What's in a Name? Naming Big Science Data in Named Data Networking

Susmit Shannigrahi
Tennessee Tech
Cookeville, TN
sshannigrahi@tntech.edu

Chengyu Fan
Colorado State University
Fort Collins, CO
chengyu.fan@colostate.edu

Craig Partridge
Colorado State University
Fort Collins, CO
craig.partridge@colostate.edu

ABSTRACT

Data naming is the most critical construct of Named Data Networking (NDN). The way a piece of content is named has profound impacts on content discovery, routing of user requests, data retrieval, and security. Besides, the naming of individual pieces of content seriously affects how *the network* behaves. While names are ubiquitous in NDN, the design choices for content names and how they affect the network have largely been overlooked. NDN applications and protocols usually name content to fit their particular application scenarios, often derived from existing naming conventions. However, these ad-hoc naming schemes often ignore the impact of these names on the network and the applications themselves. Drawing upon our experience in applying NDN to multiple science domains, we point out different exiting naming schemes in scientific communities, how we translated these names into NDN names, and the effect of naming on the network. Based on these observations, we provide a set of naming guidelines for future scientific applications and network operators supporting those applications.

CCS CONCEPTS

• **Networks → Naming and addressing**;

ACM Reference Format:
Susmit Shannigrahi, Chengyu Fan, and Craig Partridge. 2020. What's in a Name? Naming Big Science Data in Named Data Networking. In *Proceedings of 7th ACM Conference on Information-Centric Networking, Virtual Event, Canada, September 29-October 1, 2020 (ICN '20),* 12 pages.
https://doi.org/10.1145/3405656.3418717

1 INTRODUCTION

In today's Internet, data naming only affects application-level functionality; a misspelled file name or a wrong URL makes data access difficult but does not affect network functionalities (e.g., packet forwarding or routing). Naming in today's Internet is also application-specific; for example, an HTTP web server's content names do not affect the content names used by an FTP server, except when interoperability is needed.

ICN '20, September 29-October 1, 2020, Virtual Event, Canada

ACM ISBN 978-1-4503-8040-9/20/09…$15.00
https://doi.org/10.1145/3405656.3418717

NDN [41] is a future Internet architecture that relies on content names for most operations. In NDN, the network not only uses content names for publishing and accessing content but also for in-network caching, routing, forwarding of packets, verifying trust, and performing in-network operations such as failover and load balancing [1]. Consequently, poorly conceived naming schemes in NDN not only affects the applications but also has the potential to affect the network's scalability, performance, and usability.

One of the use cases that naturally fits into NDN's name based paradigm is scientific communities with large amounts of data. These communities use very large and geographically dispersed datasets (petabytes or more) managed by different institutes. The scale and the distributed nature of the data require efficient data management. We have shown in our previous work [24] that using a name-based system to publish, discover, and retrieve these distributed large datasets can reduce data management complexity [12, 32], simplify the underlying infrastructure [28, 29], and speed up content delivery. These communities already use hierarchical and semantically meaningful names that are also human-readable, making them easily translatable into NDN names [24, 30, 32].

While existing names are translatable into NDN names, utilizing NDN names for scientific data management requires careful naming considerations. For example, if a climate data producer names some climate data as `"/ClimateData/YYYY/MM/DD"`, the producer only needs to announce an aggregated namespace (`"/ClimateData"`) into the network. Naming the same data as `"/YYYY/MM/DD/ClimateData"` would require announcing n routes into the network where n is the number of years. Announcing a large number of routes has the potential to waste in-network states such as the Forwarding Information Base(FIB) and in-network cache space.

Our contribution in this paper is as follows: first, we discuss current naming schemes in scientific communities and our experience in converting these existing names into NDN names. Second, we discuss how naming affects network operations, NDN data structures, as well as application functionality. Finally, we present a list of naming recommendations for the scientific communities and network operators supporting these communities.

2 BACKGROUND AND MOTIVATION

2.1 Named Data Networking

Named Data Networking (NDN) [41] is an instance of Information Centric Networking (ICN) paradigm. NDN encourages the use of application-defined, hierarchical, and human-readable names for data publication, discovery, and retrieval. Consumer applications send "Interests" for named data that the routers forward based on the names. Upon reaching a data source (a producer, a proxy, or an

in-network cache), the Interest brings back a "Data packet" using the reverse path. The data producer digitally signs the data to establish provenance. To forward Interests towards the Data producer, NDN utilizes [1] a Forwarding Information Base (FIB) that stores name prefixes and uses longest prefix match to find one or more outgoing interfaces. The Content Store (CS) caches returning Data packets and uses exact match to satisfy future Interests with the same name. Finally, a "strategy layer" allows NDN to utilize per namespace in-network states (e.g., delay measurements, packet loss). NDN also allows forwarding of Interests to a specific repository or service by including a "forwarding hint". The first network entity that receives this Interest can then redirect it to a separate namespace. For example, an Interest with the name `"/BIOLOY/SRA/123"` with a forwarding hint of `"/NCBI"` will be forwarded to `"/NCBI"`. In other ICN architectures such as CCNx, the concept of manifest [21] can provide a list of data packet names to the requesting applications. While NDN does not support a manifest at the network layer, an application can easily provide such a service [12].

2.2 NDN Naming Studies

Several previous studies have looked into NDN naming. Thompson et al. [37] created a name-based API that enables applications to work with hierarchical, named collections of data using an abstract interface. Afanasyev et al. created NDNS [5] for name lookup using a DNS like mechanisms. Shi et al. [33] present a mechanism to filter packets based on their content names. Tehrani et al. [36] have presented a DNSSEC-like solution for ICN namespace management. Jahanian et al. present an analysis framework [16] that models and analyzes NDN data planes that perform name-based verifications. Cao et al. [8] provide a mechanism to announce content names into the network to create a CDN-like mechanism. Shang et al. looked at naming in the context of a building management system [26], and Grassi et al. looked at NDN naming for vehicular networks [14]. As part of our previous studies, we have looked into data naming in science datasets [24] [12] [19] [31] [20]. The NDN community has also published a tech report [3] that provides general guidelines for NDN application developers. However, all these studies looked at naming from specific applications' perspectives. Naming trade-offs across different application scenarios and how the naming choices affect NDN network operations have largely been overlooked. Our work addresses this gap in the literature.

2.3 Motivation

We have long collaborated with scientific communities such as climate science [24] [27], high-energy particle physics [32] [12], and genomics [30] for applying NDN to their applications. These collaborations have provided us with unprecedented insight into these communities' data management problems. The problems are in multiple dimensions - in publishing data under a consistent naming schema, in discovering geographically distributed datasets published under different namespaces, retrieving a large amount of data while reducing latency and bandwidth usage, and creating an infrastructure that hides some of the underlying data management complexity (e.g., data discovery) from the users.

Many scientific communities already use semantically meaningful, hierarchical names for organizing their data. Our experience shows that NDN's name based operations naturally match this model and can alleviate the data management problems by operating directly on the names. We also found that different scientific communities share many similarities in how they name their content and manage their data. Therefore, documenting the trade-offs of NDN's naming conventions and creating a general naming guideline will allow scientific communities (scientists and network operators alike) to name data in such a way that aligns data management operations with the underlying NDN network.

3 EXISTING NAMING IN SCIENTIFIC COMMUNITIES

This section discusses three NDN naming schemes that we have developed and deployed in collaboration with three scientific communities - climate science, high-energy particle physics, and genomics.

