Named Data Networking (NDN) Project

NDN-0001
October 9, 2010

Lixia Zhang, Deborah Estrin, and Jeffrey Burke
University of California, Los Angeles

Van Jacobson, James D. Thornton, and Diana K. Smetters
Palo Alto Research Center (PARC)

Beichuan Zhang
University of Arizona

Gene Tsudik
University of California, Irvine

kc claffy and Dmitri Krioukov
University of California, San Diego

Dan Massey and Christos Papadopoulos
Colorado State University

Tarek Abdelzaher
University of Illinois at Urbana-Champaign

Lan Wang
University of Memphis

Patrick Crowley
Washington University

Edmund Yeh
Yale University

Contents

1 Vision

In the 1960s and 70s when the core ideas underlying the Internet were developed, telephony was the only example of successful, effective, global-scale communications. Thus while the communication *solution* offered by TCP/IP was unique and groundbreaking, the *problem* it solved was telephony's: a point-to-point conversation between two entities.

The world has changed dramatically since then.

- Information-intensive business like travel, banks and financial services long ago moved onto the Internet. Today almost anything is available online as the Internet becomes the world's storefront.
- Digital coding advances have turned not just text but voice, images and video into strings of bits so an ever increasing range of content can be distributed digitally.
- The Web has made it easy for anyone to create, discover and consume content with the result that exabytes of new content are produced yearly.
- Moore's-Law-driven hardware advances have made it feasible to connect *everything* to the Internet: not just supercomputers and workstations but also factories, municipal infrastructure, phones, cars, appliances, even light switches.

While IP has exceeded all expectations for facilitating ubiquitous interconnectivity, it was designed for conversations between communications endpoints but is overwhelmingly used for content distribution. Just as the telephone system would be a poor vehicle for the broadcast content distribution done by TV and radio, the Internet architecture is a poor match to its primary use today.

The 'conversational' nature of IP is embodied in its datagram format: IP datagrams can only name communication endpoints (the IP destination and source addresses). As our project title suggests, we propose to generalize the Internet architecture by removing this restriction: the names in an NDN datagram are hierarchically structured but otherwise arbitrary identifiers. They can be used to name a chunk of data in a conversation, as the TCP/IP transport signature plus sequence number does today, but they can also name a chunk of data from a YouTube video directly, rather than forcing it to be embedded in a conversation between the consuming host and youtube.com. 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 change creates an abundance of new opportunities:

- Today's applications are typically written in terms of *what* information they want rather than *where* it is located, then application-specific middleware is used to map between the application model and the Internet's. With NDN the application's *what* model can be implemented directly, removing all the middleware and its associated configuration and communication inefficiencies.
- Since conversations are ephemeral and can be about anything, the current security approach is the one-size-fits-all model of armoring the channel between two IP addresses, which rarely meets the end-to-end security needs of data producers and consumers. In NDN, all data is secured end-to-end and the name provides essential context for security. E.g., NDN can tell if all the data on the web page one is viewing was produced and signed by one's bank; IP cannot.
- Since every chunk of data can be uniquely named, looping can be prevented using the memory already required by the router, rather than imposing the single-path forwarding constraint that today's IP routing enforces. This design choice allows any node to freely use all of its connectivity to solicit or distribute data, and removes the information asymmetries that give today's dominant providers disproportionate control over routes and thus over smaller, local providers.

The change also introduces significant intellectual challenges:

- IP addresses are hierarchically structured, which has allowed routing state to scale via aggregation. For example, IP supports direct communication between more than two billion hosts with about 300K routes in its transit core. Although the NDN namespace is bigger than IP's, we believe that by using hierarchical names, much like the URLs used to name today's web content, NDN can achieve a similar exponential reduction in space. Also, NDN's delivery model allows routing and forwarding to occur with approximate state such as Bloom filters rather than IP's exact state, potentially reducing NDN's state burden below that of IP.

- Decades of research have proven it possible to engineer ASICs to forward IP packets at wire rate, even for the fastest wires. We think that much of that research plus some new techniques can be used to achieve wire rate forwarding of NDN's longer and variable length names.
- Our fundamental, information-theoretic framework for understanding communications is based on the capacity of a point-to-point *channel*. We believe this model can be extended to describe a communication system where memory has a larger and more central role, an intellectually challenging and novel direction.
- Communications security has always been divorced from the data it secures. Securing named data potentially allows the security to be much more user-centric, expressed in terms of the user's data model and context. Finding effective, automatic and transparent mechanisms to implement and manage security of named data will be a new and more promising research trajectory than most IP security research has followed for the last two decades.

We emphasize that the NDN model is compatible with today's Internet and has a clear, simple evolutionary strategy. Like IP, NDN is a "universal overlay": NDN can run over anything, including IP, and anything can run over NDN, including IP. IP infrastructure services that have taken decades to evolve, such as DNS naming conventions and namespace administration or inter-domain routing policies and conventions, can be readily used by NDN. Indeed, because NDN's hierarchically structured names are semantically compatible with IP's hierarchically structured addresses, the core IP routing protocols, BGP, IS-IS and OSPF, can be used as-is to deploy NDN in parallel with and over IP. Thus NDN's advantages in content distribution, application-friendly communication and naming, robust security, mobility and broadcast can be realized incrementally and relatively painlessly.

2 Architecture

NDN is an entirely new architecture, but one whose operations can be grounded in current practice. Its design reflects our understanding of the strengths and limitations of the current Internet architecture.

2.1 Architectural Principles

The following architectural principles guide our design of the NDN architecture.

- The *hourglass architecture* is what makes the original Internet design elegant and powerful. It centers on a *universal* network layer (IP) implementing the *minimal* functionality necessary for global interconnectivity. This so-called "thin waist" has been a key enabler of the Internet's explosive growth, by allowing lower and upper layer technologies to innovate without unnecessary constraints. NDN keeps the same hourglass-shaped architecture.
- *Security must be built into the architecture.* Security in the current Internet architecture is an afterthought, not meeting the demands of today's increasingly hostile environment. NDN provides a basic security building block *right at the thin waist* by signing all named data.
- The *end-to-end principle* [65] enables development of robust applications in the face of network failures. NDN retains and expands this principle.
- *Network traffic must be self-regulating*. Flow-balanced data delivery is essential to stable network operation. Since IP performs open loop data delivery, transport protocols have been amended to provide unicast traffic balance. NDN designs flow-balance into the thin waist.
- *Routing and forwarding plane separation* has proven necessary for Internet development. It allows the forwarding plane to function while the routing system continues to evolve over time. NDN sticks to the same principle to allow the deployment of NDN with the best available forwarding technology while we carry out new routing system research in parallel.

- The *architecture should facilitate user choice and competition where possible.* Although not a relevant factor 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 enable competition [15].

