ndn.link

NDN Link Whitepaper

An information–centric and a self–organizing network.

NDN Link

Content

NDN Link

Introduction

As one of the important infrastructures supporting the development of modern society and technological advance, the Internet sees a development speed in scale beyond people's imagination. The content on the Internet shows an explosive increase and such a trend is accelerating. According to a report issued by Mary Meeker in 2018, the number of Internet users in the world hits over 3.6 billion, outnumbering 50% of the global population. Meanwhile, mobile videos and emerging contents such as Netflix are so attractive that users worldwide spend up to 5.9 hours daily on the Internet. As the Internet sees a growing number of users and constant expansion of business, and new application modes (like 4K/8K HD videos, AR/VR, industrial Internet, the Internet of Things and social network services) emerge, the original TCP/IP-based conventional Internet architecture designed to meet the needs of single data communication has exposed some inherent problems and limitations and fails to adapt to sharp expansion of network size in terms of scalability, controllability, security and mobility, etc.

Today, the Internet has logically become an information-centric networking (ICN), requiring efficient, large-scale and secure access, sharing and distribution of contents. Nevertheless, the conventional infrastructure, i.e. the communication model of IP network architecture, fails to fully meet such a requirement. The IP network architecture, with the design concept used in the 1960s when it was created, is the location-based end-to-end communication aiming to meet the needs of end-to-end data transfer. Such inconsistency brings about a lot of problems, especially the data exchange among various types of networks. The Internet with TCP/IP protocol as the core technology is faced with an increasingly grave challenge, showing a number of incompatibilities in network scalability, security, reliability, flexibility and mobility.

In recent years, almost all countries in the world have been focusing on how to design a brand-new Internet architecture. In this context, information-centric networking has become one of the mainstream research directions, but many researches stay at the academic level, and there is still a long way to go from academic research to commercial application. Since the advent of blockchain technology, people have never ceased their efforts to use the incentive layer of blockchain and digital currency to accelerate the construction of a new generation of distributed network. However, under the conventional TCP/IP network flow model, how to measure the Proof of Work (POW) of network nodes fairly and provide incentives based on the POW has always been a hard nut to crack in this field. So far, this problem has not been solved in all the blockchain network projects based on the original OSI7-layer protocol, with some approximate algorithms completed merely. The reason is that the IP protocol at the “narrow waist” in OSI7 layer protocol is, in essence, a point-to-point communication protocol. IP protocol is the communication between two hosts, which has nothing to do with other nodes in the network, so there is no way to form a verifiable password consensus.

At the application level, the current Internet is mainly based on Hyper Text Transfer Protocol (HTTP). As a stateless connection based on TCP, HTTP is a simple request-response protocol that usually runs on the top of TCP. HTTP specifies what message the client might send to the server and what response it might receive. The headers of the request and response messages are given in the form of ASCII code. The whole basic workflow is that the client sends an HTTP request specifying the resource the client wants to access and the requested action. After receiving the request, the server starts to process the request and makes corresponding actions to access the server resources according to the request, and finally returns the result to the client by sending an HTTP response.

Four disadvantages of HTTP:
1.Vulnerability to attack due to its centralized server.
2.High costs of data storage.
3.Leakage risk caused by centralized data.
4.Difficulty in transmission and maintenance caused by large-scale data storage. In the age of big data, the difficulties impeding the further development of HTTP include: How to store and distribute petabytes of big data, how to handle high-definition media streaming data, how to modify and iterate large volume of data and how to avoid the loss of important files.

To address these issues, projects such as IPFS/FileCoin and Lambda emerged after 2015.

IPFS, or InterPlanetary File System, is a new distributed hyper media transfer protocol based on content addressing. IPFS supports the creation of fully distributed applications, aiming to make the network faster, safer and more open. IPFS is a distributed file system designed to connect all computing devices to the same file system, thus becoming a unified storage system. At the technical level, it uses BitToreent protocol to exchange Git data objects to achieve this goal. IPFS is a protocol and a P2P network, similar to the current BT network.

Filecoin, an incentive layer running on IPFS, is a blockchain-based distributed storage network, turning cloud storage into an algorithm market. Filecoin protocol has two trading markets: Data retrieval and data storage. IPFS/Filecoin solves the problem of data storage and distribution over the Internet, providing a more cost-effective, secure and stable storage solution.

Lambda is an engineering implementation of IPFS/FileCoin, which has been put onto the main network.

According to the current progress of IPFS/FileCoin and Lambda, great progress has been made in the fields of storage network, consensus network, data integrity proof and re-storability proof, and data storage market, etc. However, there is no good solution to data retrieval.

Purposes of NDN Link in the project are:

1. To address the problem of data retrieval left by IPFS/FileCoin and Lambda projects by using NDN technology, and build a perfect distributed storage network.

At present, the solutions to data retrieval in FileCoin and Lambda projects are effective approximate methods, rather than accurate ones based on the password consensus system. This paper provides a solution to data retrieval by means of built-in cache in NDN.

2. To apply NDN technology to replace TCP/IP protocol and build a new underlying infrastructure of the Internet

In this project, we use Named Data Networking (NDN) – one kind of information network center – to replace the IP protocol at the narrow waist, thus providing solutions by smartly translating the network traffic problem into a storage problem, i.e. turning the distributed network problem into a distributed storage problem. Meanwhile, by studying the distributed storage project i.e., Proof-of-Space-Time (PoST) algorithm available for Filecoin and Lambda projects, we propose the “proof of time-space” algorithm, thus providing a fair incentive to each network node to maximize the efficiency of work. Based on the incentive, a next-generation Internet infrastructure system can be built in a quick and efficient way.

3. To use blockchain technology and digital currency as the driver of data routing strategy in NDN

We will not provide a follow-up discussion on the seventh layer, or application layer, of OSI in this paper which mainly focuses on the fourth layer of OSI model. In other words, IPFS/FileCoin and Lambda, by default in this paper, have given a better solution to the problem of application layer network.

NDN Link

I Basic Concepts and Background on Communication

Conventional Communication Network

Overall, the development of communication network has gone through two important stages. The first stage is telephone network which has been used since 1876. As for the communication of wire telephone network, a fixed dedicated line is required between the two endpoints of voice communication. The relay and transfer of telephone lines are mainly controlled by telephone exchanges. This communication mode relies on an exclu-sive physical telecommunication line for each call. However, such a line cannot be shared in the process of voice communication. The second stage, from the 1960s to 1970s, is marked by the emergence and application of the interconnection network with digital data packet switching as the communication mode which is also the basic operating mode of current Internet architecture.

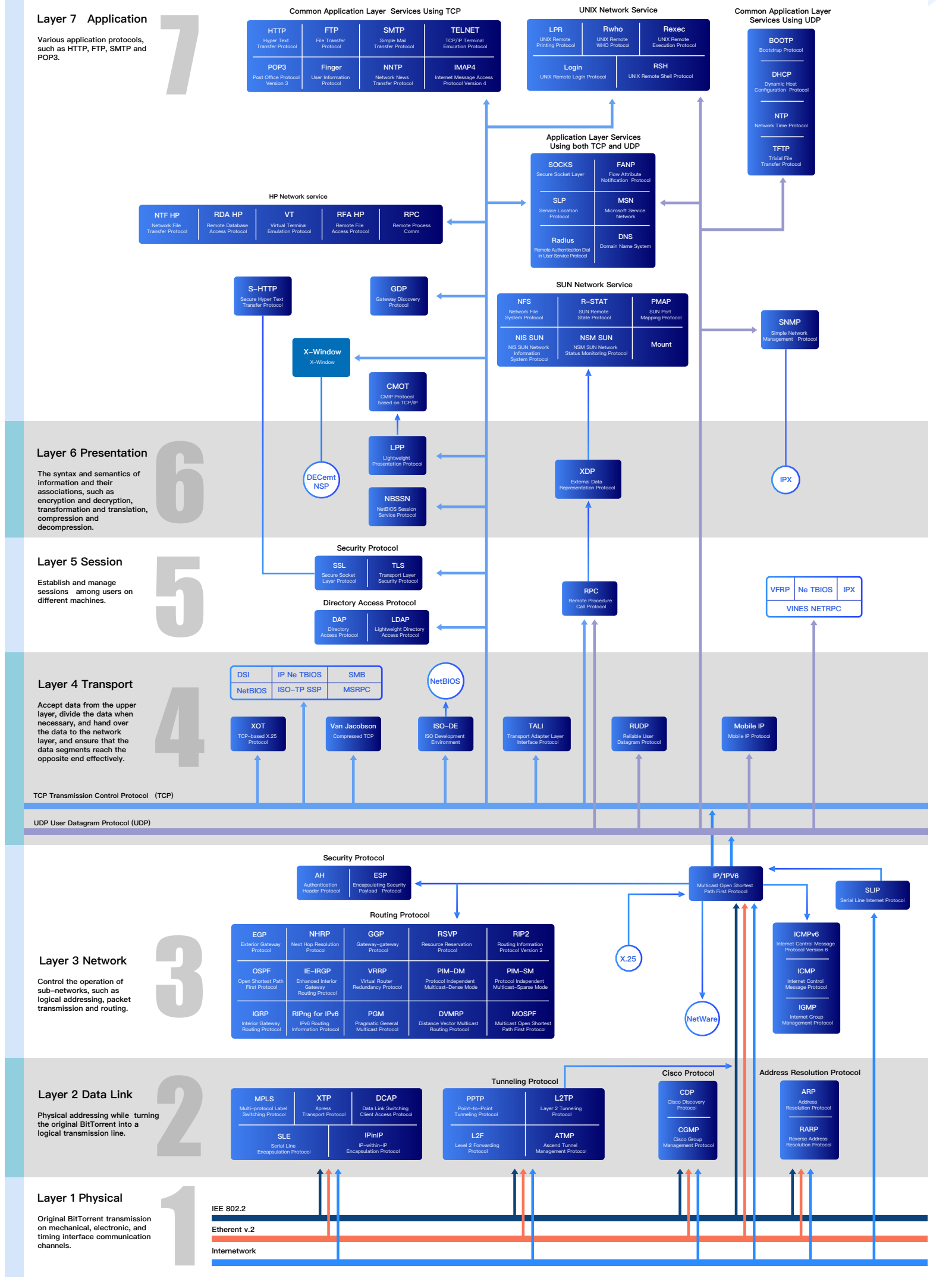
The predecessor of the Internet is ARPANET which is the first network that can actually run data packet switching, as well as the first operating network that uses TCP/IP proto-col as the communication protocol. By establishing an end-to-end connection based on network topology location, TCP/IP protocol enables data transmission through exchang-ing and relaying IP data frames on shared physical network links.