NDN naming paradigm fits nicely with the existing naming conventions of science data. NDN places few restrictions on naming, merely encouraging that, (a) the name structure is hierarchical, (b) naming rules are globally agreed upon among content users, (c) name prefixes are allocated to publishers (similar to how the current DNS system assigns domain names), and finally, (d) names are human-readable and semantically meaningful providing some additional level of assurance [41].

Currently, scientific content names can be divided into two categories. Some communities, such as the three we discuss here, have established naming conventions (though strict enforcement is often lacking). We also observe many similarities in how these communities name their data. These names usually follow a hierarchical directory structure and the file names are human-readable to allow the scientists to gather information from the names. In some other cases, names are human readable but present no information about the contents of the file. In the communities that have not established a naming convention, content naming is ad-hoc. We do not discuss those names in this work.

The fact that names are hierarchical fits naturally into the NDN paradigm. However, we found that all names are not created equal, we have often encountered names where components were arbitrarily ordered in the naming schemas, components were missing or transposed, and components disagreed with the files' actual data. In our experience, converting HEP names to NDN names requires simply replacing the delimiters. In other cases, such as climate and genomics, we needed to add additional components, remove duplicate components, and check the name components for errors. In some cases, these operations needed consulting the metadata in the files and even the data itself to mine for missing name components.

The following sections provide more insight into current data naming in scientific communities and how we converted these existing names to consistent NDN names.

3.1 Climate Data Naming

The climate communities usually access individual files [12]. Each of these file names has several components that describe various attributes of the data, such as which project generated them, the

temporal range of data in the file, and the institute name where it was generated.

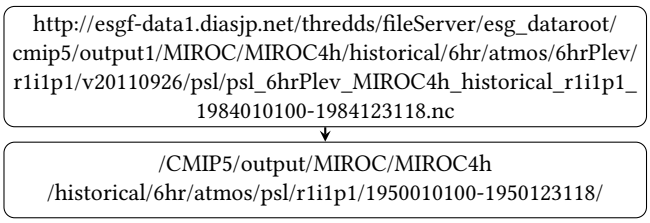

Figure 1: Naming a climate dataset into NDN. The name at the top is an existing name, the name at the bottom is the translated NDN name. Both names encode all the necessary information.

Figure. 1 shows a file name from the Coupled Model Intercomparison Project 5 (CMIP5)[35], a project that aims to model climate change. In this exaple, "`http://esgf-data1.diasjp.net`" is the host name, `/thredds/fileServer/esg_dataroot/cmip5/output1/MIROC/MIROC4h/historical/6hr/atmos/6hrPlev/r1i1p1/v20110926/psl/` is the location of the file on the filesystem and "`psl_6hrPlev_MIROC4h_historical_r1i1p1_1984010100-1984123118.nc`" is the actual file name. Note that some of the components in the original name appears multiple times (e.g., `/MIROC4h` and `r1i1p1`) The duplicity is an artifact of data organization, files need to be identified as part of a larger collection and filenames need to describe the content of the file for the applications that use it. In this example, CMIP5 is the top level activity that generates the files, output is the sub-activity (there might be many under an activity), MIROC is the organization, and the MIROC4h is the model. These name components are standard across many different climate projects [24] and these names can be generalized as `/Activity/[Sub-Activity or Product]/Organization/Model/`.

A straightforward translation of the name in Figure. 1 will start with the hostname, "`/esgf-data1.diasjp.net`". However, since we no longer need the hostname and the filesystem location, we removed these, as well as the duplicate components from the name. We take the broadest prefix from the filename (`/cmip5` in this case) and append the other components that include the necessary information (pointed out by the scientists) for this community. The resulting name is much shorter but retains all the necessary components. In this case, `/cmip5` (or `/cmip5/output`) is the routable prefix and the rest is the model-specific portion of the name. Together, these portions form a complete NDN name. The routable part of the name aims to get an Interest routed to the set of machines that host the CMIP5 data. The model-specific part allows the applications to operate on the names. In some scenarios, scientists found a few additional components that were originally missing from the naming schema. We added them into the name - the sampling rate (6hr) was one such component. The lower portion of Figure. 1 shows the data name after translation.

It is relatively easy to see how such names can be extended to support additional operations and services. If a data producer supports subsetting (a special example of on-demand data generation), retrieving a subset of a specific dataset requires the scientist to add key-value pairs such as `subset-variable=temperature` and `latitude = 30,60` and `longitude = 90,120` to the name.

3.2 High Energy Particle Physics data naming

High-energy physics (HEP) communities generally utilize "datasets," a collection of files. There is little importance of individual file names in this community, and they can often be hexadecimal strings [32]. However, the directory structure is critical since it contains vital information such as the parameters to expect in the data, when data was generated, and which experiment generated the data. Not knowing the exact filenames does not impede data access since consumers can ask for all files under a specific namespace.

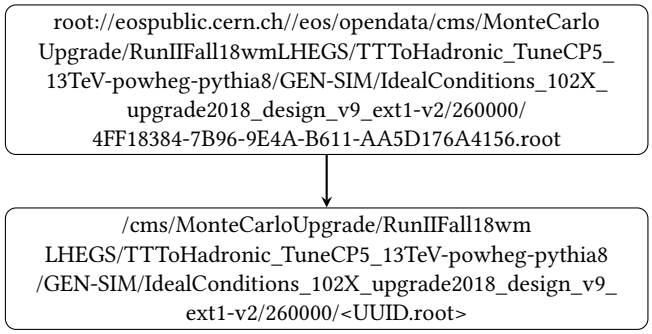

Figure 2: Naming a HEP dataset into NDN. The name at the top is an existing name, the name at the bottom is the translated NDN name.

HEP datasets are already named using a hierarchical naming schema [32]. The Compact Muon Solenoid experiment at CERN (CMS) generates two types of data: Data from the detector (D) and data from Monte-Carlo simulation (MC). Both types of data are made available to the users in the form of datasets. The dataset is a logical container and might have several files inside them [2]. Figure. 2 shows such a fle name where "`eospublic.cern.ch`" is the host name, `/eos/opendata/cms/MonteCarloUpgrade/RunIIFall18wmLHEGS/TTToHadronic_TuneCP5_13TeV-powheg-pythia8/GEN-SIM/IdealConditions_102X_upgrade2018_design_v9_ext1-v2/260000/` is the directory name, and `4FF18384-7B96-9E4A-B611-AA5D176A4156.root` is the actual file name.

The hierarchical naming structure of NDN is especially suited for HEP datasets, which are already named using a hierarchical name schema. Note that since the names are already agreed upon, the NDN names require little or no change. In this example, the original name is divided into logical segments separated with "/"s. The filename described above remains the same, except we remove the location information (cern.ch) and the filesystem information (/eos/opendata/) that we do not need. Since CMS is the organization creating these datasets, `/cms` is the first prefix in the name. CMS may decide to publish these datasets from multiple locations using NDN.

3.3 Genomic data naming

Genomic datasets can be published from centralized repositories or individual laboratories. For example, the National Center for Biotechnology Information (NCBI) in Maryland, USA, hosts 40 petabytes of high-throughput DNA sequence data with varying degrees of associated metadata. However, individual research labs maintain the data before publication and often share data with

collaborators through ad-hoc approaches or shared data grids like iRODS (the Integrated Rule-Oriented Data System). Regardless of where the dataset resides in the data life cycle, the data could be quite useful for distributed research teams and potentially thousands of genomics researchers if the datasets became discoverable. NDN can improve data discoverability by allowing the scientists to (at least initially) publish their data under an agreed-upon namespace without having to go through a centralized data publication agency.

