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

August 8, 2014

Contents

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

PROJECT SUMMARY

Named Data Networking (NDN) is a Future Internet Architecture inspired by years of empirical research into network usage and a growing awareness of persistently unsolved problems of the current Internet (IP) architecture. Its central premise is that the Internet is primarily being used as an **information distribution network**, a use that is not a good match for IP, and that the future Internet's "thin waist" should be based on **named data** rather than **numerically addressed hosts**.

This proposal continues research on NDN started in 2010 under NSF's FIA program. It applies the project team's increasingly sophisticated understanding of NDN's opportunities and challenges to two national priorities–Health IT and Cyberphysical Systems–to further the evolution of the architecture in the experimental, application-driven manner that proved successful in the first three years. In particular, our research agenda is organized to translate important results in architecture and security into library code that guides development for these environments and other key applications toward *native NDN designs*. It simultaneously continues fundamental research into the challenges of global scalability and broad opportunities for architectural innovation opened up by "simply" routing and forwarding data based on names.

Our research agenda includes: (1) **Application design**, exploring naming and application design patterns, support for rendezvous, discovery and bootstrapping, the role and design of in-network storage, and use of new data synchronization primitives; (2) **Security and trustworthiness**, providing basic building blocks of key management, trust management, and encryption-based access control for the new network, as well as anticipating and mitigating future security challenges faced in broad deployment; (3) **Routing and forwarding strategy**, developing and evaluating path-vector, link-state, and hyperbolic options for inter-domain routing, creating overall approaches to routing security and trust, as well as designing flexible forwarding and mobility support; (4) **Scalable forwarding**, aiming to support real-world deployment, evaluation and adoption via an operational, scalable forwarding platform; (5) **Library and tool development**, developing reference implementations for client APIs, trust and security, and new network primitives based on the team's fundamental results, as well as supporting internal prototype development and external community efforts; (6) **Social and economic impacts**, considering the specific questions faced in our network environments as well as broader questions that arise in considering a "World on NDN."

We choose *Mobile Health* and *Enterprise Building Automation and Management Systems* as specific instances of Health IT and Cyberphysical Systems to validate the architecture as well as drive new research. Domain experts for the former will be the Open mHealth team, a non-profit patient-centric ecosystem for mHealth, led by Deborah Estrin (Cornell) and Ida Sim (UCSF). For the latter, our experts will be UCLA Facilities Management, operators of the second largest Siemens building monitoring system on the West Coast. To guide our research on the security dimensions of these important environments and the NDN architecture more generally, we have convened a Security Advisory Council (NDN-SAC) to complement our own security and trust effort.

Intellectual Merit The NDN architecture builds on lessons learned from the success of the IP architecture, preserving principles of the thin waist, hierarchical names, and the end-to-end principle. The design reflects a recognition of the major shift in the applications communication model: from the "where" (*i.e.*, the host/location) to the "what" (*i.e.*, the content). Architecting a communications infrastructure around this shift can radically simplify application designs to allow applications to communicate directly using the name of the content they desire and leave to the network to figure out how and where to retrieve it. NDN also recognizes that the biggest weakness in the current Internet architecture is lack of security, and incorporates a fundamental building block to improve security by requiring that all content be cryptographically signed.

Broader Impacts The success of new architectures requires broad community involvement and uptake. NDN has built significant momentum through commitment to an open source model that has spurred substantial research activity in both architecture and current implementation. Project members are often invited to present at "future Internet" meetings around the world, and we have performed two high-visibility demos of NDN's ability to handle large scale distribution. Industry is also showing increasing interest in NDN. Finally, NDN has also had a significant impact on our students, yielding several current Ph.D. theses on NDN topics, industry internships involving NDN research, and graduate and undergraduate curriculum material that offer a comprehensive alternative to IP to stimulate discussion of what network architecture design really means.

PROJECT DESCRIPTION

1 Vision: A New Narrow Waist

In 2010 we proposed to explore a new vision of communication networks, inspired by years of empirical research into network usage, and a growing awareness of persistently unsolved problems of the current Internet (IP) architecture. While TCP/IP was a unique and ground-breaking solution, it was solving a problem of the telephony world: *a point-to-point conversation between two communication endpoints.* The world has changed dramatically since then, driven by the spectacular success of the TCP/IP architecture. Growth in e-commerce, digital media, social networking, and smartphone applications has resulted in the Internet primarily being used as an **information distribution network**, a use that is not a good match to the architecture. Many complexities and kludges have been introduced to mitigate this inherent misalignment between the Internet architecture and its primary use today.

The conversational nature of IP is embodied in its datagram format: IP datagrams identify communication endpoints (the IP destination and source addresses). We proposed the Named Data Network (NDN) architecture, aiming to generalize 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 anything, including a communication endpoint, as TCP/IP does today, or a chunk of data in a conversation, as the TCP/IP transport signature plus sequence number does today. But NDN names can also identify a chunk of a movie or a book, a command to turn on some lights, etc. This simple change to the hourglass model, allowing the thin waist to use data names instead of IP addresses for data delivery, **makes data rather than its containers a first-class citizen in the Internet architecture**. This conceptually simple change allows NDN networks to use almost all of the Internet's well-understood and well-tested engineering properties to efficiently solve not only communication problems but also digital distribution and control problems, and to effectively address some of the Internet's most pressing problems in security, scalability, mobility, and facilitation of applications development. In particular,

- NDN enables applications to secure data end-to-end; the name provides essential context for security, with fine-grained signing and verification, limiting the data security perimeter to the context of a single application.
- NDN networks are loop-free by design, which means any node can freely use any/all of its connectivity to solicit or distribute data. Together with NDN's support for in-network storage, this capability facilitates retrieving data from nearby locations, and reduces the control asymmetries that give today's dominant providers disproportionate control over routes and thus over smaller, local providers.
- Because NDN retrieves data by names, it circumvents the need for IP addresses and the associated issues with their scarcity. Mobility, which requires changing addresses in IP, is seamless in NDN since data names remain the same.
- Today's applications are typically written in terms of *what* information they want rather than *where* it's located, using application-specific middleware to map from application to network layer. NDN enables applications to directly address *what* they want, simplifying application development and enabling fundamentally new types of applications.

We are pursuing an application-driven approach to architecture research and development, and have deployed a 12-site testbed (an overlay on the current Internet) on which to iteratively design them. Our set of prototype applications work in their local environment (e.g. building control, file sharing) and provide immediate benefits. They also work across the testbed, and we have demonstrated several communications and data distribution applications on a global demonstration environment (described in Section 3.4).

Our application-driven investigation has enabled us to begin describing general principles and guidelines for application designs in NDN networks, and to translate these principles and guidelines into naming conventions implemented in system libraries. Library implementation operationalizes what we have learned in a form that supports consistent reuse, simplify future application development and accelerate progress, and promotes community experimentation and contribution. Broad R&D community engagement is necessary to the success of any new architecture; we are encouraged by the many information-centric networking workshops that have proliferated over the last 3 years.

In this proposal we first present a review of the current NDN architecture in Section 2. Section 3 summarizes our progress in the last three years. Section 4 describes the two network environments we have

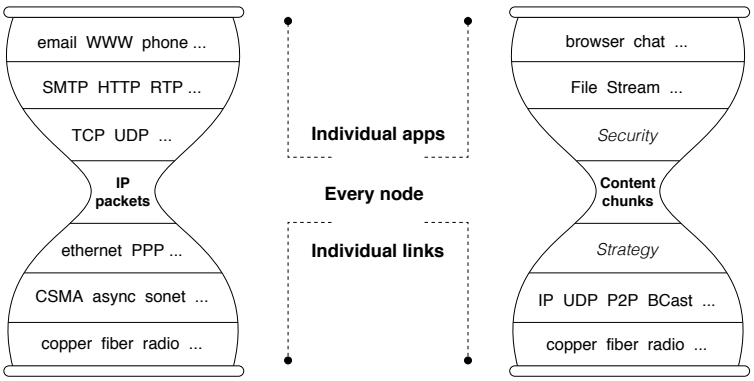

Figure 1: Internet and NDN Hourglass Architectures

selected to test and evaluate our architecture in the next phase of the FIA project. In Section 5 we provide details on our research agenda, including application, routing and forwarding strategies, scalable forwarding, software development, and social and economic impacts. Section 6 describes our approach to evaluation of the architecture, and Section 7 explains our plan for education and outreach. We summarize broader impact contributions in Section 8. Our management and collaboration plan and intellectual property policy statements are in separate documents per the FIA-NP solicitation requirements.

2 Architecture

The design principles of NDN reflect our understanding of the strengths and limitations of the current Internet architecture. The Internet's *hourglass architecture* centers on a *universal* network layer (i.e., IP) which implements the minimal functionality necessary for global interconnectivity. This thin waist enabled the Internet's explosive growth by allowing both lower and upper layer technologies to innovate without unnecessary constraints. The NDN architecture retains this extraordinarily successful hourglass-shaped architecture, but transforms the thin waist to focus on data directly rather than its location. More specifically, NDN changes the semantic of the network service from *delivering a packet to a given destination address* to *retrieving data identified by a given name* (Figure 1). NDN's design is also guided by the following principles.

- *Security must be built into the architecture.* Security in the current Internet architecture is an afterthought, not meeting the demands of today's diverse environment. NDN provides a fundamental security building block *right at the thin waist* by signing all named data.
- The *end-to-end principle* [61] underlying the TCP/IP architecture enabled development of robust applications in the face of unexpected failures. NDN expands this principle by securing data end-to-end.
- *Network traffic must be self-regulating*. Flow-balanced data delivery is essential to the stability of large systems. Unlike IP's open-loop packet delivery, NDN designs flow-balance into the thin waist.
- The *architecture should facilitate user choice and competition where possible.* Although not considered in the original Internet design, global deployment has taught us that "architecture is not neutral." NDN makes a conscious effort to empower end users and facilitate competition [17].

NDN communication consists of two types of packets: ***Interest*** and ***Data*** (see Figure 2). Both types of packets contain a *name* that identifies a piece of data that can be transmitted in one Data packet. To retrieve Data, a consumer puts the name of desired data into an Interest packet and sends it to the network. Routers use this name to forward the Interest toward the data producer(s). Once an Interest reaches a node that has the requested data, that node returns a Data packet that contains both the name and the content, and a producer signature that binds the two. NDN communication is always receiver-driven and two-way.

Conceptually each NDN router maintains three major data structures: ***Content Store***, ***Pending Interest Table (PIT)***, and ***Forwarding Information Base (FIB)*** (Fig. 3). The Content Store is a temporary cache of Data packets that the router has received. Because an NDN Data packet is meaningful independent of where it comes from or is headed, the router can cache it to satisfy future Interests. The PIT stores all Interests that have been forwarded but not satisfied yet. It records the Interest's name, incoming interface(s) and outgoing interface(s). When a node receives multiple Interests with the same name from downstream, it needs to send only the first one upstream toward the data producer. The FIB is a name-prefix-based

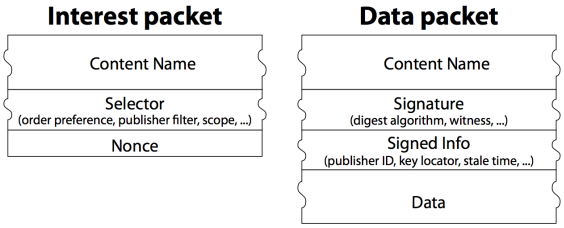

Figure 2: Packets In the NDN Architecture.

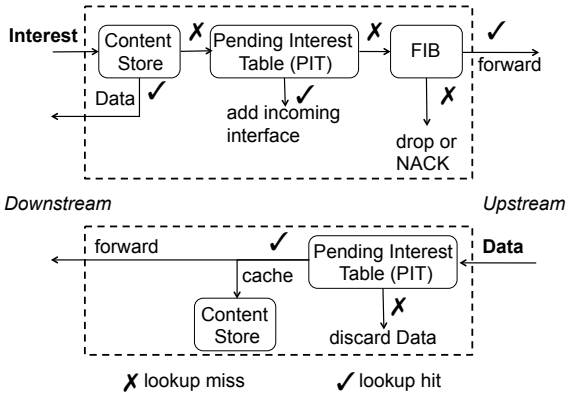

Figure 3: Forwarding Process at an NDN Node.

routing table maintained to support this communication process; if an Interest does not have a match in the PIT, the router will look up its name in the FIB and forward the Interest toward the data producer(s).

When a Data packet arrives, the router finds the matching PIT entry and forwards the data to all downstream interfaces listed in the PIT entry. It then removes that PIT entry, and caches the Data in the Content Store. Data packets always take the reverse path of Interests, and one Interest packet results in one Data packet on each link, providing *flow balance*. Neither Interest nor Data packets carry any host or interface addresses (such as IP addresses); Interest packets are routed toward data producers based on the names carried in them, and Data packets return based on the state information set up by the Interests at each hop.

2.1 Names

NDN names are *opaque* to the network, *i.e.*, routers do not know the meaning of a name (although they know the boundaries between components in a name). This allows each application to choose the naming scheme that fits its needs and allows the naming schemes to evolve independently from the network.

NDN design assumes hierarchically *structured* names, e.g., a video produced by UCLA may have the name `/ucla/videos/demo.mpg`, where '/' delineates name components in text representations, similar to URLs. This hierarchical structure allows applications to represent relationships between data elements. For example, segment 3 of version 1 of the video might be named `/ucla/videos/demo.mpg/1/3`. It also allows name aggregation: In this example `UCLA` could correspond to the autonomous system from which the video originates. Flat names can be accommodated as a special case and useful in local environments, however hierarchical name spaces are essential in scaling the routing system. Naming conventions among data producers and consumers, *e.g.*, to indicate versioning and segmentation, are specific to applications but opaque to the network.

