FIA-NP: Collaborative Research: Named Data Networking Next Phase (NDN-NP): Request for Supplement

June 1, 2016

Contents

1 Overview

Named Data Networking (NDN) (http://named-data.net/) is a new Internet architecture geared toward addressing the Internet's most pressing problems in security, management complexity, and sustainability, and meeting requirements of current and emerging applications [34]. While TCP/IP was a unique and ground-breaking architecture, it solved a problem of the telephony world: *enabling point-to-point conversations between two communication endpoints.* The world has changed dramatically in the last 30 years, driven by the spectacular success of this architecture. Growth in e-commerce, digital media, social networking, smartphone applications, and the Internet of Things (IoT) has resulted in the Internet primarily being used as an **information distribution network**. Operators, equipment designers, application developers, users, and policymakers have all struggled with the complexity (and kludges) required to accommodate the inherent misalignment between the TCP/IP architecture and its primary use today.

The conversational nature of IP is embodied in its datagram format: IP datagrams identify communication endpoints, i.e., the IP destination and source addresses. NDN generalizes today's Internet architecture by removing this restriction: names in an NDN datagram are hierarchically structured but otherwise arbitrary identifiers. The name in an NDN datagram can identify any content object, including a communication endpoint, a chunk of data in a movie or book, a command to turn on a light bulb, etc. This conceptually simple change – rooted in the recognition that **data, rather than virtual channels that contain data** is the foundation of modern communication – facilitates profound improvements in security, scalability of content distribution, support for mobility, and ease of application development.

NDN research was funded by NSF under the Future Internet Architecture (FIA) program (2010–2014), and the Future Internet Architecture – Next Phase (FIA-NP) program (2014–2016).[1] Our fundamental research challenge is to realize this data-centric vision using an architectural framework capable of solving real problems, particularly in application areas poorly served by today's TCP/IP architecture.

In the initial NDN project, our focus was to validate the basic ideas by prototyping a range of applications, and to develop the architecture by designing various network mechanisms. Early application prototypes included streaming video, serverless chatroom, peer-to-peer file sharing, multiplayer games, web browser support, and vehicular networking. Developed network mechanisms include routing protocols, forwarding strategies, fast and scalable name lookups, DDoS mitigation, and privacy preservation via name obfuscation. We developed several approaches to evaluation of research ideas and artifacts: a global overlay testbed, a simulator, and a substantial open source software base including core forwarding and routing implementations (completed during NP phase), supporting libraries, and applications.

Our application-driven focus proved successful. In the follow-on NDN-NP project we chose three network environments that required critical thinking about key components of the architecture: mobile health, building automation and management, and multimedia conferencing. Interest from physicists and climate scientists in applying NDN to some of their persistent data management problems motivated a fourth network environment. This phase of the project enabled further evolution of the architecture itself – including packet formats, protocol extensions, trust management models, and routing and forwarding algorithms – in tandem with development and refinement of applications that demonstrated its utility. In parallel, interest from the academic, commercial, and government research communities grew rapidly, so much that in 2014 we augmented our annual project retreats with international community meetings to support expanding collaborations [14, 8]. The same year, ACM SIGCOMM established the Information-Centric Network (ICN) conference series to support broader interest in NDN and other ICN-related research efforts. We also launched the NDN Consortium[2] as a forum for broader discussion of the research and its implications, including responding to industry priorities and concerns.

We made tremendous progress in the last five years, but unexpected collaborations have revealed the importance of demonstrating NDN capabilities in IoT and big data environments, and highlighted needs for accessible software platform support and emulation capabilities to facilitate R&D on both the NDN architecture and applications that leverage it. We request supplement funding to complete four tasks: 1) completing and disseminating native NDN applications and associated design patterns, 2) demonstrating NDN scala-

[1]The NDN & NDN-NP proposals and annual progress reports can be found at http://named-data.net/publications/techreports/tr001ndn-proj/, http://named-data.net/publications/techreports/ndn-np_proposal/, and http://named-data.net/project/annual-progress-summaries/, respectively.

[2]http://named-data.net/consortium/

bility; 3) documenting and releasing reference implementations, and 4) documenting NDN design decisions and lessons learned.

2 Current Status of the Project

This section summarizes the current status of the project. It also highlights emerging opportunities and collaborations that shifted our priorities in pursuit of broader impacts.

2.1 Network Environments

During the FIA-NP program, we drove our research agenda via three strategic network environments: mobile health; building automation and management, and multimedia conferencing. Interest from physicists and climate scientists in applying NDN to some of their persistent data management problems motivated a fourth network environment aiming to support scientific "big data".