Symbolizing a revolutionary step forward from the previous telephone network, the IP Ethernet has greatly reduced the cost of network communication through link sharing. However, it merely provides a solution to the end-to-end digital data exchange between two entities, the way of which is still similar to that of telephone.

OSI Seven-Layer Model

The seven layers are defined as application, presentation, session, transport, network, data link and physical.
Among them, the IP protocol mentioned herein is Layer 3 – the network layer, and TCP protocol is Layer 4 – the transport layer. HTTP and IPFS protocols are Layer 7 – the application layer.
Both NDN and ICN are Layer 3 – the network layer.

TCP/IP



TCP/IP Protocol

TCP/IP, or the Transmission Control Protocol/Internet Protocol, refers to a protocol suite that can transmit information between different networks. TCP/IP not only refers to TCP and IP, but to a protocol suite composed of FTP, SMTP, TCP, UDP and IP, etc. It is com-monly referred to as TCP/IP, because TCP and IP are the most representative among TCP/IP Protocol Suite.

TCP/IP is the most basic protocol of the Internet. The Application mainly involves Telnet, FTP and SMTP, responsible for receiving data from transport layer or transferring data to transport layer according to different application requirements and ways. The transport layer mainly has protocols UDP and TCP, serving as a channel to combine the user's platform with the internal data of the computer information network, and can enable data transmission and data sharing. The network layer mainly consists of ICMP, IP and IGMP, dealing with transmission of data packets across the network. The network access layer, also known as the network interface layer or the data link layer, is mainly composed of ARP and RARP. It is mainly used for providing link management error detection and effective processing of information details of different communication media.

II. Existing Problems of TCP/IP

About Network Scalability

Routing table

As the number of users and the demand for applications continue to increase, the growth rate of network traffic has broken Moore's law, leading to operators' constantly passive capacity expansion and increasing costs. In addition, the routing table of the backbone router has expanded dramatically, making the number of routing table entries hit more than 33.84 million in the world. As a result, the performance of routing lookup has reduced greatly and system overhead of the router has increased, which, however, can only be alleviated in the existing Internet architecture by continuously improving the performance of hardware equipment, and there is as yet no fundamental solution.

About IPv4 address exhaustion and NAT

Today's Internet is mostly based on IPv4 address protocol, which "generates" about 4.5 billion IP addresses uniformly assigned by the Internet Assigned Numbers Authority (IANA). It is estimated that by 2020 there will be around 50 billion networking devices worldwide, more than 10-fold increase from the number of IPv4 addresses available.

About Security

At the incipient stage of the Internet, the network security was neglected, and all the security measures are an afterthought. However, the patching-type security fails to meet the requirements of today's Internet environment where malicious software, distributed denial-of-service (DDoS) attack, phishing software, application vulnerabilities and other security threats exist.

About Reliability

The Internet architecture delivers IP datagrams based on the target IP address, so malicious users can insert any IP address into the source address field of each IP datagram they send, which is called spoofing. The generated datagram is delivered to the destination, but it is difficult to judge its true source.

About Mobility

In the early days, the Internet mainly provided data exchange services for fixed terminals, represented by computers, with certain processing capacity. At present, with the emergence of portable mobile devices, the Internet of Things and the Internet of Vehicles, the terminal form of the Internet has changed a lot, featuring significant increase of mobility of terminals and frequent switching of data transmission paths. The design rules of dual-type semantics of identity and address in conventional TCP/IP network are not conducive to frequent service switching, so the demands for the session scenario with low latency and no packet loss will fail to be met, especially for such service objects as fast moving vehicles. That has seriously destroyed the continuity of upper application services, and even makes it difficult to maintain the conventional end-to-end transmission. In this context, how to efficiently make the Internet be free from mobility is a problem to be solved.

III. Basic Concept and Development History of ICN

Information-centric Networking (ICN)

ICN means that everything in the network can be taken as information. It is not a host interconnected network, but a data content interconnected network with core object being information, identifying each information unit by the name of data. As for the network, the information transmitted and stored is named, and the network itself can identify the information unit. Specific information contains data which cannot be analyzed by the network system alone and the assistance from the producer and demander of the upper application is required. All nodes and programs in the entire network run under the drive of various information requests and responses. The function of ICN network is to coordinate the transmission and caching of named data, and to use intelligent optimization to query the correct data to respond to the needs of users quickly. Users or applications only focus on the information data instead of other attributes of the information.

The data communication of ICN network focuses on the information production and consumption catering to the interests of users. The network is primarily concerned with publishing, searching, and delivering information, rather than terminal host accessibility or maintenance of sessions between terminals. On the whole, the model can be divided into two function parts: Information dissemination or distribution, and information retrieval. In terms of the bottom of the ICN network, the network is a group of interconnected information content, also known as information or data objects. They choose the route by name and are served and managed by applications or middleware's at a higher level. The naming scheme of objects in CCN is used to replace the current IP naming scheme. The current IP naming scheme mixes the host location and Uniform Resource Locator (URL). In particular, the name of a content object is globally unique, regardless of its location. Content object is an abstract concept that can be of any type, including static or user-generated contents (e.g. photo, video and file) and real-time media streams (e.g. VoIP, video on demand, network TV, online video and music).

Basic Concept and Component Module of ICN Network

Information/Data Object

First, the meaning of data and information must be defined. In general definition, data is the abstraction at the lowest level and information is the abstraction at a higher level. The data that can be defined as information must be able to be interpreted and of some meaning. The concept of ICN network focuses on information itself, rather than where the information is stored. To highlight this distinction, a concept of Information Object (IO) needs to be introduced. An Information Object represents the information itself, independent of its storage location and physical representation

Naming Mechanism

The most critical part of ICN network design is a set of naming strategies, independent of location, which identify data identity. The basic considerations include the following:
(1) Uniqueness: The Information Object must be named in a unique way within the global network. This uniqueness is necessary for naming.
(2) Endurance and location independence: The name is constant, independent of the location of the host. Service providers continue to provide services, and content can be replicated or hosted from one location to another without interrupt servicing.
(3) Availability and scalability: From an information network, we can expect a large number of data objects which can be correlated and dependent on each other in the network. In addition, an object can be either static or mutable; it can be scattered in small pieces or change from one version to another, such as weather RSS. The naming scheme, with a deletion mechanism, can be used for dynamic objects and allows deletion of objects.

(4) Security: ICN network builds security directly into content, provided that there is no assumption of user trust or security provided through encrypted channels. The content–based security means that there is an encrypted binding between the content and its name, so as to ensure that the object of the information can be self–validating. It is usually possible to create a signature of a data object by using the private key of a content provider. In this regard, the content of ICN network is equivalent to the transaction on Bitcoin network.

The 5 basic data security–oriented goals within the ICN network are:

1 Confidentiality: Only authorized individuals can safely read information

2 Data integrity: The information data tampered unintentionally or intentionally and corresponding metadata can be identified, which is also known as a self–attestation capability.

3 Credibility: The identity of the owner and creator of the information data can be authenticated and identified through public/private key pairs, etc.

4 Availability: The information objects and related metadata published from the network must be accessible and obtained by authorized individuals

5 Access control: Access to information objects and corresponding metadata will be managed and restricted.

Because ICN network itself has the above capabilities, it is able to be inherently compatible with the incentives of digital currency.

Metadata

The concept of metadata is very important in ICN. It provides information to describe content objects or the relationship between the content object and other objects. For example, the metadata associated with a photo can indicate the graphics resolution, the author, the date, or any other data inserted by software.

Metadata attributes serve various purposes. First, the semantics of these attributes can benefit an application in terms of managing content objects or how they are used. For example, the search engine can use object–related metadata to do the mapping in a distributed way during a user's search based on a keyword or description.

In addition to semantics, metadata can provide the input for password operations to perform more complex security checks. For example, the metadata of a content object contains the public key and digital signature of a content provider, enabling the receiver to determine the source of the digital content. The network can also provide QoS guarantee, network access control and network monitoring capabilities by relying on metadata.

Content Addressing and Forwarding

The content retrieval in ICN network falls into two phases: Content discovery and content delivery. Content discovery includes content object addressing and query request forwarding; content delivery includes the rules for routing and transmission of content objects in a network.

Addressing, also known as name resolution, refers to the accessibility of information objects in the network by mapping the name to the host location. The process of name resolution, analogous to DNS resolution in IP, is similar to the workflow of the network layer of Internet and includes the determination of the content location, content forwarding and data delivery.

The main purpose of ICN network is to transfer data between data producers and consumers. In the model of publishing/subscribing communication, the direct linkage between content producers and consumers is canceled. Data driving is completely handled by the network itself. Most ICN architectures forward contents by publishing/subscribing network communication, involving two functional steps, namely, registration and search.

Routing and Transmission

Since an object ID is independent of location, common routing and forwarding mechanisms based on network topology cannot be used for the object ID. ICN network routing strategies are generally divided into two types, which, to a large extent, depends on the different features of the object namespace. The key point is whether the data name can be aggregated.

Transport Layer of Information Object

In the current Internet architecture, the functions of the transport layer, such as error detection, retransmission of lost data, bandwidth management, traffic control and congestion avoidance, are implemented at the terminal layer as an end-to-end communication process, which is inconsistent with the CCN. Compared with the conventional IP network communication, the role of terminals in the CCN is very different, because the session is information-centric, without relying too much on the terminal involved. Therefore, if naming and addressing enable us to implement the functions of the network layer and lower layer required by the content-centric network, we also need to consider the transport layer to completely eliminate the dependence on the terminal.

