# The First Named Data Networking Community Meeting (NDNcomm)

kc claffy
Joshua Polterock
CAIDA/UCSD
kc,josh@caida.org

Alexander Afanasyev
Jeff Burke, Lixia Zhang
UCLA
afanasev,lixia@cs.ucla.edu,
jburke@remap.ucla.edu

This article is an editorial note submitted to CCR. It has NOT been peer reviewed. The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

## ABSTRACT

This report is a brief summary of the first NDN Community Meeting held at UCLA in Los Angeles, California on September 4-5, 2014. The meeting provided a platform for the attendees from 39 institutions across seven countries to exchange their recent NDN research and development results, to debate existing and proposed functionality in security support, and to provide feedback into the NDN architecture design evolution.

## Categories and Subject Descriptors

C.2.5 [**Local and Wide-Area Networks**]: Internet; C.2.1 [**Network Architecture and Design**]: Packet-switching networks

## Keywords

information-centric networking, named data networking, architecture

## 1. INTRODUCTION

On September 4-5, 2014, the Named Data Networking (NDN) project team held the first NDN Community Meeting [4] hosted at UCLA in Los Angeles, CA. The meeting was attended by 87 participants from 39 institutions across 7 countries, and streamed live using both TCP/IP- and NDN-based video streaming. About 40% of the attendees made presentations to discuss the current state and future directions of the NDN architecture, software platform, supporting libraries, applications, and testbed. Presentations showcased recent research results inside and outside of the NDN project team, including examples of educational use and tutorial materials. The NDNcomm meeting series is part of the project team's effort to build a community around NDN research. This report summarizes discussions of the meeting [4].

The NDN project strives to use current and future applications to drive the development and deployment of the architecture and its supporting modules, to test prototype implementations, and to encourage an iterative cycle of design, real-world experimentation, and evaluation. In this spirit, the NDNcomm program committee encouraged contributions that considered specific case studies, application requirements, and real-world scenarios. The meeting also offered participants opportunities to debate existing and proposed functionality to support security and privacy at different layers of the architecture. Discussions topics included suggested evolution of the NDN architecture based on implementation or deployment experiences, and implementation artifacts including APIs.

At this meeting the NDN team also introduced the newly formed Named Data Networking Consortium [2]. The NDN Consortium is coordination framework intended to promote a vibrant open source ecosystem of NDN research and experimentation, continued openness of the core NDN architecture, access to research results, collaboration on architecture development.

## 2. HISTORY OF THE NDN PROJECT

The Named Data Networking (NDN) project formally started in 2010 as one of the U.S. National Science Foundation's Future Internet Architecture (FIA) projects [21, 20, 3], although it has its roots in an earlier idea, Content-Centric Networking (CCN), which Van Jacobson first publicly presented in 2006.[1] **Van Jacobson** (UCLA/Google) introduced the meeting by reviewing the motivation behind the NDN architecture: we have moved from phone calls to publishing and accessing information in new flexible ways. Today it is information that matters, not the plumbing (machines and links), and the NDN project embeds this concept into the basic (NDN) architecture. Specifically, NDN retains the hourglass shape of Internet's protocol architecture but evolves the thin waist to enable creation of completely general distribution networks. The core element of this evolution is removing the restriction that packets can only name communication endpoints. As far as the network is concerned, a name in an NDN packet can name anything—an endpoint, a data chunk in a movie or a book, a command to turn on some lights, etc. This conceptually simple change allows NDN networks to use almost all of the Internet's well-tested engineering properties to solve not only communication problems but also digital distribution and control problems.

The NDN project has followed an application-driven approach to architecture development, filling in architectural details while solving real problems, which serve to verify and validate the architectural direction. Developing applications over NDN requires a fundamentally new way of thinking about networking, compared to using the TCP/IP protocol stack. The NDN architecture naturally supports new *application patterns* that fit many modern communication scenarios. For example, NDN applications can easily implement multi-party communication by using in-network storage (ContentStore and Repo [6]) and distributed data synchronization protocols such as ChronoSync [22].

