it will let the interest packet forward closer to the producer, so that the interest package will be more easy to reach the target node.

In the system, the main task of network transmission is to complete the delivery of control information. After the control interest packet being sent out, the target of forwarding is to find the control device node through the interest name. The interest package ultimately needs to be forwarded to the actual geographical location of the device node. If the name can be used to define the forwarding scope, that is to say, the interest packet forwarded by the name can more easily reach the actual geographical position of the node, this forwarding scheme will improve the forwarding efficiency.

The actual scene of a smart home is in a building while the architectural scene is structured. When it is necessary to accurately describe a device in a building, the description method is usually gradually narrowed down according to the actual building hierarchy until the device is located. In the description of construction physics data, there has been a proposal using actual building hierarchy to describe data [15]. Therefore, the system considers using a multi-level device name that conforms to the building structure to implement the mapping of the real location of the device. A device in the building can be accurately positioned using a building-floor-room-device levels. Following this hierarchical structure, the system divides the device name into four levels: construction, floor, room, and device.

The namespace structure design use hierarchy device name shows in Fig. 4. Based on the original namespace structure, the system layers the device name. The system subdivides the

Device component into four layers: construction, floor, room, and device.

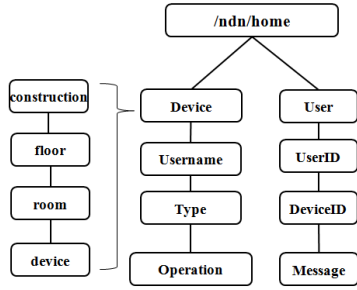


Fig. 4. Namespace structure after hierarchy device name

C. User input converts to NDN name

The system stipulates that the format of the control information input by the user is "Device Operation", and then the system needs to convert the device of user input to a complete device name.

The device name uses a strict hierarchical order and can be represented using a tree structure. The tree structure of the device name is top-down. The path of the entire device name ensures the uniqueness of the device. When searching from bottom to top based on a short name entered by the user, the uniqueness of the device path and name cannot be guaranteed. Therefore, the system introduced a referent layer to guarantee the uniqueness of leaf nodes in the tree structure.

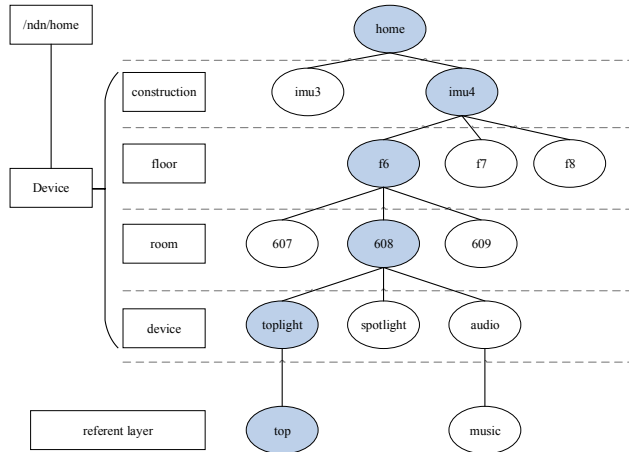


Fig. 5. Device name tree structure

The device name tree structure after adding referent layer shows in Fig. 5. /ndn/home is the application prefix, which is also the root node of the entire tree. According to the building hierarchy, the device name is divided into four levels of construction, floor, room, and device. In the case of ensuring the uniqueness of the leaf node and the device path, the leaf node can be in the device layer or the referent layer. The difference between them is whether the user has defined a referral name for this device. When a user enters a leaf node

name, the system completes the entire device name from bottom to top based on its parent node. The above description is actually implemented using a device retrieval table corresponding to the tree structure.

D. Device end determination

A series of decision conditions that the system needs to execute between the arrival of an interest packet and the execution of control operations.

- Determine whether the timestamp is correct.
- Determine whether the requesting device is current.
- Determine whether the requesting user name is in the user rights table.
- Determine whether the subsystem type matches.
- Determine whether the device state is satisfied the request operation.

IV. APPLICATION TEST

A. Test Parameters

In this paper, we mainly use 3 Android smartphones for application testing. The reason why we use three smartphones for functional testing is because three smartphones can constitute the smallest collection of NDN communications with a user end, a router and a device end. The reason why system uses this three smartphones for functional testing is because this three smartphones come from different manufacturers, and they have different system versions and CPUs, which is conducive to compatibility testing. Their device models, system versions and other parameters are shown in table I.

TABLE I. TEST PARAMETERS

Device model	MI2S	Google Nexus S	HUAWEI EVA-AL00
System version	Android 5.0.2	Android 4.1.2	Android 7.0
CPU	Qualcomm Snapdragon 600	ARM Cortex-A8	Hisilicon Kirin 955
RAM	2GB	512MB	3GB
Storage	16GB	16GB	32GB

B. Functional Test Results

Using MI 2, Google Nexus S and HUAWEI EVA-AL00, the function test examines four functions, including turn on the light, turn off the light, play the music and turn off the music. The test physical map shows in Fig. 6. The device on the right is the user end. The middle device is the router. The device on the left is the device end. The user enters "top on" in the user end. The router forwards interest package to the device end which device name is "/ndn/home/imu4/f6/608/spotlight". And then, the light on the device end are on.