Cache Mechanism

Some ICN architectures use a lot of data caches to ensure more efficient network utilization and improve data availability. There are mainly two caching methods: Caching at the network edge and caching in network. For P2P system and mirror server, they have caches at the node of the network edge. For caching in network, data is cached at intermediate nodes of the transport network path. For example, cache in the forwarding router or in combination with the name resolution service; the advantage of the latter is that it is independent of the upper application and can be utilized more efficiently, avoiding the disadvantage of failing to share caches among different application systems.

Storage and Search

According to the implementation plan of ICN network, two extra modules will initially become main components of the ICN network. First, persistent consistent storage will be more closely integrated with the network architecture, including the combination of cache and naming resolution. Second, the combination of information search and network architecture may be closer than today's Internet. There is no need to establish and maintain a large-scale independent search engine system, especially when the metadata describing data objects is stored in the ICN network.

Typical ICN Project

1 DONA
2 PSIRP/PURSUIT
3 NetInf/SAIL

Similarity

<1> The same tracing: The communication mechanism of publishing/subscribing is adopted, so that network requests and responses are decoupled in time and space.
<2> Generic caching mechanism
<3> Content-oriented security mode

Named Data Networking (NDN)

Previously, a number of researches and papers concerning ICN focused on the core mechanism of ICN, with an emphasis on design of a more advanced mechanism. In fact, these problems are no longer the focus of researches. After many years of attempts and demonstration of various early ICN prototype architectures, the core concept and

module of ICN have seen diversification and fierce competition, forming a mainstream design framework represented by NDN mechanism through initial integration. This is because some modern improvement technologies based on IP network, such as HTTP, P2P and CDN, are only a few application layer protocols developed from TCP/IP network architecture, but many of the design concepts, when integrated, have basically enabled an ICN prototype. Most of the remaining work is to continually develop, improve and perfect NDN. NDN project is an integrator, a common accumulation of network research workers of different generations all over the world.

Named Data Networking (NDN) is a network project of the basic information research center funded by the U.S. National Science Foundation (NSF) with nearly USD 10 million in 2010 for special research of the Future Internet Design (FIND). The project is mainly led by Professor Lixia Zhang of University of California, Los Angeles (ULCA) (who was a professor of University of Arizona when the paper was written) and Mr. Van Jacobson (working for Google and as a part-time professor of UCLA), both of whom are currently academicians of the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE). With participants including 12 American colleagues and universities and research institutions, the NDN project was previously known as Content Centric Networking (CCN) which was short for Palo Alto Research Center (PARC) before the NDN project was approved by NSF. PARC implemented an open source code system CCNx based on the idea of ICN, which also served as an early prototype research basis for the NDN project.

Currently, the NDN project in the lab is generally in the elementary stage of prototype research, which is mainly caused by the lack of motivation of relevant manufacturers, practitioners and clients to promote the NDN system. In this paper, we present a method, named as NDN link, for implementation of NDN framework driven by blockchain and digital currency. We have faith that NDN will be promoted and implemented in a better manner with digital currency as a trust intermediary.

Note: In addition to the implementation method of NDN Link, NDN framework also has other methods unrelated to blockchain and digital currency.

1 NDN Link, a New Narrow Waist, Driven by Blockchain

The Internet's hourglass architecture today centers on a universal network layer (i.e., IP) which implements the minimal functionality necessary for global interconnectivity. This narrow waist enabled the Internet's explosive growth by allowing both lower and upper layer technologies to innovate independently. However, IP was designed to create a communication network, where packets named only communication endpoints. Sustained

growth in e-commerce, digital media, social networking, and smartphone applications has led to dominant use of the Internet as a distribution network. Distribution networks are more general than communication networks, and solving distribution problems via a point-to-point communication protocol is complex and error-prone.

The Named Data Networking Link (NDN Link) project proposed an evolution of the IP architecture that generalizes the role of this narrow waist, such that packets can name objects other than communication endpoints (Figure. 1). More specifically, NDN Link changes the semantics of network service from delivering the packet to a given destination address to fetching data identified by a given name. The name in an NDN Link packet can name anything — an endpoint, a data chunk in a movie or a book, a command to turn on some lights, etc. This conceptually simple change allows NDN Link to use almost all of the Internet's well-tested engineering properties to solve a much broader range of problems including not only end-to-end communications but also content distribution and control problems. Based on three decades of experience with the strengths and limitations of the current Internet architecture, the design also builds in security primitives (via signatures on all named data) and self-regulation of network traffic (via flow balance between interest packet and data packet). The architecture includes functionality designed to be conducive to user choice and competition as the network evolves, such as multi-path forwarding and in-network storage.

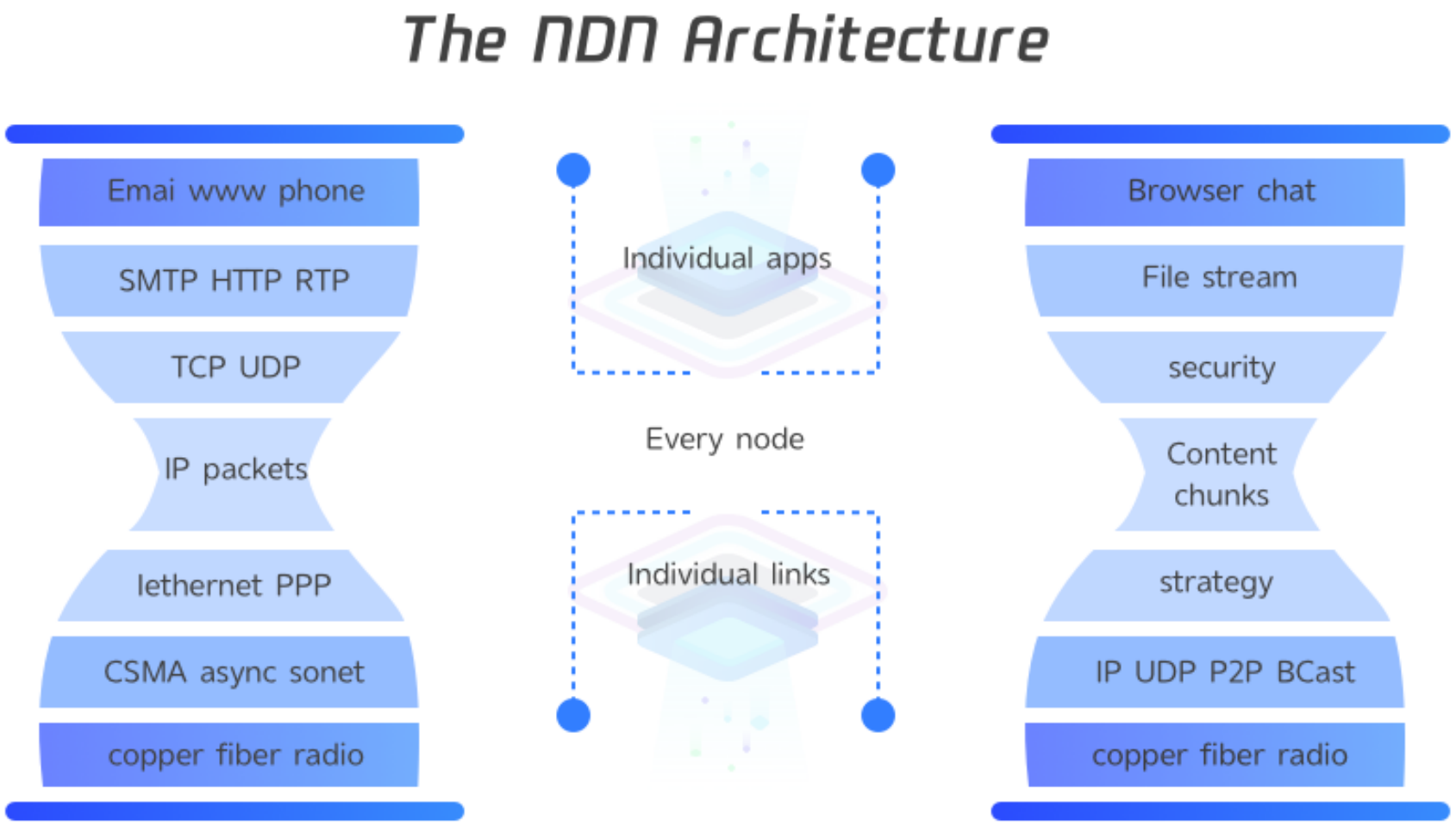


Figure 1:The main building of the NDN architecture are named content chunks,in constrast to the IP architecture's fundamental unit of communication, which is an end-to-end channel between two end endpoints identified by IP addressses.

2 NDN Link Architecture

2.1 Design Idea of NDN Link Architecture

The following principles are provided to guide the design of the NDN Link architecture:
(1) Hourglass architecture
The hourglass architecture is what makes the Internet design elegant and powerful. It focuses on a universal network layer (IP) implementing the minimal functionality necessary for global interconnectivity, serving as a key enabler of the Internet's explosive growth. NDN Link keeps the same narrow waist
(2) Security features of architecture integration
The NDN Link signature data lays a foundation for the trust of the Internet in the future, where applications can establish fine-grained and customized authentication, authorization and trust models
(3) Self-regulation of network traffic
The forwarding featuring balanced traffic is a requirement for network stabilization and NDN Link will provide a powerful hop-by-hop network traffic balance.
(4) Built-in network traffic monitoring facilitates accurate billing and promotes competition

2.2 Architecture of NDN Link

Communication in NDN Link is driven by receivers, i.e., data consumers, through the exchange of two types of packets: Interest and Data. Both types of packets carry a name that identifies a piece of data that can be transmitted in one Data packet. A consumer puts the name of a desired piece of data into an interest packet and sends it to the network. Routers use this name to forward the interest packet toward the data producer(s). Once the interest packet reaches a node that has the requested data, the node will return a data packet that contains both the name and the content, together with a signature by the producer's key which binds the two. This data packet follows in reverse the path taken by the interest packet to get back to the requesting consumer.