[1] "A New Way to Look at Networking",
`https://www.youtube.com/watch?v=oCZMoY3q2uM`

# 3. NDN PROJECT STATUS: SOFTWARE AND TESTBED PLATFORMS

In spring 2014, NSF funded the next phase of NDN (NDN-NP), which included support for development of a flexible and extensive software infrastructure and tool chain to support further exploration. Before the NDN project started, the CCN team at PARC did not know exactly what information should be carried in packets, so they created the most extensible framework using the methodology of the day: binary XML [1]. It became clear later that some of the original specification was inefficient, suboptimally implemented, or entirely unnecessary. For example, the binary XML format requires parsing of entire packets, which made forwarding optimizations practically impossible. As another example, Interest and Data scoping (limiting propagation of packets) is better handled using names, e.g., a "/localhost" prefix, instead of an opaque number buried in the packet. The NDN team fixed these weaknesses using a new Type-Length-Value (TLV) packet format [13]. The discovery and replacement of cumbersome components continues as we find better solutions to support critical functions. One such function is distributed name discovery, which currently relies on *selectors* in Interest packets, which is not obviously the best way to do it.

**Beichuan Zhang** (U. Arizona) gave an overview of the new NDN forwarding daemon (NFD) [15]. Until 2012, the NDN team had relied on a codebase developed by PARC to build applications and a testbed, complemented by an NDN simulator [8], to demonstrate the architecture's ability to solve important problems, including building automation systems and scalable video distribution. As NDN research advanced, researchers found that the PARC codebase was impeding progress, due to its complexity and rigidity that prevented modifications for experimentation. For example, the tight coupling of the codebase to the initial packet format made it impossible to introduce the new NDN-TLV packet format [13]. The monolithic nature of the codebase also prohibited real-world evaluation of Interest forwarding strategies [19, 7]. To meet the growing needs of the NSF-funded research efforts, the NDN team spent most of 2014 developing, testing, and deploying a new NDN forwarder [9, 15] that is more modular, extensible, and understandable, and allows easy experimentation with new protocol features. In addition to supporting the transition from XML binary to the new TLV packet format, NFD will enable research on NDN forwarding strategies, link-layer protocols, and caching policies and algorithms. NFD was released in August 2014 as part of the NDN platform [14] (which Jeff presented next). In addition to the code and documentation, NFD features an extensive developer's guide [9] which explains code structure, features, and hooks.

NFD packet processing has two dimensions: *forwarding pipelines* and *forwarding strategies*. Pipelines are common operations performed on Interest and Data packets at different stages of processing, and strategies are decision engines for Interest forwarding. When an instance of NFD receives a new Interest $I_{new}$, it will push $I_{new}$ into the *Incoming Interest* pipeline, which performs loop detection and consults the router's *Pending Interest Table (PIT)*, *Content Store (CS)*, and *Forwarding Information Base (FIB)*. Depending on the lookup result, NFD may invoke the *Outgoing Data* pipeline after a CS match, or forward $I_{new}$ to the forwarding strategy module for processing. The logical placement of the strategy between the two forwarding pipelines makes it the core of NDN Interest forwarding. NFD by design allows different strategies to apply to different namespaces, supporting a wide range of experimentation with Interest forwarding, yet minimizing the overhead of basic NDN packet processing.

**Jeff Burke** (UCLA REMAP) provided an overview of the NDN Platform [14], which gathers publicly supported packages of critical components developed by the NDN project for building and testing NDN applications and networks. The NDN platform includes: (1) a set of libraries that support NFD development and facilitate development of NDN applications in C++, Python, JavaScript, and Java [5, 17]; (2) an updated Named-data Link-State Routing protocol (NLSR [16]) that works with the new packet format, NFD, and uses ChronoSync [22] to synchronize the routing database; (3) a new modular implementation of an NDN data repository [6]; and (4) several NDN network management applications. To promote and encourage community development, all code in the NDN Platform is released under open-source LPGL (for libraries) and GPL licenses and is available in our GitHub repository[2]. The NDN team also embraced many other components of open source development: public issue and bug report tracking using Redmine[3], code review with Gerrit[4], coupled with continuous integration (Jenkins and Travis-CI), significant documentation for released components, and public mailing lists. Currently, the NDN Platform officially supports Ubuntu Linux and OS X operating systems, and is known to work on other desktop (CentOS, Fedora, FreeBSD) and embedded systems (Raspberry Pi, OpenWRT).