To retrieve dynamically generated data, consumers must be able to deterministically construct the complete name for a desired piece of data or use a combination of NDN's *Interest selectors* and the stack's *longest prefix matching* to retrieve the desired data in one or more iterations. Interest selectors include 1) the desired minimum and/or maximum number of name components after the prefix, 2) child name range specification based on a well-known sort ordering, and 3) scoping criteria (host, next hop, and world). In practice, a simple set of selectors is quite powerful for retrieving data with only partially known names. For example, a consumer wanting the first version of the `demo.mpg` video may request `/ucla/videos/demo.mpg/1` with the Interest selector "leftmost child" and receive a data packet named `/ucla/videos/demo.mpg/1/1` corresponding to the first segment. The consumer can then request later segments using a combination of information revealed by the first data packet, the naming convention of the publishing application, and selectors that request subsequent packets.

Data that may be retrieved globally must have *globally unique* name, however names used for local communications require only local routing (or local broadcast) to find corresponding data. In fact, individual data names can be meaningful in various specific scopes and contexts, ranging from "the light switch in this room" to "all country names in the world". How to develop efficient strategies to fetch data within the intended scope is a new research area.

Name space management is not part of the NDN architecture, just as address space management is not part of the IP architecture. However, naming is the most important part of the NDN design. Naming

data enables natural support for functionality such as content distribution, multicast, mobility, and delay-tolerant networking. We are learning through experimentation on how applications should choose names that facilitate both application development and network delivery. As we develop and refine our principles for naming, we implement them in system libraries to simplify future application development. Fortunately, the opaqueness of names to the network means that architecture development can proceed in parallel with research into namespace structure and navigation in the context of application development.

2.2 Data-Centric Security

NDN requires data producers to cryptographically sign each piece of data and its name, embedding a layer of security into the data itself, in contrast to TCP/IP's approach where security (or lack thereof) is a function of where or how the data is obtained [36]. The publisher's signature enables determination of data *provenance*, and supports fine-grained trust, allowing consumers to reason about whether a public key owner is an acceptable publisher for a particular piece of data in a specific context.

Historically, security based on public key cryptography has been considered inefficient, unusable and difficult to deploy. NDN needs flexible and usable mechanisms to sign data as well as to manage user trust. Keys are just another type of NDN data, simplifying key distribution. Secure binding of names to data supports a wide range of trust models. If a piece of data is a public key, a binding is effectively a public key certificate. Finally, NDN's end-to-end approach to security facilitates trust between publishers and consumers, and gives applications flexibility in customizing their trust model.

NDN's data-centric security has natural application to content access control and infrastructure security. Applications can control access to data via encryption and distribute (data encryption) keys as encrypted NDN data, limiting the data security perimeter to the context of a single application. Requiring signatures on network routing and control messages (like any other NDN data) provides a solid foundation for routing protocol security, protecting against spoofing and tampering. NDN's inherent multipath routing, together with the adaptive forwarding plane we describe next, mitigates prefix hijacking because routers can detect the anomaly caused by a hijack and retrieve the data through alternate paths. NDN messages can only reference data, and cannot address hosts, making it difficult to maliciously target a particular device. One possible NDN attack is a denial of service via an Interest flood, which we discuss further in Section 3.5.

2.3 Adaptive Routing and Forwarding

NDN routes and forwards packets based on the names in the packets, which eliminates four problems caused by addresses in the IP architecture: address exhaustion, address management, NAT traversal, and mobility, which requires changing addresses in IP. NDN can make use of conventional routing algorithms such as link state and distance vector, using announcements of *name prefixes* (rather than IP prefixes) of attached networks. Routers perform component-wise longest prefix match of the name in a packet against the FIB. In the first phase of our project, we designed and implemented an NDN link state routing protocol (Section 3.2), and developed efficient data structures and algorithms for fast lookup of variable-length, hierarchical names (Section 3.3).

The PIT supports forwarding across NDN's data plane, recording all pending Interests and their incoming interfaces, and removing Interests after matching Data is received or a timeout occurs. This per-packet state is a fundamental change from IP's stateless data plane. The state information enables NDN nodes to monitor packet delivery performance and loss across different interfaces, adapting to network failures and efficiently using network resources. Via a random nonce in the Interest packet, NDN nodes can identify and discard packets that have returned to the same node, preventing loops. This allows NDN nodes to freely use multiple paths, toward the same data producer.

Each NDN node also implements a per-node *forwarding strategy*, which does not exist in today's IP nodes. The role of the forwarding strategy module is to make informed decisions about which Interests to forward to which interfaces, how many unsatisfied Interests to allow, relative priority of different Interests, etc. Each node also makes local decisions to load balance their forwarding of Interests through multiple interfaces, and to choose alternative paths to avoid detected failures. The flow balance requirement means that routers can control traffic load.

2.4 In-Network Storage

Because each NDN Data packet carries a name and a signature, it is meaningful independent of its source or destination, so a router can cache the data in its Content Store to satisfy future requests. NDN treats storage and network channels identically in terms of data retrieval. Upon receiving a new Interest, the router first checks the Content Store for matching data; if it exists the router returns the Data on the interface from which the Interest came. The Content Store is analogous to buffer memory in IP routers, but IP routers cannot reuse data after forwarding it, while NDN routers can. For static files, NDN achieves almost optimal data delivery. Even dynamic content can benefit from caching in the case of multicast (*e.g.*, teleconferencing) or retransmission after a packet loss. In addition to the Content Store, the NDN architecture also (recently) supports a more persistent and larger-volume in-network storage, called a Repository (or Repo), described further in Section 5.1.

Caching named data raises different privacy concerns from those of IP. In IP, knowing who is consuming what data requires examining packet headers or possibly payload. The naming and caching of data in NDN networks may facilitate observation of what data is being requested, but without destination addresses it is harder to identify who is requesting the data (unless one is directly connected to the requesting host). This aspect of the architecture offers some natural privacy protection at a different level than IP networks.

2.5 From Transport to Distributed Synchronization

The NDN architecture does not have a separate transport layer. It moves the functions of today's transport protocols into applications, their supporting libraries, and the strategy component of the forwarding plane. NDN does not use port numbers; a host knows to which application to deliver packet based on data names, and applications handle data integrity checking, signing, and trust decisions related to their data. To provide reliable delivery across highly dynamic and possibly intermittent connectivity, such as in mobile environments, nodes will discard Interest packets that remain unsatisfied after some threshold of time. The application that originated the initial Interest must retransmit it if it still wants the data.

NDN's flow balance requirement, together with the ability of nodes to control their own traffic load by limiting the number of pending Interests (Section 2.3), means that there is no need for separate end-to-end congestion control, a typical transport layer function. If congestion losses occur, caching will mitigate the impact since retransmitted Interests may be satisfied by caches closer to the source of the Interests. NDN thus avoids the kind of congestion collapse that can occur in today's Internet when a packet is lost near its destination and repeated retransmissions from the original source host(s) consume most of the bandwidth.

To aid development of robust and efficient distributed applications, we have added a fundamentally new architectural building block that we call *Sync*, described in Section 3.1. Using NDN's basic Interest-Data exchange communication model, Sync uses naming conventions to enable multiple parties to synchronize data. By exchanging individually computed data digests, parties learn about new or missing data quickly and reliably, and efficiently retrieve data via NDN's built-in multicast data delivery.

3 Progress Toward Vision: 2010-2013

The NDN project's fundamental research challenge is to evolve the above vision into an architectural framework capable of solving real problems, particularly in application areas poorly served by today's TCP/IP architecture. We have focused our study on how NDN's narrow waist drives design choices and offers new opportunities beyond what IP can offer. Our work has led to a large number of publications, a complete NDN prototype, an NDN simulator, an overlay testbed with our prototype deployed, as well as two successful large-scale demonstrations.

3.1 NDN Application Development

Over the last 3 years, we have prototyped a wide range of different applications, the source code for which is available at `https://github.com/named-data/`. Designing, implementing, and demonstrating these applications yielded lessons about the advantages and challenges of the NDN architecture, as well as insights into what additional architectural features might be needed. Table 1 lists ways that NDN improves over IP for several common challenges in developing distributed applications.

Video streaming Our video streaming application [41] taught us that session-less streaming, which removes scaling limitations in video distribution, can indeed be achieved using NDN for both pre-recorded and

IP Challenge	NDN Improvement
Collaboratively developed	
Limited application semantics represented in the network architecture	Express application semantics in network naming, down to packet level
Iteratively deployed	
IP networks can be brittle to change or require NAT, etc. to scale	Reduce network brittleness to configuration change
Dynamically assembled	
Connection-/session-oriented models; address assignment requirement	Connection-less, session-less communication, leveraging storage in the network
Physically integrated	
IPv6 provides "room" but does not aid application development	Consistent and meaningful addressing for virtual and physical components
Asynchronously experienced	
Mobility and multicast not well-supported	Disruption tolerant and multipath-friendly
Globally integrated	
Perimeter- and channel-based security model presents challenges	Data-centric rather than perimeter-centric security

Table 1: Advantages of NDN over IP in developing distributed applications

live streams. Using a timecode-based and data quality-based namespace, NDNVideo easily offers functions that are difficult and more expensive to support in today's Internet, such as seamless viewer-specific rate adaptation, per-packet signature verification, and efficient random access into streams. Furthermore, pushing named and signed data packets into NDN's permanent data storage, i.e., Repo (Section 2.4), allowed us to integrate the handling of live and pre-recorded streaming: in either case, consumers simply issue Interests for desired data. NDNVideo has been used extensively in tests and demonstrations, as described in Section 3.3. This work also revealed missing functions in the existing Repo implementation, including access and retention control, and a lack of library support for getting the latest data from a time-series source. We discuss plans for addressing such issues in Section 5.1.

Server-less peer-to-peer applications We have developed a set of distributed collaborative applications that operate in a pure peer-to-peer manner. These applications include audio conferencing [97], a chat room [96], and file sharing ("NDN-dropbox", dubbed *ChronoShare*), all *without a centralized coordinating server*. In particular, in developing the last two applications we exploited the formulation of the Sync concept, and prototyped ChronoSync [95], a protocol that runs on top of NDN's Interest-Data exchanges to efficiently synchronize the state of a dataset among distributed users. Using appropriate naming rules, ChronoSync summarizes the dataset state in a condensed cryptographic digest form and exchanges it among the distributed parties. Each user uses this digest to detect any differences in the dataset and disseminates detected difference efficiently to all parties. With the complete and up-to-date knowledge of the dataset changes, individual applications can decide whether or when to fetch which pieces of the data. ChronoSync is still in its preliminary development stage and will be applied to other application cases to verify and validate its design, as we discuss in Section 5.1.

Multiplayer games We also explored the use of the above peer-to-peer strategies in running multi-player game prototypes [53], using the popular Unity3D game engine. Realtime 3D graphics environments such as this are a significant part of consumer experiences, professional simulations, and a variety of other applications. Our effort explored NDN distribution and security strategies in such scenarios where not only does everyone wants packet exchange with everyone else (nearby in virtual space), but interactivity requirements impose significant performance requirements. This work helped to confirm primary limitations of the initial Sync implementation, including its integration with content repository and focus on directly connected peers. We believe NDN's inherent multicast and content caching will ultimately provide immediate improvements in bandwidth requirements for shared assets and state.

Instrumented environments We have developed and demonstrated a unique lighting control application that explores actuation over NDN as a contrast to the usual focus on content distribution, demonstrating its utility as an enabler of the "Internet of Things" and a viable substrate for Cyberphysical systems [14]. Our lighting control testbed is in a television studio on the UCLA campus, where we have eleven TCP/IP-

enabled color mixing lighting fixtures controlled by the embedded processor that acts as an NDN gateway. The control system incorporates multiple keys and control namespaces for the lights, as well as, notably, "authenticated interests." Our approach to the latter incorporates a signature into Interest packets that enables them to be interpreted as authenticated RPC calls[1]. We have explored both asymmetric signatures and symmetric HMACs for verification and, after discovering that signing acknowledgment data packets burdens the embedded controller, we have come up with an encrypted challenge approach to improve performance [13]. Most recently, we created a prototype sensing application that publishes real data from UCLA chilled water and electrical demand monitors via NDN,[2] and installed an industry standard Siemens electrical demand monitoring system for the television studio, providing a dedicated testbed for building management explorations. These have been supported by UCLA Facilities Management, who will be our partner in the Building Automation and Management environment described in Section 4.2.

Web browser support Many of the applications above have motivated and incorporated new libraries including Python and C# bindings, cryptography extensions, and autoconfiguration tools. Another notable library we developed is NDN.JS, a pure JavaScript implementation of a client-side library that allows a standard web browser to use NDN for transport [65] by tunneling over WebSockets, TCP or UDP. It aims to address both the high-level goal of exploring an NDN-based world wide web and a practical need for browser support in NDN research. Pragmatically, it aims to enable more widespread dissemination of the NDN protocol and applications by reducing the complexity of usage for both users and developers. Since the release of NDN.JS code, it has been widely adopted both inside and outside our project team for various applications, including a Mozilla Firefox Add-on for accessing files published over NDN using an `ndn:/` URI scheme, and the "Smallest Federated Wiki" example mentioned in Section 3.6.

Vehicular networking Through experimenting with NDN-equipped cars [76, 75, 82], we discovered that vehicle networking via NDN allows vehicles to utilize any or all of their available channels, e.g., WiFi, WiMax, Cellular, or even vehicles themselves as data carriers, to communicate in an infrastructure-free manner. Our prototype implementation demonstrates the feasibility of a single NDN protocol framework that integrates ad hoc, delay tolerant, and peer-to-peer features all into one coherent network through the shared application namespace [29]. Information-maximizing caching and content prioritization within this network have shown a significant effect on the performance of applications that utilize content from vehicles or that exploit vehicles to ferry data. Our recent work demonstrated that, by proper name space design, NDN greatly facilitates the design and implementation of information-maximizing services that not only exploit all available channels to maximize metrics such as bit throughput, but also exploit data names to identify and reduce (an approximate measure of) content redundancy, thereby maximizing useful *information throughput* as well. The work lays the foundation for developing approaches on top of NDN that maximize application-level quality-of-information (QoI) by exploiting network knowledge of data names. An example, called Minerva, was proposed and evaluated in the context of a Beijing taxi cab mobility dataset, demonstrating the advantages of NDN in improving application-level QoI [82].