2.2 The NDN Architecture

Communication in NDN is driven by the receiving end, *i.e.*, the data consumer. To receive data, a consumer sends out an ***Interest*** packet, which carries a name that identifies the desired data (see Figure 1). For example, a consumer may request `/parc/videos/WidgetA.mpg`. A router remembers the interface from which the request comes in, and then forwards the Interest packet by looking up the name in its ***Forwarding Information Base (FIB)***, which is populated by a name-based routing protocol. Once the Interest reaches a node that has the requested data, a ***Data*** packet is sent back, which carries both the name and the content of the data, together with a signature by the producer's key (Figure 1). This *Data* packet traces in reverse the path created by the Interest packet back to the consumer. Note that neither Interest nor Data packets carry any host or interface addresses (such as IP addresses); Interest packets are *routed* towards data producers based on the names of data they request, and *Data* packets are returned based on the state information set up by the Interests at each router hop (Figure 2).[1]

NDN routers keep both Interests and Data for some period of time. When multiple Interests for the same data are received from downstream, only the first Interest is sent upstream towards the data source. The router then stores the Interest in the ***Pending Interest Table (PIT)***, where each entry contains the name of the Interest and a set of interfaces from which the matching Interests have been received. When the Data packet arrives, the router finds the matching PIT entry and forwards the data to all the interfaces listed in the PIT entry. The router then removes the corresponding PIT entry, and caches the Data in the ***Content Store***, which is basically the router's buffer memory subject to a cache replacement policy. Data takes the exact same path as the Interest that solicited it, but in the reverse direction. One Data satisfies one Interest across each hop, achieving hop-by-hop flow balance.

An NDN Data packet is meaningful independent of where it comes from or where it may be forwarded to, thus the router can cache it to satisfy potential future requests. This enables NDN to automatically support various functionality without extra infrastructure, including content distribution (many users requesting the same data at different times), multicast (many users requesting the same data at the same time), mobility (users requesting data from different locations), and delay-tolerant networking (users having intermittent connectivity). For example, consider a consumer who is watching a streaming video in a moving vehicle. The consumer may request some data but then move to a new local network. Though the Data will arrive at the old location and be dropped, it is cached along the path. When the consumer retransmits the Interest, it will likely pull the Data from a nearby cache, making the interruption minimal. Data cached close to consumers improves packet delivery performance and reduces dependency on a particular data source that may fail due to faults or attacks.

Below we describe each major element of the NDN architecture in detail, identifying benefits of the design choices and presenting the challenges.

2.2.1 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 PARC may have the name `/parc/videos/WidgetA.mpg`, where '/' indicates a boundary between name components (it is not part of the name). This hierarchical structure is useful for applications to represent relationships between pieces of data. For example, segment 3 of version 1 of the video might be named `/parc/videos/WidgetA.mpg/1/3`. The hierarchy also allows routing to scale. While it may be theoretically possible to route on flat names [11], it is the hierarchical structure of IP addresses that enables aggregation which is essential in scaling today's routing system. Common structures necessary to allow programs to operate over NDN names can be

[1] NDN supports all existing applications, including those that "push" content. For example, to send an email, the client first sends an Interest to the server to solicit server's interest in receiving the email. If the server is interested, it will send an Interest to the client and the client can then send Data packets to the server.

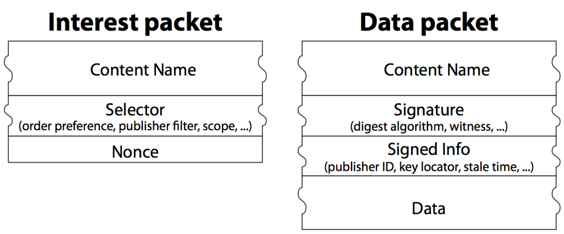

Figure 1: Packets In the NDN Architecture.

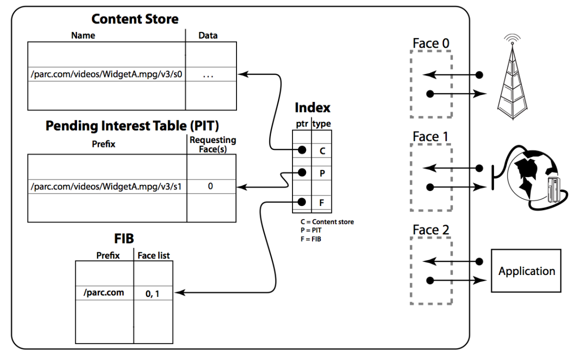

Figure 2: Forwarding Process at an NDN Node.

achieved by *conventions* agreed between data producers and consumers, *e.g.*, name conventions indicating versioning and segmentation. Name conventions are specific to applications and opaque to networks.

Names do not need to be *globally unique*, although retrieving data globally requires a degree of global uniqueness. Names intended for local communication may be heavily based on local context, and require only local routing (or local broadcast) to find corresponding data.

To retrieve dynamically generated data, consumers must be able to *deterministically* construct the name for a desired piece of data without having previously seen the name or data. Either (1) a deterministic algorithm allows producer and consumer to arrive at the same name based on data available to both, and/or (2) consumers can retrieve data based on partial names. For example, the consumer may request `/parc/videos/WidgetA.mpg` and get back a data packet named `/parc/videos/WidgetA.mpg/1/1`. The consumer can then specify later segments and request them, using a combination of information revealed by the first data packet and the naming convention agreed upon by the consumer and producer applications.

The naming system is the most important piece in the NDN architecture and still under active research; in particular, how to define and allocate top level names remains an open challenge. Not all naming questions need be answered immediately, however; the opaqueness of names to the network – and dependence on applications – means that design and development of the NDN architecture can, and indeed must, proceed in parallel with our research into name structure, name discovery and namespace navigation in the context of application development (Section 3.3).

2.2.2 Data-Centric Security

In NDN, security is built into data itself, rather than being a function of where, or how, it is obtained [44]. Each piece of data is signed together with its name, securely binding them. Data signatures are mandatory – applications cannot "opt out" of security. The signature, coupled with data publisher information, enables determination of data *provenance*, allowing the consumer's trust in data to be decoupled from how (and from where) data is obtained. It also 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.

However, to be practical, this fine-grained and data-centric security approach requires some innovation. Historically, security based on public key cryptography has been considered inefficient, unusable and difficult to deploy. Besides efficient digital signatures, NDN needs flexible and usable mechanisms to manage user trust. Section 3.4 describes how NDN offers a promising substrate for achieving these security goals. Since keys can be communicated as NDN data, key distribution is simplified. Secure binding of names to data provides a basis for a wide range of trust models, *e.g.*, 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. This offers publishers, consumers and applications a great deal of flexibility in choosing or customizing their trust models.

