NDN, Technical Report NDN-0035, 2015. http://named-data.net/techreports.html
Revision 1: November 19, 2015

Creating A Secure, Integrated Home Network of Things with Named Data Networking

Adeola Bannis
Dept. Electrical and Computer Engineering
CMU Silicon Valley
Moffet Park, CA
Email: adeola.bannis@sv.cmu.edu

Jeff Burke
Center for Research in Engineering, Media and Performance
UCLA
Los Angeles, CA
Email: jburke@remap.ucla.edu

***Abstract*—The Internet of Things promises to connect appliances which may come from different manufacturers, with different processing capabilities, all providing different and possibly overlapping sets of services. The very personal nature of home network data and the narrow focus of each appliance also present unique challenges in network design. In this paper, we design a simple smart home network protocol to demonstrate the ways in which the Named Data Networking (NDN) internet layer protocol provides better support for these aspects of network protocol design than the traditional Internet Protocol (IP). We refer to this simple NDN-based protocol as the Named Data Network of Things (NDNoT).**

***Keywords*-Internet of Things; Named Data Networking; security**

I. Introduction

Many household appliances and fixtures have gained processing capabilities, from washing machines and microwaves to light bulbs and door locks. Embedded microprocessors are added to enable complex behaviors, scheduling, diagnostics, and more recently, for communication. However, these microprocessors are typically much less powerful than those found in general-purpose devices such as smartphones and laptops, and are therefore known as constrained devices. They may be limited by memory, processor speed, or available energy (for battery-powered sensors), and these limitations must be considered when designing a home network protocol. The single-purpose nature of many appliances must also be considered when trying to link these devices together; any interaction between them will be more restricted than between general-purpose computers. Finally, the potentially private nature of a user's home data must also be considered.

The NDNoT protocol relieves the processing overhead on constrained devices with the use of a hub, called the *gateway* of the home network. This gateway is able to satisfy many administration or information requests without waking or contacting individual devices each time. This is possible because NDN data tends to be cached in the gateway, so devices may save power by sleeping until new data is available. The structure of NDN packets themselves also helps simplify packet handling. NDN packets are composed in Type-Length-Value (TLV) format, which has several advantages. Firstly, unknown type fields can be skipped easily, allowing for backwards-compatibility in the case of protocol upgrades. Secondly, TLV packets can be extended by nesting and concatenation without needing to be wrapped in another packet type. Lastly, the length and position of all fields can be determined without reading full values, so that one value can be extracted from a complex packet without holding the entire packet in memory.