3.2 Routing and Forwarding Strategy

We tackled two major challenges: (a) bounding the amount of routing state while allowing an unbounded name space; and (b) supporting intelligent forwarding of Interests over multiple paths.

Routing protocol development One of NDN's most elegant features is at the heart of most routing protocols: an information-oriented guided-diffusion flooding model that functions in the pre-topology phase where peer identities and locations are unknown. For the initial NDN testbed, we implemented OSPFN [81], an extension of OSPF, to support the propagation of name prefixes. Since 2012 we have designed and implemented NLSR, Named-data Link State Routing [35], which utilizes NDN's built-in data security to secure routing updates, and Sync to keep routers synchronized. It fully supports multipath forwarding and is more efficient than OSPFN due to its reduced message overhead.

Routing Scalability Following the well-established approach of map-n-cap [34], we proposed to use *Forwarding Hints* to scale NDN routing [5]. When a name prefix N is not propagated to the global routing table, its producer will register in DNS one or multiple *Forwarding Hint* Resource Records, each containing the

[1] The semantics are consistent with NDN: the controller is interested in the status data resulting from changing an actuator's property to a value expressed in the interest name, with a signature calculated using the controller's key over the actuator, property, and value.

[2] `http://tinyurl.com/ndnsensor`

name prefix of a provider M, i.e., Interests with the name prefix N can be forwarded using the name prefix M instead. We have finished the design and are currently implementing this DNS extension within NDN.

Hyperbolic routing represents a promising and radically different approach to scaling NDN routing. Assuming certain properties of the network topology (which the current Internet does have, e.g., small-world, scale-free) and the ability to map the observable topology to an underlying hyperbolic metric space (also demonstrated for the Internet topology [11], in the NDN case this space must be the namespace), then one can associate each name prefix with its hyperbolic coordinates to calculate the next-hop(s) for each name prefix using greedy routing. We have implemented hyperbolic routing in NLSR. NLSR maps each name prefix to its originating router's coordinates and then calculates a ranked list of next-hops using the neighbor routers' coordinates – the neighbor router closest to the name prefix in hyperbolic space is ranked the highest. We are currently testing the effectiveness of this scheme using Emulab.

Forwarding Strategy We have sketched an initial design of NDN's forwarding strategy and evaluated its data delivery performance under adverse conditions through simulations [85]. Our results show that NDN's stateful forwarding plane can successfully circumvent prefix hijackers, quickly route around failed links, and utilize multiple paths to mitigate congestion. Whenever a forwarding problem happens, forwarding strategy uses NDN's per packet, per hop state to make intelligent decisions to forward Interest toward data producers. We also compared NDN's performance with Path Splicing, an IP-based multipath routing solution, to gain insight on the advantages of a stateful forwarding plane [85].

3.3 Scalable Forwarding

Concepts, Issues and Principles We have presented the concepts, issues and principles of scalable NDN forwarding [86]. The NDN forwarding plane has some unique characteristics relative to IP, and most of the differences represent performance challenges. We have pursued several algorithmic and implementation directions to achieve speed and scale. The essential function of NDN forwarding plane is fast name lookup. By studying the performance of the NDN reference implementation, and simplifying its forwarding structure, we have identified three key issues in the design of a scalable NDN forwarding plane: 1) exact string matching with fast updates, 2) longest prefix matching for variable-length and unbounded names and 3) large-scale flow maintenance. We have also described five forwarding plane design principles for achieving 1 Gbps throughput in software implementations and 10 Gbps with hardware acceleration.

Supporting Billions of Name Prefixes The longest prefix match (LPM) lookup requirement is common to IP and NDN forwarding. However, IP-based designs need only scale to a few million prefixes, whereas the NDN namespace has no theoretical upper bound. Consequently, current LPM designs are unworkable for wide-area NDN deployment. We are pursuing a two-pronged approach. In the routing domain, we limit the forwarding table size through scalable routing mechanisms such as map-n-encap and hyperbolic routing (Section 3.2). In the forwarding domain, we are designing prefix matching algorithms (publication currently under review) that can scale to a billion or more forwarding entries (prefixes), driven by two key insights. First, we have shown that it is possible to design a forwarding structure whose size is dependent upon the information-theoretic differences between rules in a ruleset, rather than by their lengths. Second, we have shown how a distributed forwarding plane can be engineered to support namespaces whose fundamental resource requirements exceed the resources of any single system. We have intentionally avoided making strong assumptions about the precise features of future namespaces. Both flat and hierarchical namespaces admit opportunities to engineer optimizations of these core ideas. Our primary aim is to design and evaluate an approach that will work for large future namespaces, regardless of their characteristics.

Comparisons to Other Name-Based Architectures We have demonstrated architectural advantages of the NDN design by comparing content distribution performance using NDN and HTTP [87]. Among alternative name-based systems, HTTP is the most significant by any measure. A majority of today's content distribution services leverage widely deployed HTTP infrastructure, such as web servers and caching proxies. We performed an experimental performance evaluation of NDN-based and HTTP-based content distribution solutions. Specifically, the HTTP-based solution leverages popular web server lighttpd and the Squid caching web proxy. Our findings verify popular intuition, but also surprise in some ways. In wired networks with local-area transmission latencies, the HTTP-based solution dramatically outperforms NDN, with roughly 10x greater sustained throughput. In networks with lossy access links, such as wireless links with 10% drop rates, or with non-local transmission delays, due to faster link retransmission brought by archi-

tectural advantages of NDN, the situation reverses and NDN outperforms HTTP, with sustained throughput increased by roughly 4x over a range of experimental scenarios.

3.4 Network Experimentation: Testbed, Simulation, and Demonstration

To experiment with the NDN architecture, during the first year of the NDN project we established both a well instrumented ONL testbed with programmable routers, which has been used for baseline assessment of the NDN prototype implementation, and an overlay testbed among all the participating institutions, which serves both as a platform for network management research and as a collaboration tool. To test video streaming and other applications, UCLA also distributed NDN application boxes to six other sites for experimentation. These boxes currently host streaming audio/video using NDN and run web proxies for the NDN.JS javascript libraries described above.

In order to answer questions regarding large-scale NDN network properties, we have developed an ns-3 based NDN simulator, ndnSIM [4, 1], and released it to the public in summer 2012 to enable the community at large to use a *common* simulation platform for NDN research. Since then the ndnSIM user community has grown rapidly. The ndnSIM mailing list hosts active discussions on ndnSIM usage and development among over 100 members from a dozen countries [46].

In March 2012, we presented a live, large-scale demonstration of NDN's unique capabilities at the GENI engineering conference using the NDN testbed, augmented with hundreds of Amazon EC2 hosts. In May 2013, the NDN team mounted our second and most substantial demonstration for the 2013 China-America Frontiers of Engineering Symposium (CAFOE) sponsored by the US National Academy of Engineering in Beijing [19]. The live NDN demonstration involved devices spanning 5 continents, and included several communications and data distribution applications, including real-time lighting control of a performance stage during a live musical performance (the control was done from Beijing, while the musical performance was in Los Angeles). The largest demonstration component showed 1000 NDN video clients, spread across five continents, consuming the same video stream at the same time from the same source. The presentation and demonstration were very well received.

3.5 Security and Privacy

Security through Cryptographic Protection Two major challenges in NDN security are: (1) highly efficient packet signing/verification for both Interests and Data, and (2) key/trust management. Toward addressing the first challenge, we incorporated support for DSA [47] and EC-DSA [37] signature schemes in NDN. DSA allows efficient signing, at the cost of slightly less efficient verification (compared to RSA [58] signing and verification). EC-DSA is a variant of DSA over elliptic curves. It reduces both signing and verification costs while maintaining the same performance balance between the two operations. We also implemented content packet authentication using symmetric message authentication codes (MACs [8]). With MACs, packet authenticity and integrity are not publicly verifiable, since only a party that knows a symmetric key can verify a MAC. However, MAC can be implemented at low cost using an efficient hash function (e.g., HMAC [8]) or a block cipher (e.g., CBC-MAC [9]). This makes them appealing in computation- and/or bandwidth-limited settings, such as wireless/mobile networks.

Toward addressing key/trust management, we deployed a simple hierarchy trust model on NDN testbed [10], which is now used in securing NDN routing protocol [35]. In retrospect we have not moved as fast as we expected in the trust management area. We originally assumed that all development efforts would be able to incorporate a trust management component in their design, perhaps in consultation with security PIs in the team, then we would generalize later. This did not turn out to be the case. We learned that one cannot rely on application developers to derive their own trust management; this is a task that must be carried out by a joint effort between application developers, security experts, together with NDN architects, to develop a security framework that uses NDN's hierarchical naming to establish the context and scope for individual keys, we must provide tools to application developers such as easy-to-use library support for signature verification and key management. We discuss our trust management research plan in Section 5.2.

DDoS Mitigation NDN networks do not allow the most common type of IP-based DDoS attack, because the network delivers Data packets only upon request – unsolicited traffic does not make it far in the network. However, an NDN network can be subject to a new type of DDoS attack, namely Interest packet flooding. We have investigated effective solutions to mitigate Interest flooding. We showed that NDN's inherent properties of storing per packet state on each router and maintaining flow balance (i.e., one Interest packet retrieves at

most one Data packet) provides the basis for effective DDoS mitigation algorithms. Our evaluation through simulations shows that the solution can quickly and effectively respond and mitigate Interest flooding [3].

Privacy Preservation The semantic richness of NDN names are a potential privacy concern. We have examined privacy-relevant characteristics of NDN and presented an initial attempt to promote communication privacy. Specifically, we designed an NDN add-on tool that borrows a number of features from Tor; we demonstrated via experiments that it provides comparable anonymity with lower relative overhead [21].

3.6 Community Development and Outreach

A global research community is forming around NDN and related *Information-Centric Networking* architectures. ACM SIGCOMM has hosted 3 workshops on Information-Centric Networking (ICN 2011-2013), co-organized (and in 2013 co-chaired) by PI Lixia Zhang. IEEE INFOCOM has hosted 2 workshop on Emerging Design Choices in Name-Oriented Networking (NOMEN 2012-2013), also co-chaired by Lixia Zhang. Many NDN PIs served on the TPCs for both workshop series, and NDN-related work made the majority among the submissions.

Asia Future Internet Forum (AsiaFI, `http://www.asiafi.net/`) hosted an NDN Hands-on Workshop in March 2012 at Seoul National University, Korea see `http://www.asiafi.net/org/ndn/hands-on2012/`), to promote and facilitate ongoing NDN-related research efforts in the Asia-Pacific region.

One community project that impressed us is the "Smallest Federated Wiki" (`http://www.youtube.com/watch?v=xX22CgG4d18`), an open source project that picked up our NDN.JS implementation to use in building a grassroots NDN-based community-Wiki. The idea is to allow users to publish wiki content as NDN Content Objects over the NDN network and enjoy the benefit of inherent security support and in-network data caching. We hope many more such grassroots efforts will flourish in coming years as we continue building up NDN libraries to lower the bar for NDN application development efforts.

4 Network Environments & Applications

Per the FIA-NP solicitation, we have selected two network environments, **Open mHealth** and **Enterprise Building Automation & Management**, and one application cluster, **Mobile Multimedia**, to drive our research, verify the architecture design, and ground evaluation of the next phase of our project. The two environments represent critical areas in the design space for next-generation Health IT and Cyberphysical Systems, respectively. They also extend work started in the previous NDN FIA project on participatory sensing and instrumented environments to focus on specific application ecosystems where we believe NDN can address fundamental challenges that are unmet by IP. Based on the successful initial results of previous NDN research, we have identified Mobile Multimedia as an application area of cross-cutting relevance, motivated not only by the network environments above but our team's desire to further develop NDN by using it for our everyday communication. To this end, we continue development of a cluster of related applications for communication via audio, video, text, and file transmission over mobile devices such as handsets, laptops, and vehicles.

4.1 Health IT: *Open mHealth*

Mobile health (mHealth) has emerged as both an important commercial market and a key area of Health IT, a national priority. The 2013 mHealth Summit will host over 4,500 participants. Recent surveys suggest there are over 13,000 health-related apps available to Apple iPhone users, and over 6000 for Android users [24]. The Internet's role as a critical enabler of mHealth will grow further over the next decade.

To explore mHealth as a network environment for NDN, our team will collaborate with the Open mHealth project [16] led by Deborah Estrin (Cornell) and Ida Sim (UC San Francisco). Within the many applications of mobile technology to health, Open mHealth focuses on leveraging the public's everyday mobile devices (cell phones, tablets, etc.) to extend evidence-based interventions beyond the reach of traditional care and thereby improve disease management and prevention. For example, mobile applications exist or have been proposed to manage: pre- and post-natal care of mothers [32]; diabetes [66, 73]; everyday activity in stroke patients and others with chronic disease [22]; and community exposure to environmental pollutants [54].

The Open mHealth team envisions that the Internet will interconnect 1) data capture, 2) secure storage, 3) modeling and analytics, and 4) user interface components to create a modular, layered sense-making framework. Such applications will use low-level state classifications of raw data (e.g., estimated activity

states such as sitting, walking, driving from continuous accelerometer and location traces) to derive mid-level semantic features (e.g., total number of ambulatory minutes, number of hours spent out of house), that can be mapped to behavioral biomarkers for specific diseases (e.g., chronic pain, diabetes, multiple sclerosis, fatigue, depression, etc) [25].

For example, Figure 4 shows a network of open components for self-care of diabetes, which affects 25.8M people in the U.S., less than half of whom meet recommended standards, such as blood glucose index levels, for managing their own health. Type 1 diabetics continuously self-monitor blood glucose and insulin levels, and other important factors such as diet and exercise. Many developers are exploring mobile health technologies to assist self-monitoring and diabetes management, since almost all patients have access to mobile health capable technology [66]. But such applications often have proprietary or siloed designs that inhibit data exchange, e.g., data streams from apps for blood glucose and physical activity do not easily integrate, a missed opportunity to provide more comprehensive analysis and coaching to the patient and more complete longitudinal data to providers.