NDN's data-centric security can be extended 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 much-needed routing protocol security. Section 3.4 describes planned research on efficient signatures, usable trust management, network security, content protection and privacy.

2.2.3 Routing and Forwarding

NDN routes and forwards packets on names, which eliminates four problems that addresses pose in the IP architecture: address space exhaustion, NAT traversal, mobility, and scalable address management. There is no address exhaustion problem since the namespace is unbounded. There is no NAT traversal problem since a host does not need to expose its address in order to offer content. Mobility, which requires changing addresses in IP, no longer breaks communication since data names remain the same. Finally, address assignment and management is no longer required in local networks, which is especially empowering for sensor networks.

Routing can be done in a similar fashion to today's IP routing. Instead of announcing IP prefixes, a NDN router announces *name prefixes* that cover the data that the router is willing to serve. This announcement is propagated through the network via a routing protocol, and every router builds its FIB based on received routing announcements. Conventional routing protocols, such as OSPF and BGP, can be adapted to route on name prefixes. However, an unbounded namespace raises the question of how to keep the routing table sizes scalable to the number of data names. In Section 3.1 we describe several approaches to scalable routing, both conventional and new.

Routers treat names as a sequence of opaque components and simply do component-wise longest prefix match of the ContentName from a packet against the FIB. For example, `/parc/videos/WidgetA.mpg` may match both `/parc/videos` and `/parc` in the FIB, and `/parc/videos` is the longest prefix match. An important question is whether looking up variable-length, hierarchical names can be done at line rate. In Section 3.2 we introduce various hardware and software techniques for fast name lookup that we plan to explore.

NDN inherently supports multipath routing. IP routing adopts a single best path to prevent loops. In NDN, Interests cannot loop persistently, since the name plus a random nonce can effectively identify duplicates to discard. Data do not loop since they take the reverse path of Interests. Thus an NDN router can send out an Interest using multiple interfaces without worrying about loops. The first Data coming back will satisfy the Interest and be cached locally; later arriving copies will be discarded. This built-in multipath capability elegantly supports load balancing as well as service selection. For example, a router may forward the first few Interests out via all possible interfaces, measure the performance based on returning Data, and select the best performing interface(s) for subsequent Interests. Section 3.2 discusses further ideas for refining this capability, which we call *Forwarding Strategy*.

Routing security is greatly improved in NDN. First, signing all data, including routing messages, prevents them from being spoofed or tampered with. Second, multipath routing mitigates prefix hijacking because routers may detect the anomaly caused by prefix hijacking and try other paths to retrieve the data. Third, the fact that NDN messages can talk only about data, and simply cannot be addressed to hosts makes it difficult to send malicious packets to a particular target. To be effective, attacks against NDN must focus on denial of service, which will be addressed in Section 3.4.3.

2.2.4 Caching

Upon receiving an Interest, an NDN router first checks the Content Store. If there is a data whose name falls under the Interest's name, the data will be sent back as a response. The Content Store, in its basic form, is just the buffer memory in today's router. Both IP routers and NDN routers buffer data packets. The difference is that IP routers cannot reuse the data after forwarding them, while NDN routers are able to reuse the data since they are identified by persistent names. 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 packet retransmission after a packet loss (Section 2.2.6). Cache management and replacement is subject to ISP policies and will be one of our research topics as described in Section 3.2.

Caching named data may raise privacy concerns. Today's IP networks offer weak privacy protection. One can find out *what* is in an IP packet by inspecting the header or payload, and *who* requested the data by checking the destination address. NDN explicitly names the data, arguably making it easier for a network monitor to see what data is being requested. One may also be able to learn what data is requested through clever probing schemes to derive what is in the cache. However NDN removes entirely the information regarding who is requesting the data. Unless directly connected to the requesting host by a point-to-point link, a router will only know that someone has requested certain data, but will not know who originated the request. Thus the NDN architecture naturally offers privacy protection at a fundamentally different level than the current Internet (Section 3.4.4). Section 3.4.3 also discusses defense mechanisms against attacks to the Content Store and PIT.

2.2.5 Pending Interest Table (PIT)

The PIT contains the arrival interfaces of Interests that have been forwarded but are still waiting for matching Data. This information is required to deliver data to their consumers. To maximize the usage of the PIT, PIT entries need to be timed out pretty quickly, somewhere around packet round-trip time. However, if they are timed out prematurely, Data will be dropped, and it is the consumer's responsibility to retransmit his/her Interests.

The PIT state at each router has several critical functions. Since it includes the set of interfaces over which Interests have arrived, it provides natural support for multicast functionality. Second, the router can control the rate of incoming Data packets by controlling its PIT size. Together with data caching, an NDN network removes the dependency on transport protocols to avoid congestion collapse. Finally, PIT state may be exploited to mitigate DDoS attacks: the number of PIT entries is an explicit indication of the router load, an upper bound on this number sets the ceiling on the effect of a DDoS attack; PIT entry timeouts offer relatively cheap attack detection [66]; and the arrival interface information in each PIT entry gives information to implement a pushback scheme [42].

Many attempts have been made to install the above functions in the Internet over the last 20 years (e.g. Capability [74]). Each of these efforts tried to install a particular piece of state into routers, but none has succeeded on a large scale. Designing the PIT into the infrastructure can achieve all of these ends in a systematic way and offer a far superior solution than a collection of point solutions. However, it does potentially incur a state burden. We will study the feasibility of the PIT in both hardware and software implementation (Section 3.2).

2.2.6 Transport

The NDN architecture does not have a separate transport layer. It moves the functions of today's transport protocols up into applications, their supporting libraries, and the strategy component in the forwarding plane. Multiplexing and demultiplexing among application processes is done directly using names at the NDN layer, and data integrity and reliability are directly handled by application processes where the appropriate reliability checking, data signing and trust decisions can be made.

An NDN network is designed to operate on top of unreliable packet delivery services, including the highly dynamic connectivity of mobile and ubiquitous computing. To provide reliable, resilient delivery, Interest packets that are not satisfied within some reasonable period of time must be retransmitted by the final consumer (the application that originated the initial Interest) if it still wants the data. Such functionality is common to many or all NDN applications, and in NDN they will be provided by common libraries. A consumer's *forwarding strategy* works at a lower level: it is responsible for retransmission on a particular interface (since it knows the timeout for the upstream node(s) on the interface) as well as selecting which and how many of the available communication interfaces to use for sending Interests, how many unsatisfied Interests should be allowed, the relative priority of different Interests, etc.

NDN routers can manage traffic load through managing the PIT size (the number of pending Interests) on a hop-by-hop basis; when a router is overloaded by incoming data traffic from any specific neighbor, it can simply slow down or stop sending Interest packets to that neighbor. This also means that NDN eliminates the dependency on end hosts performing congestion control.

