Named Data Networking

Jack Joseph Gilbride
School of Computer Science and Statistics
Trinity College Dublin
Student Number 17340868

Abstract—**Named Data Networking (NDN) is a network architecture, proposed as a replacement to the modern Internet Protocol (IP) architecture. NDN takes a fundamentally different view of the Internet from IP; as a data distribution network rather than a communication network. For this reason the architecture is designed to distribute data identified only by name, rather than an endpoint representing its location. To achieve this, the NDN protocol stack takes some concepts from the IP stack but makes some radical changes, particularly from the network layer-up. This implementation has a positive impact on many areas of telecommunications, such as data caching, network utilization, protection against attacks and security of data. While still primarily a research project, proof-of-concepts have shown NDN to not only work in, but benefit in, many practical applications; particularly in future trends such as smart vehicles, video conferencing and the Internet of Things.**

I. Introduction

Named Data Networking (NDN) is a proposed architecture for the future Internet, intended as an improvement over the modern Internet Protocol (IP) architecture [1]. Within the IP architecture, data is referenced by location. To transfer data from one endpoint to another, a packet must contain the location (address) of its source, and its destination. In an NDN architecture, data is not referenced by location, but by a unique name which identifies that data [2].

NDN takes advantage of the fact that most Internet applications request content by name. These names, created at the application layer, are used in the network layer to identify that content.

The contents of this paper will explore the key ideas of NDN. It is designed to be tutorial in nature, to make the topic comprehensible to readers outside the specialty of the article. First we will give an overview of how NDN works; in particular the contents of the NDN protocol stack, how data is transferred, and how data is named. Next we will give an overview of security and privacy within NDN, including its key security features and resiliency against common attacks. We will move on to look at how NDN is being used in the real world, followed by a conclusion to sum up the findings of the paper.

II. Named Data Networking: Overview

By replacing identification of data objects from host location to given name, NDN shifts the semantics of the Internet from a communication network to a data distribution network. All data is requested and identified as named data; there is no reference to the location of communication endpoints.

A. The NDN Stack

When thinking of modern Internet protocols, we think of a five layer protocol stack; the physical layer, the link layer, the network layer, the transport layer and the application layer. Each of these layers contain the protocols to transmit data at increasing levels of abstraction. The network layer is the thinnest layer as it contains only one protocol; IP. Hence the modern Internet architecture is known as the IP architecture.

The NDN protocol stack retains the hourglass shape of the modern Internet stack. In the middle is the NDN network layer protocol, responsible for requesting and receiving data by name. This sits on top of the familiar layer one (physical) and layer two (link) protocols. Additionally, layer two contains TCP/UDP/IP tunnels, responsible for connecting remote NDN networks that can only reach each other via the modern Internet. Above the network layer sit simpler application and transport protocols, due to the network layer's direct use of the application namespace.

This paper will focus on the NDN network layer around which the architecture is built. It will also touch on the application layer when relevant.

B. Data Transfer

There are two types of packets at the network layer used to request and transfer information.

- NDN *Interest* packets are used to request data. These contain the name of the data being requested.
- NDN *Data* packets are sent in response to a request, containing the requested information. A *Data* object is immutable, meaning it cannot be changed after creation. If it is changed, the new version must be given a new name. If a *Data* object is too large for a packet then it is segmented and the segment number becomes part of the packet's name.

Both of these packets refer to *Data* objects solely by name, there is no reference to its source or destination location.

To transfer *Interest* and *Data* packets, network layer devices called NDN nodes are used. These nodes contain three basic components:

- The *Content Store*. This is used to temporarily cache packets that the node has received.
- The *Pending Interest Table*. This is used to store all *Interests* that a router has forwarded but not satisfied yet.
- A *Forwarding Strategy*. This includes the *Forwarding Information Base* which contains mappings from *Data* names to the interfaces to forward them to.