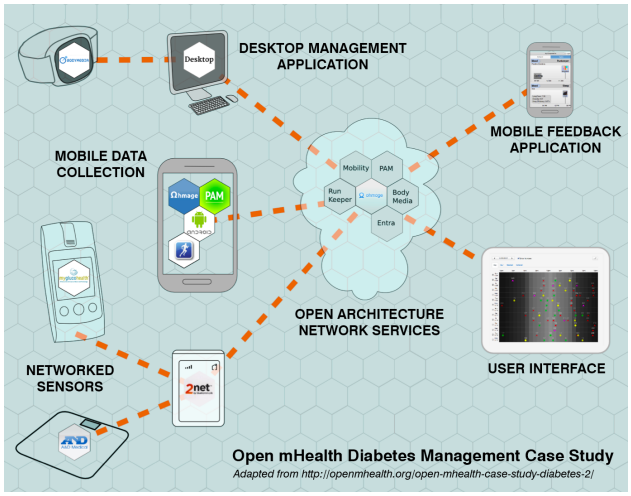

Figure 4: *Networked data producers and consumers in a diabetes management case study from the Open mHealth team, who promote interoperability between mobile health components via community-standardized data exchange.*

The Open mHealth team instead advocates an interoperable, Internet-inspired approach. They propose a thin waist of open data interchange standards that will enable an ecosystem of sensing, storage, analysis, and user interface components to support medical discovery and evidence-based care. In the same way that the Internet's IP layer enabled innovation and interoperability among distributed devices, they believe a common and open approach to mHealth data exchange will encourage the emergence of a market-supported, patient-centered landscape of innovative health applications. Central to this vision is patient-controlled, privacy-aware data exchange across device, component, and application boundaries. The focus on *data exchange as the backbone of the application ecosystem* makes open mHealth an excellent network environment to both drive and evaluate NDN.

4.1.1 Related Prior Work

Within the participatory sensing application area of the previous NDN FIA project, we extended the concept of a host-centric "personal data vault" developed at UCLA and USC [45] to create a geographically distributed *personal data cloud (PDC)* using NDN. This prototype went through three iterations reflecting increasing understanding of how to develop applications using NDN. The first version implemented data collections, key management, storage, data transfer and authentication/setup phases in a way largely analogous to TCP/IP based applications. A second revision integrated the PDC architecture into a deployed participatory sensing application at the Center for Embedded Networked Sensing, called Ohmage [55]. The most recent revision transitioned to use the new Sync primitive for transferring content between entities, and removes much of the session-like semantics initially present[3]. This experience will inform work with the Open mHealth team as they continue to develop pilot applications using Ohmage and similar platforms. With the emergence of the Sync primitive and our recently developed ChronoSync synchronization library [95], as well as lighter-weight mobile client options based on NDN.JS [65], we plan to explore end-to-end applications data dissemination via NDN.

4.1.2 Research Approach

We will partner with the Open mHealth team – both its leadership and developers – to understand the requirements and current state-of-the-art in this network environment, as well as limitations they face from the current Internet architecture. We will pick one or more applications (e.g., diabetes management or post-heart attack health management) that are representative of the envisioned ecosystem, and port existing

[3]https://github.com/remap/PDC-SYNC

software of the Open mHealth team to the NDN architecture. We will use an interactive development process, soliciting regular feedback from the Open mHealth team. As area lead for participatory sensing at the NSF Center for Embedded Networked Sensing (CENS) from 2006-2012, Burke worked extensively with Open mHealth's Co-Founder, Deborah Estrin, on the mobile and server-side applications that form the conceptual and technical foundations of the mHealth software. He will facilitate dialogue between the two teams and direct the development of NDN-based prototypes.

Naming and application design. The Open mHealth architecture focuses on data exchange rather than system interoperability, which is well-suited for NDN. Using NDN rather than IP enables Open mHealth applications to be coded using data naming directly rather than having to create abstractions to translate data names to IP-based hosts providing services. We will explore how application architecture can be simplified and map closely to network architecture. With its emerging synchronization primitives, NDN will also enhance network support for synchronizing data across multiple devices.

Trust and security. Because NDN does not rely on perimeter- or channel-based security, it will promote global health data ecosystems rather than previous walled garden approaches. This shift has direct relevance for public health, by enabling research to draw from large populations.

Storage in the network. NDN naturally supports distributed storage, which can ease the burden of fault-tolerance and load-balancing in large networks, reducing cost-of-entry and fostering innovation.

4.2 Cyberphysical Systems: *Enterprise Building Automation & Management*

The Open mHealth environment focuses on personal data collected by individuals in a self-analytic context. In contrast, our second network environment explores data from the built environment, as used in the enterprise. For our purposes, Enterprise Building Automation and Management covers the intersection of three critical sub-areas: *industrial control systems* (ICS), including supervisory control and data acquisition (SCADA) and so-called smart grid [26], *enterprise networking*, and the *Internet of Things* (IOT) movement [7]. Enterprise BAS and BMS are environments that bring both critical infrastructure considerations of ICS with exciting visions for the everyday built environment of IoT. In this domain, significant engineering challenges have emerged along with the promise offered by the convergence of networking in ICS with traditional IT, a sea change described by NIST in their ICS security review [70].

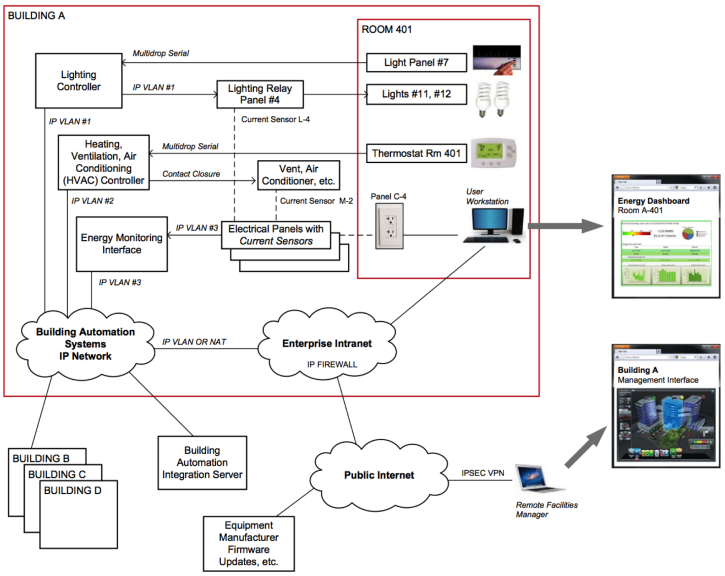

Figure 5: *Example of building automation and management components, including sensors, controllers, and interfaces to both the public Internet and an enterprise intranet. Securely integrating these heterogeneous components, while simplifying application development, is an important challenge posed by this network environment.*

BAS and BMS are software/hardware systems that perform control, monitoring and management of heating, ventilation and air conditioning (HVAC), lighting, water, physical access and other building components. Their distributed, heterogeneous nature leads to a variety of challenges. The IP protocol suite is increasingly used to network their components and as such is now a fundamental substrate of new buildings. However, IP networks suffer from limitations that impact innovation and trust in networked building systems, which we believe can be addressed with NDN.

BAS/BMS pose complementary challenges to those of mHealth. They integrate hundreds to thousands of data collection and control points often implemented by special-purpose embedded devices and managed by a single enterprise. Some devices are mobile and others are fixed; some devices are power-constrained and wireless while others are hard-wired and continuously communicating. Figure 5 shows a few components of a typical system, including thermostats used for adjusting HVAC, lighting control and

energy monitoring.

Security is a fundamental and critical challenge [40]. Like other ICS, BAS/BMS typically employ physical or logical isolation of the network as a primary security measure, which limits interoperability and integration.[4] They also use a mixture of proprietary and open protocols, only a few of which offer intrinsic security. In many networks, there is no intrinsic security at lower layers, and only channel- and perimeter-based security above that, via SSL/TLS, VPNs, and routing configuration. Given the importance of integrating subsystems for applications such as energy management, fault detection, and synchronization, network segregation and closed protocols are not a viable long-term approach. Figure 5 shows an environment that supports access to control and monitoring from intranet web sites, but makes "air gaps" impossible and logical network separation hard to engineer and maintain for normal enterprises.

For application developers, there is a fundamental mismatch in BAS/BMS between how network applications are authored (typically data-centric) vs. fieldbus and IP network abstractions (host- and device-centric). Addressing is spread across many layers, most of which are not routable. There is a great heterogeneity in protocols, especially at the fieldbus level. Typically, application developers and platform manufacturers implement abstraction layers in middleware [63], often complex, proprietary, and challenging to configure.

In addition to the protocols themselves, significant application logic is bound up in network configuration not accessible to developers or users: 1) VLANs, IP subnetting, and routing configurations enforce boundaries between systems; 2) firewall configurations describe brittle rules for system access, which can be difficult to change; 3) keys and certificates for SSL connections and VPNs may identify connections; 4) VPN configuration and enterprise authentication hold remote network access permissions. None of these are typically visible or accessible to application software in traditional systems. In fact, they represent important system control logic that is often replicated ad-hoc in application configurations. A simple example is how an application must be configured to know that 192.168.2.1/24 is lighting and 192.168.3.1/24 is HVAC, which is site-specific and meaningless to an application. Such configurations also bring brittleness to changing topologies and devices.

Our selection of digitally-controlled cyberphysical systems as a network environment is inspired by the practical goal of enabling more efficient operation of buildings, improved comfort and control for occupants, and new opportunities for understanding the interactions between elements of our built environment.

4.2.1 Related Prior Work

This network environment continues work that began in the original NDN project on authenticated lighting control and was extended through an EAGER in 2012-2013 to explore sensing and building management. The current and ongoing research, its results, and our evolving testbed are described in Section 3.1.

4.2.2 Research Plan

UCLA Facilities Management has agreed to act as domain experts and help define the practical requirements of this network environment. UCLA's currently deployed building management system has approximately 150,000 points of monitoring across the campus, potentially growing to over 400,000 points in the next five years. It is the largest installation on the West Coast for Siemens building systems after Microsoft. UCLA's IT, DDC (direct digital control), and engineering staff will interact with the NDN team to enumerate their requirements, challenges, and limitations. As part of the previous EAGER award, UCLA FM has already helped us install a dedicated Siemens electrical demand monitoring system for a laboratory space for research. We expect they will also help us install a dedicated server that will provide near real-time access to 20,000 points worth of data from campus' operational systems, probably about 10 buildings worth of data. We plan to use this server as a gateway to our own NDN testbed.

Naming and application design. The NDN architecture can reflect application knowledge (and needs) directly. Application logic for accessing devices can be captured in naming patterns and cryptographic signatures directly supported by the network. To support BA system integration, we aim to lower the cognitive distance between network protocol details and application requirements, so that complex BAS applications are easier to write, debug, and maintain. For example, an NDN-based BAS would use hierarchical naming rather than addresses and port numbers, removing the need for middleware to translate. We will have to develop consistent, granular, application-specific naming of data sources and control points, to support

[4] In many cases, systems are left exposed to the Internet inadvertently. Only a few searches on `http://shodanhq.com/` make this abundantly clear.

changing topologies and device configurations. NDN-network-based BAS could also make routing or forwarding decisions based on application-level semantics, essentially impossible in IP networks.

Trust and security. The trust model for this environment will emerge from the administrative organization of the enterprise and functional relationships among components. We believe these relationships can be expressed in the data and control namespaces, allowing straightforward trust verification in the applications. Extending our prior work in authenticated control, we plan to develop a system security approach that secures the data directly through cryptographic signatures on data packets and optional encryption of content. As a result, anyone equipped with the right credential (in the trust framework) can securely access, configure, and/or control devices using the same data name from any location in the enterprise. NDN-capable devices can then communicate on the network with authenticated messages, rather than relying on connection-level or physical segregation, or authentication present for user interface only.

Embedded and real-time support. To examine how NDN will work at all layers of BAS and BMS, we must consider embedded and real-time systems, many of which have recently transitioned to IP-based communication. Embedded platform support through both gateways and lightweight stacks for low-capability devices, including "hard" real-time communication when appropriate support exists at lower layers [44, 67].

4.3 Mobile Multimedia Application Cluster

We plan to test solutions developed for the environments above against a few important NDN-based applications that emerged in the first FIA project, as a way to understand the solutions' generalizability and to further the applications' evolution at the same time. These applications already gain significantly from NDN's core capabilities and explore NDN-based content distribution in the context of an important and common scenario: decentralized, multimodal communication and data sharing among mobile users. Interesting in themselves, the applications also provide everyday tools for the project team and building blocks for the network environments.[5]

N-way Media Streaming & File Sharing We will continue to explore peer-to-peer videoconferencing as an application that drives low-latency communication and scalability, aiming to deliver a usable application for the project team via the emerging WebRTC support in modern browsers, e.g., Chrome or Firefox. We will change only the rendezvous and transport, and take advantage of echo cancellation, bandwidth adaptivity (which may require different measurements for NDN), automatic gain control, noise reduction and suppression, etc. This work will incorporate our previous video streaming application [41], multi-user chat [96], and audio conferencing [97] work introduced in the previous section. We will focus on mobile deployments given rapid development in this area (Mozilla, AT&T and Ericsson recently demonstrated WebRTC-based calls via Firefox on a mobile device.[6]). We will also extend and apply ChronoShare, our NDN-based Dropbox-style application (first developed as a project in PI L. Zhang's seminar class) that supports file sharing among a group of potentially mobile users in a completely distributed fashion (i.e., without a centralized server in the cloud). These applications have provided an opportunity to explore both the new NDN Sync primitive (for efficient namespace reconciliation across multiple nodes) provided in the CCNx platform. The resulting image and video libraries will be applicable to "imager-as-sensor" applications in mobile health and surveillance in building management applications.