**John DeHart** (WUSTL) presented an update on the growing NDN Testbed: a shared resource and vehicle to explore NDN research ideas. The testbed consists of software-based routers, application host nodes, and other devices at NDN project institutions; As of November 2014, the testbed spans 21 sites across the globe (12 in US, 4 in Asia, and 5 in Europe) and is open for external sites (beyond the NSF-funded institutions) to participate.[5]

The NDN architecture has a fundamental security building block: each Data packet is cryptographically signed. Otherwise, NDN allows applications to develop their own trust models. The testbed has recently prototyped a hierarchical trust model [10], which is used by a number of applications. **Alex Afanasyev** (UCLA) introduced the automated NDN PKI system[6] created to support namespace assignment and public key certification of NDN testbed users. The web-based system assigns users sub-namespaces under "/ndn" based on their email addresses (e.g., if a user owns user@testbed.site email, the system assigns "/ndn/site/testbed/user" namespace) and provides step-by-step instructions on how to generate keys and install approved certificates.

# 4. NDN INFRASTRUCTURE RESEARCH

**Beichuan Zhang** (U. Arizona) and **Lan Wang** (U. Memphis) described the current efforts in NDN routing research and development. NDN's intelligent data plane relaxes the fast convergence requirement on routing protocols, allowing them to focus on disseminating long-term topology and policy information, which promotes scalable performance and enables routing schemes that are difficult for IP to handle. Lan Wang introduced the routing protocol currently used on the testbed: Named-data Link State Routing Protocol (NLSR). NDN naming allows entities in the routing system to associate with each other; router and key names can reflect trust relationships among routers. NLSR is implemented in C++, using the ndn-cxx library [5]. NLSR version 0.1.0 [16] was released in August 2014. It supports both link state and hyperbolic

[2] `http://github.com/named-data`

[3] `http://redmine.named-data.net`

[4] `http://gerrit.named-data.net`

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

[6] `http://ndncert.named-data.net/`

routing, and uses ChronoSync [22] to maintain decentralized state synchronization.

Research in routing and forwarding will benefit from the flexibility of NFD. One challenge of network architecture research is the continual tradeoff among efficiency, comprehensibility, fungibility, and extensibility. Maintaining and evolving a software platform enables realistic exploration and experimentation with these tradeoffs. For example, NFD's strategy module allows research on strategies for hyperbolic routing or for ad hoc (e.g., vehicular) environments with varying media (Bluetooth, WiFi, etc).

**Junxiao Shi** (U. Arizona) described the use of NDN in local area networks (e.g. home/office networks, data centers) to locate requested content, including policy support for separation of traffic on different links, and self-learning strategies to adapt to path availability and link quality in LANs. His group is also working to get Hadoop running natively on an NDN LAN.

**Giovanna Carofiglio** (Alcatel-Lucent, now Cisco) presented an analytical model for pending interest table (PIT) dimensioning in NDN. Her model assumed Poisson Interest arrivals, and responding flows multiplexed over 100Mbps or 1Gbps links, based on traffic traces from Orange networks. The model output the expected rate of congestion at the receiver. Her main result was that all PIT tables prior to a bottleneck become empty, and all PITs after the bottleneck are characterized by delayed differential equations. She used this formula to help determine the dimensions of the PIT.

**Haowei Yuan** (WUSTL) presented his work with Patrick Crowley to scale high-performance PIT design, given the requirement to update the PIT for each packet. Their design uses optimized hash tables that store only fingerprints in core router PITs rather than full names, to minimize computational cost of PIT updates and lookups. Edge routers, which have lower throughput requirements, will still hold the full names of objects, and support full prefix matching.

**G.Q. Wang** (Huawei Technologies) led a lively discussion of business issues for large-scale ICN deployment for carriers, including the need for data packets large enough (5GB) to support high-end video (e.g., 8K UHDTV).