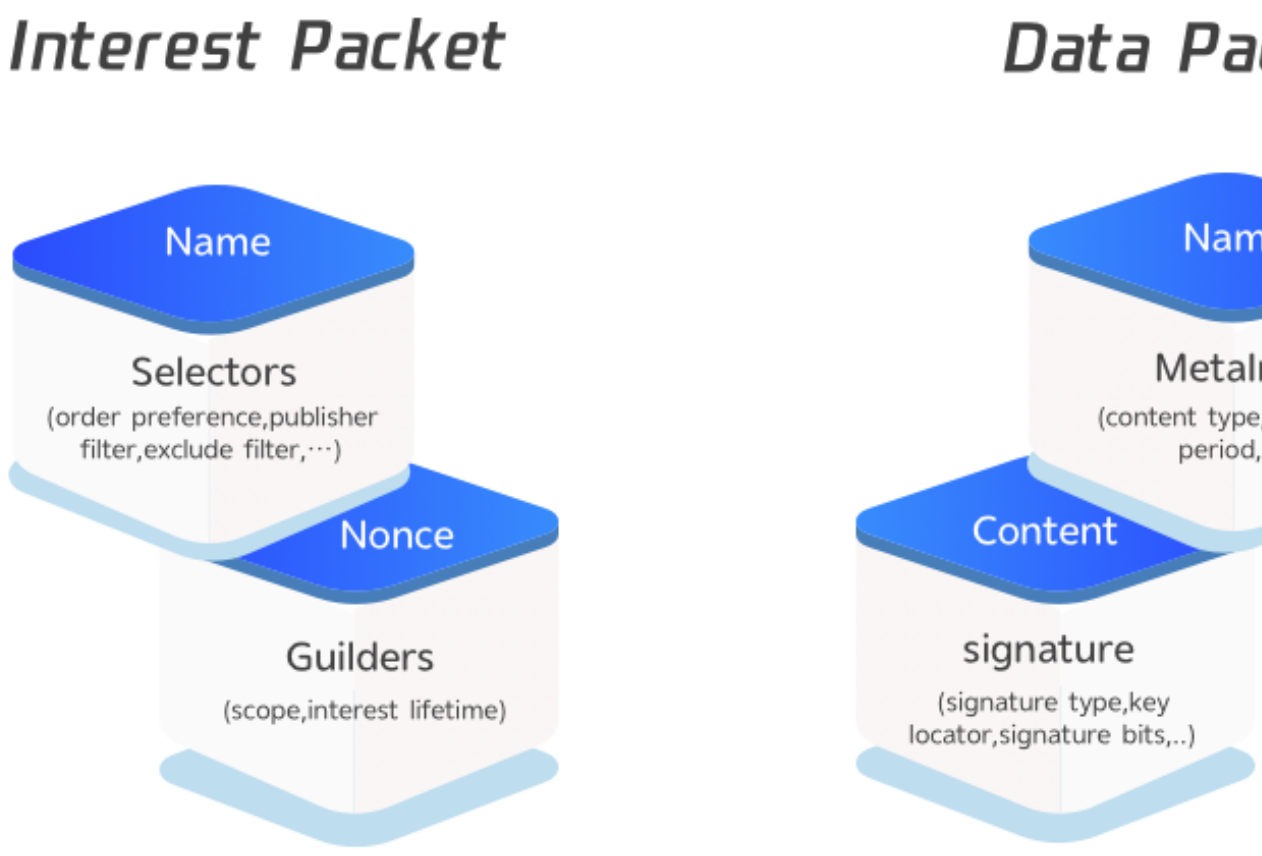


Figure 2: Paket in the NDN Architecture

To forward the interest packets and data packets, each NDN Link router maintains three data structures: A Pending Interest Table (PIT), a Forwarding Information Base (FIB), and a Content Store (CS) (Figure 3), as well as a Forwarding Strategy module that determines whether, when and where to forward each interest message. The PIT stores all the interest packets that a router has forwarded but not satisfied yet. Each PIT entry records the data name carried in the interest packet, together with its incoming and outgoing interface(s). When an interest packet arrives, an NDN router first checks the Content Store for matching data; if it exists, the router returns the data packet on the interface from which the interest packet came. Otherwise the router looks up the name in its PIT, and if a matching entry exists, it simply records the incoming interface of this interest packet in the PIT entry. In the absence of a matching PIT entry, the router will forward the interest packet toward the data producer(s) based on information in the FIB as well as the router's adaptive Forwarding Strategy. When a router receives interest packets for the same name from multiple downstream nodes, it forwards only the first one upstream toward the data producer(s). The FIB itself is populated by a name-prefix based routing protocol, and can have multiple output interfaces for each prefix.

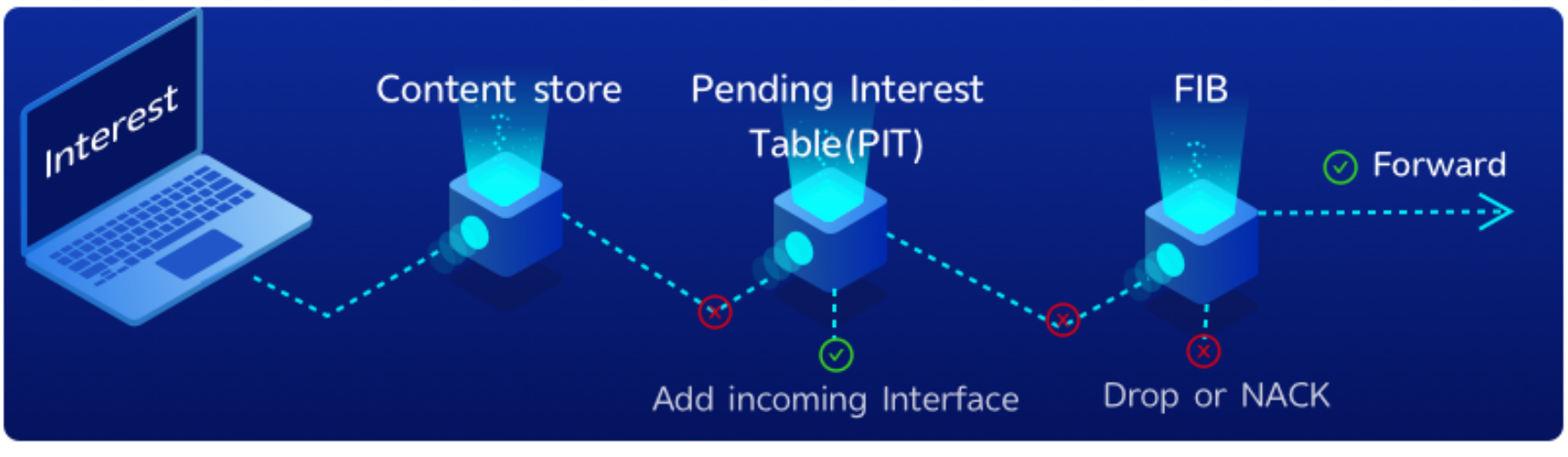


Figure 3: Forwarding Process at an NDN NODE

FIB is used to route request packets to a potential matching data source. The FIB of NDN Link stores <prefix, interface list> tuple, with the longest match for query. NDN Link supports one routing item pointing to multiple potential matching interfaces at the same time. That is to say, data can be obtained from multiple sources to realize parallel query. The information in FIB is updated by manual configuration or the name–based routing protocol.

CS is similar to the content caching in IP. In the IP cache mechanism, data transmission takes the form of flow due to use of point–to–point dialog, and the intermediate router forwards data only. When a packet is sent out, it will be invalidated, and the content in the cache area cannot be reused. On the router side, NDN can identify and process con–tents by name, and reuse data by content name. Therefore, it is necessary to keep the received data as long as possible to maximize sharing and reduce bandwidth consumption.

PIT saves the uplink information sent by the interest packet to ensure that when the data packet is received, it can be returned to the requester correctly. After data is sent to the request interface, the PIT item matching the data should be deleted, and the table items that have not received the return data for a long time should be cleared after timeout. PIT allows aggregation of request packets: The same request from different interfaces are merged in PIT, and only the first incoming request packet is routed to a potential responder.

Face is an abstraction of the transmission interface, which can be a connection with other network nodes, or a connection established with local applications through inter-process communication. Face can realize multiple transmission modes through configuration: Data packets can be broadcast or multicast over a network card, or received or sent by using the point-to-point addresses or constructed tunnels in the underlying transport layer. All data packets of NDN Link must be sent or received through Face.

NDN Link is of In-Network Caching mechanism. All data packets are cached in the routing node (CS) for as long as possible. The content packet in the CS meeting the requirements of the request packet can be used to respond to the request.

The Forwarding Strategy may decide to drop an interest packet in certain situations, e.g., if all upstream links are congested or the interest packet is suspected to be part of a DoS attack. For each interest packet, the Forwarding Strategy retrieves the longest prefix matched entry from the FIB, and decides when and where to forward the interest packet. The Content Store is a temporary cache of data packets the router has received. Because an NDN Data packet is meaningful independent of where it comes from or where it is forwarded, it can be cached to satisfy future interest packets.