Open mHealth. Inspired by how Cornell's Open mHealth project uses named data for interoperability in the current Internet, we are developing a mobile fitness app (NDNFit) to drive research within the broader domain of consumer health and wellness. NDNFit has distributed capture, storage, processing, and visualization components. We designed its application namespaces to simultaneously describe data, support forwarding, express trust and access control, and conserve storage. This work generated new collaborations including one with the Univ. of Basel, who is integrating their Named Function Networking (NFN) [20] as a framework for distributed processing.

Enterprise Building Automation & Management System (EBAMS). We used the EBAMS environment to demonstrate the value of the NDN architecture to improve security and deployment in traditional industrial control systems as well as emerging cyberphysical systems. This environment demonstrated NDN's support for distributed data processing and aggregation and provided a second driver for trust management and access control designs. More importantly, the integrated security capabilities that NDN provides to traditional industrial control are inherently aligned with the needs of the exploding area of the Internet of Things (IoT). This environment thus inspired our effort, not planned in the original proposal, to port the NDN platform to constrained devices [26]. This effort attracted interest and support from Qualcomm and Huawei to focus on IoT.

Mobile Multimedia. To understand the requirements of mobile multimedia over NDN, we developed a functional real-time video and audience conferencing solution, NDN-RTC [12], that incorporates adaptive rate control co-developed with Panasonic Research. This effort drove development of latency-sensitive, real-time applications over NDN [13], which illustrated the need for congestion control functionality now being pursued collaboratively with Cisco Research Paris (§3.2.2). Delivering a usable application required unanticipated design and implementation effort (e.g., scalable integration of a contemporary video codec and complete audio pipeline) but has informed design of the architecture as well other apps relying on low-latency communication.

Scientific Big Data. In coordination with effort funded under an NSF-CCNIE proposal, we used NDN technology developed under the NDN and NDN-NP projects to facilitate data management in climate and high-energy physics (HEP) applications. This unforeseen project constituted our most exciting real-world demonstration thus far of NDN's utility to the NSF-funded scientific research community. One NDN site (CSU), deployed an 8-node 10GB testbed across ESnet sites around the world, and collaborated with scientists to use NDN to implement a robust and resilient data catalog that allows publishing, search and retrieval of $\sim$100K (as of January 2016) scientific datasets. This deployment showed how NDN's flexible multipath, multicast data retrieval and request/response model enables robust performance optimization, how its hierarchically structured namespaces incentivize common, domain-wide naming systems, and how its security model and mechanisms simplify the persistent challenge of data integrity and provenance in big data domains.

2.2 Architecture

The FIA-NP program's vision to focus architecture research on specific use cases, such as those described above, provided positive framing and momentum for NDN evolution in the last two years. We made fundamental advances in the areas of architectural prototyping, security, routing, forwarding, and data synchronization, and created a functional prototype of an NDN network with a set of applications. We also developed technologies to enable experimentation and evaluation of the architecture.

Prototyping. The NDN architecture is embodied in a set of packet formats and protocols extended via conventions for standard naming and data types, all supported by a reference implementation that evolves with our understanding of NDN's use in applications. In 2014, we released an open source NDN forwarder (NFD) implementation that reflects our understanding of the NDN architecture [24, 6]. Since then NFD has gone through seven more releases. NFD is the centerpiece of our commitment to making NDN's core technology free and open, to facilitate research, development, and use. The NDN testbed runs NFD and NDN routing protocols natively. Though still evolving, the packet format, NFD codebase, and global testbed provide a solid foundation for research experimentation with NDN.

- *NDN Packet Format.* We developed a standard NDN packet format and completed extensive documentation [22], as well as articulating fundamental design principles and requirements [5, 21].
- *Packet Format Extensions.* Application requirements have driven extensions of the NDN packet format to support 1) the LINK object, which supports routing scalability [7] and enables applications such as NDNFit to name data independently of the devices that capture, store, and process it; and 2) the Negative Acknowledgement (NACK) object, used to enhance NDN's intelligent forwarding plane [29].[3]
- *Link Adaptation Protocol.* Supporting real applications on a functional testbed motivated us to develop the second version of the NDN Link Adaptation Protocol to match the current NDN packet format, support network-generated NACK, and implement hop-by-hop packet fragmentation and reassembly.
- *Multiparty Sync Protocols.* To support our own and other NDN development efforts, we explored multi-party synchronization based on set reconciliation techniques. Sync offers a disruption-tolerant information dissemination approach that is better suited to modern network environments than the point-to-point semantics of typical TCP/IP protocols.[4] In addition to a log-based implementation [36], we also developed a two-level invertible Bloom filter structure to support efficient exchange of set differences [11], and PartialSync [3] which uses Bloom filters to synchronize *subsets* of data in specific scenarios.