**Christian Tschudin** (U. Basel) gave a brief lightning talk on *named function networking* (NFN), which he has explored as a natural extension to NDN. NFN supports the idea that people usually want cooked rather than raw data, e.g., video at a specific transcoding, query to a database, or a summary of heart rate monitor data. Clients name the desired computation results, server-agnostically. NFN could then orchestrate the computation by (1) locating data, function, and resources, (2) triggering the execution, and (3) collecting the results. Since NDN shields data locations from users by design, one could leverage this feature by putting NFN over the NDN platform and providing an API. This talk inspired a long discussion on the tradeoffs of embedding such functionality into the architecture, and more generally what level of abstraction is appropriate for a given mechanism. There was concern about allowing long lifetimes of Interests (due to computation delay), which could lead to a storage attack on the network.

# 5. NDN APPLICATION RESEARCH

Participants exchanged experiences and lessons learned from developing NDN-based applications, including two efforts in the area of the Internet of Things (IoT). **Adeola Bannis** (UCLA REMAP) presented her work on developing an NDN IoT software toolkit running on the Raspberry Pi platform to support experimentation in sensing and monitoring.[7]. **Wentao Shang** (UCLA) reported his work on an NDN Sensor Network Emulator that started during his summer internship with Cisco. The emulator enables exploration of NDN features in sensor networks, e.g., name discovery, routing/forwarding and security, without having to build a real network.

**Yong-Jin Park** (Waseda University) introduced a recent Japan-EU collaboration (GreenICN)[8], the goal of which is to enable networks and end devices to operate in a highly scalable, resilient, and energy-efficient way by exploiting the advantages of information centric networking (ICN) architectures.

**Peter Gusev** (UCLA REMAP) presented the NDN-based real-time conferencing application *ndnrtc*. He showed an instance of *ndnrtc* with data producers at UCLA and UCSD, with consumers pulling from both, and simulated a link break to see traffic flow through alternate paths. The current software is a C++ prototype that uses the NDN-CPP and WebRTC libraries, and provides interfaces for publishing media streams and fetching (MacOS only right now). Soon *ndnrtc* will support chat and conference discovery.

**Christos Papadopolous** (Colorado State University (CSU)) presented an overview of his project using NDN support for climate applications, funded by NSF's Campus Cyberinfrastructure program. Climate applications present an ideal case for NDN research, illustrating many core architecture challenges: content discovery, retrieval, replication, synchronization, versioning, and security. There are many independent implementations of climate applications, but no consistent naming structure among them, which poses a major impediment to data sharing. The climate community recognized the importance of a structured namespace and started moving to CMIP5 data reference syntax, which shares several commonalities with NDN naming. The CSU project has developed tools to enable naming climate data in a standard, structured, hierarchical fashion to support scalable and efficient content discovery and retrieval over NDN. The positive interactions with climate scientists have revealed many opportunities for other big data disciplines.

**Eric Osterweil** (Verisign Labs) presented some early stage thinking of how NDN may facilitate managing and mining very large, diverse, and distributed data sets, which are often today compartmentalized into silos prevented from mutual visibility for security reasons. He considered how NDN could support a big data computation environment without requiring a central storage (i.e., leave the data where it is), which will require decoupling data ontology and access, perhaps using named-based scoping.

The NDN NP project includes two specific network environments as drivers, along with several mobile multimedia applications. The two environments – Enterprise Building Automation and Monitoring and Open mHealth – are used to motivate and validate research [3]. **Jeff Burke** (UCLA REMAP) introduced the newest area, Open mHealth over NDN. Open mHealth is a non-profit organization dedicated to leveraging everyday devices for sources of health data.[9] This network environment highlights issues of selective access to personal data flows from mobile publishers, including trust models and anonymization. The team plans to develop or adapt existing applications that monitor physical activity and other information specific to the location or context of the person or activity. The mHealth environment has inspired conversations with researchers focused on values in design of network architectures, including how cultural/social uses of data by the public should drive practical design decisions.