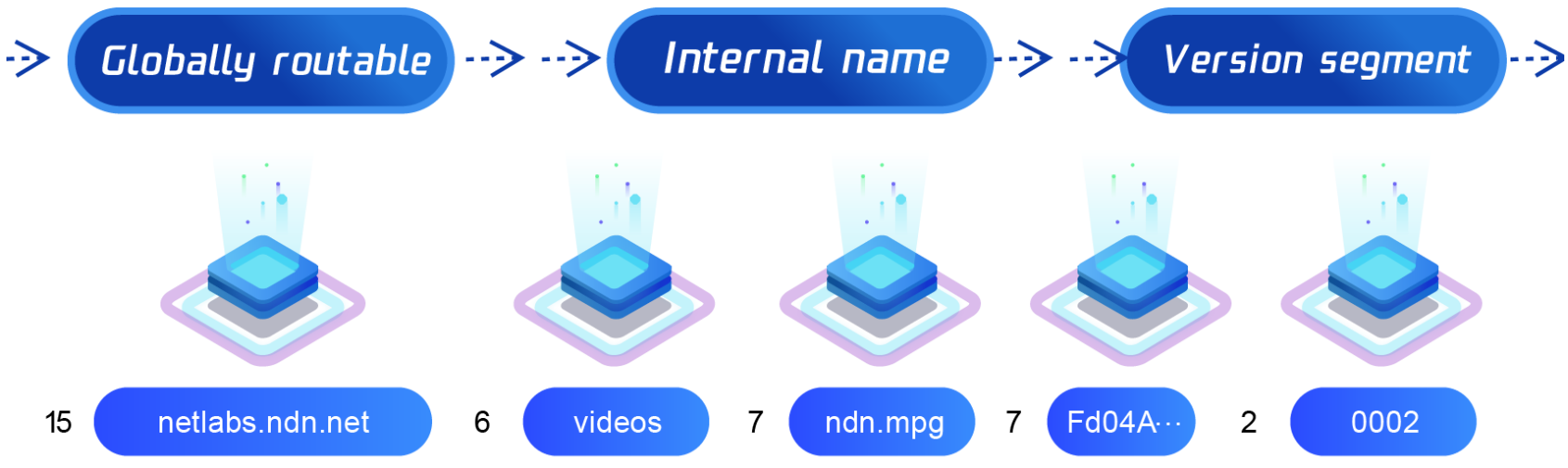
When a data packet arrives, an NDN Link router finds the matching PIT entry and forwards the data to all downstream interfaces listed in that PIT entry. It then removes that PIT entry, and caches the data in the Content Store. Data packets always take the reverse path of interest packet, and, in the absence of packet losses, one interest packet results in one data packet on each link, providing flow balance. To fetch large content objects that comprise multiple packets, interest packets provide a similar role in controlling traffic flow as TCP ACKs in today's Internet: A fine-grained feedback loop controlled by the consumer of the data. Neither interest nor data packets carry any host or interface addresses; routers forward interest packets toward data producers based on the names carried in the packets, and forward data packets to consumers based on the PIT state information set up by the interest packets at each hop. This interest/data packet switching symmetry induces a hop-by-hop control loop, and eliminates the need

2.3 Naming Mechanism of NDN Link

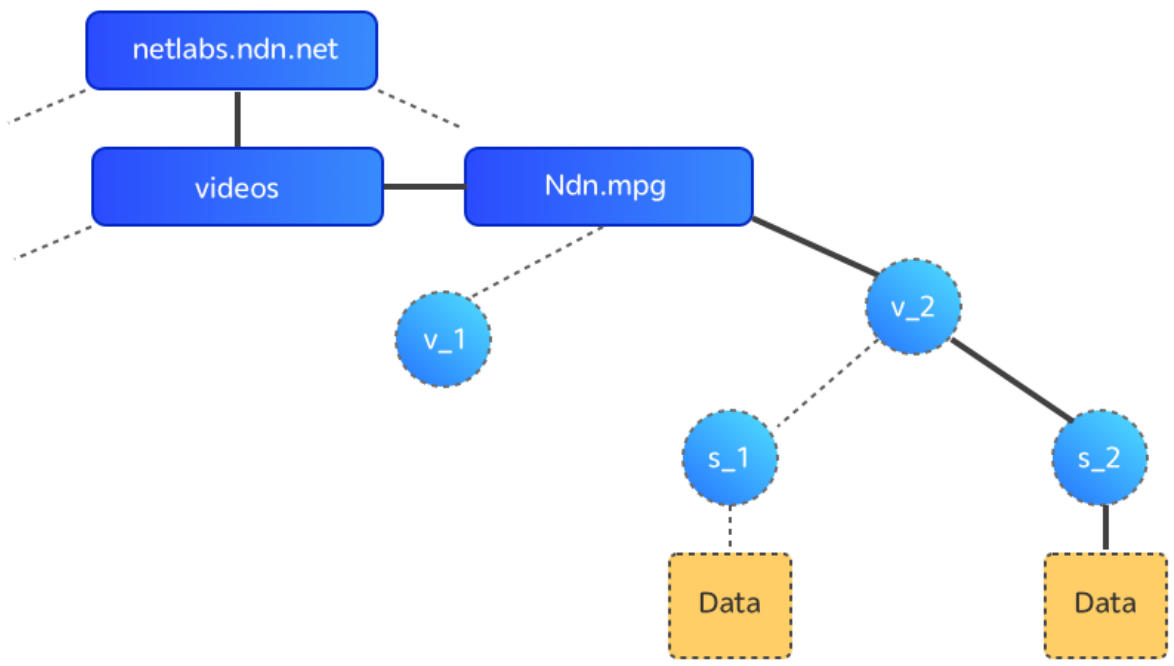
The name of the data used in NDN Link transmission is completely opaque to the underlying network. A router in the network does not know the exact meaning of a name, which allows each application to choose the naming scheme that fits its needs. The data names contained in the Interest packet or Data packet in NDN Link are essentially binary objects composed of a series of components and are of a hierarchical structure similar to that in conventional IP, which facilitates the match according to the longest common prefix of the names during transmission. Just like IP addressing, data naming itself contains no information about “where it is from” or “where it will go”, thus making the transmitted data packet irrelevant to the communication endpoint. Data packets can be cached by a router during the transmission process to meet the same data resource requests in the future, thus achieving data reuse.

As shown in the figure below, NDN Link data is named in URI format

```
/netlabs.ndn.net/videos/ndn.mpg/_v<timestamp>_s2
```



Data packets, in essence, match Interest packets according to the longest common prefix of their respective data names, which means that the name of matched Data packets is the subtree of the Interest packet name, and the search process of data is the process of traversing the name spanning tree of NDN Link, as shown in the figure below:



NDN Link uses hierarchical structured naming to name its contents, which allows researchers to quickly locate the information they need by using a method similar to the mechanism of IP address prefixes. In NDN Link, a name is usually composed of several parts. Generally, the content name can be divided into two parts: Content name and fragment name, as shown in the previous figure.

To retrieve dynamically generated data, consumers must be able to determine the name of the data without having previously seen the name or the data. Therefore, a deterministic algorithm can be used to make the data producer (s) and consumer (s) get the same name, or the consumer (s) can retrieve data according to some names. Data that may be retrieved globally must have globally unique names, but names used for local communications may require only local routing (or local broadcast) to find matching data. Individual data names can be meaningful in various scopes and contexts, ranging from “the light switch in this room” to “all country names in the world”.

Namespace management is not part of the NDN Link architecture, just as address space management is not part of the IP architecture. However, naming is the most important part of NDN application designs. Naming data enables support for functionality such as content distribution, multi-cast, mobility, and delay-tolerant networking.

2.4 Data-centric Security

At present, the security of the Internet based on TCP/IP architecture depends on the security of the data channel, while NDN Link guarantees the security of contents. Such a design separates the credibility of the data from that of the host. In NDN link, security is based on data, without depending on the source and acquisition method of the data. Each piece of data is signed by the publisher. Data signature is mandatory. After the data is signed, plus the information released by the data, the credibility of the data can be determined, so that consumers’ trust in the data will no longer depend on how and where it is obtained.

Fine-grained trust is also supported, allowing consumers to determine in certain circumstances whether the owner of a public key is an acceptable publisher for a particular part of the data.

The security based on public key encryption has such disadvantages as low efficiency and difficulty in deployment and use. In addition to efficient digital signature, NDN Link also needs a flexible and feasible mechanism for trust management. Since a key can be forwarded as data, distribution of the key is simplified, and the binding between name and data can support multiple forms of models. For example, if some data content is a public key, then the binding between the name and the content is the authentication of the public key. The end-to-end security of NDN is able to promote the trust between producers and consumers, providing more flexibility for producers, consumers and applications in selecting and customizing their trust models.

NDN Link's data-centric security can provide support in content access control and network architecture security. NDN Link enables applications to control access to contents and maintain data security within each program by encrypting the contents and distributing encrypted keys. In terms of routing, NDN Link requires the control information between routers to provide signatures, just like other data, which can resist spoofing and tampering between routers. NDN Link's inherent multi-path routing strategy, together with the adaptive forwarding plane, can resist prefix hijacking attack. When an abnormality caused by prefix hijacking is detected, information can be obtained from other paths.

In contrast to TCP/IP, which leaves responsibility for security to the endpoints, NDN secures the data itself by requiring data producers to cryptographically sign every Data packet. The publisher's signature ensures integrity and enables determination of data provenance, allowing a consumer's trust in data to be decoupled from how or where it is obtained. It also supports fine-grained trust, allowing consumers to reason about whether a public key owner is an acceptable publisher for a specific piece of data in a specific context. The second primary research thrust is designing and developing usable mechanisms to manage user trust. We have experimented with both a hierarchical trust model where a key namespace authorizes use of keys (a data packet carrying a public key is effectively a certificate, since it is signed by a third party) to sign specific data, and web-of-trust to enable secure communication without requiring pre-agreed trust anchors.

NDN's data-centric security has natural applications to content access control and infrastructure security. Applications can control access to data via encryption and distribute (data encryption) keys as encrypted NDN Link data, limiting the data security perimeter to the context of a single application. Requiring signatures on network routing and control messages (like any other NDN data) provides a solid foundation for securing routing protocols against, e.g., spoofing and tampering. NDN's use of multi-path forwarding, together with the adaptive forwarding strategy module, mitigates prefix hijacking because routers can detect anomalies caused by hijacks and retrieve data through alternate paths. Since NDN packets reference content rather than devices, it is trickier to maliciously target a particular device, although mitigation mechanisms will be needed against other NDN Link-specific attacks, e.g., interest packet flooding DoS.

2.5 Routing and Forwarding Strategies of NDN Link

NDN routes and forwards packets based on names, which eliminates three problems caused by addresses in the IP architecture: Address space exhaustion, NAT traversal, and address management. There is no address exhaustion problem since the namespace is unbounded. There is no NAT traversal problem since NDN does away with addresses, public or private. Finally, address assignment and management are no longer required in local networks.

NDN can use conventional routing algorithms such as link state and distance vector. In-stead of announcing IP prefixes, an NDN router announces name prefixes that cover the data the router is willing to serve. The routing protocol propagates these announcements across the network, informing each router's construction of its own FIB. Conventional routing protocols, such as OSPF and BGP, can be adapted to route on name prefixes by treating names as a sequence of opaque components and doing component-wise longest prefix match of a name in an interest packet against the FIB table.