Security. During the NP period, we developed effective data-centric security solutions that avoid the brittleness of IP Internet's channel- and perimeter-based security for modern network environments. The NDN architecture mandates digital signatures on all data packets, but each application must decide which keys to trust and how to handle access control. Building several driver applications simultaneously, each with security from the ground up, helped us identify systematic approaches to trust management and confidentiality, which we codified in a set of approaches for re-use by application developers.

- *Schematized Trust.* In NDN, trust decisions can use the structure of names to schematize verification on a packet-by-packet basis. We developed *schematized trust* [30] to 1) formally specify trust relationships based on NDN's hierarchical namespaces, 2) automate verification using those relationships, and 3) distribute verification rules across the network. These schemas simplify application development and reduce the opportunity for error. The trust requirements of our network environments motivated development of these schemas, which we are incorporating into each driver application using a regular-expression-based policy language. Although we did not originally anticipate trust schemas for the NP scope of work, developing and integrating them into pilot applications has constituted a major advance in security for NDN.
- *Name-based Access Control.* We intentionally selected network environments to motivate work on confidentiality. As with schematized trust, our approach to access control both leverages data and key naming and decouples data confidentiality from hosts, connections, and sessions. The outcome of this effort is *Name-based Access Control* (NAC) [31], which we applied first in the NDNFit driver application and second in EBAMS. The granular access requirements of these driver applications motivated initial design choices. NAC's design distinguishes among data owners that specify access control policy, producers that generate encrypted data following the owner's instructions, and authorized consumers that access the data via decryption. NAC leverages NDN's hierarchical namespaces to describe relationships between keys, data, and security principals and automates the distribution

[3] One role to be played by network-generated NACKs is in support of congestion control.

[4] For example, EBAMS can use sync to efficiently enumerate the names of devices and data present on a large network, and NDNFit can employ it for multiparty publishing into the same health namespaces.

of decryption keys.

- *Certificate Format.* Certificates are critical data objects in practical NDN systems used by both trust management and access control. Over the past two years, we evaluated the old NDN certificate design and developed new solutions to minimize processing overhead and facilitate certificate distribution through naming conventions. This new understanding yielded a new NDN certificate format. We also developed an NDN certificate issuing and publishing system for the NDN testbed and NDNFit network environment. Managing testbed and application certificates helped us articulate open research questions in certificate management, for example, how to automate certificate rollover/revocation, how to support short-lived certificates, and how to handle long-lived data packet with short-lived certificate. The first two questions will be addressed during the supplement effort; the last one is a longer term research topic.

2.3 Forwarding & Routing

In addition to many benefits they draw from local NDN communication, all of our network environments are built on the assumption of globally scalable forwarding and routing of NDN packets. Meeting this fundamental need motivates the research described below.

Forwarding Strategy. NDN's forwarding plane selects among multiple interfaces to forward a packet. This capability brings new opportunities and introduces challenges for designing an intelligent strategy to select the best interface to optimize application performance. We implemented modular strategy support in the NDN forwarding daemon (NFD), enabling researchers to easily develop and experiment with their own strategies. Example strategies[5] have been adjusted based on evaluation within the network environments, for example, to accommodate fast application-side retransmissions used in NDN-RTC.

Forwarding Scalability. We implemented and evaluated a scalable FIB longest name prefix lookup design based on the binary search of hash tables [33]. We implemented the proposed design in software, using Intel's DPDK [1] packet processing framework, and demonstrated 10 Gbps forwarding throughput with one billion synthetic longest name prefix matching rules, each containing up to seven name components. We explored in-network caching design issues and evaluated performance of an NDN repository based on the Redis key-value store [32], showing that existing storage systems and databases, such as Redis, can be employed to implement terabyte-scale repositories [27].

Hyperbolic Forwarding (HF). HF offers a new approach to scalable routing that does not exchange routing updates upon connectivity changes. It is greedy geometric forwarding based on hyperbolic coordinates of nodes that encode network geometry. HF's major drawbacks are sub-optimal routes or local minima for some destinations, which NDN's intelligent forwarding plane can mitigate. We compared the performance of HF vs. link-state routing (LSR) over NDN, under various forwarding strategies, failure conditions, and topologies, to determine its viability for inter-domain routing [2]. Our initial results suggest that HF's delay stretch is minimal and the overhead (due to probing alternative paths) is much lower than with LSR.

Named-Data Link State Routing (NLSR) Protocol. To provide practical support to the NDN testbed and an opportunity to evaluate routing *in situ*, we developed NLSR. The first version was deployed in August 2014, and the latest version in January 2016 [16, 4]. NLSR is a native NDN application that supports name-based multi-path forwarding and secures routing information using schematized trust. It enables us to evaluate both link-state and hyperbolic forwarding algorithms by disseminating hyperbolic coordinates in link-state announcements. Our evaluation motivated important design revisions including runtime advertisement and withdrawal of name prefixes, and on-demand publishing and retrieval of NLSR's Link-State Database.