Ilya Moiseenko (UCLA), presented the latest Consumer-Producer API for NDN [12], which provides generic programming interface to NDN communication protocols and architectural modules. Consumer context associates NDN name prefix with consumer-specific

[7] https://github.com/remap/ndn-pi

[8] http://www.greenicn.org/

[9] http://openmhealth.org

data transfer parameters and controls how Interests are expressed and how returning Data packets are processed. Producer context associates NDN name prefix with producer-specific data transfer parameters and controls how data is produced and secured, and how Interests are demultiplexed. Both contexts are extendable to admit functionality of newly developed protocols and modules.

**Ryan Bennett** (Colorado State) presented his work with NDN in Javascript (NDN-js), which he used to build an expanded class library. NDN-Contrib[10] is a modular set of javascript classes for building NDN-enabled applications in Node.js, the browser.

## 6. POSTERS AND DEMONSTRATIONS

The evening session of the first day showcased ten posters and demos (some of which also had lightning talks), and on-site technical support to those wanting to get the recent NDN *nfd* software release and NDN ChronoChat application working on their laptops.

- Alexander Horn (UCLA REMAP), *Applications of the NDN IoT toolkit*
- Giovanni Pau (UPMC/UCLA), *Using GeoFaces to route Interests and Data in Vehicular Networks*
- Golnaz Farhadi (Fujitsu Labs), *PnC: Predict and Cache in Content Centric networks*
- Jeff Thompson (UCLA REMAP), *NDN-CCL Libraries*
- Peter Gusev (UCLA REMAP), *ndnrtc: Real-time conferencing*
- Niky Riga (GENI Project Office), *Using GENI*
- Steven Dale (Thoughtworks // parallels.io), *Parallels: An Exploration Engine for The Discovery of Ideas*
- Yingdi Yu (UCLA), *ChronoChat* (NDN chat application)
- Zhehao Wang (UCLA REMAP), *Matryoshka: NDN Multiplayer Online Game*

## 7. SECURITY IN NDN

**James Kasten** (U. Michigan) presented an update on NDN security research. The Raspberry Pi toolkit already supports signing and verification, and NLSR (Named-data Link State Routing) has an initial security implementation, although there are unresolved usability issues, many of which have been well-studied (although not solved) in the security community, e.g., trust management, key distribution. The team is moving away from the OS-provided Trusted Platform Module (TPM) for online keys, and providing support for elliptic curve cryptography (ECC) in addition to RSA to provide cryptographic strength with higher performance. Another performance optimization is to aggregate signatures over multiple packets.

**Van Jacobson** noted that NDN's explicit shift away from a conversational, opaque model of communication to an exposed one creates an opportunity to integrate security capabilities that naturally emerge from aspects of the application or environment, rather than as retroactive security add-ons the IP architecture requires. For example, one can embed routing security functionality as part of the operation required to configure an NDN router; similarly with IoT applications or transactional security features. Imagine a relationship with an entity on the network, e.g., bank, doctor, that involves exchanging some key(s). NDN can make that security arise from existing behavior that is inherent to the relationship (or configuration). NDN is still developing the analogy to a set of application design patterns for trust management.

Van also emphasized that because TCP/IP requires viewing security as a wrapper around connections between two arbitrary points, security solutions tend to require global roots of trust. The NDN communication model affords more naturally usable security models that are tightly coupled with (i.e., congruent with) applications and their configuration. For example, NDN does not require a global system of trust to achieve security in a single building – NDN allows the name-based configuration of the devices within the building to implement the security.

**Yingdi Yu** (UCLA) described a proposed signature logging system for validation of long-lived data. The lifetime of a data packet depends on its usage – it may even exist forever – but the lifetime of a signing key is limited, which restricts the validity of data packets signed with it. Maintaining long-lived data by re-signing it with new keys over time leads to complicated processes involving key rollover and publication of re-signed data. An alternative approach is to ascertain whether a key (and thus a signature it made) was valid when it signed a given piece of data. A trustworthy logging system could support such verification at scale. Yingdi is exploring a MerkleTree-based design of such a logging system for NDN, with multiple loggers to provide redundancy and mutual auditing.