Modern genomic DNA data comes in the form of "static" reference genomes with coordinate-based annotation files, and "dynamic" measurements of genome output (e.g., RNAseq data files that contain RNA molecule snapshot strings in the tens of millions of sequence records) [10]. A common aspect of genomics datasets is that they can be named in a community agreed, evolution-based, hierarchical manner that are easily mappable into an NDN framework. For example, an NDN name take the agreed-upon form of "`/Genome/genus/species/infraspecific-name/assembly-name`".

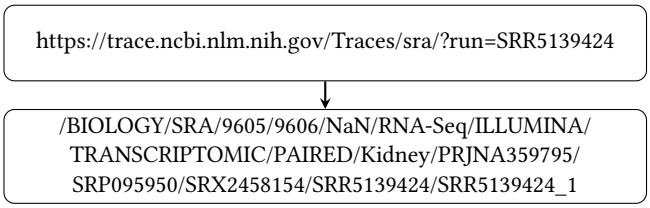

Figure 3: Naming a genomics dataset into NDN. The existing file names do not embed any information.

In Genomics, these files are combined into logical containers. These container names are concise and do not provide any useful information. Figure. 3 shows such a name. `https://trace.ncbi.nlm.nih.gov/` is the hostname, "`/Traces/sra`" is the location and `SRR5139424` is the SRA-ID (the container). The data name provides little information about the contains of this container. In collaboration with the community, we created a naming scheme that encodes useful components into the name [30]. The NDN name derived from file directory information and metadata provides more information (that it contains RNA sequence of kidney, along with other domain-specific information) to the scientists and is easier to index.

The logical container hosts multiple files with the same base name with different extensions (see Table 1). The common part of these names is the base portion of the name, and the extensions denote what types of data they contain. For example, one file might contain the base pair of a Genome, and another might content annotations. Since all files in the dataset have the same base name with different extensions, applications that use these names might be able to infer the existence of various files if they know the base name. Using only the base name for inferring filenames is convenient but implicitly assuming the existence of files based on a base name might be problematic if files are missing.

4 THE EFFECT OF NAMING ON NDN OPERATIONS

In NDN, in-network operations such as caching of content, packet forwarding, routing, and measurements are name-based. The following sections discuss how naming choices affect these *in-network* operations. In this section, we also introduce the concept of a minimum usable data unit (MDU) in the context of scientific datasets.

4.1 Name Length and Expressiveness

NDN encourages names to be hierarchical, human-readable, and semantically meaningful. The same content can have short names such as "`/BIOLOGY/SRR5139424_1`" or longer and more descriptive such as `/BIOLOGY/SRA/9605/9606/NaN/RNA-Seq/ILLUMINA/TRANSCRIPTOMIC/PAIRED/Kidney/PRJNA359795/SRP095950/SRX2458154/SRR5139424/SRR5139424_1`. Both names work well in NDN though the longer name is more useful for most operations (albeit at a higher storage and processing costs and reduction of available space for the payload in the returning data packets) as we discuss below. In the subsequent tables, we use the terms *expressive names* and *shorter names*. *Expressive names* are longer and hold more semantic information while *shorter names* are brief and do not contain as much information. The expressive names allow the users, applications, and network operators to better interpret the information necessary to perform various functions. Note that we use these terms to specify a design continuum rather than the exact length of the names.

4.2 Naming and Minimum Usable Data Units

Table 1: An example MDU in the Genomics Community. It contains the genetic sequence of a fungal pathogen. All these files are necessary to run a computation, a single file is not important. We have shortend these names in the interest of brevity.

```
/Trichosporon/asahii/1.1/ht2
/Trichosporon/asahii/1.2/ht2
/Trichosporon/asahii/1.3/ht2
/Trichosporon/asahii/1.4/ht2
/Trichosporon/asahii/1.5/ht2
/Trichosporon/asahii/1.6/ht2
/Trichosporon/asahii/1.7/ht2
/Trichosporon/asahii/1.8/ht2
/Trichosporon/asahii/1/fa
/Trichosporon/asahii/1/gff3
/Trichosporon/asahii/1/gtf
/Trichosporon/asahii/1/meta.json
/Trichosporon/asahii/1/Splice_sites
```

In many science communities, such as high energy particle physics and genomics, individual files are not sufficient for experiments. Rather, a set of such files (often referred to as a dataset) are used together for experiments. We introduce the term "Minimal Usable Data Units" (MDU) to refer to these datasets in NDN. The goal of an MDU is to specify a unit of data that should be kept together for network and application efficiency as we will discuss in the following sections.

Specifically, an MDU is the unit of data that is used by domain applications. An MDU can be a directory with a number of files with various extensions and used together. Table 1 shows such an MDU in a genomics community that contains the genetic sequence of a fungal pathogen, "Trichosporon asahii". The files are as follows - ht2 files hold the actual data, meta.json is the metadata, .fa file

contains nucleotide sequences, gff3 describes the genes and other features of the DNA, .gtf holds information about gene structure, and Splice_sites identifies potential locations of mutations. All these files are **always** used together, and a single ht2 file is not useful without the other files. The concept of MDU occurs in various other domains such as climate science[34], high energy particle physics[32], astronomy[7], and genomics[30]. However, the number of files in an MDU varies in different communities. In special cases, such as in some climate communities, each file can be an MDU. MDU is a higher-level concept that can be implemented by lower-level mechanisms such as catalogs and manifests, a network layer strategy, or any other application-specific mechanism. In this work, we do not dictate which mechanisms the communities should adopt since it will vary by the community and specific use cases.

In the science data context, the goal of NDN should be to handle (e.g., forward, cache and evict) the items of an MDU together. Note that we *do not* propose that the network layer pre-fetch or pre-cache the data blindly. However, as the files are fetched in response to the user requests, the Interest and Data packets should be handled consistently (e.g., same upstream and downstream, same strategy) and cached together. The concept of an MDU should be a hint to the network layer, rather than a strict requirement.

One easy way to accomplish this is to create a large file (e.g., tar or a zip file) with all these smaller files. However, since each of these files can be several terabytes, creating a very large binary MDU can lead to wasted resources and negate some of NDN's benefits, such as parallel retrieval or selecting a better upstream when a route degrades. NDN does not provide a way to enumerate all files in an MDU. That is, we can not specify at the network layer that a request for `/Trichosporon/asahii` should bring in all files under that namespace (`/Trichosporon/asahii/*`). Since the individual Interests have no notion of an MDU, NDN can (potentially) forward them over multiple paths and create smaller caches that do not have the full MDU. When the next request comes in, there is no way of directing the Interests towards a cache that has the full MDU. Here is an example that illustrates this problem using Figure 4. Imagine a client is requesting two files - `/Trichosporon/asahii/1.1/ht2` and `/Trichosporon/asahii/1.2/ht2`. The first file follows the path $D \rightarrow B \rightarrow A$ and the second file follows the path $E \rightarrow C \rightarrow A$. For subsequent requests, there is no way to direct the Interests for `/Trichosporon/asahii/1.1/ht2` to go through B where it is cached. If this request goes through C, then the cache space in B is wasted, and the client (A) receives no benefit from NDN's caching. A similar problem happens with cache eviction too - not sending an Interest past a nearby cache causes cache eviction faster even when content is present nearby.

Since this concept is useful for the applications (at least for the scientific applications), the NDN community should explore how to integrate MDU's notion into the network layer. This integration of MDU into the network layer can be community-specific and first deployed on scientific networks such as CERN's LHC Optical Private Network (LHCOPN) or ESnet.