Networked 3D environments Networked 3D environments represent a crucial building block of simulations, visualizations, games, and educational experiences. We plan to extend preliminary work in supporting multi-player online games with NDN [53] to peer-to-peer Massively Multiplayer Online Games (MMOGs), which require high interactive responsiveness, availability, and security, as well as state consistency across distributed nodes [64]. These requirements call for robust application architectures and efficient synchronization mechanisms. Consider the basic scenario of a player traveling through virtual space, encountering objects instantiated by remote peers. The application must efficiently discover these objects with progressive, prioritized download and synchronization with views of peers nearby in the virtual space. Many IP-based peer-to-peer MMOG frameworks use DHT-based overlays to implement application-level multicast between nearby players [15]. NDN's intrinsic multicast support and name-based routing can improve performance and simplify application development, e.g., names can encode (virtual) locality information through

[5] Consider the increasing use of mobile phone cameras in mHealth for applications such as dietary self-monitoring [56] and teledermatology [43], the application of surveillance and observation systems [83] to building monitoring and access control, the potential for 3D environments in building data visualization, and the role of vehicles in both daily life and the fleet services of major enterprises.

[6] `http://www.ericsson.com/thecompany/press/releases/2013/02/1680640`

techniques such as locality-sensitive hashes [20] or hierarchical spatial representations [62]. Results for this application can likely be applied in other location-based applications; for example, nearest neighbor (k-NN) queries in "virtual space" are a key part of other mobility applications, such as vehicle-to-vehicle communications, described below. Further, we plan to explore the use of these techniques to provide interactive visualization of building management data shared over NDN in our first network environment.

Vehicular Applications As a special case of mobility and intermittent environments, vehicular applications offer a great context for experimenting with network support for information maximization and other mechanisms for prioritizing data transfer. We will explore the exchange of both sensor data and media in support of collision avoidance, traffic monitoring, and location-based notifications [38]. Though their communication is resource-constrained, co-located devices collect largely redundant data, as they often share (and sense) the same environment. Opportunities for information transfer are limited between different clusters of devices, such as cars moving in opposite directions or meeting at intersections. A network utility maximization problem can define utility as a function of delivered information rather than transmitted flow rate; redundant information from different sources would have a lower utility than non-redundant information from the same source. The solution to this information maximization problem will exploit network parameters such as forwarding and cache replacement policy. Our goal is to demonstrate that generic solutions are possible that operate based solely on (similarity) relations among content names, without any semantic interpretation of names or any application-specific knowledge. To experiment with vehicular applications, we shall exploit an existing collaboration with UIUC facilities and services (F&S) department, which owns 100 vehicles available for research instrumentation as part of on an ongoing NSF-funded CRI project (CNS-1059294).

5 Research Agenda for Next Phase

Our research agenda is driven by the requirements of our selected network environments, as well as the need to transform the results into architectural components and an overall NDN system realization. This requires that we address both the research issues raised by the environments described in Section 4, as well as issues that, although not directly related to those environments, are fundamental to globally scaling the NDN architecture. The following sections describe a research agenda for application development, security, and scalable routing and forwarding. We also describe the creation of libraries and tools that operationalize what we have learned about the architecture and support new application development, both for ourselves and the larger community. Finally, we examine social and economic implications of the NDN architecture. We will also continue to support and extend our testbed, simulation, and demonstration outreach activities, although they are beyond the scope of the specific research agenda we describe in this section.

5.1 Applications

Our research agenda on applications focuses on four critical, cross-cutting areas, motivated by our previous experience and the requirements of the network environments: 1) naming and application design patterns; 2) rendezvous, discovery, and bootstrapping; 3) in-network storage; and 4) data synchronization.

Naming & Application Design Patterns Namespace organization is at the heart of NDN application design. A primary research task will be to develop proper naming structures for the environments, generalize them where possible, and test against both increasingly sophisticated application requirements and entirely different use cases. Such iterative namespace design will explore key research questions. For example, given Open mHealth's patient-centric focus, we will begin with namespace design in which prefixes reflect patient identity, such as `/healthvault/`⟨`patient_id`⟩, where `healthvault` is the prefix of a compatible storage provider. This has limitations: it will require network mobility support for patients to publish consistently to such namespaces as they move and obtain connectivity in different ways. A simple solution could be through a mapping service (e.g., from their `healthvault` name to `/verizon.net/`⟨`username`⟩ or `/ucla.edu/client/`⟨`client_id`⟩), a research topic for NDN routing group discussed further in Section 5.3. Further, anonymization of patient identity may require *name* encryption. Meeting HIPAA regulations in communication with professional care providers will require *data* encryption. Exploring solutions to each of these requirements will occur in collaboration with the security group's research, discussed in Section 5.2.

Seeking further benefit from NDN, as we build prototype applications, we will think outside the box of

today's default patterns, considering data exchange in a future "world running NDN" instead[7]. For example, Open mHealth can be seen as a data distribution challenge in which network processes exchange filtered subsets of data traceable back to and controlled by the patient. Rather than mapping this to an assortment of client-server systems that interoperate on a connection-by-connection basis as is done by the Open mHealth group currently, we plan to implement all communications directly on NDN, with research to select appropriate cryptography techniques and naming approaches for data aggregates. Not only could this better support patient control of their own health data and system interoperability, but also enable wholly new opportunities, such as efficient participation in population-based studies that access the entire data space [8], something extremely difficult in today's connection-oriented HTTP/TCP/IP architecture.

We will also explore the interplay between naming, application design and routing/forwarding. When the network uses names meaningful to the application for forwarding decisions, it can directly satisfy application-level semantics; e.g., it is straightforward to configure core routers in an NDN BAS to forward appropriately signed data and interests in the namespace ⟨`enterprise_root`⟩`/bas/`⟨`building`⟩`/lighting/`⟨`room`⟩`/`⟨`light`⟩ to the correct edge router based on building and subsystem (`lighting`) information in the name. We will experiment with such environment-specific configuration in Year 2 to meet the practical needs of integrating a gateway to the existing UCLA BAS/BMS with the NDN testbed.

We will use environments and applications to test what we observed in the previous FIA project: that NDN's value is most evident when 1) namespaces map to specific strategies and needs of applications, 2) communication is redesigned rather than replicating the approaches of IP, and 3) network configuration is organized using application-generated namespaces. Our goal is to establish a palette of specific ways that NDN can enhance development of complex, secure distributed applications.

Rendezvous, discovery & bootstrapping Before processes can participate in an NDN application, they must often perform rendezvous, discovery, and or bootstrapping functions. NDN application and routing groups will work together to develop solutions to enable a device to learn its own names or prefixes and those of data that it wishes to access. For example, instances may need to 1) obtain a globally routable prefix, 2) inform the network of which prefix they wish to publish in, 3) discover the names that other authorized consumers will use to request content or side effects from them. Because a node does not require a name to issue an Interest for data, this can be done as soon as connectivity is available. We plan to use predefined, locally scoped names, e.g., `/local/discover/device/light` or `/broadcast/mhealth/commons`) to obtain knowledge necessary to communicate. As suggested in Section 4.3, we will explore how established techniques such as hierarchical spatial partitioning and locality-sensitive hashing may prove useful in mapping higher-dimensional features and indexes into NDN names, simplifying discovery by enabling Interests to directly express feature similarity or virtual/physical location.

In-Network Storage We will explore how NDN's incorporation of storage into communication impacts application design. As introduced earlier, "repositories"–persistent data stores that respond to Interests in specific namespaces–have emerged as fundamental components of most NDN applications, and may often replace file systems or databases. To expand on previous work, we will develop a more complete semantic definition of what storage in the NDN architecture means, and how repositories are distinguished from other applications that answer Interests. The proposed network environments require persistent storage of highly granular and often sensitive data as a key feature; we will explore the requirements this places on repository design and implementation. We believe the architecture should provide basic storage primitives, as well as potentially offering standard methods for applications to discover whether they can write to a persistent store in a given namespace, validate that they trust who is providing the storage, and enumerate basic capabilities of different stores. Additionally, options for local APIs, configurability, and performance optimization need to be explored[9]. We plan to explore different storage models such as: 1) *NDN native,* in which the API to repositories is built solely on Interest/Data exchange. 2) *File system,* in which a proxy utility or component of the network stack makes available existing hierarchical file systems using NDN. 3) *Database,* in which local access is through a database-like library interface. An important research challenge is to explore how to

[7] E.g., a client-driven approach and timecode-based data names enabled session-less random access in video streaming [41].

[8] Imagine "donating a decryption key to science", enabling access to de-identified subsets health data for a genome-wide association study, without requiring centralized data storage as needed today.

[9] Previous video streaming work generated several challenges for the Repo, including the need for very efficient packet writes, signing and verification as well as large storage capacities and "rollover" when storage limits were exceeded.

express common queries directly in Interests, enabling NDN repositories to replace certain databases, such as key/value stores, and provide intrinsic scaling and redundancy more easily than conventional solutions.

Data synchronization We will continue evaluation and design of the synchronization (SYNC) primitive based on the needs of the environments. We expect the design to mature through use in solving real problems, and foresee that it could play a role in the NDN architecture that is similar to TCP's in the IP architecture, helping bridge the gap between applications' need to synchronize distributed datasets and the datagram fetching provided by NDN network layer. Both BAS/BMS and Open mHealth will use NDN namespace SYNC to obtain a "picture" of the data available (via their names) and then selectively retrieve data objects based on what is expressed in those names. For BAS/BMS, this might be to retrieve data named under a "electrical demand monitoring" subtree `.../emon/`, for example, while in Open mHealth, it might be to synchronize a list of patient free-text journal entries, `/healthvault/⟨patient_id⟩/free-text/`. We will flesh out what the network primitive should do in our environments in order to determine whether there is only one "SYNC" or different, application-specific synchronization schemes. As an example, UIUC's mobile and vehicular applications desire to maximize information coverage by multiple nodes observing the same event ("shared situation awareness"). From this, the idea of prioritizing synchronization has emerged: If names reflect similarity between objects, it is possible to design generic protocols that achieve approximate information maximization as a function of similarity relations among content names (such as length of their common prefixes). We have already developed a set of information-maximizing caching [74, 23] and data transport [82, 74] schemes that test the name-based information maximization hypothesis. They serve as a proof of concept that naming data allows formulating and solving network utility maximization problems involving information-based utility functions, but it is not clear yet whether this requires additional support in the architecture, or can be achieved through application-level naming conventions.

5.2 Security and Trustworthiness

NDN "collapses" communication layers traditionally separated in the TCP/IP Internet, through its use of data names and signatures in every (data) packet. In this way, as introduced in Section 4, NDN provides *application security* by enabling fine-grained *data security* at the "collapsed" communication layer, rather than through *channel-based security* approaches implemented above the network layer as in IP. Our security research agenda is informed by three challenging aspects of this vision faced in the previous FIA project: 1) *key generation, distribution, and revocation* tools need to be available and easy to use in order to incorporate security into realistically distributed test applications, rather than limited evaluations on a single host; 2) standardized support for flexible *trust models* should be available to steer developers down architecturally appropriate and secure paths, guiding the development process and avoiding common security pitfalls; 3) similarly, standardized support for *encryption-based access control* is necessary to build applications that have expectations of confidentiality. Working through these "chicken-and-egg" challenges in the initial FIA project—by concurrently writing applications, exploring security implementations, and checking the fit between the two—has prepared us for the work proposed here. We will transition existing prototype security code into a common security framework (set of libraries and tools), on which we will build applications for the proposed networked environments and evaluate the fit, with initial stages of this work already underway.

We believe that the security framework can be built on well-known cryptographic techniques, and we will focus the majority of our effort on mapping these techniques to the needs of NDN applications, so the security and trustworthiness of our approach to the network environments can be tested in practical designs and running code. We wish to minimize common errors by developers in typical application scenarios while guiding developers toward data-centric security design. Additionally, we will continue research to address emerging security challenges of a future widely deployed NDN architecture.

5.2.1 Trust and Security Building Blocks

Key Management To be able to develop, test, and run trust management features, we must first provide simple and easy-to-use tools to generate, sign, publish, store, fetch, and revoke keys. While NDN supports key/certificate fetching in exactly the same way as fetching any other type of data, the main research and engineering challenge is the proper key certification, publication, storage and revocation designs. For application prototyping on the NDN testbed, we have developed a number of simple tools [10] to generate public/private key pairs, issue public key certificates, and publish them in globally synchronized repositories. These tools are just a first step, suitable for our testbed (e.g., supporting only RSA keys and limited key

publishing authentication), and currently work only within the specific hierarchical trust model described below. We will expand this work and develop, as part of an NDN security framework, more flexible general-use APIs and user-friendly tools to more completely address key management issues. For example, for key publication and storage we will test the utility of the recently developed NDNS [6], a DNS-like distributed database for NDN, which combines the flexibility of the hierarchical namespaces with strict control over publication to a specific namespace. Only data (keys) signed by the namespace (zone) owners can be published and stored in the specific NDNS namespace.

Trust Management For NDN applications and application developers, we envision trust management as the primary interface to the security framework, providing logic that organizes use of various types of cryptography and its relationship to data namespaces, to provide provenance, encryption-based authentication, and data access control. That is, each application instance needs to be able to obtain certified keys of his or her communication parties, as well as securely validate these keys based on some trust model. We will start with a simple and specific trust management solution that suits the network environments, generalizing at later stages. We will extend a simple hierarchical trust model recently deployed on the NDN testbed [10], which is now used to secure several developing NDN applications, including the routing protocol [35] and file sharing application [94, 42]. In this trust model, each data packet—including all keys, which are also data packets—needs to be signed by a key with a name that is prefix of a data name. This way, names can be directly used to authorize use of keys to sign specific data packets and limit the usage scope of the keys. We believe this approach may directly apply to some applications within the BAS/BMS environment and plan to test it there first. For example, `/ucla/bms/melnitz_hall/electrical_demand` should be and can be signed only with a key that has name `/ucla/bms/melnitz_hall`, `/ucla/bms`, or `/ucla`, while the key with name `/ucla/bms/melnitz_hall/electrical/2013-07-02` is eligible to sign only electrical demand measurements produced on July 2, 2013.[10]

While the deployed hierarchical trust model works well for our routing and file sharing applications and perhaps some of the network environments, it is not expressive and flexible enough to reflect trust relations for all possible applications. We will build on available key management tools to develop a certificate validation engine that can support customized and distributed trust models. This engine will enable application developers (and perhaps end-users) to specify their specific trust management policies and rules, which will be used to automatically validate public keys and application data.