Since appliances are mostly special-purpose devices, it is

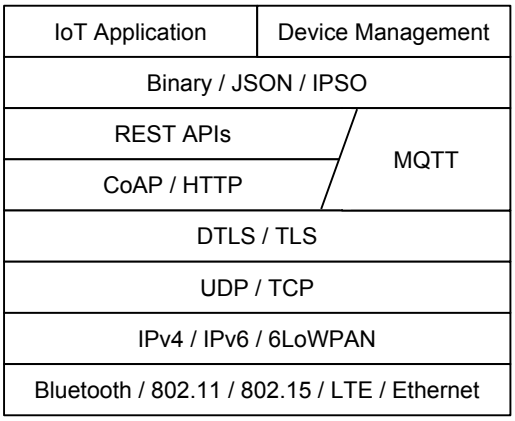

Figure 1. Typical IoT Stack

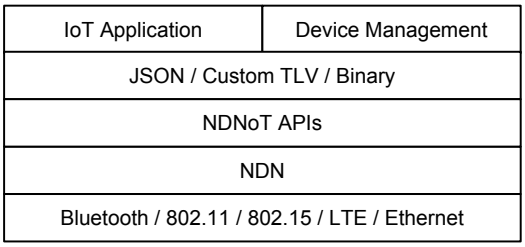

Figure 2. NDNoT Stack

reasonable to assume that a network of appliances should not operate exactly like the Internet of general-purpose computers. Each appliance has its own region of influence, e.g. locking a door, or monitoring electricity use, which is of much more importance to the user than the identity of the appliance itself. Indeed, in an ideal world, the user would not be aware of the devices in their home, but rather what data can be monitored or what appliances can be activated. In the smart home, a device's data output (where actions taken can be considered a kind of 'physical data') is much more important than its identity or network address. For example, temperature sensors and thermostats can be replaced or upgraded, but the user's intention to monitor and control temperature still remains. The smart home network appears to the user as less of a network, and more like a super-appliance comprising heterogenous devices held together by a local network. Existing IoT protocols such as MQTT are successful partly because the *subscription model* they use allows devices to be abstracted away into subscription topics (called services in NDNoT). In fact, MQTT topics are hierarchical strings, similar to NDN names[**?**]. However, MQTT must add packets to impose subscription semantics over a point-to-point communication model, whereas NDN's name-based data addressing model translates directly into a subscription model. This greatly simplifies the network stack needed to manage an NDNoT-enabled home network; Fig. 1 shows a typical IoT network stack, while Fig 2 shows the equivalent stack for NDNoT.

The unique environment of the home places important security constraints on network design as well. A person's home is, by definition, very private, as is any information about what is located or occurring in it. Exposing appliances to the Internet may allow a user the convenience of accessing and monitoring her home from any location, but it also opens up the threat of strangers (malicious or otherwise) gaining the same access. Although remote access is an important feature to many users, we have decided to limit access to the home network to users within the local area network (LAN) environment of the home. However, just as the physical home can be broken into, so can the LAN traffic be sniffed or even modified by an attacker in physical proximity. Therefore, the NDNoT protocol includes authentication for all packets, and optional encryption for increased privacy. This built-in security is NDNoT's primary advantage over other available IoT frameworks. Authentication increases the computational overhead of packet handling for all devices in the network, whether signing or verifying. However, NDN data packets are required to carry some kind of signature information; any device capable of handling NDN packets is therefore capable of handling authenticated NDN packets (in theory). This is in contrast to pprotocols such as the Constrained Application Protocol (CoAP), which does not provide authentication or privacy guarantees on its own. Furthermore, proposed schemes such as Datagram Transport Layer Security[**?**], which is an encryption-only scheme, require the original packets to be further wrapped in DTLS frames. NDNoT guarantees at least authentication for all packets, without adding another transport layer for the appliance to parse.

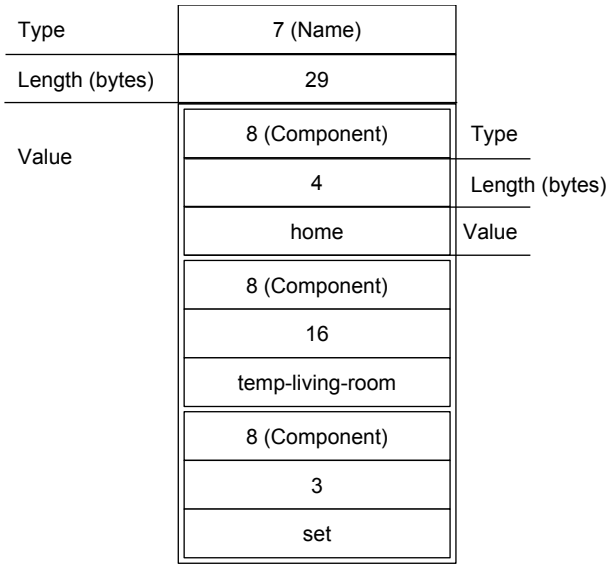

Figure 3. TLV Representation of NDN Name Address

II. Terminology

A. Named Data

Named Data Networking[**?**] is a proposed future Internet architecture that shifts the common layer of the network from host-based addressing to data-based addressing.