The test results are shown in table II. In the three tested smartphones, all four tests were successful. Taking lighting and music player as examples, the system realized smart home control system test based on NDN.



Fig. 6. Test physical map

TABLE II. TEST RESULTS

Function	MI2S	HUAWEI GRA-TL00	HUAWEI EVA-AL00
turn on the light	Done	Done	Done
turn off the light	Done	Done	Done
play the music	Done	Done	Done
turn off the music	Done	Done	Done

C. Existing Problem

In the process of testing, NFD-Android hides into the background process after a period of time, application and NFD-Android disconnection may occur in HUAWEI EVA-AL00. This is due to NFD-Android automatically entering the dormant state after being hidden to the background process for a period of time. The workaround is to remove the background process restrictions in the developer option of smartphone settings.

V. CONCLUSION AND FUTURE WORK

In this paper, we designed and implemented a user-driven smart home control system based on NDN. The system takes the initiative control needs of users as the starting point, and uses NDN to realize remote real-time control of home devices. The system uses the device naming of the building hierarchy. This data names combine the forwarding strategies to define the forwarding range, and make it easier for interest packages to reach the actual location of the node. The system integrates the tree structure and device retrieval table to automatically complete the entire device name. This design method provides users with good interaction and reflects the advantages of NDN naming combined with application functionality. The system used NFD-Android to implement the NDN application on the Android platform, and tested in the actual environment. The test results show that the interaction between user and devices in the NDN environment is feasible in the actual environment.

Obviously, this paper is a step forward in the feasibility of NDN in the real environment. Future study will be conducted on NDN applications and home access controls.

ACKNOWLEDGMENT

This paper is supported by the National Natural Science Foundation of China (Project No. 61363079).

REFERENCES

- [1] Zhang L, Estrin D, Burke J, et al. Named data networking (ndn) project[J]. Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC, 2010.
- [2] V. Jacobson, J. Burke, L. Zhang, T. Abdelzaher, B. Zhang, k. claffy, P. Crowley, J. Halderman, C. Papadopoulos, and L. Wang, "Named Data Networking Next Phase (NDN-NP) Project May 2015 - April 2016 Annual Report", Tech. rep., Named Data Networking (NDN), Jun 2016.
- [3] Datta S K, Bonnet C. Integrating Named Data Networking in Internet of Things architecture[C]// IEEE International Conference on Consumer Electronics-Taiwan. IEEE, 2016:1-2.
- [4] Shang W, Ding Q, Marianantoni A, et al. Securing building management systems using named data networking[J]. IEEE Network, 2014, 28(3): 50-56.
- [5] Amadeo M, Campolo C, Iera A, et al. Information Centric Networking in IoT scenarios: The case of a smart home[C]// IEEE International Conference on Communications. IEEE, 2015:648-653.
- [6] Amadeo M, Campolo C, Iera A, et al. Named data networking for IoT: An architectural perspective[C]// European Conference on Networks and Communications. IEEE, 2014:1-5.
- [7] Ahmed S H, Kim D. Named data networking-based smart home[J]. Ict Express, 2016, 2(3):130-134.
- [8] Burke J, Gasti P, Nathan N, et al. Securing instrumented environments over content-centric networking: the case of lighting control and NDN[C]//Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on. IEEE, 2013: 394-398.
- [9] Bannis A, Burke J A. Creating a secure, integrated home network of things with Named Data Networking[J]. NDN Technical Report NDN-0035, 2015.
- [10] De Silva U, Lertsinsruttavee A, Sathiaselalan A, et al. Named Data Networking Based Smart Home Lighting[C]//Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference. ACM, 2016: minutes
- [11] Briante O, Amadeo M, Campolo C, et al. eDomus: User-home interactions through Facebook and Named Data Networking[C]// Eleventh IEEE International Conference on Sensing, Communication, and NETWORKING. IEEE, 2014:155-157.
- [12] Phanama Y A, Ekadiyanto F A, Sari R F. Serverless Mobile Multiuser Chat Application Based on ChronoSync for Named Data Networking[J]. Transactions on Networks and Communications, 2016, 4(4): 32
- [13] Afanasyev A, Shi J, Zhang B, et al. NFD developer's guide[J]. Technical Report NDN-0021, NDN, 2014.
- [14] Yi C, Afanasyev A, Moiseenko I, et al. A case for stateful forwarding plane[J]. Computer Communications, 2013, 36(7):779-791.
- [15] Ortiz J, Culler D. A system for managing physical data in buildings[J]. tech. rep. EECs, 2010: 2010-128.