Mobility support. NDNFit, NDN-RTC, and IoT applications all demand support for consumer mobility, which NDN provides natively, and publisher mobility, which remains an active area of our research. We designed a Secure Namespace Mapping mechanism that uses LINK object and Forwarding Hints to support mobility and to scale global routing [7], which is now supported by NFD. We also evaluated different design options for producer mobility support, identifying the most promising approaches [35].

[5]Strategies developed during the NP phase include best-route, interest multicast, ncc, access gateway, and context-specific approaches, such as the adaptive SRTT-based strategy to support hyperbolic forwarding [2] and a collaboration with Dan Pei of Tsinghua University on a strategy for video broadcast over WiFi.

2.4 Evaluation Toolset

Pursuing NDN architectural development, along with performing basic research described above, motivated us to develop a robust set of platforms and tools to support NDN development, experimentation, and evaluation at scale.

- *NDN Platform: Forwarder and Libraries.* The team's software platform supported development of the architectural research results described above. We ported NFD to multiple operating systems,[6] many motivated by the unanticipated research needs of ourselves and our collaborators. We developed libraries that provide cross-language support in C++, Java, Javascript, and Python. These libraries include experimental support for current research in schematized trust, access control, and sync.
- *Testbed.* Our prototype applications all connect to the global NDN testbed, which runs the NDN Platform, including NFD and NLSR routing among 30+ sites across three continents, and is operated and managed by the NDN team at Washington University. The testbed has been used for various NDN development and application experimentations [23]. Recently we also enhanced testbed monitoring tools [15] and converted them to native NDN communications. For example, the NDN map tool[7] shows the real-time usage of the testbed collected via NDN from client daemons on each node.
- *Emulation.* Because Emulab lacks the availability and scale needed for our routing evaluations, as another unanticipated task we developed a light-weight emulation-based tool called Mini-NDN [19] for moderate-scale experimentation and automatic integration testing of code. This has in turn supported the building automation environment by enabling it to roll-out a multi-node test without separate hardware for each node. Mini-NDN uses the NDN Platform to emulate a configurable NDN network on a single system, including NLSR, NFD, library support, and Python-based experiment orchestration.
- *Simulation.* To enable easy-to-use, flexible, and large-scale experimentation with the NDN architecture beyond the above means, we developed, and continue to support and extend, the NDN simulation framework ndnSIM [18]. ndnSIM version 2,[8] released in 2015, directly utilizes the codebase of NFD and its supporting libraries, and enables simulations driven by the source code of real NDN applications. This extension significantly improves fidelity of simulation results and allowed simulation experimentation by directly transferring code from emulation and testbed environments. ndnSIM has gained wide adoption; as of February 2016, over 230 research projects worldwide have used ndnSIM.

3 Proposed Work for the Supplement

NDN is a large-scale, collaborative endeavor. Our 2nd NDN community meeting in September 2015 was attended by over 100 participants from 57 institutions across 13 countries. NDN's growth requires engagement on four fronts: 1) addressing technical research problems through new research proposals[9]; 2) industry partnerships such as the NDN consortium, with a planned proposal to NSF's I/UCRC program; 3) mission-specific development for agencies such as DARPA, DOE, and NIST; and 4) collaborations with proven-sustainable projects in the open source development community. We propose four tasks for the supplement period that will advance our position on these fronts. First, we will **complete and demonstrate native NDN applications** that showcase usable security features such as NDN certificate management, schematized trust, and access control. Second, we will **complete and document our open source reference implementations** to facilitate broader impact via community involvement and uptake. Third, we will **demonstrate and quantify the scalability of NDN**, validating hyperbolic forwarding work through experiments deployed on the NDN testbed and using cloud resources. Finally, we will **document our design decisions and lessons learned** over the years, to engage the broader community in design discussions on NDN as a future Internet architecture. Each of these elements is fundamental to fulfilling the FIA-NP program goal of transitioning NSF's future Internet architecture research investments toward practice. Our expected progress was delayed due to fruitful but unforeseen new collaborations described in the previous sections, unanticipated dependencies in implementation timelines, and shifts in focus (e.g., toward IoT) motivated by interaction with the research community and industry. We organize our proposed tasks around these goals.

[6]Desktops: Ubuntu (12.04, 14.04, 15.10), OS X (10.9, 10.10, 10.11), other Linux and BSD flavors (Fedora, CentOS, FreeBSD), and MS Windows; home routers: OpenWRT, DD-WRT; Mobile and Embedded: Android (released to GooglePlay store), Raspberry Pi.