4.3 Naming and In-network Caching

Content caching in NDN is based on names, and by default, NDN utilizes exact name matching. We can use a special flag, "canBePrefix," to utilize a partial prefix match. For example, "`/Trichosporon/asahii_1/1/ht2/timestamp,flag=CanBePrefix)`" will match `/Trichosporon/asahii_1/1/ht2/`.

Table 2: Effect of Naming on Caching

	Expressive Names	Shorter Names
Popularity Based Caching	No effect	No effect
Probabilistic Caching	Better context for decisions	Less information, worse decisions
Predictive Caching	More information for prediction, better decisions	Worse decisions
Context-aware Caching	Better context, better decisions	Less information, worse decisions

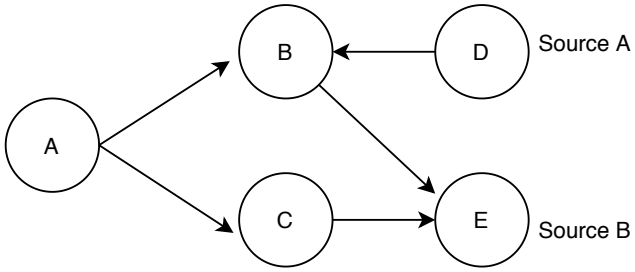

Figure 4: An NDN Topology with multiple sources and routes.

NDN caching schemes can be categorized into three broad categories - popularity based [18, 22], probabilistic[11, 25], and prediction based[4, 13]. In popularity-based caching, the cache eviction policies are based on how a popular content is. In this case, naming has no bearing on cache eviction as long as content names are not made unique (e.g., by appending timestamps).

In the context of science data, looking at only the individual content popularity does not always yield the best result. Instead, context-aware caching is useful. For example, probabilistic and prediction based caching methods can use content names to cache content speculatively. In these cases, a longer and more expressive name might provide hints about which content to cache. For example, if a user wants to study Trichosporon Asahii (see Table 1), any content that has either "Trichosporon" or "Asahii" in the name can be speculatively cached. A caching scheme may use n name components or utilize a similarity function to cache the files of an MDU. Similar to caching, files in an MDU can be evicted together.

In science data, the MDU concept also has a huge impact on how forwarding should be done. If some files in an MDU is large, once it is cached, there is a real incentive to forward future requests past the cache that holds the MDU. This is likely to reduce the aggregate network load. First, there is currently no mechanism in NDN to accomplish this. Second, the "tussle" in this context is that this

decision may not be in the requester's best interest if another faster data source is available or if the cache holding the MDU has a slower link. For optimal forwarding to a cached MDU, proximity to the cached content and network characteristics should be considered. When content sizes are large, arbitrary forwarding (e.g., round-robin) of Interests towards data sources should be avoided. Another observation is that if the network caches different parts of an MDU, the forwarding decision becomes more difficult. A manifest/catalog that enumerates the content names in an MDU should be useful to resolve this problem.

4.4 Naming and Interest Forwarding

Table 3: Effect of Naming on forwarding

	Expressive Names	Shorter Names
Default route	Finer grained forwarding	aggregated forwarding
Passive Measurement based	Finer grained forwarding, low overhead, fast	aggregated forwarding, low overhead, fast
Active Measurement based	Finer grained forwarding, high overhead, fast	aggregated forwarding, low overhead, fast
Probabilistic	More accurate than shorter names, slow	Less accurate, fast

Interest forwarding in NDN is also dependent on naming[6][38]. In NDN, forwarding can be classified into three categories - default route (dictated by routing, such as the shortest path or lowest weight), default route complemented by measurements (e.g., lowest delay, least congested), and probabilistic. Table 3 summarizes how naming affects forwarding. Longer, more granular names are useful in all three forwarding methods.

In the default routing based approach, content producers (or their proxies) announce prefixes that populate the forwarding tables. For example, if a producer publishes several files beginning with "`/Trichosporon/asahii`", it can announce a prefix "`/Trichosporon`" into the network.

In measurement-based forwarding[17], the forwarding plane utilizes the default announcements but refines the forwarding decisions based on throughput, delay, and packet loss link. NDN can also probabilistically forward packets with the hope that it will bring data back[9].

Naming plays a crucial role in efficient forwarding, especially in scientific communities. A huge amount of science data is already replicated and can benefit from NDN's measurement based forwarding. For example, a dataset is fully replicated and published (e.g.,"`/Trichosporon`") from multiple sources, the default forwarding scheme forwards Interests to one of the servers, which may not be the optimal source at that time. However, naming granularity is crucial since a generic namespace's measurement may not yield the best possible forwarding path. A content publisher that replicates content but only announces a very high-level prefix runs the risk of aggregating all sub-namespaces' performance under the larger namespace. For example, a smaller prefix of "`/Trichosporon`" may be the best route for "`/Trichosporon/asahii`" but not for "`/Trichosporon/vadense`". Rather, more specific prefixes such as "`/Trichosporon/asahii`" allow more granular handling of measurement and Interest forwarding.

Similar to the measurement-based forwarding, probabilistic forwarding can use arbitrary name component matching [9] (compared to the longest prefix match) for request forwarding. In these scenarios, more descriptive names that make sense to the users would be appropriate. If an application wants the network to forward packets based on a name component (such a rule might look for the occurrence of "`/asahii`" in a name) occurring anywhere in the name, aggregated forwarding tables that only contain "`/Trichosporon`" will not be able to forward an Interest. However, if the forwarding table contains a prefix "`/algie/amorphic/trichosporonaceae/asahii`", the Interest can be probabilistically forwarded. In this case, an expressive name should be preferred over a shorter name. Note that forwarding hints can be used to override any forwarding decision we discuss above. The usage of forwarding hints is an active research question. For example, forwarding hints might still redirect the Interests to another namespace, which may utilize one of the forwarding schemes we described above.

4.5 Naming and routing

Table 4: Effect of Naming on Routing

	Expressive Names	Shorter Names
Route Selection	More granular	Less announcements
Producer's control over forwarding	Combined longer and shorter prefix announcements provides control over forwarding	Less control over forwarding
Number of annoucements	More route annoucements may require more update messages	Fewer announcements

Routing influences how packets are forwarded in the network by populating the forwarding table (FIB)[15], therefore indirectly influences how packets are forwarded and which data sources are selected. An essential difference between forwarding and routing in NDN is that forwarding is based on FIBs in NDN routers that the network operators control. On the contrary, routing announcements are controlled by the data publisher. The data publisher does not directly get to manipulate the FIB but can indirectly influence route selection. For example, when a publisher has two data servers, and it announces "`/Trichosporon/asahii`" from both servers, the forwarding table may select either one of these possible routes [39], and the publisher has little control over which route it chooses (it may attempt to influence routing by varying the weight but the local policy may ignore that). On the other hand, if the publisher announces "`/Trichosporon/asahii`" from the preferred server and "`/Trichosporon`" from the backup server,

NDN's longest prefix match utilizes "/Trichosporon/asahii" over "/Trichosporon". Even when forwarding is probabilistic, the data publisher can announce name prefixes that aid probabilistic forwarding (e.g., a name prefix "/fungi" may be useful for forwarding "/genome/<organism name>/fungi"). Therefore, we observe that a more granular routing table ensures greater control over traffic patterns.