Once congestion occurs, data retransmission is aided by caching. For example, if two congested links exist along the path between a producer and a consumer, and a Data packet gets through the first congested link but is dropped at the second congested link, then after the consumer times out and retransmits the Interest, caching will allow the Data packet to be retransmitted over only the second congested link. In the current Internet, the retransmission of data will happen all the way back at the producer, and the packet has to try to pass the first congested link again. In NDN, data make steady progress towards the final destination, as the cached copy is used to satisfy the original Interest as well as retransmitted Interests. Thus NDN avoids congestion collapse that can occur in today's Internet when bandwidth is mostly consumed by repeated retransmissions and effective throughput drops to minimal.

2.3 How NDN Adheres to Architectural Principles and Benefits Society

NDN realigns the architecture with application needs by adopting named data as the thin waist of the hourglass architecture. Today's applications have to rely on complex middleware to map from IP's host-based abstractions to the content that they care about. NDN greatly simplifies application development (Section 3.3), and new applications in turn will drive further growth and success of the future Internet.

NDN signed data provides the essential building block for the trustworthiness of the future Internet. Applications can build fine-grained, customized authentication, authorization, and trust models.

NDN adheres to the end-to-end principle. This signature of NDN data provides both integrity protection and data origin authenticity, so that when a consumer receives the data and verifies the signature, he knows for sure that he has received a copy of the original data published by the right producer. Thus NDN offers a very strong notion of secure end-to-end data transmission, even though the producer and consumer do not communicate directly.

NDN provides a strong hop-by-hop network flow balance by matching every Data to every Interest at each link. Thus NDN networks are able to self-regulate traffic flows for both unicast and multicast traffic without relying on transport protocols. *NDN also separates routing schemes and forwarding mechanisms.*

NDN facilitates choice and competition by empowering users. As shown by a network economic model due to Laskowski and Chuang [50], monitoring delivery performance is a key requirement to achieve ISP accountability. However in today's global routing system, IP uses only a single path to each destination, and that path is often asymmetric due to "hot-potato" routing. It is difficult to measure and compare performance through different service providers simultaneously. In contrast, with NDN's built-in multipath forwarding capability and feedback loop (*i.e.*, sending one Interest out, receiving one Data back over the same path), users can explore multiple paths, monitor delivery performances, and make their choices. For example, multihoming users and small ISPs can choose to use providers with the best performance. This will encourage innovations and investments into the network infrastructure through competition.

NDN democratizes content distribution, which is another way NDN significantly facilitates choice and competition. One key societal impact of the Internet is disseminating content and knowledge. Though Internet is no doubt successful, the amount of content we create still dwarfs what today's Internet is able to disseminate. NDN's built-in caching capability enables content producers, be they CNN or a home user, to distribute their content at global scale efficiently without special infrastructure such as CDNs, which will have far-reaching impacts on the society, especially for people in underdeveloped regions and in underrepresented groups. It will also have a positive feedback effect, encouraging people to create and produce original content.

2.4 Comparison

An often heard question about NDN is "Isn't NDN another overlay/CDN/search/pub-sub system?" NDN is an overlay in the same sense that IP is an overlay on top of all the different transmission networks deployed today. NDN only requires simple best-effort packet transport between adjacent NDN nodes; it can run on top of any layer-2 technology or above.

Today's content distribution networks (CDNs) are essentially a massive overlay infrastructure, deploying a large number of machines to cache and serve contracted data. The service is expensive, and specific to only contracted applications specially modified to use it. Different CDNs are isolated from each other, and each one's performance is limited by its own server coverage. Nonetheless the widespread CDN services deployed by large content producers highlight the increasing need for efficient content distribution. NDN overcomes CDN's limitations and democratizes content distribution.

NDN itself is not a search engine. Rather, users may get application/data names from a combination of memory, guessing, family and friends in social circles, and search engines, and then use NDN to retrieve the corresponding data from the network. Furthermore, if one may consider the routing announcements from data producers as "publish" acts and Interest packets as user "subscribe" requests, then NDN builds a scalable and efficient server-less publish/subscribe system. In contrast to existing publish/subscribe system designs, which utilize special rendezvous servers, all NDN routers cache both Interests (in the PIT) and Data packets (in the Content Store), thus no redundant Interest is sent upstream over the same paths, and no redundant Data is sent downstream.

3 Research Agenda

The NDN design introduced in the previous section represents a novel architectural blueprint with both unique opportunities and many challenges. Even a team as large as ours cannot hope to fully address all of these in one project, as is to be expected for a significant architectural change. We therefore must prioritize the topics most critical for realizing and testing the NDN design. We must develop both **scalable routing solutions** and **fast forwarding engines** that can handle NDN traffic at wire speed to show that NDN is deployable at a global scale. We must develop **applications** that run on top of NDN both to drive development of the design and to verify NDN feasibility and usefulness in supporting existing applications and facilitating new generations of applications. We must demonstrate how the **security foundations** in NDN can be *effectively* and *efficiently* used to secure applications, protect privacy, and defend the infrastructure itself. The new communication model of NDN, which contains not only transmission but also *storage*, requires a new fundamental theory of communication. In all of these areas, we must investigate new evaluation methodologies to gauge the correctness and effectiveness of the NDN design. Given its fundamental departure from today's IP Internet, finding the right measurements is a daunting research challenge on its own. Each of these topics is part of our research agenda and described further in the remainder of this section, along with a short summary of implementation and deployment plans including instrumentation to support network traffic management.

Prior to NDN and the NSF FIA (Future Internet Architecture) program, nine of us have been conducting research funded by the NSF FIND (Future Internet Design) program. These projects include (a) Patrick Crowley: An Architecture for a Diversified Internet; (b) Dmitri Krioukov, KC Claffy: Greedy Routing on Hidden Metric Spaces as a Foundation of Scalable Routing Architectures without Topology Updates; (c) Jeff Burke, Deborah Estrin: Network Innovations for Personal, Social, and Urban Sensing Applications; and (d) Daniel Massey, Lan Wang, Beichuan Zhang, Lixia Zhang: Enabling Future Internet innovations through Transit wire (eFIT). These prior research findings can be directly applied to the routing, forwarding and application design in NDN.

3.1 Routing

We propose to tackle two major challenges in routing: (a) bounding the amount of routing state while allowing an unbounded name space; and (b) supporting intelligent forwarding of Interests over multiple paths. By design, 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 of networking where peer identities and locations are unknown. By transforming all communications to use names, NDN provides a robust information security model that also applies to the routing infrastructure itself (Section 3.4). NDN's separation of the forwarding and routing planes allows us to deploy and test other parts of the architecture as soon as possible using extensions of existing routing protocols, while the routing research team can spend most of the project duration to learn from actual deployment and investigate more scalable solutions to support large-scale deployment.