Named Data Networking addresses are called *names*. Names consist of segments called *components*, each of which is a sequence of arbitrary bytes, of arbitrary length. All NDN packets are composed in Type-Length-Value (TLV) format, where the values may contain embedded TLV packets themselves. For example, a name address containing three components: "home", "temp-living-room", "set", would be transmitted in TLV format as shown in Fig. 3. The segments of an NDN name are taken to form a hierarchical structure; that is, data packets which share a name prefix are expected to be related in some way. NDN names are typically written as URIs in order to emphasise this hierarchical structure, e.g. */home/temp-living-room/set*.

NDN uses a request-response model, where request packets are called interests and response packets are called data. A simplified view of the contents of these packets is given in Fig. 4. There are other optional parameters that can be included in either packet type to help with forwarding or caching. Both interest and data packets contain name addresses, which are used to pair interests to data. NDN data packets are immutable, and as data packets are retrieved

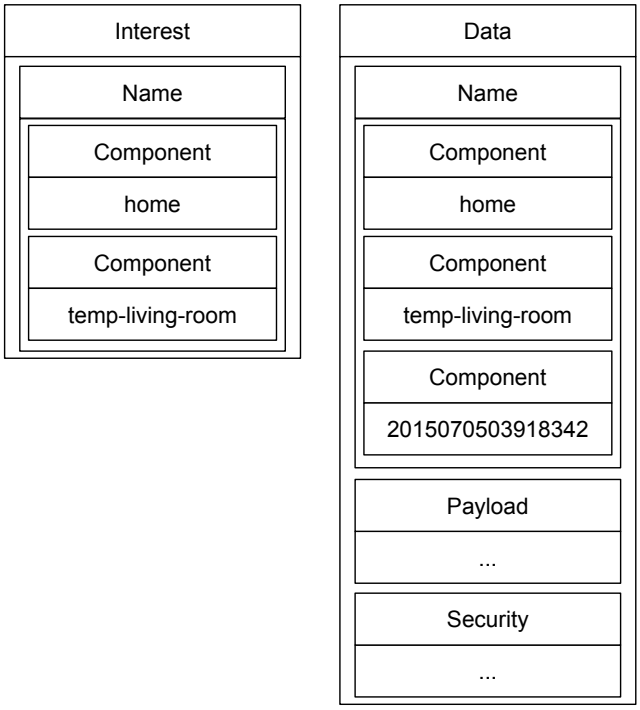

Figure 4. Simplified NDN Packet Contents. The length field is omitted, and types are given as strings.

by name only, the mapping of NDN name addresses to data packets is immutable. That is, two data packets with the same name address must be bitwise identical. This is necessary for caching of packets, so that interests for the same name will return consistent data to all consumers.

However, it is not necessary to know the full name of a data packet in order to retrieve it. Interests are then routed based on longest prefix matching. This means that a listener is able to respond to interests that fall below its prefix name in a hierarchy. For example, if a device registers the prefix */a/b/c*, it will receive interests for */a/b/c/d* or */a/b/c*, but not for */a/b* or */a/b/c1*. Data is also routed by longest prefix matching. If the device listening at */home/kitchen/fridge* receives an interest for */home/kitchen/fridge/temp*, it can respond with data named */home/kitchen/fridge/temp/now*, but not with */home/kitchen/fridge/status*. The advantages of longest prefix matching, rather than exact matching, come into play when a service must monitor a changing resource, i.e. when different content is a reasonable response to identical interests. Instead of allowing for mutable data, which would prevent caching, version numbers or timestamps can be appended to the returned data packet name as suffix components to differentiate data packets. An added benefit is that any cache that holds responses to */home/kitchen/fridge/temp* can automatically collect an ordered log of previous values with no additional configuration.

One final advantage of NDN name addresses is that the 'raw addresses' can be chosen to be human-readable, and even contain semantic information. The hierarchical nature allows users to express their internal view of the smart home, which in turn can help developers of IoT applications discover usage patterns or create integrated services.

B. Network Components

The physical NDNoT home network consists of the *gateway* and *appliances*. The logical network abstracts all the capabilities of both device types into network *services*, which *applications* can use alone or in combination to perform tasks for the user.