[7]`http://ndnmap.arl.wustl.edu`

[8]`http://ndnsim.net/2.1/`

[9]Examples include trust anchor bootstrapping, cache poisoning defenses, scalable forwarding, secure and robust mobility support.

3.1 Disseminating NDN-Native applications and design patterns

With supplement funds, we will finalize our designs and pilot implementations, and then disseminate our results on how to design applications that leverage NDN's capabilities.

3.1.1 Applications

Open mHealth. Our overarching task for the Open mHealth network environment is to finish and document the NDNFit driver application, which includes a mobile capture app, data storage, processing (via NFN), and visualization components, as well as certificate issuing support, all communicating via NDN and employing/validating our current approaches to 1) schematized trust, 2) name-based access control, and 3) LINKs as forwarding hints enabling namespace independence from device or service providers. We will publish a technical report and implementation and test all components with end users. This process will also drive work in NDN autoconfiguration (§3.2.1).

Enterprise Building Automation and Management. Our initially successful integration with UCLA campus building systems hit an unexpected snag when they upgraded their system in late 2015. With supplement support we will re-connect the campus data to our Mini-EBAMS testbed, and complete the design, implementation, demonstration, and documentation of approaches to trust, aggregation, access control, and visualization that leverage hierarchical naming of real UCLA building sensors and their data. We will also demonstrate system integration of IoT devices that use native NDN communication [26].

Mobile Multimedia. Our primary task will be to finalize NDN-RTC as the NDN team's primary videoconferencing tool, a long-standing project goal that will drive autoconfiguration and congestion control research (§3.2.1 and 3.2.2). We will apply our experience with NDN-RTC to sketch a solution to just-in-time selection and navigation of multidimensional content, which is uniquely well-supported by NDN and a building block of next generation in mobile augmented reality applications–a major interest of collaborators and partners.

Scientific Big Data. We propose two tasks in this area: 1) Investigate the NDN-based data catalog's applicability to existing software widely used in climate and physics communities to distribute scientific data, such as ESGF and xrootd; and 2) Investigate integration of the catalog with layer 2 reservation systems such as OSCARS and potentially GENI. Since NDN can run directly over layer 2, integrating NDN with these reservation mechanisms can potentially leverage NDN routing and forwarding strategies.

Named Data Link State Routing (NLSR). NLSR development yielded not only a routing protocol running natively over NDN, but also a pilot application that revealed insights into NDN naming design, verified the utility of our synchronization protocol, and validated our schematized trust solution. We plan to resolve, implement, and document two open issues: 1) Upon starting, NLSR needs mechanisms to discover available physical and logical interfaces; other applications share this issue, which further informs work on autoconfiguration. 2) NLSR uses sync (§2.2) to retrieve routing updates in a naive way, without a clear notion about which neighbors may have the data, resulting in unnecessary broadcast messages. This issue may affect other applications on multihomed devices; resolving it will improve sync usability and performance.

3.1.2 Security

With supplement funding, we will finish testing and release a usable security library and set of security tools that help developers tackle data-oriented security challenges in NDN applications. We will finish the design of specific schematized trust policies and access control approaches, refine and simplify the APIs and schema languages, and identify data-centric security principles derived from experience with the network environments. Our specific tasks include improved automation of 1) certificate management, 2) data signing and verification based on schematized trust, and 3) encryption/decryption using name-based access control. Each of these requires consideration of impact on namespace design, APIs for developers, and how to handle and report error conditions.

3.2 Reference Implementations

Software reference implementations have been critical to our hands-on exploration of the network environments, and to engaging others in exploring NDN. During the supplement period, we will provide reference implementations for two gaps in our architecture, motivated by the requirements of the network environments: autoconfiguration support and congestion control.

3.2.1 Autoconfiguration Support

Experience with our pilot applications has made it clear that painless autoconfiguration is essential for usability, especially for mobile and intermittently connected environments. Autoconfiguration involves: 1) establishing NDN connectivity; 2) obtaining appropriate keys; and 3) producers registering name prefixes to attract interests for their data. In local environments where NDN can run directly over layer-2, nodes can locally broadcast Interest packets to discover neighbor gateways and available data, and figure out where to forward Interests by observing from which direction data packets return. When the NDN stack runs as an overlay on IP, it can perform local NDN gateway discovery in IP multicast-enabled domains or using DNS and DHCP-derived local domains, or use *home agent* gateways based on pre-configured settings. We have implemented these features in various prototype forms.

During the supplement period, we will use widely supported and emerging service discovery technologies (mDNS, DNS-SD) to extending existing automatic prefix registration mechanisms for NDN publishers. We will use naming schemas to infer home agent gateways from configured cryptographic key names. Assuming NDN nodes are configured with keys to publish data, the naming structure of keys can either directly or indirectly reveal the home agent gateway. To bootstrap keys, we will design an automation system similar to Let's Encrypt [17]. Our existing implementation limits the namespaces under which keys can be generated, and requires manual intervention. We will develop a system with automated and modular support for different methods to claim a namespace, called "challenges" in Let's Encrypt.