3.1.1 Initial Deployment: Extending Existing Routing Protocols

For the initial roll-out of NDN, we propose to extend the implementations of the OSPF (intra-domain) and BGP (inter-domain) routing protocols to support name prefixes and multipath forwarding of Interests. Immediate implementation will allow us to study two issues in multipath routing that will inform our subsequent research strategy: preventing multi-path forwarding of Interest packets from inducing loops and determining the optimal number and diversity of paths to maximize the probability and performance of data delivery while minimizing computation, latency, and overhead.

3.1.2 Long-Term Deployment: Achieving Routing Scalability

We plan two long-term routing research directions: one using provider-assigned names, which are amenable to aggregation, and a more scalable approach that exploits the small-world property that is predicted in a large-scale deployment (see Section 3.7).

Approach 1: ISP-based Aggregation This approach has two basic components: (a) hierarchical provider-assigned names to facilitate aggregation; and (b) a mapping service to map user-selected names to provider-assigned names. This reduces the FIB size to be proportional to the number of ISPs rather than users, similar to the separation of core and edge prefixes we previously proposed for IP routing scalability [51, 45]. We will also investigate whether it is feasible to employ local FIB aggregation techniques such as those we developed to scale IP forwarding tables [75].

Conceptually, provider-assigned names are similar to provider-assigned IP addresses, but unbounded in size. For example, a user at AT&T may be assigned the name `/att/location/user`. These names can be aggregated to `/att/location` and ultimately to `/att`. Alternatively, users may choose names that are easier to remember, *e.g.*, `/aliceblog`, and use a mapping service to map the name, or name prefix, to a provider-assigned name, e.g., `/att/atlanta/alice/blog`. For Bob to read Alice's blog, Bob's computer will use the mapping service to retrieve the provider-assigned name (or names) for `/aliceblog`.

Two architectural features of NDN profoundly reduce the incentive for network traffic engineers to de-aggregate or otherwise inject longer name prefixes than necessary into the global routing system. NDN's native multipath forwarding capability provides natural support for multihoming, and its inherently symmetric routing – data is only sent back traversing the Interest path – allows routers to monitor "path" performance by maintaining per-interface throughput statistics, and adapt forwarding path (interface) selection strategy accordingly. Furthermore NDN naturally secures routing updates. These two factors remove the incentive of IP prefix de-aggregation to support traffic engineering or to reduce the chance of being hijacked.

We will investigate *how to design the mapping service from user-selected names to provider-assigned names.* One option is to design a new system from scratch. Another option is to augment the existing DNS system with a new NDN record containing the mapping information, possibly including a new top-level domain .ndn to hold mappings for user-selected names e.g., `aliceblog` would map to `aliceblog.ndn`.

Approach 2: Exploiting the Name Space and Network Structure to Scale Routing Ultimately we want to route directly on application names without resolving them to provider-assigned names. The NDN architecture is sufficiently flexible to allow us to explore the most ambitious routing research idea to emerge from NSF's FIND program: greedy routing based on underlying metric spaces. Assuming routers can be assigned coordinates in a name-based metric space, they can compute the name-space distances between their directly connected neighbors and the destination name in the Interest packet. In standard greedy routing, each intermediate router then forwards the Interest packet to its neighbor router closest to the destination name. The key to the scalability of this approach is the hierarchical, or tree-like, name space structure. Even an approximately tree-like structure can be mapped to an underlying hyperbolic metric space [33], which we have recently shown supports theoretically optimal routing performance characteristics when greedy routing is used [48, 47, 63]. Unfortunately, standard greedy routing does not always succeed in reaching the destination [7], sometimes getting stuck at local minima, i.e., routers having no neighbors closer to the destination than themselves. We propose to overcome this problem by augmenting standard greedy routing with elements of random walks, and evaluate the cost to path length. Specifically, intermediate routers can forward Interests not only to the neighbor closest to the destination, but to any neighbor with higher probability of being closer to the destination. This approach is conceptually similar to emerging "social overlay" routing in a variety of networks [19, 53, 41, 54, 12, 55] where the destination of information propagation is a specific individual or content. Forwarding decisions rely on social distances to the destination, while network connectivity is provided by the highly dynamic "underlay network." In our case, the underlay network is the NDN router network, and the hierarchical name space serves as a type of social overlay.

Open research questions include: (1) how routers can compute the name-space coordinates using only local information in their FIBs, PITs, and/or Content Stores, and potentially the name-space coordinates of their neighbors; (2) how routers can optimize their performance with probabilistic random walk behavior; (3) determining specific properties that the structure of the name space and router topology must have to ensure efficiency; and (4) what mechanisms might induce these properties.

3.2 Forwarding

Due to its fundamental focus on named data, NDN implies a substantial re-engineering of forwarding plane devices to provide fast name lookup, intelligent forwarding strategy, and effective caching policies. Our design must meet the following three requirements.

1. Variable-length, hierarchical names. In NDN, all lookup operations reply upon finding the longest matching prefix for a given name, which differs from IP's longest prefix match in two substantial ways. First, NDN names are explicitly hierarchical, consisting of a series of delimited components (i.e., length-annotated bit-strings), while IP addresses can match a prefix at any bit position. Second, IP addresses are fixed length, while NDN name lengths are variable, with no externally imposed upper bound. While methods for high-speed IP longest prefix match can be applied to NDN names, we will pursue other directions that show greater promise.

2. Fast updates to prefix table. Logically, the NDN forwarding path contains three tables: the FIB, the PIT, and the Content Store. In memory these tables can be organized as a single structure, with table entries recording their type in a mixed structure. This prefix table must support fast inserts, deletes and modifications, after the routing protocols recompute the FIB and Interest/Data packets arrive.

3. Very high capacity. Suppose a system is provisioned with packet buffers 2KB in size with 2 GB of total buffer space; this means Content Store capacity for 1 million distinct packet names. If an NDN name were 200 bytes in length, then storing the names alone would require a further 200 MB. There may be as many PIT entries, requiring $(bandwidth X RTT)$ worth of Interests. FIB entries are likely fewer. In this example, an NDN forwarding device would require perhaps 10s of gigabytes of main memory, and would need to support a prefix table several gigabytes in size.

Based on past enthusiasm for improving IP data plane mechanisms, e.g., longest prefix match and packet classification, we optimistically hope these engineering challenges will be met enthusiastically and successfully by the scientists and engineers charged with building the NDN forwarding plane. Below we present our research agenda in each area.