The PIT at each router supports forwarding across NDN's data plane, recording each pending interest packet and the incoming interface(s), and removing the interest packet after the matching data is received or a timeout occurs. This per-hop, per-packet state differs from IP's stateless data plane. Based on information in the FIB and performance measurements, an adaptive forwarding strategy module in each router makes informed decisions about: Which Interests to forward to which interfaces, how many unsatisfied interest packets to allow in the PIT, the relative priority of different interest packets, load-balancing interest packet forwarding among multiple interfaces, and choosing alter-native paths to avoid detected failures. If a router decides that the interest packet cannot be satisfied, e.g., the upstream link is down, there is no forwarding entry in the FIB, or extreme congestion occurs, the router can send a NACK to its downstream neighbor(s) that transmitted the Interest. Such a NACK may trigger the receiving router to forward the interest packet to other interfaces to explore alternate paths. The PIT state enables routers to identify and discard looping packets, allowing them to freely use multiple paths toward the same data producer.

The PIT state serves other valuable purposes. First, since it records the set of interfaces over which the interest packets for the same data name have arrived, it naturally supports multi-cast data delivery. Second, since each interest packet retrieves at most one data packet, a router can control the traffic load by controlling the number of pending interest packets to achieve flow balance. Third, the number of PIT entries is an indicator of router load; constraining its size limits the effect of a DDoS attack. Finally, PIT entry timeouts offer relatively cheap attack detection, and the arrival interface information in each PIT entry could support a push-back scheme.

2.6 In-Network Storage of NDN Link

After receiving an Interest packet, NDN Link router first checks CS to see if there is a data. If there is one and its name is included in the name of the Interest, the data will be returned as a reply packet. The basic form of CS is equivalent to a router's cache memory, but an IP router cannot reuse packets after forwarding them to the destination, while an NDN router can reuse the data, because their names are constant. For static files, NDN achieves almost optimal data transfer. Dynamic content can be transmitted well in the case of data packet retransmission after multi-cast or a packet loss.

NDN Link uses the existing packet cache in an IP router as its cache. A cache hit means a decrease in bandwidth usage and use of a reasonable caching strategy. We assume that NDN Link performs better in static data than IP caching does in terms of reducing bandwidth requirements and the load of the origin server. It also outperforms IP caching in dynamic data.

NDN Link treats storage and network channels equally according to data acquisition. For static files, NDN achieves almost optimal data delivery. Even dynamic content can benefit from caching in the case of multi-cast (e.g., real-time teleconferencing) or retransmis-sion after a packet loss.
In addition to the Content Store, the architecture now supports a more persistent and larger-volume in-network storage, called a Repository (Repo for short). This type of storage can support services similar to that of today's Content Delivery Networks (CDNs), without having to engineer them as an application layer overlay using creative protocol tricks (e.g., DNS manipulation) to make them work.

2.7 Function of Transport Layer

The NDN architecture does not have a separate transport layer. It moves the functions of today's transfer protocols into applications, supporting libraries, and the strategy module of the forwarding plane. The reuse and demultiplexing of an application process is the name directly used in NDN Link. The integrity and reliability of data are directly handled by the application. The reliability check, data signature and trust, and determination are all made in the application.
The NDN Link is designed to deliver unreliable data packets, including highly dynamic connections and pervasive computing. To provide reliable and flexible transmission, the returned Interest packet is not received within a reasonable period of time. If data is still needed, it must be retransmitted by its final consumer. Such a function is very common for many or all NDN applications. In NDN Link, this function will be provided by a common library. The forwarding strategy of consumers is completed by a lower layer which is re-sponsible for re-transmission through a specific interface, and choosing an available communication interface, the number of available communication interfaces to send In-terest packets, the maximum number of unsatisfied Interest packets, and the priority level of different Interest packets, etc.

The NDN Link router can control load traffic by managing the size of PIT. If a router is overloaded due to the incoming traffic from some particular neighbor, it can simply slow down or stop sending Interest packets to the neighbor. That also means NDN has eliminated the dependence on the terminal host to perform congestion control.
Once congestion occurs, data retransmission will be completed with the help of cache. For example, if there are two congestion links in the path between the producer and consumer and one data packet passes the first one, but is discarded at the second one, then the cache will only allow the second one to retransmit the packet when the consumer resends the Interest packet after timeout at the first one. In the current Internet, data retransmission will occur from the producer's location along the whole path, so the packet must try to pass the first congestion link again. In NDN Link, the returned data moves steadily to the destination step by step, so the cached copy is used to satisfy both the original and the retransmitted Interest packets. Therefore, the disadvantage of data retransmission, which is a key focus of bandwidth consumption in today's Internet, can be avoided in NDN Link which also reduces throughput effectively.

3 Development of NDN Architecture

An NDN protocol specification requires standard formats for the two basic packet types (Interest packet and Data packet) and description of the functions supported by the network layer, i.e., the new narrow waist. Building an operational NDN also requires software libraries to support naming, high performance forwarding and routing, forwarding strategy, and trust management. Similar to IP's supporting components (address allocation, routing protocols, DNS), these libraries are not part of the core architecture but intrinsically support it, and all involve daunting research challenges. This section describes the project's application–driven, experimental approach to designing and developing the architecture, including examples that illustrate its capabilities, and open research challenges.

3.1 Application Research

The project's approach is to design and build a variety of applications on NDN to drive the development and deployment of the architecture and its supporting modules, to test prototype implementations, and to encourage community use, experimentation, and feed–back into the design. Application–driven development also allows verification and valida–tion of performance and functional advantages of NDN, such as how routing on names promotes efficient authoring of sophisticated distributed applications, by reducing com–plexity, opportunities for error, and time and expense of design and deployment. A few years of designing and developing prototype applications on NDN has revealed 5 key areas of application research that map to important features of the architecture: (1) Namespaces; (2) trust models; (3) in–network storage; (4) data synchronization; (5) ren–dezvous, discovery, and bootstrapping. These challenges arise within and across applica–tions. Namespace design must also recognize the interplay between application–specific requirements for data distribution and organization of trust–related information, together with those imposed for efficient routing/forwarding. Similar challenges exist in name dis–covery, bootstrapping, and mobility support. This commitment to application development paid off early in the project: It uncovered the unanticipated importance of both a per–node repository for persistent storage, and synchronization as a general building block for applica–tions. A few examples of early applications illustrate NDN's benefits and challenges.

Video streaming: One of the first NDN Link applications was a functional video streaming application that demonstrated the practical benefits of NDN Link–based media delivery, which inherently supports caching and multi–cast. NDN Video streams live and pre–re–cord HD video via NDN, and have been tested and demonstrated over both UDP and Eth–ernet transport. The NDN Video application does not require direct communication be–tween publisher and consumer, enabling publisher independent scalability through NDN Link's use of in–network storage. Applications that perform on–the–fly assembly of con–tent or selection of video sections, i.e., frame–level random access requirements, are supported directly through namespace design.

Real–time conferencing: The Chrono Chat multi–user text chat application has provided a platform to explore data synchronization techniques that can support a peer–to–peer chat service.

Vehicular networking: Vehicular networking is another domain where the NDN architec–ture offers advantages, enabling location–based content retrieval and new trust models to support ad–hoc, opportunistic communication. Experimentation with vehicular applica–tions has also led to updates to the NDN protocol stack itself, including support for other media (e.g., 3G/LTE, DSRC/WAVE, WiFi, WiMAX) and network–layer support for data muling, where vehicular NDN nodes cache data packets heard over a broadcast channel that do not have matching pending Interest packet in their PIT, in order to later provide them to other vehicles or pass them to infrastructure.

New architecture component: Sync. As a direct result of trying to build robust, efficient and truly distributed (i.e., serverless) peer–to–peer NDN applications, the architecture now supports a new building block called Sync. Using NDN's basic interest–data packet switching communication model, Sync uses naming conventions to enable multiple parties to synchronize their datasets. By exchanging individually computed data digests, each party learns about new or missing data quickly and reliably, and then can retrieve data efficiently via NDN's built–in multi–cast delivery.

3.2 Trust Management

To verify a data packet's signature, an application can fetch the appropriate key, identi–fied in the packet's key locater field, just like any other content. But trust management, i.e., how to determine the authenticity of a given key for a particular packet in a given application, is a primary research challenge. Consistent with an experimental approach, NDN trust management research is driven by application development and use: Solving specific problems first and then identifying common patterns.

For example, the security needs of NLSR require development of a simple hierarchical trust model in which keys are published with names that reflect their trust relationship. A root key is owned by the network domain's administrator, and below the root are site keys, each owned by a single site's administrator, signed by the root key and published in the next level of the hierarchy. Each site key then signs the site's operator keys, which in turn sign router keys, which in turn sign the key of the NLSR process on that router. Fi–nally, the NLSR key signs the routing data originated by NLSR. In this trust model, the namespace matches the hierarchy of trust delegation, i.e., (conceptually) /root/site/oper–ator/router/process. Publishing keys with a particular name in the hierarchy authorizes them to sign specific Data packets and limits their scope. Other applications where real world trust tends to follow a hierarchical pattern, such as in our Building Management

Systems (BMS), may use two separate hierarchies for building operators and for application data, to facilitate fine control over who has access to which data. More flexible and expressive trust relations, such as with our chat application, have motivated experimentation with a web-of-trust model. A current chatroom participant can introduce a newcomer to others by signing the newcomer's key. Future applications will implement a cross-certifying model (SDSI), which provides more redundancy of verification, allowing data and key names to be independent, which more easily accommodates a variety of real-world trust relationships.

4 NDN and P2P System

4.1 NDN and P2P

When it comes to content distribution and sharing, P2P cannot be ignored. In the current TCP/IP network, P2P traffic will continue to keep an important position. Take P2P file-sharing traffic for example, it still accounts for 19% of the total Internet traffic in 2019, and the impact it brings to operators' network due to its special traffic distribution mode cannot be belittled.