We also plan to initially implement a cross-certifying model (SDSI) [59, 33, 2], which provides more redundancy of verification, allowing data and key names to be independent of each other, which easily accommodates any trust relation in the real world. Based on the experience gained though development of hierarchical and SDSI trust models, we will further generalize trust management relations, exploring how best to define: 1) naming conventions and data formats to help application developers easily express their own trust models, 2) standards to express trust rules and trust policies, 3) semantics and formats of certificates in terms of NDN data packets, 4) interfaces for validation engines, as well as 5) reference security implementations and documentation.

Encryption-based Access Control To support data privacy and access control, we will develop an encryption library built from standard cryptographic implementations as an additional NDN security building block. The basic mechanism of encrypting NDN data with the public key of authorized requesters will be coupled with a variety of encryption mechanisms to accommodate different security and privacy requirements. We plan to incorporate into this library more complex privacy models relying on advanced encryption techniques, such as group and broadcast encryption [39, 27] that, among other benefits, may preserve caching benefits of NDN otherwise lost in more naive per-consumer encryption.

The access control library will also need to provide capabilities for further enhancement of privacy protection by supporting NDN Interest encryption, first prototyped in [13]. An important building block for this library is encryption of the data name using the publisher's public key, so that only the data publisher can tell exactly what data is being requested, exposing to the public only a prefix part that is used to guide Interests toward the producer. To support application and user privacy, we will incorporate and improve a Tor-like interface, previously prototyped [21], that hides requested data names from unauthorized third-parties.

[10] Note that the actual key names need to contain additional components to identify the specific version of the key, e.g., `/ucla/bms/key/⟨key-id⟩/...`, but these can be transparent to the trust model.

5.2.2 Environment-Specific Security

Applications for each network environment will use these building blocks, along with scope control in routing and forwarding, to achieve their basic security goals. The environments will also prompt deeper security challenges that can benefit from techniques that are less common or emerging state-of-the-art.

Open mHealth With genome sequencing and other health data collection likely to begin at birth, the mHealth environment should aspire to provide a human lifetime of secure data storage, which will not be achieved by basic encryption techniques. We plan to review challenges and possible solutions to long-term secure archival storage, which have been widely studied [12] but have become more prominent in NDN, where long-term data availability is a practical reality. On a more immediate scale, we will examine the design challenges for protecting anonymity in Open mHealth's ecosystem of multiple interacting applications, each using descriptive (but potentially encrypted) names for data. Relevant research includes both a variety of privacy-preserving data publishing techniques, as summarized in [28], as well as guidelines for secondary uses of health data such as in [60].

Building Automation and Building Management Systems Our focus will be on Enterprise BAS/BMS like those used on university campuses, which have vital business implications but do not represent so-called critical infrastructure. Important additional challenges may include security approaches for low-capability devices and low-latency feedback control loops implemented over the network, which could be informed by previous applications of pairing-based cryptography and elliptic curve cryptographic in resource constrained nodes such as sensor networks [48, 72]. At the network level we believe tradeoffs between the expression of capability information in namespaces vs. in certificates may generate design challenges unique to NDN, and plan to explore this even as the implementations described above are underway.

5.2.3 Addressing Future Security Challenges

Given the paramount importance of security to a future Internet architecture, we have also established a Security Advisory Council of esteemed security experts (initially Matt Blaze, Bill Cheswick, Steve Bellovin, and George Kesidis) to participate in meetings and retreats and give feedback on design and implementation decisions, as well as to identify unique challenges caused by NDN and possible solutions. They will provide high-level as well as systems-level advice and ideas on trust management schemes for different applications, including naming conventions for crypto keys, and integration of key usage and management into library APIs for application developers.

5.3 Routing and Forwarding Strategy

We propose to develop and deploy an **inter-domain** routing protocol that supports Internet-scale deployment of NDN with the following properties and capabilities:

- Scalability: the protocol can scale to a large topology and a large number of name prefixes. Assuming a mapping system is used to convert user/application names to ISP names (Section 3.2), the protocol needs to scale, at a minimum, to the number of ISP networks and their name prefixes;
- Security: every node can verify the authenticity of received routing information;
- Multipath: the protocol can provide multiple paths to each name prefix if redundant paths exist; and
- Policy support: network operators can express and apply routing policies to influence traffic paths.

We will investigate three candidates for NDN's inter-domain routing: path-vector, link-state, and hyperbolic routing. Each approach has its own strengths and weaknesses in addressing the above issues. Our past experiences will guide us in this investigation and development. We have conducted extensive studies on the path-vector routing protocol BGP (Border Gateway Protocol [57])(e.g., [93, 52, 92, 79, 80, 91, 77, 51, 90, 89, 49, 88, 78]). We also have a solid understanding of link-state and hyperbolic routing in NDN based on the development of NLSR [35]. Below we first briefly discuss the pros and cons of the three potential approaches to inter-domain routing in NDN. We also present the research goals for each approach if it is adopted. We then describe the routing security model that we will explore.

The *inter-domain path vector* approach is most similar to today's Internet interdomain routing protocol BGP. BGP supports a wide range of routing policies, but has several issues: (1) it computes and propagates a single path to each destination (we would need multiple); (2) it converges slowly during routing instability (which NDN's adaptive forwarding plane may avoid altogether); and (3) it has no working solution for routing

security despite years of research. The *inter-domain link state* approach has several advantages over path-vector routing, although NDN's adaptive forwarding plan may lessen their importance: (1) it separates the topology from name prefixes, which reduces routing churn when a link fails (only the topology, not the name prefix, needs to be updated); and (2) it converges faster. Unfortunately, link state protocols also require ISPs to disclose private business relationships with their neighbors; promising previous research on inter-domain link-state policy routing such as IDPR [68, 69] may help us find a viable compromise between information disclosure and policy support.

The third candidate approach is *inter-domain hyperbolic routing*, which can greatly increase the scalability of the routing system as there will be no FIB (or a small cache for storing recently used routes). Extending hyperbolic routing to an inter-domain scenario requires: (1) a mapping system to convert name prefixes to hyperbolic coordinates, one of which we have developed for converting user/application names to ISP names (Section 3.2); (2) a forwarding plane that supports greedy forwarding based on coordinates carried in Interest messages. The biggest research issue with this approach is which routing policies can actually be supported by hyperbolic routing, since routers cannot express policies in their coordinates and they do not have full topology information to apply policies. On the other hand, there are no routing advertisements to authenticate, so the routing protocol is secure as long as we can secure the mapping system.

Security The path-vector or link-state approaches will require a way to secure routing updates, using NDN's built-in data security. We will name every element in the routing system (from ASes to router interfaces), and associate names with public/private key pairs. Routers verify the provenance of routing messages by verifying their signatures using trusted keys. Public keys and their certificates are distributed along with the routing messages that they sign. However, our design still needs to address the following issues: (1) how to name various players in the routing system? (2) what is the trust model to ensure that keys are authentic? Since the relationship between routers, routing processes and routing data is inherently hierarchical, we expect to use a hierarchical naming scheme for them consistent with the naming scheme in NLSR. For the trust model, instead of mandating global trust in a single entity as in most PKI designs, we will leverage the contractual provider-customer relationship to locally control and administer trust.[11]

The research of forwarding strategy will focus on the design, implementation, and deployment of strategies customized for specific purposes or network environments. Our existing work has developed a general strategy that was shown in simulations to be superior over IP networks. The next step is to develop strategies that can take advantage of specific environment contexts, support particular goals, and be integrated under one extensible framework.

Forwarding Strategy and Mobility Support Our research on forwarding strategy (similar to IP routing policy) will focus on the design, implementation, and deployment of strategies customized for specific purposes or network environments, and then attempt to integrate them under a common framework. Many local area networks (LAN) differ from typical ISP networks in that they may not run a routing protocol but prefer zero/auto configuration, and can afford occasional network-wide broadcasts. We plan to support LAN environments by developing a self-learning strategy, similar to Ethernet bridge's self-learning, that uses local broadcast to learn where to forward in the absence of any routing protocol or configuration.

To effectively support mobile devices, we will use a combination of wireless broadcast, custodian service, and the conventional map-n-encap approach we are using to support routing scalability. When a mobile node needs to fetch data, it starts by broadcasting an Interest locally. Any nearby node N with the requested data or who knows where to forward the Interest (such as a router), will send a reply (or forward the Interest) after a short random wait to minimize reply collisions; if no answer is received within some reasonable time, the mobile node can query a mapping service to get the *Forwarding Hint* (FH) for the namespace in the Interest, and include this FH when retransmitting the Interest. The forwarding strategy will keep recent data fetching history to learn what data sources may be locally available, as well as the learned FHs, until their lifetime expires. We can use DNS to implement this mapping service within the same system used for routing scalability (Section 3.2). Mobile devices can keep DNS servers updated on

[11] A provider's AS key can certify its customers' AS keys. If two customers share the same provider, they will be able to verify each other's key using the provider's key. If two customers share the same grandparent provider, they can verify each other's key since they know that the same grandparent certifies the keys of their parent providers. In other words, any ISP can be the trust anchor for the customer tree rooted at this ISP. Tier-1 ISPs would sign each other's key in a pairwise way, so that networks in different trees can verify each others' keys.

their whereabouts by dynamically updating their *Forwarding Hint* resource records[12]. NDN's built-in data authentication automatically provides crypto protection for both dynamic FH updates and DNS queries.

Our research agenda on forwarding strategy is motivated by open issues from our existing work as well as by the needs of our two network environments. In Open mHealth, most data will be collected from, and possibly stored in and shared across, mobile devices, so support for device mobility is crucial. In contrast, the BAS environment emphasizes collection of and access to data produced by many sensors, as well as controlling a large number of actuators. Our research focus will be efficient self-learning strategies to minimize manual configuration in establishing connectivity among nodes as well as necessary context in which sensors and actuators can be properly named and controlled.

5.4 Scalable Forwarding

Our research agenda in scalable forwarding will focus on supporting real-world deployment, evaluation, and adoption of NDN. We will continue our efforts in the development, evaluation and release of an operational, scalable forwarding platform. WUSTL will develop, maintain, and support the software routing platform at the heart of the NDN testbed and routing/forwarding research. WUSTL, UCLA and UIUC will collaborate to ensure that the platform supports and is responsive to evolving application needs. WUSTL and UCLA will jointly ensure integration between client library development (Section 5.5) and the core platform.

5.4.1 Core Forwarding Node Design, Algorithms & Data structures

The NDN forwarding plane needs to support fast packet name lookup, intelligent forwarding strategies and effective cache replacement policies. Our initial focus has been on fast packet name lookup in particular since name-lookup rates directly impact the scalability of the forwarding plane. To deliver a scalable platform, as our design ideas are supported by research results, we will migrate our enhancements into the NDN-curated code base. To begin, we will integrate our recent work on data structures (Section 3.3) into an operational platform. Our measurements have shown that the peak throughput of the reference implementation is much lower than the 1 Gbps link rate that we consider a minimum requirement for software platforms. The challenges of designing a scalable forwarding plane derive primarily from scalable forwarding of variable-length names and the read/write nature of packet forwarding.

Notably, we have devised a forwarding structure whose size is dependent upon the information-theoretic differences between rules in a ruleset, rather than by their lengths as is the case with all IP-inspired LPM data structures. This approach provides a compact representation of forwarding prefixes which enables fast real-world software implementations by keeping data structures in cache and main memory. For example, a Web-based namespace with 1 million prefixes requires less than 6 MB, and can comfortably fit within the on-chip caches found in server CPUs. This recent work (publication under review) has resulted in the design of a single forwarding node that can sustain forwarding performance of 1us per lookup in traditional servers and 20ns in ASICs, while supporting up to one billion prefixes in the forwarding information base (FIB). Furthermore, we have designed a distributed forwarding plane, which makes use of multiple individual nodes, that can sustain natural multiples of these rates. We have validated this work at the level of data structure requirements, but as we illustrate in [86], a functioning NDN node must ensure that the FIB, pending interest table (PIT) and content store (CS) data structures sufficiently support and complement the several ancillary tasks and support data structures that are currently active in our reference implementation. We plan to explore the effectiveness of our new data structures through systems implementation and evaluation in the context of an operational platform.

Distributed router design Future namespaces may have resource requirements that exceed the capabilities of a single system. Our distributed forwarding plane design integrates smaller routers in order to scale up. We consider three router granularities: forwarding engines within a router, multi-routers in a rack and distributed routers in a network. To be clear, our distributed router design aggregates multiple individual routers together to support larger namespaces and higher overall throughput. This is not a distinct software platform. Our distributed design is enabled by our novel data structures, and so our aim in this project is to develop, evaluate and release the code necessary to operate an NDN router in distributed fashion.

Name Encoding At present, a significant portion of packet processing time is spent on packet decoding, which is unusual for high performance router design, where string lookup is the bottleneck. Our reference

[12]A similar solution was successfully deployed in the late 2000's to support Apple's MobileMe service with millions of users [71].

platform employs a complicated XML format to encode packets, which is good for network architecture and protocol design as it is very flexible, but it slows down the entire data plane. There are two ways to solve this problem. The first way, which we think is very challenging, is to develop a packet decoding algorithm that can decode XML encoded packets quickly and efficiently. A simpler way is to define a new packet format that favors fast packet encoding and decoding. For instance, a simple and efficient packet format could be similar to IP packets, where each field of the packet has fixed length. We are aware of at least 4 distinct proposals for new name encodings for NDN. Our aim is to evaluate each in our platform, and to use experimentation and data to inform a name encoding consensus amongst the community.

5.4.2 Closely-coupled Subsystems

Our experience in scalable forwarding has revealed that the NDN strategy layer, repository, and synchronization service require greater modularity. We (and others) have discovered the importance of these subsystems, along with their serious limitations. For each, we believe that modular integration with the core forwarding node logic is the most significant and important design and engineering challenge.