3.2.2 Congestion Control

To support high-throughput applications (e.g. scientific big data) and low-latency applications (NDN-RTC), we need congestion control in the reference implementation and deployed on the testbed. Unlike TCP, NDN data retrieval happens among consumer(s), producer(s), and router cache(s), with no fixed source or destination. The problem becomes generalized multiparty-to-multiparty dissemination, and congestion control solutions from TCP/IP do not directly apply. NDN's stateful forwarding plane provides a unique opportunity for effective network-layer congestion control [29], especially necessary for the target applications. However, because the network NACK functionality, a necessary ingredient in NDN congestion control, only became available recently, we have not yet completed an implementation.

During the supplement period, we will develop an initial hop-by-hop (instead of end-to-end) solution that essentially avoids congestion by flow control at each hop.[10] All existing work on ICN congestion control [29, 25, 9, 28], including ours, assumes the link capacity is a known constant. This assumption fails for shared links like Ethernet or overlay links, such as tunnels, due to cross traffic. We plan to 1) design initial mechanisms for congestion detection, back-pressure signaling, multipath strategies, and rate adjustment for hop-by-hop rate control that relax these assumptions; 2) simulate and evaluate these mechanisms; 3) implement them in NFD, and deploy them on the NDN testbed, and 4) develop library APIs to provide rate control functionality to applications, and integrate them into the two applications: NDN-RTC, and bulk transfer of scientific data.

3.3 Demonstrating NDN Scalability

To be viable for the proposed network environments and more generally as a future Internet architecture, NDN must provide scalability in forwarding and routing, performance, and application deployment. During the supplement period, we target three areas of completion that are important milestones in the effort to transition NDN to practice.

Forwarding and Routing. Progress toward scalable forwarding has met our objectives for the NP project, and we pursue additional work in that area in separate proposals. We focus our effort in the supplement period on demonstrating hyperbolic forwarding, a promising candidate to scale inter-domain routing in NDN (§2). We have tested HF in Mini-NDN but have not tried it over the real NDN testbed, due to concerns about

[10] The downstream node N of a link controls the Interest-sending rate to the upstream node, so that the returning data rate will be within the link's capacity, and dynamically adjusts to accommodate variation in delays, packet sizes, and cross traffic. If N's outgoing Interest queue grows, N will send excessive Interests to alternative paths. If all of N's outgoing links are full, N will send a back-pressure signal to its downstream. Eventually the back pressure will reach the consumer application, which will reduce the rate of Interest generation. A complete congestion control solution consists of various mechanisms that detect congestion, adjust sending rates, signal back-pressure, and trigger multipath forwarding strategies.

potential disruption to the testbed. Now that the hyperbolic forwarding code is stable, we plan to do real testbed experiments to discover potential issues that may not surface in an emulated environment.

Implementation performance. So far, performance optimization of NDN implementations has been secondary to fundamental architectural research and development of demonstration applications by our team, although it is pursued vigorously by our industry colleagues. Through the work of the NP project, we have identified areas where improved implementation performance is critical to illustrating the value and viability of the architecture to potential users, and we will work towards these in the supplement phase. One is congestion control to support multimedia applications, as described in §3.2.2. The second is to improve bulk data transfer performance for big data applications. In a preliminary evaluation of the performance of the current codebase in bulk data transfers, we transferred large objects at speeds up to 5Gbps over a single hop and 3Gbps over multiple hops. We narrowed the bottleneck to inefficient data copying inside NFD, which we can eliminate with non-trivial software development effort. We will develop effective solutions to speed up NFD and retest it with big data applications.

Application Scalability. Our experience has shown that carefully selected capstone demonstrations of NDN application scalability are critical to sharing progress with the outside world, as well as testing implementation limits. The original FIA project included a 1000-node video playout test [10], which included a comparison with TCP/IP-based solutions, that was an important milestone in discussions with industry. During the supplement period, Washington University will lead the team in completing another milestone demonstration, whose centerpiece will be a wide-area application of the NDN real-time video conferencing application, NDN-RTC, operating over a network of more than 10,000 globally distributed clients and NDN gateways. (This demonstration has been delayed by application development challenges and other testbed roll-out priorities.) NDN-RTC has a significantly more challenging set of requirements due to its low-latency traffic; we are eager to see its performance in practice as it demonstrates the viability of distributed real-time communication, which is very hard to achieve with IP-based approaches. To achieve the target scale, the scenario will make use of the global NDN testbed and thousands of public-cloud server instances, both of which have footprints spanning the globe. Through this effort, the team will undertake performance evaluations of NDN services and applications at this scale. We will document the approaches used to obtain evidence and help others perform similar tests. We plan to develop and monitor the test with input from Cisco Research Paris and Panasonic Research, both are users of NDN-RTC.