When a Node receives an *Interest* packet, it first checks the Content Store to check if it contains the corresponding *Data*. If so, it responds with the *Data*. If not, it checks whether the name is in the Pending Interest Table. If the name is already here, then the node has already forwarded that *Interest* on the network from a different requester, so the new requester is simply added to the list of requesters for that entry. Otherwise, the *Interest* is added to the table and forwarded on the network based on the Forwarding Policy and the entries in the Forwarding Information Base.

When *Data* is forwarded in response to an *Interest*, is is forwarded hop-by-hop over the reverse path of the *Interest*. When there are multiple requesters of this *Interest*, the *Data* is forwarded to each of them. After forwarding the *Data*, the node may cache it in its Content Store to serve future requests. Caching is possible due to the immutability of *Data*; as *Data* can never be changed it will always be the "up to date" version, no matter what physical location it is read from.

C. Naming

As the core concept of NDN is the referencing of *Data* by name, it is important to understand exactly what forms names can take.

NDN nodes attribute no meaning to *Data* names; names are opaque to the network. This allows application layer programs to chose naming schemes independently of the network. Thus naming schemes can evolve naturally.

While names are opaque in NDN, some structure is assumed. A hierarchical name is made up of a number of components separated by '/'. An example is a video produced by UCLA which might be named /ucla/videos/demo.mpg.

Hierarchically structured names have a number of benefits:

- Application layer programs can represent relationships and contexts of named *Data*, e.g. the group to which it belongs.
- A *Data* object too big for one packet can be broken up into segments, and its segment number captured in the hierarchy.
- It allows for name aggregation, e.g. the first component of a name could correspond to the automated system that produced the *Data*.

For NDN to be a suitable replacement for the modern web architecture, users must be able to fetch data without having previously seen its name. The first solution for this is a deterministic naming algorithm on the side of both the *Data* producer and consumer, that would enable them to create the same name based on information available to them. The second solution involves a concept known as an interest selector, which uses longest prefix matching across one or more iterations. Here a consumer sends a prefix, i.e. the first n components of the data name, along with an interest selector, which is some metadata to determine which packet should be sent back out of the ones which match the prefix. The producer will respond with *Data*. The consumer can then fetch more *Data* using information learned from the previous packet.

To avoid the wrong *Data* being fetched, names must be unique in the context that they will be used. I.e. *Data* that will be referenced globally must have a globally unique name. But *Data* that will only be referenced on a local NDN network need only have a unique name locally.

Namespace management is not part of the NDN architecture, and it is up to those building NDN applications to define namespaces. This has the benefit of increasing the closeness of mapping between data names at the application layer and the network layer, which makes the entire process simpler overall.

III. Privacy and Security

A. Inherent Protection Against DoS & DDoS

Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) are common and notorious types of attack on the Internet. These attacks are generally easy to instantiate and can be difficult to mitigate. For this reason it is important to analyse their effect on NDN and see what measures need to be taken to prevent them [3].

A common form of DDoS attack is Bandwidth Depletion, which floods its victim with IP packets to saturate their network or server resources. Thankfully, due to the intrinsic nature of NDN, this attack would be limited. Once requested *Data* is initially produced, it would be cached at the NDN nodes through which it is forwarded. Subsequent *Interest* packets would retrieve the *Data* from these nodes, thus not overloading the intended victim.

Another form of DDoS is a Reflection Attack, in which numerous forged IP packets are created with the source address of the intended victim. These packets are sent to one or more separate hosts. These hosts will then route all responses to what they think is the sender. The content of the packets sent to the host is chosen so that the content of the responses will be much larger, thus overloading the intended victim. Again the intrinsic nature of NDN would limit the effect of this, as all *Data* responses are always sent along the channel that the *Interest* came from, thus the original sender would be the one to receive each of the responses, effectively overloading itself.