Most of the network logic for the NDNoT protocol is contained in the *gateway*. The main functions of the gateway are to

- act as a router and cache for NDN packets,
- manage the addition and removal of devices and services,
- create, distribute and revoke security credentials, and
- provide a directory of available services.

All other devices connected to the home network are considered to be *appliances*. As physical objects, they may serve multiple functions, some of which may overlap with other appliances. However, the smart home super-appliance is only concerned with the ability to monitor some aspect of the home, e.g. room occupancy or temperature, or to perform some action to change the state of the home, e.g. washing dishes in a dishwasher. Appliances are therefore replaced in the NDNoT network model with the services they provide, and each service is given a distinct NDN name address. The gateway also provides some basic administration services, including a directory of all other available services.

Services are not simple mappings of names to devices. It is possible for multiple appliances to provide the same service, with the same name; for example, a smart lightbulb and a smart alarm clock may both be able to report the ambient light level in a room, and therefore should provide this data under the same service name. The gateway maintains a mapping of human-readable keywords to service names, allowing applications to locate the services they need. The more sensitive services in the smart home can be protected with authentication or even encryption, while less-sensitive services requiring faster access can be left unauthenticated.

Applications, while not traditionally considered a network component, are integral to the NDNoT network model. They combine the various monitoring and action services provided by appliances to create more complex behaviors and contexts to help the user understand and manage the state of the house. For example, a thermostat application may monitor not just the ambient temperature of the room it controls, but the temperature in adjacent rooms, the occupancy status of the room and adjacent rooms, and even the ambient light level, as the user may enjoy cooler temperatures for sleep. Another power management application may monitor the

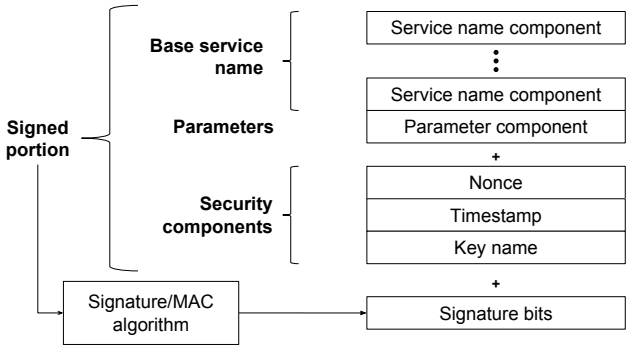

Figure 5. Composing a Command Interest Name

thermostat and other appliances' behaviour and offer the user energy-saving tips. Applications may have an associated name address and security credentials, reflecting their status as first-class members of the NDNoT network model.

III. Security

Although NDN provides per-packet security fields, it is still up to a network operator to choose a signature or message authentication algorithm and define the best security model for her network. The default authentication method in NDN networks is RSA public key authentication[**?**]. Public key authentication first requires a trusted certificate authority to issue certificates for new services. By restricting the smart home network to an intranet, we remove the need to establish trust in an external entity. Instead, the gateway acts as the certificate authority for the home network.

When a new device is added to the network, however, it still needs to locate a trustworthy root certificate. A malicious device may be able to register the name address of the root certificate and intercept the interest-data exchange. To prevent the new device from trusting a bad gateway, user intervention is required, in the form of a pairing code. The new device displays a pairing code, which the user must enter into the gateway. This pairing code is also used to derive key for use in hashed message authentication codes (HMAC). This HMAC key is used to sign all the initial (onboarding) messages between gateway and appliance, and ensures that the appliance that is being added is the one the user intended.

Once the appliance has been added to the network, it needs to generate an RSA public/private key pair, which will be used to identify the appliance and its services. In particular, the public key will be used to sign any further messages between the appliance and the gateway. The public key is presented to the gateway, which creates a digital certificate that can be downloaded by any appliance or application in order to authenticate messages from the new appliance. Figure 6 gives an overview of the onboarding process.