Scaling down as well as scaling up. We believe that NDN provides transformative support for constrained devices in challenging communication conditions [26]. Given expanding interest in IoT as discussed previously, we plan to complete foundational work in designing and demonstrating NDN support for constrained IoT devices. This has evolved into a significant area of interest for the NDN team, motivated by potential impact on the field, industry and collaborator interest, and experience with our building management network environment and related work. We have developed other proposals that target software infrastructure development in this area. During the supplement phase, we plan to lay the groundwork for such future efforts by ourselves and our collaborators by designing and providing reference implementations that enable NDN communication over radio protocols like Bluetooth Low Energy, and integrating communication from Arduino-class IoT devices with the building automation and management network environment.

3.4 Documentation

Finally, one of our most important tasks is to complete documentation of the state of NDN research, including both the architecture and its implementation, as the FIA phase closes. This documentation will include articulation of design principles we have identified and applied, design decisions about the NDN packet format; documentation of software components; and taxonomizing open issues in taking NDN to the next stage, for use by funding agencies, other researchers, and ourselves.

An example of necessary design documentation involves two particularly nuanced and under-articulated sets of tradeoffs related to 1) in-network caching, and 2) the use of names for network data delivery. NDN enables network caching that facilitates scalable distribution, but caching introduces new application design issues. We need to describe these issues and effective approaches to enabling real-time applications to fetch the most recent data in the face of network caching. Similarly, NDN breaks new ground by having the network share the same namespace with applications, empowering them to express their own communication patterns at the network delivery level but introducing tighter coupling between applications and the network, with associated implications for namespace design. As part of the supplement period's effort, we plan to carefully describe these constraints and offer guidelines for naming patterns.

For NDN software, our most successful documentation has been the NFD Developer's Guide [6] which explains how to interpret, use, and modify the NFD codebase. During the supplement period, we will follow its model to expand documentation on other important components, particularly NLSR, whose current documentation only offers simple routing configuration information. More comprehensive documentation for NLSR will benefit researchers by 1) describing how NLSR can support basic dynamic routing functionality in non-trivial testbed, emulation or simulation experiments; 2) describing how its software design and source code can serve as a template for others to prototype new routing protocol implementations; and 3) explaining how researchers may develop new algorithms to improve the performance of NLSR. These use cases all require both detailed documentation on configuration and execution and a developer's guide that describes the overall design of the software, so that others can understand and change the code.

References

[1] DPDK: Data Plane Development Kit. http://www.dpdk.org.

[2] Hyperbolic vs. Link-State Routing in ICN. Online: https://www.caida.org/workshops/ndn/1509/slides/ndncomm2015_vlehman.pdf, 2015.

[3] PartialSync: Synchronizing a Partial Namespace in NDN. Online: https://www.caida.org/workshops/ndn/1509/slides/ndncomm2015_mzhang.pdf, 2015. The implementation code can be found at https://netwisdom.cs.memphis.edu/gitlab/mzhang4/partial_sync.

[4] Named-Data Link State Routing (NLSR) Protocol. Online: http://named-data.net/doc/NLSR/0.2.2/, 2016.

[5] Alexander Afanasyev, Ravi Ravindran, GQ Wang, Lan Wang, Beichuan Zhang, and Lixia Zhang. ICN Packet Format Design Requirements. Internet-draft, ICN Research Group, September 2015.

[6] Alexander Afanasyev, Junxiao Shi, et al. NFD Developer's Guide. Tech. Rep. NDN-0021, NDN, 2015.

[7] Alexander Afanasyev, Cheng Yi, Lan Wang, Beichuan Zhang, and Lixia Zhang. SNAMP: Secure namespace mapping to scale NDN forwarding. In *Proc. of IEEE Global Internet Symposium*, April 2015.

[8] Alexander Afanasyev, Yingdi Yu, Lixia Zhang, Jeff Burke, kc claffy, and Joshua Polterock. The second named data networking community meeting (NDNcomm 2015). *ACM SIGCOMM Computer Communication Review (In submission)*, 2016.

[9] Giovanna Carofiglio, Massimo Gallo, and Luca Muscariello. Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks. In *Proceedings of ACM SIGCOMM ICN Workshop*, 2012.

[10] Patrick Crowley. Named data networking. In *China-America Frontiers of Engineering Symposium*, Frontiers of Engineering, 2013.

[11] Wenliang Fu, Hila Ben Abraham, and Patrick Crowley. Synchronizing Namespaces with Invertible Bloom Filters. In *Proceedings of the Eleventh ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ANCS '15, 2015.