Content naming directly influences the aggregability of prefixes. If two species of "/Trichosporon" are named as "/Trichosporon /asahii" and "/genome/Trichosporon/vadense", the routing announcements are not aggregatable. On the contrary, "/Trichosporon /asahii" and "/Trichosporon/vadense" can be aggregated under "/Trichosporon". Note that the aggregated prefix announcement is only possible for fully replicated namespaces. In the case of partial replication, longer namespaces must be announced. For example, the Genus "Trichosporon" has seventeen subspecies. When the datasets are partially replicated, all the seventeen names must be individually announced into the network. If some organism has hundreds of subspecies such announcements can cause a FIB explosion. Using forwarding hints can retain shorter tables sizes in these scenarios. Publishers should utilize forwarding hints to a catalog that can then direct the request to a server that holds the actual data.

Table 4 summarizes these observations. While announcing a high-level prefix keeps the routing table smaller, it adversely affects other forwarding functions such as fine-grained load balancing, traffic engineering, and QoS, all of which are based on names. On the other hand, the expressiveness comes at the expense of more routing announcements and larger tables. Expressive names also allow for a more granular route section and allow the data producer to retain greater control over source selection and packet forwarding.

4.6 Effect of Naming on NDN data structures

Table 5: Effect of Naming on NDN Data Stuctures

	Expressive Names		Shorter Names	
	Size	**Control**	**Size**	**Control**
Interest Packets	Larger	N/A	Smaller	N/A
Data Packets	Smaller payloads	N/A	Larger payloads	N/A
FIB table	Larger	Fine-grained	Smaller	Coarse-grained
RIB table	Larger	Fine-grained	Smaller	Coarse-grained
PIT Table	No effect	No effect	No effect	No effect
Measurement tables	Larger	Fine-grained	Smaller	Coarse-grained

In the previous sections, we discussed how naming affects network operations. NDN utilizes two packet types and three primary data structures for communication. Table 5 summarizes how naming affects these entities. Larger names in the Interest takes resources for forwarding and reduces the available space for returning payload. Further, large names or data appended to the name or put in the "Application Parameters" field may lead to high resource usage in the network. Generally, Interests should not be too large.

Cache sizes are independent of how content is named. However, in probabilistic and context-aware caching, naming affects how content is evicted and cached. The FIB and RIB tables are larger when names are more granular. The size of the Pending Interest Table (PIT) depends on the request pattern and is not affected by content naming. Finally, in the strategy layer, measurement tables are larger but are populated with more fine-grained measurement (see the discussion in the forwarding section above) when names are expressive, allowing more granular control over route selection and forwarding.

5 NAMING AND APPLICATION LAYER FUNCTIONALITY

Having discussed how naming affects network layer functionality, we now discuss how naming in NDN affects different application layer functionality. At the end of this section, Tables 11 and 12 summarize these observations and provide a multi-dimensional view of how naming affects application as well as in-network functionality.

5.1 Data Publication

Table 6: Effect of Naming on Data Publication

	Expressive Prefixes	Shorter prefixes
Replication	Needed for partial replication	Difficult to support partial replication with higher level prefixes
Publication using forwarding hints	No effect	No effect, forwarding hints should be preferred in this case

In science communities, data is often replicated - either fully or partially. When data is partially replicated, the routing prefixes from each content provider must reflect the partial namespace they serve. Take for example, a genomic dataset that has the following naming format - "/genus/species/infraspecific-name/assembly-name". An example name in this format might be "/Trichosporon/asahii /var_asahii/cbs_2479". If all datasets under "/Trichosporon" are fully replicated, all data publishers can announce the "/Trichosporon" prefix from all repositories. However, if one repository holds the "/Trichosporon/asahii" data and another holds the "/Trichosporon /vadense" data, they must announce both prefixes.

If NDN forwarding hints[5] are used, the publication granularity can be independent of names. In this case, the names can be looked up in a catalog (or by a service) that tells the user which repository holds the actual content. For example, when a user looks for "/Trichosporon/asahii", it can use the forwarding hint to send an Interest to a predefined repository, e.g., "/Genome/Catalog". The catalog maintains a database that maps a content name to the actual repository holding the data. For example, upon looking up the repository for "/Trichosporon/asahii", the catalog might redirect the user to "/Genome/Repository/Instance1", which holds the actual data. The publishers may also bypass the catalog and directly

Table 7: Effect of Naming on Data Discovery

	Expressive Names	Shorter Names
Keyword-based search	More information, slower	Less information, faster
Autocomplete	More information provides better context, Slower	Less context, lower accuracy, faster
Predictive	More information provides better context, slower	Less context, lower accuracy, faster

include a forwarding hint to the repository. Table 6 shows naming trade-offs for data publication.

5.2 Data Discovery

Data discovery is an application-layer function. While it was (recently removed from NDN) possible to enumerate content names by (re-)expressing Interests into the network and excluding already discovered content (e.g., ask for "`/Trichosporon/asahii`", retrieve some content, reexpress the same Interest excluding the name of the already retrieved content), this approach was non-deterministic. Science communities already use name catalogs (databases that hold content names) for data discovery. These catalogs use myriad techniques for finding names- keyword search, name trees, predictions, autocomplete. Table 7 enumerates these options. Descriptive names increase the efficiency of name discovery regardless of the techniques used.

Descriptive, human-readable names are important when publishing scientific datasets. For example, `/BIOLOGY/SRA/cellular_organisms/Eukaryota/Opisthokonta/` provides enough context for the domain scientist to understand the name and the name catalog to index it. Each of these organisms also has a unique identifier associated with them (taxon IDs). So the same name can also be expressed as "`/BIOLOGY/SRA/cellular_organisms/2759/33154`". However, the second name does not provide enough information to the user for an efficient search function even though these IDs are universally agreed upon.

Table 8: Effect of Naming on Services

Special Namespace	Special Name component
Interests follow different paths for data and service	Same path for data and service
Service name must be the first component of announced prefix	Service component location in the name must be agreed upon

5.3 Data Retrieval

In NDN, content retrieval is directly based on the names. Retrieval efficiency is directly related to routing, forwarding, and content caching efficiency. Expressive naming can enable efficient forwarding, in-network load balancing, failover, and intelligent strategies, as we discussed earlier.

Load balancing A special case of data retrieval is retrieving from multiple data sources[23]. NDN can support this operation by storing multiple paths to the same namespace. However, load-balancing is more efficient when content names are more granular, and endpoints are uniquely identified. Take for example the topology in Figure 4 where both D and E announces "`/NCBI`" prefix. Even if A utilizes routes through both B and C, the Interests might end up at node E, potentially eliminating any load-balancing benefits. If the endpoints are identified either as a name component or use forwarding hints, the network can utilize both sources, albeit at the expense of location transparency.

Subsetting Subsetting is another special service where computation (or subsetting request) is sent to the data[24], and a smaller portion of the data is retrieved. Table 8 summarizes the naming trade-offs. The subsetting service can be accessed in two ways. First, it can be advertised as a service using the routing system (e.g., "`/ESGF/subsetting`"). An Interest, `/ESGF/subsetting/<data name>/subset-variable=temperature and latitude = 30,60 and longitude = 90,120`, sent to this service will bring back a data subset. However, this requires a special namespace for subsetting that does not necessarily follow the same path as data; "`/ESGF/subsetting/<data name>/<arguments>`" might follow a different path than "`/ESGF/<data name>/<arguments>`". It also requires multiple routing announcements.

Another option is to append the service name at the end (or at an arbitrary location) of the name "`/<data name>/<arguments>/subsetting`". This way, both the service requests and data follow the same path. However, this process can be fragile - a name component coinciding with a service name can cause unpredictable behavior.