3.2.1 Fast Name Lookup

Current-generation TCAMs may effectively support the matching function of name lookup in a *Hybrid TCAM Solution*. Distribution of NDN prefix lengths may be approximately bi-modal: relatively short names, say, of 10 or fewer components, that correspond to globally routable FIB prefixes, and longer components that refer specifically to names of Data packets. A 640-bit wide TCAM could store the relatively short FIB entries in their entirety, but not the relatively long PIT and Content Store entries. Suppose, however, that the first 640 bits of the full name would yield a result that both indicated that this was a partial match, and some unique identifier (such as a 16-bit hash of the first 640 bits) that could be used as the initial prefix for a subsequent lookup starting with the 641st bit of the full name. With each partially-matched key yielding a unique identifier for use as a prefix in a subsequent lookup, sets of full length names that only differ in their suffix bits would, through multiple TCAM lookups, eventually be uniquely identified. For perspective, consider that 10 sequential lookups performed in this manner, which used 640 name bits in the first lookup and 640-16=626 in the 9 subsequent lookups would uniquely identify names 6880 bits in length (or 1420 octets). Key insertion would also take multiple operations, but fortunately would be entirely deterministic and only require simple pre-processing of the name and its length to determine: A) how many partial keys must be inserted, and B) the value of the unique identifier (i.e., hash value) to use as the prefix in all insertions but the first.

Other design alternatives may lead to dramatically higher levels of performance. Since each component in an NDN name is known, NDN names can be encoded in *nested hash tables*. A hash is computed over the first component, and the result is a pointer to the next hash table, which is keyed with the hash of the second component, and so on. In the simplest incarnation, if a name consists of *k* components, then in the absence of collisions, *k* hash lookups would be required in the worst case to identify the longest matching prefix. (Since methods for provisioning hash tables to reduce collisions are well understood, in this discussion we assume for simplicity that hash accesses will require one memory access.) While dependence on *k* is much better than dependence on the number of bits in the name (as would be the case with a trie-based implementation), it is still far from a constant time lookup.

To see how we might move closer to a constant-time solution, consider that the nested hash lookups must occur sequentially. If you know a given name only matched prefixes of component lengths 3 and 11, then with a properly populated single hash table, you could perform lookups only with keys of component lengths 3 and 11. It is possible to design a prefix lookup implementation with an on-chip oracle that indicates which prefix lengths should be checked. Bloom filters are compact filters that determine set-membership in constant time. It is possible to maintain a Bloom filter for each prefix length, and when a prefix consisting of *k* components is inserted into the prefix table, it would also be programmed into the *k*th Bloom filter. The filter subsystem could be organized to check all of the length-specific on-chip filters concurrently. The 1st filter uses the first component as its key, the 2nd uses the first and second components together as its key, and so on. The result of the filtering stage is the set of prefix lengths, and hence name prefixes, that should be looked up in the off-chip prefix hash table. Since the objective is to find the longest matching prefix, only the largest prefix needs to be sought off-chip. Hence, only a single off-chip hash table lookup would be needed, resulting in a constant time longest-prefix match.

Two facts impinge on this ideal outcome. First, Bloom filters are correct but probabilistic, meaning that there is a chance for false positives in which a filter falsely indicates that a prefix match exists for a given length. So, more than one lookup may be necessary on occasion. (They will never miss one however, so this will not exhibit false negatives.) The odds of experiencing a false positive depend on occupancy of the hash table, which in turn depends on the number of keys inserted and the overall capacity. It is well understood how to provision Bloom filters for a given probability of false positives but since the length of NDN names is unbounded, there is no upper bound on the number of Bloom filters that may be needed (one for each prefix size). This basic scheme has already been evaluated in the context of IP lookup for IPv4 and IPv6 [20], demonstrating that modern, cost-effective ASICs have resources sufficient to maintain between 32 and 100 concurrent on-chip Bloom filters. We plan to explore the effectiveness of this *Bloom-filter accelerated longest prefix match* approach to achieve constant time operation on very large tables. We will build on our recent research on designing segmented hash tables [49], which included careful design and analysis of high-level concerns such as collision resolution policy all the way down to low-level Bloom filter circuit optimization strategies.

3.2.2 Forwarding Strategy

Forwarding strategy is a key component in NDN nodes that makes them more powerful than their IP counterparts. Instead of relying solely on routing to calculate a single best path for each destination, the forwarding strategy layer in a NDN node (either a router or an end node) can dynamically select multiple interfaces from the FIB to forward each Interest packet. This real-time decision enables nodes to fully utilize their rich connectivity, and to defend against route hijacking attacks – if no data returns over a particular interface for a particular name, that interface may not lead to a valid path for that name. In addition, the strategy layer provides traffic control by adjusting the number of unsatisfied Interests allowed. At an end node, the strategy component decides when to retransmit an unsatisfied Interest, and through which interface, providing support for multihomed hosts.

The simplest strategy is to send an Interest on each of the interfaces in a FIB entry in sequence – if there is no response to the Interest, then try the next interface. We can also send Interests on all the interfaces at once and see which interfaces receive the data first – these interfaces will be used for a period of time and their performance monitored. If their performance level degrades below a certain level, another experiment can be performed. Open research issues include (1) selecting performance metrics to rank interfaces, e.g. delay or throughput; and (2) avoiding instability (frequent oscillation of paths) while maintaining good data delivery performance. A more flexible design is for each FIB entry to contain a program specialized to make Interest forwarding decisions. The instructions for this machine should include a small subset of the normal load/store, arithmetic, and comparison operators that can be used to invoke actions when significant events occur. We will first experiment with the simpler strategies to understand their performance and then investigate the feasibility of the programmable strategy.

3.2.3 Caching Policy and Storage Management

NDN routers use packet buffers that already exist in IP routers as caches; a cache hit means reduction in bandwidth usage. With a reasonable caching policy, we expect NDN networks to perform better than IP for static data, and no worse than IP for dynamic data, in reducing both bandwidth demand and original server load. We discuss how we will address four caching issues. First, we will investigate ***cache replacement policies*** for NDN routers. While prior work on this subject suggested that a least-recently-used (LRU) or least-frequently-used (LFU) policy would be desirable, we hypothesize that a far simpler random replacement policy would perform nearly as well. We plan to study this issue both analytically and experimentally. We will base any more advanced schemes on a rich set of literature on caching [70]. Second, we can avoid storing ***stale data*** by letting publishers assign a TTL with each piece of data, a proven approach used by the DNS. Third, to prevent ***cache pollution attacks***, the routers need to verify data signatures. We plan to investigate fast verification schemes that can significantly reduce the chance of a successful attack without necessarily detecting all bad signatures (Section 3.4). Fourth, ***DDoS attacks against caches*** are much more difficult in NDN than in IP since an attacker cannot simply flood useless data without preceding Interests. We discuss DDoS attacks in more depth in Section 3.4.