**Pedro de-las-Heras-Quirós** (U. Rey Juan Carlos) talked about his team's exploration of solutions for encryption-based group access control for NDN applications. They used NDN.js, NDN-CCL, ndnd-tlv, ndncert, and Mini-CCNx (adapted to ndnd-tlv) to develop prototypes of *codecaps* (secure abstraction with code capabilities [18]) and *macaroons* [11] in NDN.js to see if they could improve consumer anonymity in NDN when compared with signed Interests. Their experiment stored and published raw sensor data to a service that transformed and republished it to a different group with different rights. Each codecap includes a certificate chain that the codecap's owner (principal) can extend by adding new rights function that attenuate the original rights. *Macaroons* are cookies with contextual caveats for decentralized authorization, and are similar to codecaps but act as credentials rather than capabilities. His team is currently adding features to macaroons such as frequent revocation to support anonymity, and group access control where a macaroon acts as a store of a session key. These two abstractions enable scalable, flexible service composition in NDN applications: the producer does not store state proportional to the number of consumer principals (as ACLs require), and intermediate principals can design their own access control policy before delegating capabilities, not constrained by the original producer's policies.

**Aziz Mohaisen** (VeriSign) discussed how additional timing information and access controls can overcome some ICN-based privacy risks, such as timing attacks on access privacy.

## 8. OPEN CHALLENGES IN NDN

There was a discussion panel in the afternoon of the second day to engage industry views of challenges and gaps in the NDN architecture, starting with a focus on the security and privacy landscape. The panelists were **Eve Schooler** (Intel), **Ignacio Solis** (PARC), **Leonce Mekinda** (Orange), **Massimo Gallo** (Alcatel-Lucent), and **Mark Stapp** (Cisco), who provided their views on strengths and weaknesses of the current NDN architecture.

The first issue addressed was privacy, specifically policy associated with data access. The NDN model naturally supports data integrity and authenticity with signatures, but names in Interests can reveal information about requested data, albeit not the identity of the requester. Must the architecture support network awareness of requester location, in order to facilitate laws that restrict flow of certain information across national boundaries? Certainly access ISPs will know if their customers have accessed or expressed Interest in certain named data. Using digital object identifiers may

[10] http://npm.taobao.org/package/ndn-js-contrib

mask the identity of the content being requested, but also lose the advantages of using hierarchical names.

The NDN architecture, unlike IP, is agnostic to names. The network layer may use whatever names one wants, but names constrain how applications work, and vice versa – a namespace for a given app should be designed with an understanding of how data gets from producer to consumer in that app. The real privacy challenge is not decrypting the data, but figuring out who should be able to decrypt it, i.e., establishing an identity framework and trust model to support a privacy framework. In other words, if communication is shared convention, and public conventions are a recipe for decoding, then a recipe for privacy is to hide the common part of conventions, and do things locally per community. A designer always faces this trade-off: not using the common conventions means one cannot use the library support. But NDN can naturally support confining data to a local enclave better than TCP/IP, because NDN can create (named) information-based boundaries, in addition to encryption-based privacy.

The discussion continued with several technical questions:

1. What is the biggest show-stopper in the NDN architecture for the problem(s) you are trying to solve? How would you like to see it addressed?
2. What other holes do you believe exist in the current description of the NDN architecture?
3. What do you consider to be the highest priorities in addressing the unanswered questions? Any suggestions on how to address them? Any offers to help?
4. What is the most interesting / challenging application domain for NDN to consider now? What are the key challenges there NDN can help address?

For those building commercial routers, verification of packet signatures at wire speed was a major concern. Other gaps identified include: tracking cache accesses and communicating information back to the producers, converging on a standard packet format across different efforts, including what to put directly in the name.

**Ignacio Solis** (PARC) discussed motivations for the design divergence occurring in CCN, such as its removal of nonces and selectors, and making some implicit name types explicit.

**Mark Stapp** (Cisco) suggested more documentation of discussions that lead to design choices in terms of scalable routing, naming, and congestion control. Cisco is interested in application layer problems (e.g. how to support the REST paradigm to ICN), and in understanding and optimizing costs for having different functions in different places in the network, costs that will not reveal themselves in a small testbed. Mark also encouraged the community to agree on standard definition of the explicit NACK, so different elements would know how to respond. Type/name conventions should be explicit and standardized, even as they evolve over time. He thought the architecture might still need ACKs for cases of a mismatch between app service time and network RTT, i.e., the one value currently in the PIT is not enough to handle all scenarios.