5.4 Security and Privacy

Table 9: Effect of Naming on Security and Privacy

	Expressive Names	Shorter Names
Schematized trust	Opportunity to synergize trust schemes with data names, added assurance, simplified trust models	May not align well with data names
Name Based Access Control	Granular access control to a (sub)-prefix	Higher level access
Privacy	Reveals more information	Reveal less information

In NDN, security aspects such as authentication, trust verifications, and access control are name-based[42]. Additionally, naming conventions can be utilized to automate trust checking and verification (schematized trust)[40]. Naming does not directly affect trust at the network layer since NDN routers typically do not verify trust. However, a longer and more expressive name can support a set of rich and flexible trust schemas. One example might be the following - to perform a gene annotation, the "`gene-annotation`" component has to occur anywhere in a data name and must be signed

by "/human-genome/gene-annotation" key (or a sub-key signed by this key). This schema is well defined and easy to understand for the users.

Well-formed, hierarchical names are essential for schematized trust - access to a content with a name "/NCBI/Trichosporon/asahii" or "/Trichosporon/asahii/../NCBI" can be restricted only to the scientists from NCBI by requiring a user to present a key (e.g., "/NCBI/Scientist/Key") that is signed by NCBI's key (e.g., "/NCBI /Key"). While not necessary, this model provides added assurance to the users and simplifies the trust model by aligning the content names and the key(s). Unstructured names without enough context make these security associations difficult to make. Table 9 summarizes the implications of naming on security. Expressive names provide more information for trust verification, name-based access control, and security associations. At the same time, these names expose more information about the data, reducing privacy. Note that NDN does not require key names to match data names. However, consistent naming makes it easier for the communities to create a trust schema that is easy to understand and maintain.

Resource Reservation

Table 10: Effect of Naming on Reservation

	Expressive Names	Shorter Names
Per-namespace Resource reservation	Granular control over reservation	Resources reservered for a larger namespace possibly with several sub-namespaces
Per-application Resource Reservation	Name/prefix must be unique to the application, length does not matter	No effect

Resource reservation[28] is widely used in data-intensive science to ensure timely completion and data transfer performance. In big-data science, two types of resource reservation are possible, per-namespace and per-application. Similar to forwarding, aggregating a larger namespace makes it difficult to reserve resources at a finer granularity. If two genomics applications/services "/NCBI /gene-comparison" and "/NCBI/gene-annotation" are combined under "/NCBI", any resources reserved for "/NCBI/data-transfer" will also be usable by "/NCBI/gene-annotation", which may not be desirable. For resource reservation, names with finer granularity should be preferred to avoid other applications' unwanted resource usage.

Quality of Service

Similar to resource reservation, QoS in NDN is still a mostly unexplored topic. QoS mechanisms typically utilize Interest names for packet marking and shaping on a per-namespace basis. QoS over a shorter prefix potentially applies the same policy over several sub-namespaces which may belong to different applications. When longer and more expressive names are used by applications, applying per-prefix QoS for individual applications is easier.

In this section, we look at how naming affects application and network functionality. We summarize these observations in Tables 11 and 12.

6 GENERAL NDN DISCUSSION AND NAMING RECOMMENDATIONS

This section summarizes the lessons we have learned from naming science data and generalizes them into a set of recommendations for both science applications and the network operators. The recommendations are specific to science communities - other application domains might warrant a slightly different set of recommendations.

6.1 Considerations for Science Applications

(1) Scientific data names should be built as a set of well-defined *components*. Data names should be hierarchical, human-readable, and semantically meaningful. We observe that existing names are often terse and do not provide enough information for scientific applications. These names should be longer and incorporate some metadata into the names. Having more information in the names helps an NDN network to make better decisions in terms of caching, forwarding, and performance measurement. They also help the users to understand the content of the files better. Longer names with more domain-specific information make it easier to index components for searching. For example, the name "/BIOLOGY/SRA/cellular_organisms/Eukaryota /Opisthokonta" provides more information than "/SRR5139424" and should generally be preferred. However, names should include just enough (up to the scientists to decide) information to make them useful. Names should not include an exhaustive list of components - we discuss the reasons below.

(2) While science applications should prefer longer and more descriptive names, longer names are at odds with what the network should prefer. Longer names in an NDN network require additional space and computation. Fortunately, NDN allows multiple names to refer to the same object, and also provides the concept of a "link object" (a redirection from a shorter name to a longer name, or vice-versa). For example, a shorter name with essential name components might help the scientists to understand the data, and at the same time, it might point to a larger name with additional components that can be used for cataloging and indexing. Furthermore, while longer names are generally beneficial, they occupy space on the return Data packets. The default Data packet size in NDN is 8800 bytes. Care should be taken not to make the names too long since it reduces the amount of available payload space.

(3) More general (more common) name components should appear earlier in the hierarchy. In our previous example, */CMIP5* is the first component since it is the project that is generating the data. Making "CMIP5" the first component makes it easier to aggregate the namespace. Some of the scientific names can have tens of components [34]. Consequently, naming designs must weigh the trade-offs of placing various name components earlier vs. later.

(4) Science application should name data in such a way that individual datasets are identifiable as part of a larger collection (MDU). The network and the applications using these names

Table 11: Effect of expressive, longer names on application functionality

Application Functionality	Caching	Forwarding	Routing
Data Publication	(+) Useful for probablistic and predictive caching	(+) Granular forwarding (-) Larger table	(+) Better route selection, control over forwarding (-) Larger table
Data Retrieval	(+) Granular control over caching	(+) Fine grained forwarding, more information for strategy layer (+) better multipath support (-) larger forwarding table	(+) better control over forwarding (-) Larger routing table
Load Balancing	N/A	(+) Fine grained per namespace FIB entries (+) fine grained load balancing	(+) fine grained RIB entries (+) more influence over forwarding
Resource Reservation	(+) Cache reservation for smaller namespace (+) smaller cache requirement	(+) FIB/Bandwidth reservation for smaller namespaces (+) less unused resources	N/A
Privacy	N/A	(-) FIB exposes more information	(-) RIB exposes more information
QoS	N/A	(+) Fine grained, per namespace QoS	N/A

Table 12: Effect of shorter names on application functionality

Application Functionality	Caching	Forwarding	Routing
Data Publication	N/A	(+) Shorter table (-) Lesser control over forwarding	(+) Shorter routing table
Data retrieval	(-) Less information for predictive/probablistic caching	(+) smaller forwarding table (-) Coarse grained forwarding (-) less information for strategy layer (-) multipath support only for fully replicated content	(+) smaller routing table (-) less control over forwarding
Load balancing	N/A	(-) FIB entries for larger namespace (-) coarse grained load balancing	(-) less influence over forwarding
Resource reservation	(-) Cache can grow huge	(-) More resources needed (-) potentially underutilized	N/A
Privacy	N/A	(+) FIB only exposes high level prefix	(+) RIB only exposes high level prefix
QoS	N/A	(-) Aggregatated QoS for many applications	N/A

should be able to utilize these datasets together. For example, MDUs should be cached and evicted together. If one file of an MDU is selected for eviction from the cache based on some probability, the same probability can be applied to all other files in the same MDU.

(5) Naming has implications on how replication and origin selection work in NDN. In a fully replicated scenarios, the names can begin with anything (e.g., /cmip5/). In the case of partial replication, the names should be classified under a few higher-level prefixes (`/cmip5/output`).

Granular naming and route announcement ensures better control over data movement. For example, if a producer announces two namespaces "`/CMIP5/MIROC`" and "`/CMIP5`" from two servers, NDN prefers the longer prefix (by default) that can allow the data producer control which server is more utilized for content delivery.