The idea of content distribution, decentration and data-based indexing and retrieval is first proposed for P2P which serves as a transitional technology for content distribution and sharing and as a predecessor of the architecture of information-centric networking system. P2P technology has changed the original client server mode (CS mode) of the Internet, allowing resources not to be concentrated on a few expensive servers, but widely distributed in many terminal nodes, and there is no longer to depend on the capability of the central server to obtain resources. The scalability of P2P is achieved by reducing and balancing the central load, which makes full use of the self-organization ability, adaptivity and fault-tolerant distribution mechanism among nodes. In various P2P systems, the client actively serves as an agent to parse resource names for other candidate nodes, retrieve and download data segments from the selected nodes. In terms of topological structure, P2P, from the original central topological structure to the completely distributed structure, has a variety of structures.

Distributed Hash Table (DHT) is used by most P2P networks as a basic protocol to improve the scalability of the architecture. The representative DHT protocols include Chord, CAN and Kademlia. In terms of the field of application, P2P technology has covered distributed scientific computing, file sharing, voice communication, streaming media on-demand and live online game support platform, and mail system, etc. However, P2P is only suitable for providing a proprietary data transmission and processing model for specific applications and has a strong dependence and attribution on the applications, so it is not a universal solution.

4.2 NDN and CDN

In 2019, the traffic transferred by the Content Delivery Network (CDN) accounted for more than half of the Internet traffic. Currently, large Internet content providers use CDN services to accelerate content distribution of static and dynamic data. The core idea of CDN is to build a coverage network and actively schedule contents to the network edge closest to users, so that they can get the required content nearby. In addition, it aims to solve the congestion problem of the Internet, improve the speed of response to users, and enhance processing performance by increasing storage. For live video streaming, the overlay structure constitutes a multi-cast tree to reduce bandwidth consumption and improve performance. The distributed content deployment of CDN avoids the bottleneck on the Internet that may affect the speed and stability of data transmission wherever possible, and replicates the data content for different operators' networks to reduce interconnected traffic, thus reducing loan costs. CDN provides more control for service providers who can make macro-control based on regions, user density, content popularity or other factors to make content transmission faster and more stable. Meanwhile, CDN system can intelligently schedule users' requests and assign them to their nearest service node according to system parameters, such as network traffic, connected load of each node, data popularity, network distance from users and response time, so as to improve efficiency and performance.

The current CDN infrastructure requires a range of complex management tools and scheduling strategies. Private high-speed links are even used among some key data centers to configure the location, path and management performance of data objects, and to evaluate usage. That leads to high cost of capital. The solution is exclusive and depends on the specific CDN network, which means that there is no universal method. Therefore, there is no interoperability among different CDN systems, and it is impossible to share data among different CDN networks, or connect them. In this case, cross-network storage and communication resources cannot be effectively utilized. At the same time, due to the high cost of synchronization among CDN data centers, it is difficult to support some applications with strong real-time performance, small data granularity and frequent interaction, such as twitter, social network and video conference.

Currently, consumers are generally identified by their location, such as an IP address. However, in general, information should not be interpreted by an address. For example, the service of BCC iPlayer can only be accessed by a British IP address, so it's very difficult for a legitimate citizen in the country to deal with mobility when temporarily using a foreign connection. BCC iPlayer selects the best content copy according to the IP address of a visitor, even if the visitor may change his position in the future. In contrast, NDN explicitly separates location and identification, thus providing a good mechanism for nodes to be repositioned without artificial manipulation of their permanent address. Thus, the location of a node can be changed seamlessly, with a consistent identity maintained.

4.3 Correlation and Comparison of NDN, CDN and P2P

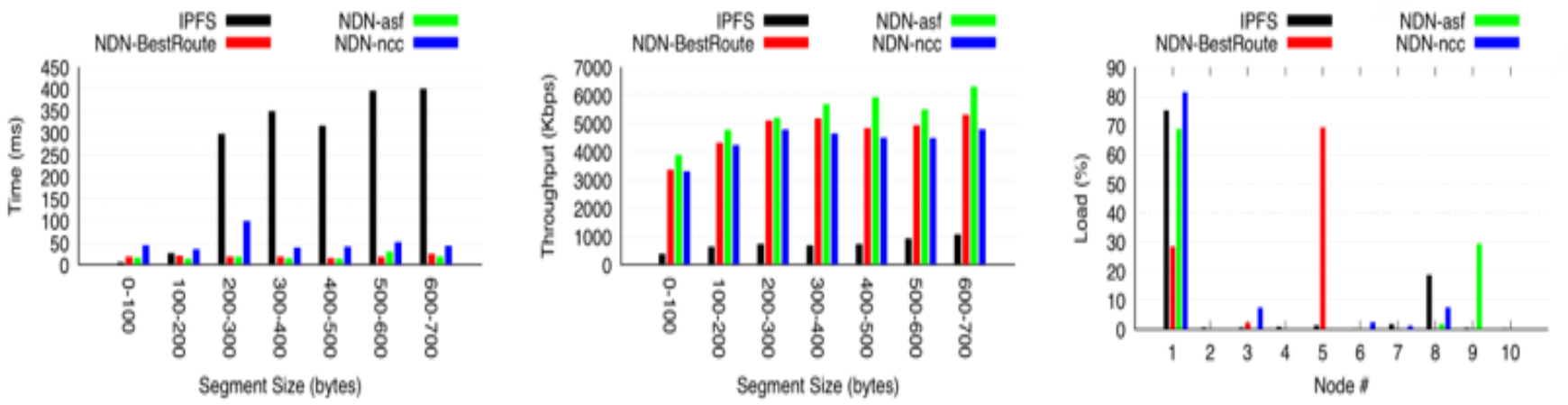
Comparison	P2P	CDN	NDN
Networking protocol	Connection–based End–to–end communication Self–organizing topology Dynamic regulation and planning Distributed storage High jitter Complex structure and high management requirements	Connection–based Most are hierarchical Static topology Static deployment Distributed storage of data, active scheduling of content, and moder–ate dynamic planning High requirements for management and monitoring	Content–centric Interest driven Pull method Flexible routing strategy Nearby acquisition Fault–tolerant and disruption–tolerant Low management requirements
Architecture	Structured or hybrid architecture Central control is still required Network coverage at the application layer Dependent on application Poor universality, general availability and medium scale	Tree and star topologies are in the majority Complete central control Network coverage at the application layer Dependent on application Poor universality, high availability and local scale	Completely decentralized Serverless Transport layer network Friendly to the top layer Strong universality, high reliability and availability, large–scale processing capacity
Security	Additional system–level protection mechanism Passive data management Difficulty in identification	Private network and DNS control scheduling ensure access security. Proprietary security protection capability of data center DDOS attacks still exist	Data naming and signature Self–encryption of data Self–protection capability, unrelated to transmission
Use of broadband	High data redundancy and massive data repli–cation Main cause leading to network congestion	Dedicated high–speed bandwidth connected to the data center. Reduced impact on back–bone network, less occu–pation of backbone network, great benefit at the edge	Nearby acquisition, intelligent data replication and shared transmission. Demand–based dynamic self–reg–ulation of loads, intelligent routing strategy and multiple routing mechanism
Copyright control	Rampant piracy, lack of effective copyright control Operator rejection	Based on central certification and control Bound with applications and services Additional DRM technology	Data naming and signature Public and private key mechanisms Unrelated to storage and transmission

Green computing	Bandwidth waste and duplicate transmission Much energy waste	High bandwidth require–ments High energy consumption Dedicated high–perfor–mance storage solutions High energy consumption in data center	High rate of resource utilization and reuse Saved investment, especially for multimedia content distribution, such as video playing and conference
Business model	No effective billing mode Low cost of capital Unclear business model	Effective billing mode, without accurate quantification of effects High cost of capital	Effective billing mode High granularity, accurate quantification High cost performance New routing equipment, OS design and application service market, with the great potential

5 Use of NDN to Enhance Data Retrieval Capability of IPFS

In SIGCOMM 2019, a paper named《Towards Peer–to–Peer Content Retrieval Mar–kets: Enhancing IPFS with ICN》provided descriptions on the feasibility of using NDN to enhance IPFS retrieval capability, summarized as follows:

	IPFS	NDN
Naming	Content hash	Hierarchical name
Human–readable names	No	Yes
Name Lookup and Versioning	IPNS + DHT	Naming Conventions + NDNS
Collections	Collection File	Naming convention or Manifest
Routing	DHT (Kademlia)	FIB + NDNS
Routing Table Size	O(d/n))	O(d)
Lookup speed	O(log(n))	O(1)
PDU	Bitswap Messages	Bitswap Messages Interest + Data
Security	Merkle DAG	Signatures



(a) Resolution time (b) Throughput (c) Load distribution

Figure 3: Preliminary results

In the model experiment in this paper, the retrieval efficiency of IPFS network with NDN is more than ten times that of an ordinary IPFS network. The main reason is that in the IPFS network, due to lack of an effective scheduling mechanism, the valid data accounts for 10% merely of all the data received by most retrievers, and the rest are duplicate packets.