In order to prevent an attacker from issuing commands, appliances must be able to ensure that the interests they receive are valid. However, interest packets (as of time of writing) do not contain digital signature fields as data packets do. Previous work has addressed this with authenticated interest names[**?**], created by supplementing a base name with an extra component derived from some device-internal state and a digital signature or MAC. The NDNoT extends this approach in its own authenticated interest format. First, an optional parameter component is added to the base service name, to provide any data the service may need, such as a target temperature. A timestamp and nonce are appended after the parameter component to prevent replay attacks. The next component, the key name, depends on the choice of security model. When RSA public key signatures are used, it is an encoded name address that can be used to download a digital certificate for the public key used to sign the command. In the NDNoT protocol, HMAC is also available for authentication, and in this case the key name is assigned by the gateway to map appliances to HMAC keys. Once these three extra components are appended to the original name address, the entire name is signed (or an MAC is computed), and the digital signature (or MAC) bits are appended as a final component. The recepient can then authenticate the interest by locating the key named in the key name component, splitting off the final name component as the signature/MAC, and verifying with the rest of the name.

For comparison, we consider one of the few IoT solutions that incorporate security, the AllJoyn framework. AllJoyn's onboarding process begins by allowing appliances to connect to an intermediate soft access point (AP), which is then used to distribute credentials for the password-protected home network[**?**]. However, the current specification does not seem to authenticate the messages between the soft AP and the onboarded device. This enables two man-in-the-middle (MITM) attacks: one where the appliance is fed incorrect credentials to force it to connect to another AP, and a more serious one where a malicious appliance inserts itself into the home network using sniffed credentials. In fact, the AllJoyn Security 2.0 framework only governs interactions after the initial onboarding handshake[**?**].

The NDN data (response) packets already contain built in security fields, which are used in the NDNoT protocol for authentication. These fields include a digital signature or MAC computed over the rest of the data packet. Authentication may not provide sufficient security for all services, however, as data about the home can be very private. The NDNoT protocol also allows for encrypted data payloads. In this case, the public key named in the data security field is used for RSA PKCS #1 decryption instead of signature verification. It is also possible to use 128-bit AES encryption to support more constrained devices, but this encryption scheme is not exposed in the current NDNoT implementation.

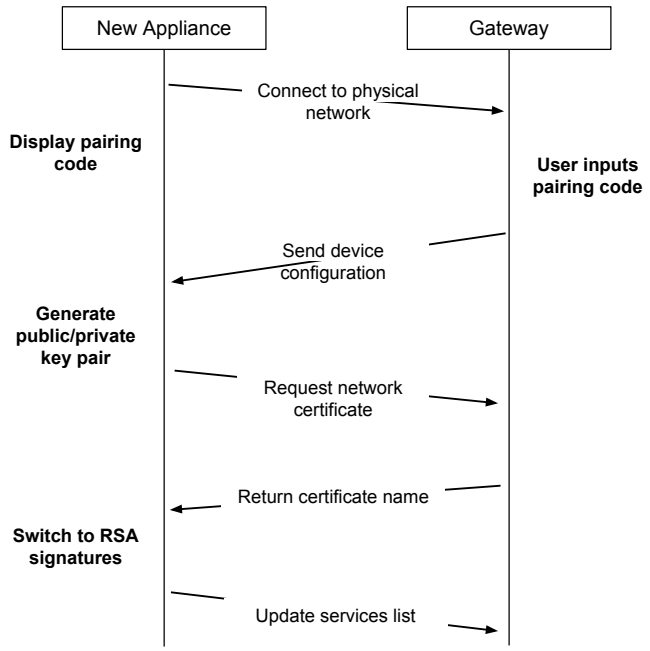

Figure 6. NDNoT onboarding flow

```
[
  {
  "tv" : [
       {
          "name" : "/home/living_room/tv/set_channel",
          "signed" : true
       },
       {
          "name" : "/home/living_room/tv/set_on",
          "signed" : true
       },
       {
          "name" : "/home/living_room/tv/status",
          "signed" : false
       }
    ],
  "presence" : [
       {
          "name" : "/home/living_room/presence",
          "signed" : true
       },
       {
          "name" : "/home/bedroom/presence",
          "signed" : true
       }
    ],
    ...
  }
]
```