3.3 Driver Applications

Our application research agenda aims to (1) drive architecture development based on a broad vision for future applications; (2) drive and test the prototype implementations of the architecture; (3) demonstrate performance and functional advantages of NDN in key areas; and (4) show how NDN's embedding of application names in the routing system promotes efficient *authoring of sophisticated distributed applications*, reducing complexity and thus opportunities for error, as well as time and expense of design and deployment. Burke has argued since 2002 that named data networking would transform *authoring* of hybrid physical and digital environments [10]. We plan to develop and deploy four prototype applications of increasing societal interest: a web browser protocol handler, a media streaming application, instrumentation to support a heterogeneous, media-rich cyber-physical system; and a *participatory sensing* application that can leverage commodity mobile phone platforms to enable global, personally-regulated sensing and information distillation.

3.3.1 Mainstream or "traditional" applications

We plan to use actual application traffic across the NDN testbed to drive the experimentation and evaluation of the core NDN design and implementation early on. First, we plan to implement a protocol handler for a standard web browser (e.g., Firefox), so that the browser can use NDN for transport. This will give us the first experience of running applications use the communication functions provided by NDN, as well as providing NDN traffic to exercise the forwarding machinery and observe both the performance of the forwarding strategy and the impact of caching. Second, we will leverage PARC's recent experience implementing Voice-Over-CCN to create a bi-directional secure high definition (HD) video and audio streaming service that uses named data packets, leverages content caching to provide multicast as well as point-to-point media distribution, offers multiple quality levels through the use of different name suffixes, and supports key management and time synchronization. The implementation of these services on top of NDN will help us gain further experience in utilizing the communication functions provided by NDN. Once ready, they can be used for our research coordination across different campuses and also drive the security research by providing an opportunity to study application-specific trust models. This synchronized, authenticated media multicast service will also form a key part of our more advanced media-rich instrumented environments described below.

3.3.2 Media-rich instrumented environments

Many future Internet applications will expand the vision of ubiquitous computing [71] to high definition content and interactivity, integrating sensing and control, distributed processing, and user interfaces, at scales and complexity far beyond today's applications. Educational, creative, and simulation environments should be enabled by the future Internet to easily, efficiently, and securely incorporate components crossing all of our emerging cyber-physical infrastructure. (Consider even the most trivial example of how difficult it is to have an authorized slide presentation on a guest laptop dim a room's lights in an IP networked environment, and how simple it could be in NDN with local naming, broadcast communication, and signed data.) Although the subsystems already exist and even have IP connectivity, applications have been severely limited in their ability to coherently and securely incorporate them. We expect such subsystems to become more prevalent in the future, motivated by concerns about issues such as energy management and physical intrusion detection. For example, future buildings are likely to be constructed with digitally-controlled, addressable lighting and environmental systems, access control systems with a variety of presence, flow, and identity sensing, touch-activated networked displays and projection, paging or sound systems, and video recording or broadcasting capabilities. Practical public deployment of applications on such *heterogeneous, experiential cyber-physical systems* has been limited by the host- and connection-based approach of the current Internet architecture, with its finite addressing schemes and security management difficulties, that impose major authoring challenges requiring substantial development resources. NDN's intrinsic support for naming data, broadcast, caching, and fine-grained authentication provide obvious advantages to future content-centric application developers.

We will create NDN interfaces for representative, IP-enabled SCADA, sensing, and multimedia subsystems, and connect them to the testbed NDN network to explore security, performance, and addressing features enabled by the architecture. Exploration of this area expands on UCLA's experience in building interactive spaces of the kinds found in museums, theme parks, live performances, and physical simulation spaces. We will use these components to implement simulation, education, or creative experiences in collaboration with UCLA's theater and engineering schools, which have incorporated sensor research into film sets, performances, and architectural projects. We imagine work on this application will also help prioritize directions for NDN architecture research: (1) application library support for name prefix publishing; the effective use of multiple namespaces, and other naming challenges; (2) the mDNS-style discovery of names by embedded devices; (3) support for distributed, synchronized state management that leverages the properties of NDN; and (4) effective key management and distribution for control packet authentication.

3.3.3 Participatory sensing

A growing ecosystem of human-centric sensing applications that use personal and community-scale participatory data collection have been fueled by the proliferation of sensing devices accessible to large consumer populations, starting with the billions of deployed mobile phones and now expanding into active RFIDs, smart residential wireless power meters, in-vehicle GPS devices, sensor-enhanced entertainment platforms (e.g., Wii-fit), and activity monitoring sportsware (e.g., the Nike+iPod system). These applications are reaching mature market penetration, offering unprecedented opportunities for data collection and sharing, and reflect the impedance mismatch between the exponentially growing ability of technology to generate new data, and the fixed human capacity to absorb it. An ecosystem of new applications is arising whose main purpose is to distill such data into actionable information [10]. Data collected for such applications includes GPS trajectories to monitor traffic patterns [28], pollution traces to assess environmental impact [29], vehicular fuel-efficiency measurements to find green routes [27], and health data. Deployment of participatory sensing applications on IP networks is challenged by the need to (1) publish and subscribe to information in the presence of mobility, (2) locate and extract information from diverse data, and (3) satisfy heterogeneous requirements for data privacy, fidelity, legitimacy, and security. NDN removes the above hurdles from application development by enabling communications on names and secure data directly.

The named-data paradigm also facilitates application-level performance optimization. In a network whose main function is information distillation, the value of information becomes an important optimization metric. For example, in an application where both images and vehicular GPS traces are used to estimate traffic speed, if sufficient GPS data exists then the pictorial data may not be cost-effective. By controlling interests in logical data names in a named-data network, information distillation applications can automatically reinforce or discourage collection of the right types of data, or support multiple levels of fidelity via network coding across different name prefixes, such that the network is optimized for the value of information.

We will build a secure data ecosystem for participatory sensing, composed of mobile data source and user devices, secure data storage, and privacy-preserving data processing and sharing services. An anchor data type will be individual mobility patterns (time-location "traces") around which to organize and correlate other collected personal data. We will extend the host-centric "personal data vault" concept developed at UCLA and USC to create a geographically distributed *personal data cloud (PDC)* that protects an individual's data, bound by contractual or statutory legal protections against unauthorized use, similar to banks, telecom conversations, or privileged communications. Note that the deployment of these new applications will have *mobility* support embedded in. The basic operation of NDN, *i.e.*a consumer expresses interest for communication and data flows back following the state set up by the Interest packet, can be directly used to provide mobility support for moving receivers. For moving senders, on the other hand, the mobile's Interest packet will inform the receiving end to send Interest towards the mobile to retrieve the data.

Creating the PDC will require the implementation of a distributed database on NDN, providing a widely applicable test case that leverages the architecture's features and supports content distribution systems implemented for the media-sensing application. The PDC will also inform NDN requirements for data discovery, caching, and diverse models for trusted communication among mobile users, PDCs, and applications, including: (1) disruption tolerant upload from sensors to the data store; (2) internal communication between nodes of the PDC; (3) filtered data sharing with authorized third-party applications providing personal services to the individual; (4) filtered, often anonymous data sharing to aggregators.