[12] Peter Gusev and Jeff Burke. NDN-RTC: Real-time videoconferencing over Named Data Networking. In *Proceedings of the 2nd International Conference on Information-Centric Networking*, pages 117–126. ACM, 2015.

[13] Peter Gusev, Zhehao Wang, Jeff Burke, Lixia Zhang, Takahiro Yoneda, Ryota Ohnishi, and Eiichi Muramoto. A Real-Time Streaming Data Delivery over Named Data Networking. In *IEICE Transactions on Communications*, May 2016.

[14] kc claffy, Josh Polterock, Alexander Afanasyev, Jeff Burke, and Lixia Zhang. The First Named Data Networking Community Meeting (NDNcomm). *ACM SIGCOMM Computer Communication Review (CCR)*, April 2015.

[15] Ze'ev Lailari, Hila Ben Abraham, Ben Aronberg, Jackie Hudepohl, Haowei Yuan, John DeHart, Jyoti Parwatikar, and Patrick Crowley. Experiments with the emulated ndn testbed in onl. In *Proceedings of the 2Nd International Conference on Information-Centric Networking*, ICN '15, 2015.

[16] Vince Lehman, Mahmudul Hoque, Yingdi Yu, Lan Wang, Beichuan Zhang, and Lixia Zhang. A Secure Link State Routing Protocol for NDN. Technical Report NDN-0037, NDN, January 2016.

[17] Let's Encrypt. https://letsencrypt.org/.

[18] Spyridon Mastorakis, Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. ndnSIM 2.0: A new version of the NDN simulator for NS-3. Tech. Rep. NDN-0028, NDN, 2015.

[19] Mini-NDN GitHub. `https://github.com/named-data/mini-ndn`.

[20] Named Functions. `http://www.named-function.net/`.

[21] NDN Protocol Design Principles. `http://named-data.net/project/ndn-design-principles/`.

[22] NDN Project. NDN Packet Format Specification. Online: `http://named-data.net/doc/ndn-tlv/`.

[23] Experimentations Using the NDN Testbed. `hhttp://named-data.net/ndn-testbed/experimentations/`.

[24] NFD - Named Data Networking Forwarding Daemon. `http://named-data.net/doc/NFD`.

[25] N. Rozhnova and S. Fdida. An effective hop-by-hop interest shaping mechanism for ccn communications. In *Proceedings of IEEE INFOCOMM NOMEN Workshop*, 2012.

[26] Wentao Shang, Adeola Bannis, Teng Liang, Zhehao Wang, Yingdi Yu, Alexander Afanasyev, Jeff Thompson, Jeff Burke, Beichuan Zhang, and Lixia Zhang. Named Data Networking of Things. In *Proceedings of the 1st IEEE International Conference on Internet-of-Things Design and Implementation*, April 2015.

[27] Tian Song, Haowei Yuan, Patrick Crowley, and Beichuan Zhang. Scalable Name-Based Packet Forwarding: From Millions to Billions. In *Proceedings of the 2nd International Conference on Information-Centric Networking*, ICN '15, 2015.

[28] Yaogong Wang, Natalya Rozhnova, Ashok Narayanan, David Oran, and Injong Rhee. An improved hop-by-hop interest shaper for congestion control in named data networking. In *Proceedings of ACM SIGCOMM ICN Workshop*, 2013.

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

[30] Yingdi Yu, Alexander Afanasyev, David Clark, kc claffy, Van Jacobson, and Lixia Zhang. Schematizing and automating trust in Named Data Networking. In *Proceedings of 2nd ACM Conference on Information-Centric Networking*, September 2015.

[31] Yingdi Yu, Alexander Afanasyev, and Lixia Zhang. Name-Based Access Control. Technical Report NDN-0034, NDN, October 2015.

[32] Haowei Yuan. Scalable NDN Forwarding. Doctoral Thesis. Washington University in St. Louis, Department of Computer Science & Engineering, December 2015.

[33] Haowei Yuan and Patrick Crowley. Reliably Scalable Name Prefix Lookup. In *Proceedings of the Eleventh ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ANCS '15, 2015.

[34] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, et al. Named data networking. *Comp. Comm. Review*, 2014.

[35] Yu Zhang, Alexander Afanasyev, Jeff Burke, and Lixia Zhang. Supporting Mobility in Named Data Networking. In *Proceedings of IEEE INFOCOM 2016 Workshop on Named-Oriented Mobility: Architecture, Algorithms and Applications*, April 2016.

[36] Zhenkai Zhu and Alexander Afanasyev. Let's ChronoSync: Decentralized dataset state synchronization in Named Data Networking. In *ICNP*, pages 1–10, 2013.