Figure 7. Partial NDNoT directory listing

Of course, this security model still cannot thwart all attackers with access to the physical network. Anyone who connects to the physical network is still able to read all unencrypted packets, although he cannot modify or spoof them. Due to the lack of access control in the current implementation, an attacker with physical access to the gateway itself can also install a malicious appliance to enable him to issue commands to the rest of the network. However, physical access to a user's home is a much more serious problem than network access alone, and cannot be prevented by any network design.

IV. Integrating Services

The NDNoT protocol is only useful if it enables us to create rich IoT applications more easily than other available IoT protocols. We will therefore highlight some of the most important network management operations as they compare to other available solutions.

In order to provide its services, a new appliance needs to be admitted to the smart home network, often referred to as *onboarding*. The new appliance must first join the physical network created by the gateway. The current NDNoT protocol assumes an open home WiFi network. Although the appliance now has access to the physical network, and may be able to sniff packets, it has no authentication credentials and therefore no way to access the services of the other appliances in the network. The user must indicate to the gateway that a new appliance is ready to be added in order to begin the onboarding process. The rest of the onboarding process involves creating and certifying the appliance's security credentials, and providing the gateway with a list of services to cache (Fig. 6).

Once appliances have been added to the network, they need to be discoverable by applications. The gateway provides a directory service from which an application can fetch a listing of *capabilities* and associated service names (Fig. 7), compiled from the service lists presented by appliances at the end of the onboarding stage. Capabilities are human-readable strings whose main purpose is to group related services. There is currently no standard for capability names, but eventually best practices will emerge as IoT network design matures.

The current NDNoT implementation does not require appliances to describe the parameters taken by a service name, nor to describe the data type of the responses. All responses are currently given in JSON, a simple and widely used data format.

V. Discussion

A. Implementation

The NDNoT protocol has been implemented in Python as part of NDN-Pi, a smart home toolkit for the Raspberry Pi. This toolkit is intended to allow researchers and developers to familiarize themselves with NDN using real-world hardware. It also provides examples that can be extended to create new service types. The current examples include simple lighting control, data monitoring and collection, and

a full-loop example that switches a television on or off depending on room occupancy.

Source code and system images for the Raspberry Pi implementation of the NDNoT protocol are available at https://github.com/remap/ndn-pi.

B. Future Work

The primary security concern with HMAC-SHA1 is the need to refresh authentication keys periodically. A key update protocol is a vital next step to further securing the NDNoT protocol.

An AES-based encryption key naming scheme needs to be exposed for use by the most constrained appliances, which cannot perform RSA PKCS #1 encryption.

Once constrained devices are guaranteed secure authentication and encryption schemes, we may begin to explore finer-grained access control, to prevent applications from gaining access to unneeded, private data.

The certificate delivery system currently requires constrained appliances to serve their digital certificates on request, and although they may be cached in the gateway, it is still unreasonable to expect all devices to maintain full digital certificates. The naming scheme for network certificates can be modified to place all digital certificates under the gateway's hierarchy, and therefore have all certificates served from the gateway.

VI. Conclusion

The NDNoT protocol outlined here is very simple, but manages to support essential IoT network management operations, including onboarding of new appliances, service discovery, and application development. It requires a greatly reduced network stack due to the reliance on NDN's special transport semantics, which translate well to a network of things. The use of NDN also provides security guarantees that force the use of additional transport layers in other IoT frameworks. Using the NDNoT protocol, developers can create frameworks and applications to easily connect a variety of appliances to form a smarter, more user-centric Internet of Things.

Acknowledgments

Partial support for this work was provided by Qualcomm Research. Additional support was provided by the National Science Foundation (award CNS-1345318).