A third form of attack is a Prefix Hijacking attack, where an autonomous system advertises invalid routes to other routers. As a result the routers forward their traffic through the system to this invalid route, and this traffic is simply discarded. This result is known as "black-holing". Once again intrinsic features of NDN make it resilient to this. Routing updates are signed and verifiable, and *Data* being forwarded back along the path of the *Interest* means that a node can detect black-holing. This enables NDN to try alternative paths, and, as *Data* was cached in previous nodes, it can be recovered.

B. Cryptographic Security

A vital part of any modern telecommunications system is the secure transfer of data. For NDN to be a suitable architecture for the future Internet, its users need to be able to encrypt *Data* so that it can only be decrypted by its intended recipients. Those who receive the *Data* must also be able to check whether it comes from a reliable source. A security framework that deals with these requirements is not only part of NDN, but is a core concept of it. This framework is built on public key cryptography [4]. There are three concepts at the core of this framework:

- *Digital Keys*. Keys are treated as named *Data*, meaning they are retrieved with *Interest* packets and returned in *Data* packets.
- *Certificates*. Issued as an endorsement between the name and the public key. A *Data* packet carrying a public key.
- *Trust Policies*. Defined by applications. State which entities can be trusted to produce which *Data*, and in what cases specific keys should be used.

To utilize these concepts, a system must be able to verify the authenticity of a certificate with some form of "trust anchor". NDN assumes that the authority of each network establishes its own trust anchor. All devices under that authority have access to this trust anchor; it is either pre-configured into them or they can access it through some secure means outside the NDN network. An application with a name can apply for a certificate from the trust anchor, which certifies the application's ownership of that particular name (i.e. certifies that the application is who it says it is).

After obtaining a trust anchor, an application can obtain trust policies from trusted sources. Trust policies can be defined using a trust schema, which is simply named *Data* containing the policies. To collect trust policies, an application must have a default trust policy (i.e. only accept trust policies signed directly by the trust anchor).

So trust anchors provide the ability for entities to attain certificates, to show that they are trustworthy, and to attain trust policies, so that they know who to trust. These provide application developers with a secure framework to share *Data* in NDN.

To help to realize this framework, developers can utilize the concept of structured naming in NDN. Keys and certificates are constructed as delimited entities that can be parsed for specific information. The convention for naming a certificate is the following:`/<prefix> /KEY /<key-id> /<issuer-info> /<version>`. Here we see a practical use of structured naming in NDN: an application can accept a certificate like regular named *Data* and parse out the public key and any other relevant information.

The use of encryption keys, trust anchors, certificates and structured naming enables applications to create authenticatable *Data*. When a packet is received, its receiver validates it by checking it against its trust policies; i.e. if the packet carries the appropriate name and key name, and also if that key name is valid for that particular *Data* name. Following validation by trust policy, the receiver must then verify the signature. This involves retrieving the certificate *Data* packet for that particular signature in the network. This certificate points to the certificate of its issuer, and so on until a trust anchor is reached. If each certificate in this chain is valid, and the packet has passed the trust policies, then it is considered valid.

Interest packets are not signed by default, but may be signed when necessary to do so, i.e. when the authenticity of the requester must be determined before *Data* is sent back.

On top of allowing for *Data* authentication, NDN's security model also enables *Data* confidentiality. Again utilizing the concept of structured naming, this can be achieved through Name-Based Access Control.

We can illustrate the example using a simple producer-consumer model. A trust anchor for the network generates two keys:

- A public, plaintext *key encryption key (KEK)*
- A private, encrypted *key decryption key (KDK)*

The producer generates one key:

- A symmetric *content key (CK)* for content encryption

The producer encrypts the content with the *CK*. It then fetches the *KEK* from the trust anchor, which it uses to encrypt the *CK*. So the producer has produced two pieces of content; the *CK*-encrypted-*Data* and the *KEK*-encrypted-*CK*.