When data is expected to be fully replicated, care should be taken not to bind names to locations. The top-level prefix should describe the data, not an organization. For example, a name "`/CSU/CMIP5`" binds the data to a location (CSU), but a name

like "/CMIP5/CSU" does not. While NDN is agnostic of data locations, existing applications are not. The familiarity with location-based naming (e.g., URLs) may influence the application developers and users to name data in a location-dependent fashion - this should generally be avoided unless there is a good justification to identify the endpoints.

(6) When datasets are partially replicated and the namespace is large, instead of announcing a large number of prefixes in the network, publishers should utilize forwarding hints to a repository that holds the actual data. This approach prevents state explosion in the network while maintaining most of NDN benefits. However, note that some NDN features, such as automatic failover, do not work well with this approach. Imagine a repository serving *n* files of name "/BIOLOGY/filename" under the namespace "/NCBI". If half of the file is located in a cache along the normal forwarding path (e.g., "/BIOLOGY"), adding the forwarding hint will bypass these caches. If the repository is broken, the forwarding hints will still bring the Interests to this repository.

(7) More expressive names provide less privacy since they expose more information. This problem can be solved by encrypting the names. However, encrypting the names might remove any caching benefits along with forwarding benefits. To address these problems, sensitive name components should be separated into public and private portions. The routing portion of the names (first *n* components) should not contain private information and should not be encrypted. The non-routing part can be encrypted without any loss in functionality.

6.2 Naming Considerations for Scientific Network Operators

(1) From the network's point of view, shorter names save space and speed up forwarding. NDN's forwarding speed is tied to the name length, utilizing longer names require more processing and in-network state, and is generally slower. In networks where devices are resource-constrained, or very high forwarding performance is needed, shorter names should be preferred. However, expressive names are preferred for one-to-many scenarios, since they help with caching and forwarding decisions.

(2) Science organizations spanning multiple sub-organizations have two options to announce name prefixes - location-dependent and location independent. For example, two sub-organizations CSU and UCLA can announce /cmip5/csu and /cmip5/ucla prefixes, respectively. The alternate approach might be to have /cmip5 as a sub-namespace of participating organizations - such as /CSU/cmip5 and /UCLA/cmip5. However, the second approach creates a fragmented namespace. More critically, the second approach also binds data to locations. This should generally be avoided since creating location-transparent in-network mechanisms with these prefixes (e.g., transparent failover) can become tricky.

(3) In the context of science data, the network's goal should be to cache and forward an MDU together. If some files in an MDU are large and they are cached, there is a real incentive to forward future requests past the cache the holds the MDU. This is likely to reduce the aggregate network load. First, there is currently no mechanism in NDN to accomplish this. Second, this decision may not be in the best interest of the requester if another faster data source is available or if the cache holding the MDU has a slower link. For optimal forwarding to an MDU, both proximity to the cached content and network characteristics to the cache should be considered. When content sizes are large, arbitrary forwarding (e.g., round-robin) of Interests towards data sources should be avoided. Another observation is that if the network caches different parts of an MDU, the forwarding decision becomes more difficult. Since scientific applications benefit from the concept of an MDU, the NDN community should consider incorporating it into the network layer as a hint.

(4) Location transparency is often at odds with efficient content forwarding and resource utilization. For example, if load balancing between multiple sources is desired, the endpoints serving the content should be uniquely identified. Without knowing either the exact endpoint or the whole network topology, it is impossible in an NDN network to specify an exact data source. Figure. 4 provides such an example. In this case, the use of forwarding hints might be useful.

This section presents general naming guidelines for NDN based scientific communities and network operators. These guidelines should simplify the naming of scientific data and allow the applications to benefit from intelligent functions in NDN networks.

7 CONCLUSIONS

Naming is a critical aspect of NDN since it dictates how the network and the applications perform. In this work, we present a study in utilizing NDN names for scientific data management and discuss the various trade-offs of naming choices and how they affect scientific networks and applications.

In this work, we find that the concept of MDU can have significant benefits such as lower cache utilization, faster data download, and less cache eviction in networks supporting big science. The concept of MDU opens up several new research directions in NDN - joint caching and forwarding decisions, caching decision optimization, and efficient forwarding of requests past existing caches. We observe that expressive names that encode application-specific information are useful to both applications and the network. However, utilizing such names in an NDN network requires more resources in the network, is expensive to process, and reduces the available space for the payload in the Data packets. In this work, we point out these trade-offs and present several naming recommendations for both the scientific community as well as the scientific network operators. We do not try to dictate the exact naming schemes but acknowledge that the exact naming choices will depend on the specific domain. We also do not explore how the names can be formalized but plan to address it in a future work.

ACKNOWLEDGMENTS

We gratefully acknowledge support from the National Science Foundation grants OAC-1659403 and OAC-2019163. We also thank the anonymous reviewers and Jeff Burke for their valuable comments and extensive feedback.

REFERENCES

[1] 2018. NFD Developer's Guide. *Technical Report, NDN-0021* Revision 9 (2018).

[2] 2019. (Jul 2019). http://opendata.cern.ch/record/12300/files/CMS_MonteCarloUpgrade_RunIIFall18wmLHEGS_TTToHadronic_TuneCP5_13TeV-powheg-pythia8_GEN-SIM_IdealConditions_102X_upgrade2018_design_v9_ext1-v2_260000_file_index.txt [Online; accessed 25. Aug. 2020].

[3] 2020. NDN Technical Memo: Naming Conventions - Named Data Networking (NDN). (May 2020). https://named-data.net/publications/techreports/ndn-tr-22-2-ndn-memo-naming-conventions [Online; accessed 19. May 2020].

[4] Noor Abani, Torsten Braun, and Mario Gerla. 2017. Proactive caching with mobility prediction under uncertainty in information-centric networks. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*. 88–97.

[5] Alexander Afanasyev, Xiaoke Jiang, Yingdi Yu, Jiewen Tan, Yumin Xia, Allison Mankin, and Lixia Zhang. 2017. NDNS: A DNS-like name service for NDN. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 1–9.

[6] Alexander Afanasyev, Cheng Yi, Lan Wang, Beichuan Zhang, and Lixia Zhang. 2015. SNAMP: Secure namespace mapping to scale NDN forwarding. In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 281–286.

[7] Massimo Brescia, Stefano Cavuoti, George S Djorgovski, Ciro Donalek, Giuseppe Longo, and Maurizio Paolillo. 2012. Extracting knowledge from massive astronomical data sets. In *Astrostatistics and Data Mining*. Springer, 31–45.

[8] Jianxun Cao, Dan Pei, Xiaoping Zhang, Beichuan Zhang, and Youjian Zhao. 2016. Fetching popular data from the nearest replica in NDN. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 1–9.

[9] Kevin Chan, Bongjun Ko, Spyridon Mastorakis, Alexander Afanasyev, and Lixia Zhang. 2017. Fuzzy interest forwarding. In *Proceedings of the Asian Internet Engineering Conference*. 31–37.

[10] Deanna M Church, Valerie A Schneider, Tina Graves, Katherine Auger, Fiona Cunningham, Nathan Bouk, Hsiu-Chuan Chen, Richa Agarwala, William M McLaren, Graham RS Ritchie, et al. 2011. Modernizing reference genome assemblies. *PLoS biology* 9, 7 (2011).

[11] Gang Deng, Liwei Wang, Fengchao Li, and Rere Li. 2016. Distributed probabilistic caching strategy in VANETs through named data networking. In *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 314–319.