6 Economic System Design

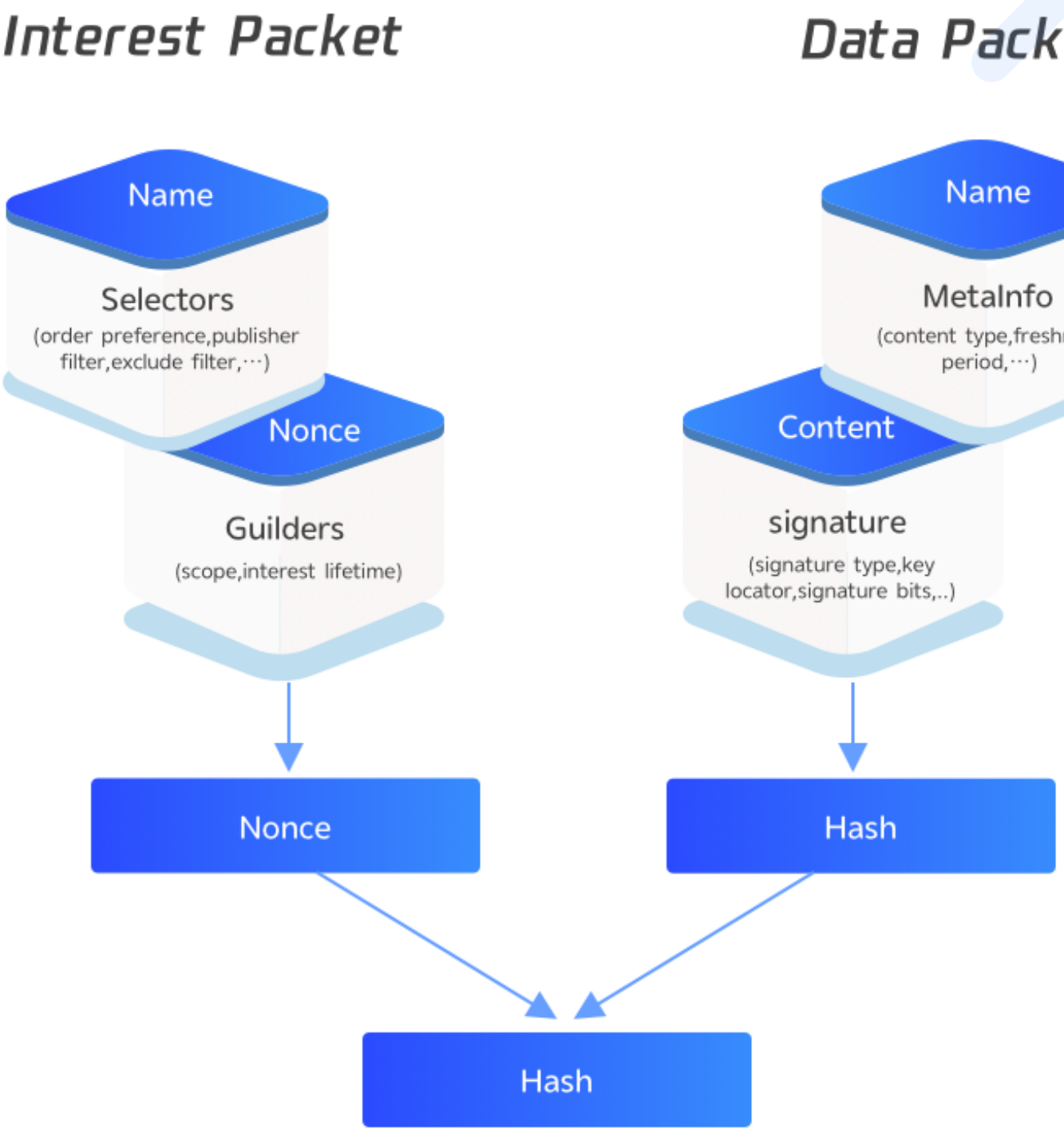
6.1 Token System Design

The Token name of NDN Link project is NDN with a total number of 5 billion.

For the distribution of NDN, the POW strategy is adopted, and the routing node needs to provide PoW based on Interest packet Nonce and Data packet Hash. Since both the In–terest and Data packets can't be forged, the challenge posed by the design of the con–sensus part is relatively small.

In the NDN link project, we use the POW consensus mechanism without random Nonce, which is called Reusable Proofs of Work (RPOW). When running random Hash operation, we will take the Hash generated by the content of Data packet and the Hash value com–posed of Nonce of Interest packet as the POW in each block generation cycle. Since all data packets in the hop–by–hop network of NDN are signed by the producer of content, and the network has no concept of an address, there is no way to transfer forged pack–ets in the network by cheating to form POW. Accordingly, we do not need to introduce random Nonce in our POW.

After the forwarded POW is generated, the whole consensus system will be responsible for block packaging and generation.



The second purpose of Token is to be paid by the content's consumer to the producer.

7 References

[1] NS–3–based NDN simulator.http://ndnsim.net.
[2] A. Ghodsi, T. Koponen, J.Rajahalme, P. Sarolahti, and S. Shenker. Naming in con–tent–oriented architectures.In ACM SIGCOMM Workshop on Information–Centric Network–ing (ICN), 2011.
[3] M. Abadi. On SDSI'slinked local name spaces. Journal of Computer Security, 6(1–2):3{21, Oct. 1998.
[4] A. Afanasyev, P.Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang. Interest ooding attack and countermeasuresin Named Data Networking. In Proc. of IFIP Networking, May 2013.
[5] C. Bian, Z. Zhu, A.Afanasyev, E. Uzun, and L. Zhang. Deploying key management on NDN testbed.Technical Report NDN–0009, Rev.2, Feb 2013.
[6] M. Bogu~n_a, F.Papadopoulos, and D. Krioukov. Sustaining the Internet with Hyperbolic Mapping.Nature Comms, 1:62, 2010.
[7] J. Burke, P. Gasti, N.Nathan, and G. Tsudik. Securing instrumented environments over Content–CentricNetworking: the case of lighting control. In IEEE INFOCOM 2013 NOMEN Workshop,Apr. 2013.
[8] CCNx. Ccnx software. http://www.ccnx.org.
[9] P. Crowley. Named DataNetworking (Demo). In China–America Frontiers of Engineering Symposium, Frontiersof Engineering, 2013.
[10] S. K. Fayazbakhsh, Y.Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S.Shenker. Less pain, most of the gain: Incrementally deployable ICN. SIGCOM–MComput. Commun. Rev., 43(4), Aug. 2013.
[11] G. Grassi, D. Pesavento,G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang. VANET via Named Data Networking.In IEEE INFOCOM NOMEN Workshop, Apr. 2014.
[12] T. R. G. Green and M.Petre. Usability analysis of visual programming environments: a \Cognitive dimensions"framework. Journal of Visual Languages and Computing, 7(2), 1996.
[13] J. Y. Halpern and R. vander Meyden. A logic for SDSI's linked local name spaces. In IEEE Computer SecurityFoundations Workshop, 1999.
[14] A. Hoque, S. O. Amin, A.Alyyan, B. Zhang, L. Zhang, and L. Wang. Named–data link state routing protocol.In ACM SIGCOMM ICN Workshop, 2013.
[15] V. Jacobson, , D. K.Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Bray–nard. Networkingnamed content. In CoNEXT, 2009.
[16] D. Krioukov, F.Papadopoulos, M. Kitsak, A. Vahdat, and M. Bogu~n_a. Hyperbolic ge–ometry ofcomplex networks. Physical Review E, 82:036106, 2010.
[17] D. Kulinski and J.Burke. NDN Video: Live and Prerecorded Streaming over NDN. Tech–nical Report NDN–0007,Sept 2012.

[18] P. Mahadevan, D.Krioukov, M. Fomenkov, B. Hu_aker, X. Dimitropoulos, kc cla_y, and A. Vahdat.The Internet AS–level topology: Three data sources and one de_nitive metric. ComputCommun Rev, 36(1), 2006.
[19] I. Moiseenko and L.Zhang. Consumer–Producer API for NDN. Technical Report NDN–0017, Feb 2014.
[20] NDN Team. Named DataNetworking (NDN) Project 2012 – 2013 Annual Report, Sept 2013.
[21] NDN team. NDN ForwardingDaemon, 2014. http://named–data.net/doc/NFD/current/.
[22] NDN team. NDN Platform,2014. http://named–data.net/codebase/platform/.
[23] W. Shang, Q. Ding, A.Marianantoni, J. Burke, and L. Zhang. Securing building manage–ment systems usingnamed data networking. IEEE Network Special Issue on Informa–tion–CentricNetworking, Apr 2014.
[24] W. Shang, J. Thompson,M. Cherkaoui, J. Burke, and L. Zhang. NDN.JS: A JavaScript Client Library for NamedData Networking. In IEEE INFOCOM 2013 NOMEN Workshop, Apr 2013.
[25] K. Shilton, J. Burke, kccla_y, C. Duan, and L. Zhang. A World on NDN: A_ordances and Implications of NDN.Technical Report NDN–0018, April 2014.
[26] W. So, A. Narayanan, andD. Oran. Named data networking on a router: Fast and DoS–resistant forwardingwith hash tables. In ACM/IEEE Symposium on Architectures for Networking and CommunicationsSystems (ANCS), Oct 2013.
[27] M. Varvello, D. Perino,and J. Esteban. Caesar: A content router for high speed for–warding. In ACM SIGCOMMWorkshop on ICN, 2012.
[28] L. Wang, A. K. M. M.Hoque, C. Yi, A. Alyyan, and B. Zhang. OSPFN: An OSPF–based routing protocol forNDN. Technical Report NDN–0003, July 2012.
[29] G. Xylomenos, C.Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsa–ros, andG. Polyzos. A survey of information–centric networking research. IEEECommunica–tions Surveys Tutorials, 2013.
[30] C. Yi, J. Abraham, A.Afanasyev, L. Wang, B. Zhang, and L. Zhang. On the role of rout–ing in NamedData Networking. Technical Report NDN–0016, Dec 2013.
[31] C. Yi, A. Afanasyev, I.Moiseenko, L. Wang, B. Zhang, and L. Zhang. A case for stateful forwardingplane. Computer Communications: ICN Special Issue, 36(7):779{791, April 2013.
[32] C. Yi, A. Afanasyev, L.Wang, B. Zhang, and L. Zhang. Adaptive Forwarding in Named Data Networking. ACMSIGCOMM CCR, 42(3), 2012.
[33] H. Yuan and P. Crowley.Scalable pending interest table design: From principles to practice. IEEE INFOCOM,2014.
[34] H. Yuan, T. Song, and P.Crowley. Scalable NDN forwarding: Concepts, issues and principles. In ICCCN,2012.
[35] Z. Zhu and A. Afanasyev.Let's ChronoSync: Decentralized dataset state synchroniza–tion in NDN. In ICNP,2013.
[36] Z. Zhu, C. Bian, A.Afanasyev, V. Jacobson, and L. Zhang. Chronos: Serverless multi–user chat over NDN.Technical Report NDN–0008, October 2012.

[37] Z. Zhu, J. Burke, L.Zhang, P. Gasti, Y. Lu, and V. Jacobson. A new approach to secur–ing audio conferencetools. In Asian Internet Engineering Conference, AINTEC, 2011.

ndn.link