Modular strategy layer In NDN, the strategy layer decides which output interface to use when multiple options exist. Currently, when a forwarding decision yields multiple valid output interfaces, the current forwarding strategy chooses only one to use with a bias toward the best performing interface (i.e., the one with lowest average Data delivery time calculated over a window) based on recent experience; the best performer is chosen most often, but all alternatives are used periodically to help ensure that performance estimation is accurate over time. This clever and surprisingly robust strategy has been shown to work well in both local and wide-area contexts. But recent work has explored ideas of how to adapt forwarding strategy to different context and environments, and we want to develop and add a modular interface to NDN core forwarding node for implementing and invoking strategies.

Sustainable Repository The NDN repository has emerged as a critically important component in the architecture, providing significant non-volatile storage to NDN nodes. From storing public keys and configuration information required at system boot, to storing 10s or 100s of gigabytes of files and videos, the repository is now considered a core NDN component. Unfortunately, the current repository design (similar to a log-based file system) has critical limitations, most significantly lack of support for delete operations and substantial write performance limitations. There is an entanglement between the repository design and the current synchronization primitive which has increased the consequences of making changes in the near term. For the NDN-NP project we will design and release a sustainable repository that both better supports application storage needs and synchronization.

5.5 Library and Tool Development for Application-Driven Research

Our first three years of research showed that it is crucial to develop and release libraries and tools, created based on application development experience, that promote ongoing team and community experimentation with NDN. This effort is a necessary part of identifying application needs and design patterns and then testing solutions; it also has an impact on protocol design. We plan to expand such development and support of libraries and their corresponding documentation to encourage individuals, companies, research groups and our entire project team, especially those who may be only part-time NDN app developers for now, to write innovative, experimental NDN applications.

Reference implementations. Significant effort, including full time staff developers at UCLA, will be put into evolving the NDN reference implementation and other libraries. We will focus on making them easier to use for both applications and architecture researchers, as well as incorporating new features developed across the various groups. We will fork PARC's open source CCNx implementation and continue to provide new open source bug fixes and enhancements, as well as developing the following reference implementations:

- *Client libraries / APIs.* Based on our experience developing with CCNx and building the NDN Simulator [4] and other "clean-slate" implementations, we plan to build a portable C++ library that is compatible with the CCNx packet format. The library will decouple the API from the wire format, not currently the case in CCNx, and provide modularity that makes it easier for architectural experimentation. We also will create a parallel Java implementation to support development on Android mobile devices for the Open mHealth environment. Design of these libraries will provide experience from which to prototype a lightweight

C or C++ library suitable for embedded devices in our building automation and management network environment.

- *Trust management and security libraries.* The applications, architecture and security teams will co-develop libraries and tools for key generation, signing, publishing, and revocation, as well as exploring generalized support for common trust models expressed in naming and implementation of encryption-based access control and name encryption for our specific environments and applications.
- *Synchronization and other new primitives.* We will develop libraries to explore the fit of different SYNC approaches and features to our applications. While the simplest synchronization primitive between two nodes is of the form "give me all content you have that I do not have", application performance may be improved by optimizations such as prioritized content exchange and preferential expiration/deletion.[13]

Prototype facilitation. The first FIA project demonstrated that experimentation by students and the research community can be further fostered by providing NDN support in popular and easy to use languages such as Python and Javascript. Within a few months of making the NDN.JS Javascript library available [65], we had our first community-built application, a peer-to-peer Wiki prototype by open source developers that we have never met. Our own students quickly created user interfaces for NDN-based file sharing, sensor visualization, and routing status. We would like to enable much more of such experimentation. By providing better documentation, examples, and bindings for popular programming languages, we can advance our research and test our ideas effectively within our own team and in today's "maker culture". We will continue to support Python and C# (for the Unity 3D engine), and will explore support for other languages that would specifically benefit our target network environments. Additionally, the web is both a crucial application domain – a cornerstone of the Open mHealth vision – and a ubiquitous, well-known user interface platform for applications like BAS/BMS. We will expand research and development support in using NDN in browsers and web-style content publishing.

Autoconfiguration. To simplify internal use of our NDN-based applications, we have already developed (in an ad hoc manner) preliminary node autoconfiguration tools as well as package deployments (e.g., via MacPorts) that include common tools and applications. We plan to re-examine the existing tools systematically and to fully incorporate trust management into autoconfiguration, to support increasing use of test applications, including by our partners in the network environments.

5.6 Social and Economic Impacts

During the first phase, the project team collaborated with Values-in-Design researchers to identify values embedded in the developing architecture, and their implications for social issues such as free speech, user interaction and trust, privacy, law enforcement, network neutrality, and data management. These will be published in forthcoming articles, including a collaboratively authored piece called "A World on NDN." As described in the management plan, we will continue to consult with the ViD group in this next project period. We plan to explore manifestations and implications of four values that represent a distinct departure from TCP/IP: 1) a default toward publication of data; 2) an emphasis on the provenance of data; 3) intrinsic support for decentralized communication; and 4) an emphasis on semantic classification through naming at the network layer. Specifically, our applications team will perform an empirical analysis of data naming strategies in Open mHealth and BAS/BMS, to understand how existing practices might impact semantically-meaningful naming practices in NDN. We will also examine how these environments' privacy and security requirements can best be reflected in naming practices, routing protocols, and network storage polices. Finally, we plan to investigate social questions emerging from more general architectural choices, such as how prioritization decisions in data-centric routing protocols could impact network neutrality–for example, if optimizing solely for the most popular content encourages a "tyranny of the majority" or "filter bubble" [50] that dampens our goal of not only a more efficient, reliable and trustworthy Internet but also one that more fundamentally enables democracy and equity of information access.

[13] For example, when two mobile devices meet, the encounter time might be short. The order in which content is to be exchanged can have impact on maximizing information flow. We plan to explore when general policies could be helpful, such as most recent first, most diverse first, or most corroborated first. A flexible library of common prioritization techniques could significantly facilitate application development.

6 Evaluation Plan

Our criteria for success are rooted in the need to generalize from specific implementations and applications to the core network architecture and its ability to serve a wide range of other environments. Per Wroclawski's discussion of architectural evaluation at the Spring 2013 FIA meeting [84], we believe evaluation has at least two steps, the first of which is to clarify what the architecture actually is, including its principles, as opposed to artifacts of system development. The second step is evaluation itself. We plan an experimental approach that includes a mix of use case, emulation, simulation, and actual deployment, as well as organizing student red teams to stress/attack the implemented systems. For example, this spring UCLA began graduate student red team attack efforts on NDN lighting control systems.

Primary evaluation criteria. In our selected environments and applications, as well as a global Internet, the NDN architecture should enable:

- Emergence and evolution of (name-based) data exchange standards from within practical, use-case driven approaches rather than top-down specification.
- Interoperability and feedback loops among heterogeneous personal, enterprise, and global systems.
- Communication patterns not previously well-supported by IP, such as large scale peer-to-peer data exchange, and efficient information distillation from large data gathering networks.
- Persistent, fault-tolerant, secure storage for a lifetime's worth of data–whether for a person or a building.
- Granular privacy and confidentiality of data, managed at the data source, with corresponding trust models and security implementations.
- Intrinsic security in low-level protocols, rather than reliance on physically or logically isolated networks.
- Embedded systems through both gateways and lightweight stacks for low-capability devices, including real-time communication where appropriate [44, 67].
- Robust, simple solutions to mobility, disruption tolerance, and adaptation to network conditions.

We will perform side-by-side comparisons of existing IP-based systems in the selected environments and NDN prototypes based on these criteria. Many of these criteria incorporate or suggest a notion of "fit" between applications and the architecture. In addition to quantitative measures typically used by the networking community, our starting point for establishing measures of such "fit" from the application perspective will be Green and Petre's *cognitive dimensions* framework [30, 18], applied first to notation and programming languages but extended to many other domains. The framework was intended "to improve design practice by making it easier to talk about design usability at an appropriate level of abstraction" [31]. We thus expect it to be very useful in formalizing application-based evaluation approaches. We plan to undertake structured interviews and surveys of users both in our selected environments and the community of NDN application users. We will differentiate feedback that is system-specific from feedback related to the core architecture itself, and integrate the resulting insights into our final report.

7 Education and Outreach

A key objective, and opportunity, of the NDN project is to teach students *architectural thinking*, a type of *computational thinking* that encompasses system principles, invariants and design trade-offs. Contrasting IP and NDN as architectures offers a way to teach students about architectural evolution, from individual components to the larger social context of communication networks. We plan to carry out three specific tasks in support of this goal: (1) incorporate NDN material into courses at all participating campuses; (2) develop and publish NDN introductory material to promote broader understanding and educational impact of the architecture and applications; and (3) continue our project-wide seminar series on architecture discussions, including comparing NDN with other proposed architectures. We have produced a variety of educational material (see `http://www.named-data.net/education.html`). Most NDN collaborating institutions have already incorporated NDN and other FIA project material into classroom teaching. We will continue to publish our conference, tutorial, and panel presentation content on our web site.

Outreach We have undertaken several outreach directions described in Section 3.6, which we will continue and expand in the next phase. We plan to offer NDN tutorials at conferences such as SIGCOMM, INFOCOM, and ICNP. We also plan to host annual "World on NDN" community workshops, for participants from around the world to share their experiences using NDN. CAIDA has extensive experience hosting such events and will either act as a host or be closely involved with the organization. We will continue to

engage industry, including regular contact with Huawei and Cisco to exchange ideas about implementation issues in the architecture. Cisco (David Oran's group) is already porting NDN to one of their router platforms. There is an emerging collaboration with Panasonic Research and initial dialogues with the Cinegrid industry consortium and NTT, spurred by WUSTL's 2013 demo of NDNVideo in Beijing [19].

8 Broader Impacts

The NDN architecture builds on lessons learned from the success of the IP architecture, preserving principles of the thin waist, hierarchical names, and the end-to-end principle. The design reflects a recognition of the major shift in applications communication model: from the "where" (*i.e.*, the host/location) to the "what" (*i.e.*, the content). Architecting a communications infrastructure around this shift can radically simplify application designs to allow applications communicate directly using the name of the content they desire and leave to the network to figure out how and from where to retrieve it. We summarized the differences between IP and NDN in Section 3 and Table 1. NDN also recognizes that the biggest weakness in the current Internet architecture is lack of security, and incorporates a fundamental building block to improve security by requiring that all content is signed.

The success of new architectures requires broad community involvement and uptake. NDN has built significant momentum already by adhering to an open source platform that many have deployed and are experimenting with, either locally or as part of larger testbeds. In addition to our own active education and outreach activities that we will continue (Section 7), our commitment to an open source model has spurred substantial research activity in both architecture and current implementation. Project members are often invited to present at "future Internet" meetings around the world, and we have performed two high-visibility demos of NDN's ability to handle large scale distribution. Industry is also showing increasing interest in NDN; e.g., Cisco now has a dedicated NDN implementation group and participates in many of our meetings, while Panasonic Research is actively collaborating on video streaming research.

NDN has also significant impacted our students, yielding several current Phd theses on NDN topics. Several students have gone to industry internships, and many are involved in interdisciplinary research. Undergraduate students have been involved through individual projects and classes, and can now present a comprehensive alternative to IP to stimulate discussion of what network architecture design really means. The project involves a high percentage of female PIs, which gives us a unique edge in addressing the underrepresentation of women and minorities. The team includes an EPSCOR state participant (U. Memphis).

References

[1] ndnSIM: NS-3 based Named Data Networking (NDN) simulator. ndnSIM website, `http://ndnsim.net`.

[2] Martin Abadi. On SDSI's linked local name spaces. *Journal of Computer Security*, 6(1-2):3–21, October 1998.

[3] 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.

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

[5] Alexander Afanasyev, Cheng Yi, Lan Wang, Beichuan Zhang, and Lixia Zhang. Scaling NDN Routing: Old Tale, New Design. Technical Report NDN-0004, NDN Project, July 2012.

[6] Alexander Afanasyev, Yingdi Yu, and Lixia Zhang. NDNS: scalable and distributed name mapping service for NDN. Poster, UCLA Tech Forum 2013, May 2013.

[7] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.

[8] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *CRYPTO 96*, pages 1–15, 1996.

[9] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *J. Comput. Syst. Sci.*, 61(3):362–399, 2000.

[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] Marián Boguñá, Fragkiskos Papadopoulos, and Dmitri Krioukov. Sustaining the Internet with Hyperbolic Mapping. *Nature Comms*, 1:62, 2010.

[12] Johannes Braun, Johannes Buchmann, Ciaran Mullan, and Alex Wiesmaier. Long term confidentiality: a survey. *Designs, Codes and Cryptography*, pages 1–20, 2012.

[13] J. Burke, P. Gasti, N. Nathan, and G. Tsudik. Securing instrumented environments over Content-Centric Networking: the case of lighting control. In *Proc. of IEEE INFOCOMM 2013 NOMEN Workshop*, April 2013.

[14] Jeff Burke, Alex Horn, and Alessandro Marianantoni. Authenticated lighting control using named data networking. Technical Report NDN-0011, NDN Project, October 2012.

[15] Chris Carter, Abdennour El Rhalibi, and Madjid Merabti. A survey of aoim, distribution and communication in peer-to-peer online games. In *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, 2012.

[16] Connie Chen, David Haddad, Joshua Selsky, Julia E Hoffman, Richard L Kravitz, Deborah E Estrin, and Ida Sim. Making sense of mobile health data: An open architecture to improve individual-and population-level health. *Journal of medical Internet research*, 14(4), 2012.

[17] David D. Clark, John Wroclawski, Karen R. Sollins, and Robert Braden. Tussle in cyberspace: defining tomorrow's Internet. In *Proc. of SIGCOMM '02*, pages 347–356, 2002.

[18] Steven Clarke. Describing and measuring api usability with the cognitive dimensions. In *Cognitive Dimensions of Notations 10th Anniversary Workshop*. Citeseer, 2005.

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

[20] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. of the twentieth annual symposium on Computational geometry*, pages 253–262, 2004.

[21] Steven Dibenedetto, Paolo Gasti, Gene Tsudik, and Ersin Uzun. ANDaNA: Anonymous Named Data Networking application. In *Proc. of NDSS Symposium*, February 2012.