**Eve Schooler** (Intel) reminded us of the need for a methodology and metrics for comparison of architectures, not just between traditional IP and NDN but also across different flavors of ICN.

## 9. NDN CONSORTIUM

**Jeff Burke** reported the launch of NDN consortium and its intended purpose. The recent release of NFD served as an inflection point for the NDN team and the community it wants to build. The consortium is a significant component of the team's effort to facilitate and promote wider conversations about technology, business, and policy issues.

The primary goal of the consortium is to grow and support a community that is collaborating to build a vibrant open source ecosystem of research and experimentation around Named Data Networking, including design, implementation, and evaluation. The consortium will provide developer support tools, organize community meetings, pursue outreach activities, host working groups for both industry verticals and cross-cutting activities, as well as provide general communication and legal support not typically funded through universities. It will also serve as an umbrella for coordination that is independent of individual sponsored research projects.

Jeff emphasized that there was no IPR component of the current consortium membership agreement, nor a standardization component.[11] Consortium members will discuss what activities the consortium should undertake; it may in the future become an independent organization. As funding allows, the consortium will hire staff to support program management, membership liaison, and documentation assistance.

## 10. MOVING FORWARD

The second day had an hour devoted to sharing reflections on the most interesting things participants learned on the previous day. There was general appreciation for the technical information offered in presentations, the opportunity for exchanges between applications and architecture developers, and the tremendous effort in releasing a free and open source software platform to maximize experimentation and research opportunities. However, some people also wanted to engage in deeper technical discussions that the packed program did not make possible, e.g., group debate of design decisions in the architecture. The NDN team emphasized the need for feedback on the first official release of the forwarding software NFD, as well as feedback on usability and functionality of the API and applications. Now that NFD was out and in use, some hoped to take advantages of NFD's modularity in architectural experimentation, and to devote more effort to applications and services development, enabling the rich NDN research agenda to further increase momentum.

Some participants expressed interest that future meetings include more discussions about mobility, scalability, managing complexity in the architecture, and demonstration of applications that are much easier to develop and deploy with NDN than with TCP/IP. Others underscored the critical importance of namespace design, since it allows application developers to take advantage of the architecture's unique strengths. One participant made the point that in order to gain traction, a new Internet architecture would have to show dramatic improvements in scalability, security, and robustness without much work on the part of developers. Both academics and industry participants were happy to hear about the testbed opening up to the larger community.[12]

Many expressed excitement for the new NDN consortium's potential to facilitate community participation in the evolution of the design, including navigating the natural tension between the need for flexibility to support architectural research now and the recognition that routers in the core will ultimately have performance demands that merit consideration early in the design phase. **Lixia Zhang** (UCLA) acknowledged her hope that the more formal structure a community consortium provides would promote intellectual exchange among various ICN efforts around the world.

[11] Details on the fee and voting structure for participation in the consortium is available at http://named-data.net/consortium. For 2014-2015, there is no fee for academic participants.

[12] http://named-data.net/ndn-testbed/policies-connecting-nodes-ndn-testbed/

Finally, in the last session of the meeting, the NDN team solicited specific suggestions about what the community would like to see at future meetings, either the next NDN Community Meeting or smaller more technically focused retreats. Suggestions included: 1) **technical breakout sessions** for sharing experiences, discussing core design decisions, developing evaluation methodologies to compare ICN architectures and approaches to problems; 2) more attention to **mobility**; 3) a third day for a **hackathon**; 4) better ways for members of the community to efficiently **tap into conversations of interest**, e.g., via technical memos or links to mailing list threads or redmine discussions; 5) more representation of (non-networking) **application perspectives**; 6) discussion of how to garner sufficient attention to attract **sustained funding** when current NSF support completes in 2016. 7) use of **visualization/animation/video to better communicate the benefits of NDN**, such as the layers of complexity added to the TCP/IP architecture over the last 25 years to cope with its fundamental deficiencies, and how the simpler NDN architecture can benefit modern Internet uses and users; and 8) a **monthly newsletter** of research and development activities.