The consumer wants to consume the *Data* produced by the producer. It expresses an *Interest*, and the producer responds with the *CK*-encrypted-*Data*. It then fetches the *KEK*-encrypted-*CK*. It wants to decrypt the *Data*, but first it must decrypt the *CK*. The producer combines the *KEK*, the encrypted *CK* and its own name into an *Interest* packet for the trust anchor. If the trust anchor deems that the consumer's name is valid and is entitled to decrypt the *Data*, it sends back the corresponding *KDK*. The consumer derypts the *CK* with the *KDK*, and the *Data* with the *CK*.

As mentioned at the start of the paper, the point of NDN is as an improvement over the IP architecture. In terms of security, NDN provides the following improvements:

- As each *Data* item is secured directly, it can be cached anywhere in the network, regardless of whether the cache location is trustworthy. In other words, provided the *Data* packets themselves are secured properly, they can be cached anywhere, reducing the complexity and improving the efficiency of data distribution.
- NDN certificates are also represented as *Data* which leads to similar efficiencies. Furthermore, a sequence of certificates which eventually point to the trust anchor's certificate can be stored together in an NDN certificate bundle, meaning they can all be accessed at once, further improving efficiency.
- The focus on securing *Data* rather than connection channels makes the information more secure; e.g. data could be altered before entering a channel, and loses cryptographic protection once it leaves it. With NDN, the *Data* is protected from creation at the producer's application layer, to being processed at the consumer's.
- The concept of a local trust anchor for a system means that users have more confidence in certificates as they are not issued by some commercial entity, but rather some entity with a view of the relevant network.
- Trust policies based on structured naming mean that *Data* from an entity not just based on the entity's identity, but

other factors such as the type of *Data*, and whether an entity has the right to produce it. This gives more granular control over security policy.

IV. NDN In Practice

Named Data Networking is a growing area of research in telecommunications. It is a primary research area for sixteen investigators funded by the National Science Foundation (NSF) across twelve campuses [5]. A global NDN test bed for this research is formed by more than 30 institutions. This research covers a wide range of topics, including routing, forwarding, driver applications, security, privacy, implementation and deployment [6].

One such example is the application of NDN to the Internet of Things, known as the Named Data Networking of Things [7][8]. Research into this area has shown that NDN's minimal gap between network and application layers, as well as its data-centric security model, make it a natural fit to this area where properly securing the network is a challenge and overhead must be kept low. An application of the Named Data Networking of Things could be a smart home, with a number of devices (things) with *Data* to share. These devices could share their *Data* in the network, taking advantage of the architecture's native encryption and verification features [9]. This home network could be connected to an outside network (e.g. the Internet) from an access point using IP/TCP/UDP tunneling as mentioned earlier in the paper. This would allow external devices with the appropriate permissions (e.g. the homeowners smartphone when they are away from home) to interact with the IoT devices using NDN.

Practical research of NDN has been applied to many other areas of telecommunications. One such example is the design, implementation and testing of real time videoconferencing over NDN through C++ [10]. Another is the design, implementation and testing of realtime streaming over NDN in Python [11]. A third is the design and demonstration of smart vehicle internetworking with NDN, known as V-NDN [12]. Each of these are hot topics in technology at the moment, so when looking at the potential of NDN as a future Internet architecture, its successful application to these areas looks promising.

While there are plenty of examples of NDN being successfully applied to real-world scenarios, there is still plenty of ongoing research to realize it as a replacement to the IP architecture. For this reason many of the so called "real-world" scenarios have been implemented with research in mind. These scenarios are interesting test cases, as they remove some of the abstraction from NDN and show practically how it can work in the modern world.

V. Conclusion

This paper aimed to give an oversight to the core concepts of Named Data Networking. It has laid out NDN as an architecture; a protocol stack which replaces IP with the NDN network protocol. This network protocol references *Data* by name, with no reference to communication endpoints. These names are structured, but are opaque to the network, and their management is left to applications which use the architecture. These names are used to fetch information with *Interest* packets, which are forwarded by NDN nodes according to their Content Store, Pending Interest Table, Forwarding Strategy and Forwarding Information Base. The information is returned in a *Data* packet, which takes the exact inverse path of the *Inverse* packet. This architecture makes NDN intrinsically resistant to common security attacks, and NDN's use of public key cryptography build on this to provide a solid security framework for the distribution of *Data*.