[12] Chengyu Fan, Susmit Shannigrahi, Steve DiBenedetto, Catherine Olschanowsky, Christos Papadopoulos, and Harvey Newman. 2015. Managing scientific data with named data networking. In *Proceedings of the Fifth International Workshop on Network-Aware Data Management*. ACM, 1.

[13] Hesham Farahat and Hossam Hassanein. 2016. Optimal caching for producer mobility support in named data networks. In *2016 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[14] Giulio Grassi, Davide Pesavento, Giovanni Pau, Rama Vuyyuru, Ryuji Wakikawa, and Lixia Zhang. 2014. VANET via named data networking. In *2014 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. IEEE, 410–415.

[15] AKM Mahmudul Hoque, Syed Obaid Amin, Adam Alyyan, Beichuan Zhang, Lixia Zhang, and Lan Wang. 2013. NLSR: named-data link state routing protocol. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*. 15–20.

[16] Mohammad Jahanian and KK Ramakrishnan. 2019. Name Space Analysis: Verification of Named Data Network Data Planes. In *Proceedings of the 6th ACM Conference on Information-Centric Networking*. 44–54.

[17] Vince Lehman, Ashlesh Gawande, Beichuan Zhang, Lixia Zhang, Rodrigo Aldecoa, Dmitri Krioukov, and Lan Wang. 2016. An experimental investigation of hyperbolic routing with a smart forwarding plane in NDN. In *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*. IEEE, 1–10.

[18] Jun Li, Hao Wu, Bin Liu, Jianyuan Lu, Yi Wang, Xin Wang, YanYong Zhang, and Lijun Dong. 2012. Popularity-driven coordinated caching in named data networking. In *2012 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*. IEEE, 15–26.

[19] Huhnkuk Lim, Alexander Ni, Dabin Kim, Young-Bae Ko, Susmit Shannigrahi, and Christos Papadopoulos. 2018. NDN construction for big science: Lessons learned from establishing a testbed. *IEEE Network* 32, 6 (2018), 124–136.

[20] C at alin Lordache, Ran Liu, Justas Balcas, Raimondas Širvinskas, Yuanhao Wu, Chengyu Fan, Susmit Shannigrahi, Harvey Newman, and Edmund Yeh. 2020. Named Data Networking based File Access forXRootD. In *J. Phys. Conf. Ser.*

[21] Marc Mosko, Christopher Wood, Christian Tschudin, and David Oran. 2019. File-Like ICN Collections (FLIC). (Nov 2019). https://tools.ietf.org/html/draft-irtf-icnrg-flic-02 [Online; accessed 18. Aug. 2020].

[22] Muhammad Ali Naeem, Muhammad Atif Ur Rehman, Rehmat Ullah, and Byung-Seo Kim. 2020. A Comparative Performance Analysis of Popularity-Based Caching Strategies in Named Data Networking. *IEEE Access* 8 (2020), 50057–50077.

[23] Ashok Narayanan and David Oran. 2011. NDN and IP routing: Can it scale?. In *Proposed Information-Centric Networking Research Group (ICNRG), Side meeting at IETF-82*.

[24] Catherine Olschanowsky, Susmit Shannigrahi, and Christos Papadopoulos. 2014. Supporting climate research using named data networking. In *2014 IEEE 20th International Workshop on Local & Metropolitan Area Networks (LANMAN)*. IEEE, 1–6.

[25] Yang Qin, Weihong Yang, and Wu Liu. 2018. A probability-based caching strategy with consistent hash in named data networking. In *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*. IEEE, 67–72.

[26] W. Shang, Q. Ding, A. Marianantoni, J. Burke, and L. Zhang. 2014. Securing building management systems using named data networking. *IEEE Network* 28, 3 (2014), 50–56.

[27] Susmit Shannigrahi, Chengyu Fan, and Christos Papadopoulos. 2017. Request aggregation, caching, and forwarding strategies for improving large climate data distribution with NDN: a case study. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*. ACM, 54–65.

[28] Susmit Shannigrahi, Chengyu Fan, and Christos Papadopoulos. 2018. Named Data Networking Strategies for Improving Large Scientific Data Transfers. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops): Information Centric Networking Solutions for Real World Applications (ICN-SRA) (ICC 2018 Workshop - ICN-SRA)*. IEEE.

[29] Susmit Shannigrahi, Chengyu Fan, and Christos Papadopoulos. 2018. SCARI: A Strategic Caching and Reservation Protocol for ICN. In *Proceedings of the Asian Internet Engineering Conference*. ACM, 1–8.

[30] Susmit Shannigrahi, Chengyu Fan, Christos Papadopoulos, and Alex Feltus. 2018. NDN-SCI for managing large scale genomics data.. In *ICN*. 204–205.

[31] Susmit Shannigrahi, Spyridon Mastorakis, and Francisco R Ortega. 2020. Next-Generation Networking and Edge Computing for Mixed Reality Real-Time Interactive Systems. In *ICC 2020, ICN-SRA*.

[32] Susmit Shannigrahi, Christos Papadopoulos, Edmund Yeh, Harvey Newman, Artur Jerzy Barczyk, Ran Liu, Alex Sim, Azher Mughal, Inder Monga, Jean-Roch Vlimant, et al. 2015. Named data networking in climate research and hep applications. In *Journal of Physics: Conference Series*, Vol. 664. IOP Publishing, 052033.

[33] Junxiao Shi, Teng Liang, Hao Wu, Bin Liu, and Beichuan Zhang. 2016. Ndn-nic: Name-based filtering on network interface card. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. 40–49.

[34] Martina Stockhause and Michael Lautenschlager. 2017. CMIP6 data citation of evolving data. *Data Science Journal* 16 (2017).

[35] Karl E Taylor, V Balaji, Steve Hankin, Martin Juckes, Bryan Lawrence, and Stephen Pascoe. 2010. CMIP5 data reference syntax (DRS) and controlled vocabularies. (2010).

[36] Pouyan Fotouhi Tehrani, Eric Osterweil, Jochen H. Schiller, Thomas C. Schmidt, and Matthias Wählisch. 2019. The Missing Piece: On Namespace Management in NDN and How DNSSEC Might Help. In *Proceedings of the 6th ACM Conference on Information-Centric Networking (ICN '19)*. Association for Computing Machinery, New York, NY, USA, 37–43. https://doi.org/10.1145/3357150.3357401

[37] Jeff Thompson, Peter Gusev, and Jeff Burke. 2019. Ndn-cnl: A hierarchical namespace api for named data networking. In *Proceedings of the 6th ACM Conference on Information-Centric Networking*. 30–36.

[38] Yi Wang, Keqiang He, Huichen Dai, Wei Meng, Junchen Jiang, Bin Liu, and Yan Chen. 2012. Scalable name lookup in NDN using effective name component encoding. In *2012 IEEE 32nd International Conference on Distributed Computing Systems*. IEEE, 688–697.

[39] Cheng Yi, Alexander Afanasyev, Ilya Moiseenko, Lan Wang, Beichuan Zhang, and Lixia Zhang. 2013. A case for stateful forwarding plane. *Computer Communications* 36, 7 (2013), 779–791.

[40] Yingdi Yu, Alexander Afanasyev, David Clark, Van Jacobson, Lixia Zhang, et al. 2015. Schematizing trust in named data networking. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking*. ACM, 177–186.

[41] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. 2014. Named data networking. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 66–73.

[42] Zhiyi Zhang, Yingdi Yu, Haitao Zhang, Eric Newberry, Spyridon Mastorakis, Yanbiao Li, Alexander Afanasyev, and Lixia Zhang. 2018. An overview of security support in named data networking. *IEEE Communications Magazine* 56, 11 (2018), 62–68.