*ACKNOWLEDGMENTS. The workshop was supported by the National Science Foundation CNS-1457074, CNS-1345286, and CNS-1345318. We thank the NDNcomm Program Committee members for their effort of putting together an excellent program. We thank all participants for their insights and feedback at the workshop.*

## 11. REFERENCES

[1] CCNx binary encoding (ccnb). `http://www.ccnx.org/releases/latest/doc/technical/BinaryEncoding.html`, 2013.

[2] Named Data Networking Consortium, 2014. `http://named-data.net/consortium`.

[3] Named Data Networking Next Phase (NDN-NP) Proposal. Technical Report NDN-0026, NDN, 2014.

[4] NDN Community Meeting (NDNcomm 2014): Architecture, Applications, and Collaboration, September 2014. `http://www.caida.org/workshops/ndn/1409/index.xml`.

[5] ndn-cxx: NDN C++ library with eXperimental eXtensions. `http://named-data.net/doc/ndn-cxx/`, 2014.

[6] repo-ng: Next generation of NDN repository. `https://github.com/named-data/repo-ng`, 2014.

[7] Alexander Afanasyev, Priya Mahadevan, Ilya Moiseenko, Ersin Uzun, and Lixia Zhang. Interest flooding attack and countermeasures in Named Data Networking. In *Proc. of IFIP Networking*, May 2013.

[8] Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005, NDN Project, July 2012.

[9] Alexander Afanasyev, Junxiao Shi, Beichuan Zhang, Lixia Zhang, Ilya Moi-seenko, Yingdi Yu, Wentao Shang, Yi Huang, Jerald Paul Abraham, Steve DiBenedetto, Chengyu Fan, Christos Papadopoulos, Davide Pesavento, Giulio Grassi, Giovanni Pau, Hang Zhang, Tian Song, Haowei Yuan, Hila Ben Abraham, Patrick Crowley, Syed Obaid Amin, Vince Lehman, , and Lan Wang. NFD developers guide. Technical Report NDN-0021, NDN Project, July 2014.

[10] Chaoyi Bian, Zhenkai Zhu, Alexander Afanasyev, Ersin Uzun, and Lixia Zhang. Deploying key management on NDN testbed. Technical Report NDN-0009, Revision 2, NDN, February 2013.

[11] A. Birgisson et al. Macaroons: Cookies with contextual caveats for decentralized authorization in the cloud. In *Network and Distributed System Security Symposium*, 2014.

[12] Ilya Moiseenko and Lixia Zhang. Consumer-producer api for named data networking. Technical report, NDN Project, 2014. `http://named-data.net/publications/techreports/tr17-consumer-producer-api/`.

[13] NDN Project. NDN Packet Format Specification. Online: `http://named-data.net/doc/ndn-tlv/`, 2014.

[14] NDN Project. NDN platform. Online: `http://named-data.net/codebase/platform/`, 2014.

[15] NDN Project. NFD - named data networking forwarding daemon. Online: `http://named-data.net/doc/NFD/0.2.0/`, 2014.

[16] NLSR - Named Data Link State Routing Protocol. `http://named-data.net/doc/NLSR/0.1.0/`, 2014.

[17] Jeff Thompson and Jeff Burke. NDN common client libraries. Technical Report NDN-0024, NDN, September 2014.

[18] R. van Renesse et. al. Secure abstraction with code capabilities. In *Int. Conference on Parallel, Distributed and Network-Based Processing*, 2013.

[19] 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.

[20] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang. Named data networking. *ACM SIGCOMM Computer Communication Review*, July 2014.

[21] Lixia Zhang et al. Named Data Networking. Technical Report NDN-0001, NDN, 2010.

[22] Zhenkai Zhu and Alexander Afanasyev. Let's ChronoSync: Decentralized dataset state synchronization in Named Data Networking. In *Proceedings of the 21st IEEE International Conference on Network Protocols (ICNP 2013)*, 2013.