[22] B. H. Dobkin and A. Dorsch. The promise of mhealth: daily activity monitoring and outcome assessments by wearable sensors. *Neurorehabil Neural Repair*, 25(9):788–798, Nov 2011.

[23] William Dron, Alice Leung, Md Uddin, Shiguang Wang, Tarek Abdelzaher, Ramesh Govindan, and John Hancock. Information-maximizing caching in ad hoc networks with named data networking. In *2nd IEEE International Workshop on Network Science (NSW)*, 2013.

[24] D. S. Eng and J. M. Lee. The promise and peril of mobile health applications for diabetes and endocrinology. *Pediatr Diabetes*, 14(4):231–238, Jun 2013.

[25] Deborah Estrin. Sensemaking for mobile health. In *Proc. of IPSN*, 2013.

[26] Xi Fang, Satyajayant Misra, Guoliang Xue, and Dejun Yang. Smart grid—the new and improved power grid: A survey. *IEEE Communications Surveys & Tutorials*, 14(4), 2011.

[27] Amos Fiat and Moni Naor. Broadcast encryption. In *Proc. of Advances in Cryptology, CRYPTO'93*, pages 480–491, 1994.

[28] Benjamin Fung, Ke Wang, Rui Chen, and Philip S Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (CSUR)*, 42(4):14, 2010.

[29] G. Grassi, D. Pesavento, L. Wang, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang. Vehicular internetworking via named data. In *ACM HotMobile 2013 (Poster)*, 2013.

[30] Thomas R. G. Green and Marian Petre. Usability analysis of visual programming environments: A 'cognitive dimensions' framework. *Journal of visual languages and computing*, 7(2):131–174, 1996.

[31] TRG Green, AE Blandford, L Church, CR Roast, and S Clarke. Cognitive dimensions: Achievements, new directions, and open questions. *Journal of Visual Languages & Computing*, 17(4):328–365, 2006.

[32] T. A. Gurman, S. E. Rubin, and A. A. Roess. Effectiveness of mhealth behavior change communication interventions in developing countries: a systematic review of the literature. *J. Health Commun.*, 17 Suppl 1, 2012.

[33] Joseph Y. Halpern and Ron van der Meyden. A logic for SDSI's linked local name spaces. In *In Proc. of IEEE Computer Security Foundations Workshop*, 1999.

[34] R. Hinden. New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG. RFC 1955, 1996.

[35] AKM Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang. Named-data link state routing protocol. In *Proceedings of the ACM SIGCOMM ICN Workshop*, 2013.

[36] Van Jacobson et al. Networking named content. In *Proc. of CoNEXT*, 2009.

[37] Don Johnson, Alfred Menezes, and Scott A. Vanstone. The elliptic curve digital signature algorithm (ECDSA). *Int. J. Inf. Sec.*, 1(1):36–63, 2001.

[38] Georgios Karagiannis, Onur Altintas, Eylem Ekici, Geert Heijenk, Boangoat Jarupan, Kenneth Lin, and Timothy Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *Communications Surveys & Tutorials, IEEE*, 13(4):584–616, 2011.

[39] Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Group encryption. In *Proc. of ASIACRYPT'07*, 2007.

[40] Eric D Knapp. *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*. Syngress, 2011.

[41] Derek Kulinski and Jeff Burke. NDN Video: Live and Prerecorded Streaming over NDN. Technical Report NDN-0007, NDN Project, September 2012.

[42] Jared Lindblom, Ming-Chun Huang, Jeff Burke, and Lixia Zhang. FileSync/NDN: Peer-to-peer file sync over Named Data Networking. Technical Report NDN-0012, NDN Project, March 2013.

[43] R Littman-Quinn, C Mibenge, C Antwi, A Chandra, and CL Kovarik. Implementation of m-health applications in botswana: telemedicine and education on mobile devices in a low resource setting. *Journal of telemedicine and telecare*, 2013.

[44] Jork Loeser and Hermann Haertig. Low-latency hard real-time communication over switched ethernet. In *Proc. of Euromicro Conference on Real-Time Systems*, pages 13–22, 2004.

[45] Min Mun, Shuai Hao, Nilesh Mishra, Katie Shilton, Jeff Burke, Deborah Estrin, Mark Hansen, and Ramesh Govindan. Personal data vaults: a locus of control for personal data streams. In *Proc. of Co-NEXT*, 2010.

[46] ndnSIM user mailing list. `http://www.lists.cs.ucla.edu/mailman/listinfo/ndnsim`.

[47] NIST. Digital signature standard (DSS). FIPS 186-3, June 2009.

[48] Leonardo B Oliveira, Diego F Aranha, Conrado PL Gouvêa, Michael Scott, Danilo F Câmara, Julio López, and Ricardo Dahab. TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks. *Computer Communications*, 34(3):485–493, 2011.

[49] Ricardo Oliveira, Rafit Izhak-Ratzin, Beichuan Zhang, and Lixia Zhang. Measurement of highly active prefixes in BGP. In *IEEE GLOBECOM*, 2005.

[50] Eli Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin, 2011.

[51] D. Pei, , X. Zhao, D. Massey, and L. Zhang. A study of BGP path vector route looping behavior. In *International Conference on Distributed Computing Systems*, March 2004.

[52] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang. Improving BGP convergence through consistency assertions. In *Proceedings of the IEEE INFOCOM 2002*, June 2002.

[53] Zening Qu and Jeff Burke. Egal car: A peer-to-peer car racing game synchronized over Named Data Networking. Technical Report NDN-0010, NDN Project, October 2012.

[54] N. Ramanathan, M. Lukac, T. Ahmed, A. Kar, P.S. Praveen, T. Honles, I. Leong, I.H. Rehman, J.J. Schauer, and V. Ramanathan. A cellphone based system for large-scale monitoring of black carbon. *Atmospheric Environment*, 45(26):4481 – 4487, 2011.

[55] Nithya Ramanathan, Faisal Alquaddoomi, Hossein Falaki, Dony George, C Hsieh, John Jenkins, Cameron Ketcham, Brent Longstaff, Jeroen Ooms, Joshua Selsky, et al. Ohmage: an open mobile system for activity and experience sampling. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on*, pages 203–204, 2012.

[56] Nithya Ramanathan, Dallas Swendeman, W Scott Comulada, Deborah Estrin, and Mary Jane Rotheram-Borus. Identifying preferences for mobile health applications for self-monitoring and self-management: Focus group findings from hiv-positive persons and young mothers. *International journal of medical informatics*, 2012.

[57] Y. Rekhter, T. Li, and S. Hares. A border gateway protocol 4 (BGP-4). *RFC 4271*, January 2006.

[58] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[59] Ronald L. Rivest and Butler Lampson. SDSI - a simple distributed security infrastructure. Technical report, MIT, 1996.

[60] Charles Safran, Meryl Bloomrosen, W. Edward Hammond, Steven Labkoff, Suzanne Markel-Fox, Paul C. Tang, and Don E. Detmer. Toward a national framework for the secondary use of health data: An american medical informatics association white paper. *Journal of the American Medical Informatics Association*, 14(1):1 – 9, 2007.

[61] J. Saltzer, D. Reed, and D. Clark. End-to-end arguments in system design. *ACM Transactions in Computer Systems*, 1984.

[62] Hanan Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2):187–260, 1984.

[63] T. Sauter. The three generations of field-level networks - evolution and compatibility issues. *IEEE Transactions on Industrial Electronics*, 57(11):3585–3595, 2010.

[64] Gregor Schiele, R Siiselbeck, Arno Wacker, Jorg Hahner, Christian Becker, and Torben Weis. Requirements of peer-to-peer-based massively multiplayer online gaming. In *Proc. of CCGRID 2007*, pages 773–782, 2007.

[65] Wentao Shang, Jeff Thompson, Meki Cherkaoui, Jeff Burke, and Lixia Zhang. NDN.JS: A javascript client library for Named Data Networking. In *Proceedings of IEEE INFOCOMM 2013 NOMEN Workshop*, April 2013.

[66] J. C. Sieverdes, F. Treiber, and C. Jenkins. Improving diabetes management with mobile health technology. *Am. J. Med. Sci.*, 345(4):289–295, Apr 2013.

[67] Tor Skeie, Svein Johannessen, and Oyvind Holmeide. Timeliness of real-time IP communication in switched industrial Ethernet networks. *IEEE Transactions on Industrial Informatics*, 2(1):25–39, 2006.

[68] M. Steenstrup. An architecture for inter-domain policy routing. *RFC 1478*, June 1993.

[69] M. Steenstrup. Inter-domain policy routing protocol specification: Version 1. *RFC 1479*, July 1993.

[70] Keith Stouffer, Joe Falco, and Karen Scarone. Guide to industrial control systems (ICS) security. Technical Report 800-82, National Institute of Standards and Technology (NIST), June 2011.

[71] Cheshire Stuart, Zhenkai Zhu, Ryuji Wakikawa, and Lixia Zhang. Understanding Apple's Back to My Mac service. RFC 6281, 2011.

[72] Piotr Szczechowiak, Leonardo B Oliveira, Michael Scott, Martin Collier, and Ricardo Dahab. NanoECC: Testing the limits of elliptic curve cryptography in sensor networks. *Wireless sensor networks*, pages 305–320, 2008.

[73] T. Tamrat and S. Kachnowski. Special delivery: an analysis of mHealth in maternal and newborn health programs and their outcomes around the world. *Matern Child Health J*, 16(5):1092–1101, Jul 2012.

[74] Md Yusuf Sarwar Uddin, Hongyan Wang, Fatemeh Saremi, Guo-Jun Qi, Tarek Abdelzaher, and Thomas Huang. Photonet: a similarity-aware picture delivery service for situation awareness. In *Proc. of Real-Time Systems Symposium (RTSS)*, pages 317–326, 2011.

[75] J. Wang, R. Wakikawa, R. Kuntz, R. Vuyyuru, and L. Zhang. Data naming in vehicle-to-vehicle communications. In *Proceedings of INFOCOM 2012 Workshop on Emerging Design Choices in Name-Oriented Networking*, 2012.

[76] Jiangzhe Wang, Ryuji Wakikawa, and Lixia Zhang. DMND: Collecting data from mobiles using named data. In *Proceedings of the Second IEEE Vehicular Networking Conference (VNC 2010)*, 2010.

[77] L. Wang, D. Massey, K. Patel, and L. Zhang. FRTR: A scalable mechanism for global routing table consistency. In *Proceedings of the International Conference on Dependable Systems and Networks*, June 2004.

[78] L. Wang, M. Saranu, J. Gottlieb, and D. Pei. Understanding BGP session failures in a large ISP. In *Proceedings of the IEEE INFOCOM 2007*, may 2007.

[79] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. Wu, and L. Zhang. Observation and analysis of BGP behavior under stress. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop 2002*, November 2002.

[80] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, and L. Zhang. Protecting BGP routes to top-level DNS servers. *IEEE Transactions on Parallel and Distributed Systems*, September 2003.

[81] Lan Wang, A K M Mahmudul Hoque, Cheng Yi, Adam Alyyan, and Beichuan Zhang. OSPFN: An OSPF based routing protocol for Named Data Networking. Technical Report NDN-0003, NDN Project, July 2012.

[82] Shiguang Wang, Shaohan Hu, Shen Li, Hengchang Liu, Md Yusuf Sarwar Uddin, and Tarek Abdelzaher. MINERVA: Information-centric programming for social sensing. In *Proc. of International Conference on Computer Communications and Networks (ICCCN)*, 2013.

[83] Xiaogang Wang. Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, 2012.

[84] J. Wroclawski. Issues related to the evaluation of architecture. Presentation at FIA Spring 2013 meeting, March 2013.

[85] 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.

[86] H. Yuan et al. Scalable ndn forwarding: Concepts, issues and principles. In *Proc. of ICCCN*, 2012.

[87] Haowei Yuan and Patrick Crowley. Experimental evaluation of content distribution with NDN and HTTP. In *Proc. of the 32th Annual IEEE Conference on Computer Communications (INFOCOM'13), mini-conference*, 2013.

[88] Beichuan Zhang, Vamsi Kambhampati, Mohit Lad, Daniel Massey, and Lixia Zhang. Identifying BGP routing table transfers. In *ACM SIGCOMM Mining the Network Data (MineNet) Workshop*, August 2005.

[89] Beichuan Zhang, Daniel Massey, and Lixia Zhang. BGP dynamics during route flap damping. Technical Report 03-805, USC-CSD, November 2003.

[90] Beichuan Zhang, Daniel Massey, and Lixia Zhang. Destination reachability and BGP convergence time. In *Proc. of IEEE Globecom Global Internet Symposium*, December 2004.

[91] X. Zhao, D. Massey, S. F. Wu, M. Lad, D. Pei, L. Wang, and L. Zhang. Understanding BGP behavior through a study of DoD prefixes. In *Proc. of DISCEX*, April 2003.

[92] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang. Detection of invalid routing announcements in the Internet. In *Proceedings of the International Conference on Dependable Systems and Networks*, June 2002.

[93] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. An analysis of BGP multiple origin AS (MOAS) conflicts. In *IMW*, 2001.

[94] Zhenkai Zhu, Alexander Afanasyev, and Lixia Zhang. ChronoShare: a new perspective on effective collaborations in the future Internet. Poster, UCLA Tech Forum 2013, May 2013.

[95] Zhenkai Zhu, Alexander Afanasyev, and Lixia Zhang. Let's ChronoSync: Decentralized dataset state synchronization in Named Data Networking, 2013. under submission.

[96] Zhenkai Zhu, Chaoyi Bian, Alexander Afanasyev, Van Jacobson, and Lixia Zhang. Chronos: Serverless multi-user chat over NDN. Technical Report NDN-0008, NDN Project, October 2012.

[97] Zhenkai Zhu, Jeffrey Burke, Lixia Zhang, Paolo Gasti, Yanbin Lu, and Van Jacobson. A new approach to securing audio conference tools. In *Proceedings of the 7th Asian Internet Engineering Conference*, AINTEC'11, 2011.