Having examined NDN, it seems to be a serious contender for the architecture of the future Internet. While there is still a lot of research to do before it can be implemented, the reasoning behind it is solid, and each of the concepts explored provide a great benefit over the current IP model. It is clear that the researchers behind NDN are taking tried-and-tested concepts from IP, such as a skinny network layer, the distribution of information in packets, and the same link layer and physical layer protocols. However, while taking inspiration from IP concepts, it is clear that NDN is a clean-slate approach at designing an architecture, fundamentally changing how we think of the Internet from an end-to-end communication network to a data distribution network. NDN benefits from taking a fresh approach to how we think of the Internet's architecture, using the strengths and weaknesses of IP to build something much better suited to the modern world. It certainly seems like NDN will become an option for networks in the future; the use of IP/UDP/TCP tunnels in the link layer means that, for example, your home network could be set up as an NDN network, with the outside link through IP meaning that your devices could still communicate with external networks.

Overall I see a very promising future for NDN and will be sure to follow the research to see if becomes a viable contender for the architecture of the entire Internet.

References

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, kc claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," in *ACM Computer Communication Review*, July 2014.

[2] A. Afanasyev, J. Burke, T. Refaei, L. Wang, B. Zhang and L. Zhang, "A Brief Introduction to Named Data Networking" in *Military Communications for 21st Century (MILCOM 2018)*, October 2018

[3] P. Gasti, G. Tsudik, E. Uzun and L. Zhang, "DoS & DDoS in Named Data Networking" in *Proceedings of ICCCN 2013*

[4] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev and L. Zhang, "An Overview of Security Support in Named Data Networking" in *IEEE Communications Magazine*, November 2018

[5] Named Data Networking: Next-Phase Participants [*Online*], Available http://named-data.net/project/participants/

[6] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J.D. Thornton, D.K. Smetters, B. Zhang, G. Tsudik, kc claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley and E. Yeh, "Named Data Networking Project" [*Online*], Available http://named-data.net/techreport/TR001ndn-proj.pdf, October 2020

[7] W. Shang, A. Bannis, T. Liang, Z. Wang, Y. Yu, A. Afanasyev, J. Thompson, J. Burke, B. Zhang and L. Zhang, "Named Data Networking of Things" in *Proc. 1st IEEE Intl. Conf. on Internet-of-Things Design and Implementation*, April 2016

[8] W. Shang, Z. Wang, A. Afanasyev, J. Burke and L. Zhang, “Breaking out of the cloud: Local trust management and rendezvous in Named Data Networking of Things” in *ACM/IEEE International Conference on Internet-of-Things Design and Implementation (IoTDI)*, April 2017

[9] Y. Li, Z.Zhang, X. Wang, E. Lu, D. Zhang and L. Zhang, “A Secure Sign-On Protocol for Smart Homes over Named Data Networking” in *IEEE Communications Magazine*, July 2019

[10] P. Gusev and J. Burke, “NDN-RTC: Real-Time Videoconferencing over Named Data Networking” in *2nd International Conference on Information-Centric Networking (ACM ICN), San Francisco, CA*, September 2015.

[11] D. Kulinski, J. Burke and L. Zhang, “Video Streaming over Named Data Networking” in *Multimedia Communications Technical Committee IEEE Communications Society E-Letter*, Vol.8, No.4, July 2013

[12] G. Grassi, D. Pesavento, L. Wang, G. Pau, R. Vuyyuru, R. Wakikawa and L. Zhang, “Vehicular Inter-Networking via Named Data”, *ACM HotMobile 2013 Poster*