3.4 Security and Privacy

A fundamental security primitive is embedded in the "thin waist" of NDN: the name in each NDN packet is bound to packet content with a signature. This basic feature provides data integrity and origin authentication, as well as machinery to support trust and provenance by mapping between the packet signer and its source (e.g., an individual or an organization). Named and signed content also forms a more solid foundation for building secure applications (Section 3.3), but poses two major scaling challenges: cost-effective fine-grained signature operations, and functional and usable trust management infrastructure. Our primary security research approach is to design and implement practical and general-purpose security mechanisms to support particular applications and routing components created in the project (Sections 3.3, 3.1), and to evaluate and improve these mechanisms based on testbed experimentation. We recognize that NDN is not a security panacea and does not offer solutions for all network security challenges. NDN provides fundamentally new and powerful mechanisms for validating content and determining its provenance, but malicious or unwanted content such as spam can (*e.g.*, due to malware) originate at a legitimate content source and bear a legitimate signature.

3.4.1 Efficiency of Signatures

The need to sign and verify every packet suggests an ominous need for efficiency in signature generation, transmission, verification, and possibly storage. Fortunately, recent research suggests that per-packet RSA signatures for real-time data (*e.g.* voice) are practical on commodity end-user platforms today [43]. Furthermore, NDN does not expect or require signatures to be verified by core routers.

We plan to develop computation- and bandwidth-efficient signature schemes for NDN. We will apply work in fast signature schemes (*e.g.*[59]), techniques that allow semi-trusted proxies to sign on behalf of weak devices [22, 23, 21], as well as methods for *aggregate* signature generation and verification (from simple and efficient Merkle Hash Trees (MHTs) [52] to recent cryptographic research using bilinear maps over elliptic curves [8, 57] and related *batch verification schemes e.g.* [1, 39]). We will also investigate techniques for spreading signature meta-data over multiple packets. Some of these schemes have not yet seen practical deployment, but variants of them are promising for NDN. Verification cost will likely be the most important factor among signature-related challenges, since a signature is generated once but may be verified many times.

3.4.2 Usable Trust Management

Signature verification of NDN content merely indicates that it was signed with a particular key. Making this information useful to applications requires managing *trust* – allowing content consumers to determine acceptable signature keys in a given context. NDN provides an excellent platform for deploying both accepted and new trust management models. Keys can be treated as named NDN data and signed NDN data items effectively function as certificates. NDN can express secure *links* between pieces of content, allowing certification of not only keys, but of content itself. This provides a rich substrate where many pieces of linked "evidence" can support consumer trust in a particular piece of content. For example, a consumer might verify the front page of the *N*ew York Times because it is signed with a well-known certified key. She can then verify individual articles because the front page links securely to them. One advantage of NDN is that it does not require a "one size fits all" trust model: trust is end-to-end, between producer and consumer. Different consumers and different content may require varying levels of assurance. However, to make NDN accessible and deployable, it must come "out of the box" with a set of usable trust mechanisms applicable to a wide range of applications.

Prior research in trust management for large-scale deployment of public key cryptography has resulted in two main approaches: hierarchical Public Key Infrastructure (PKI) [17], and peer-level PGP web of trust [76], both with significant usability problems [37, 73]. Recent research shows that these approaches can be made more user-friendly [38, 5] and constructs new ways for automatically developing trust in keys through observation and experience [36, 61, 72]. One model of particular relevance to NDN is SDSI/SPKI [64, 26, 2], which maps a small-world model of trust onto a notion of local "namespaces" for naming keys, which in turn can map directly into the NDN notion of content namespaces. We will experiment with such models and evaluate them in the context of the applications and routing work.

We also plan to design solutions for **revocation**, the problem of deciding when a key (or credential) can no longer be trusted, which is particularly acute in NDN due to caching. Faced with signed content that may have been cached for a while, we need to determine whether the corresponding key (1) has been revoked, and (2) was valid at the time of signing. Although current revocation approaches (CRLs, OCSP [56]) can be used with NDN, they will likely be even less effective than in more traditional contexts.

3.4.3 Network Security and Defense

Any future Internet architecture must offer improved protection and resilience over today's network, which is subject to pervasive and persistent attacks. NDN network security is based on the establishment of a trustworthy routing mesh, relying on signed routing messages and an appropriate trust model unlike today's Internet. NDN packets address content instead of end-points, making it difficult to target a particular host, and the fact that NDN nodes forward data only in response to an incoming interest makes it impossible to flood unsolicited data through an NDN network.

The research challenges for NDN network security are 1) designing a trust model to defend against attacks on the routing mesh while supporting common providers' practices and policies, and 2) designing defenses against new types of attacks. We will design trust models appropriate to each of our routing research approaches (Section 3.1), and implement and evaluate them in prototype routing components and experimental deployments. We will address **Interest Flooding Attacks** (mirroring traditional denial of service (DoS) attacks) which send large numbers of new and distinct interests that cannot be aggregated or satisfied from caches, and **Content Pollution Attacks** which introduce malicious content purporting to match legitimate requests. For Interest Flooding Attacks, we plan to conduct experiments with routers that throttle the number of unsatisfied interests they will hold for a given target domain. For Content Pollution Attacks, the consumer should always use signature verification to reject malicious content, but we also plan to evaluate the burden of ingress filtering and egress filtering in (non-core) routers to protect against simulated attacks. We recognize other possible attacks, such as "hiding" content from legitimate requesters and abusing cryptographic operations to mount DoS attacks, which we hope to enable other researchers to investigate.

3.4.4 Content Protection and Privacy

Integrity, provenance and trustworthiness of content are necessary but not sufficient; content publishers want fine-grained access restrictions on sensitive or valuable content, and consumers want to maintain their privacy by not exposing information about content they retrieve. Since NDN consumers are likely to obtain desired content from caches rather than the original publisher, the latter cannot rely on entities (hosting caches) to enforce its access control policies. Therefore, NDN adopts the familiar content-based approach to access control, obtained primarily via content encryption. Encryption is end-to-end and largely opaque to the network layer, handled by applications or libraries. Most schemes for protecting content by encryption (*e.g.* broadcast encryption [30, 46, 58, 9] and encryption-based access control schemes [3, 6, 4]) can be adapted to NDN by choosing appropriate naming schemes and data representations for keys. We plan to select and implement appropriate schemes for specific applications (Section 3.3), prioritizing efficiency and compatible support for revocation. We will also examine *content firewalls*, whose atomic unit of protection is content referenced by name, and which provide another method of user-friendly perimeter control for restricted content.

Although an NDN interest refers to a potentially human-readable name, NDN implicitly offers better end-to-end privacy, since it only tracks what data is being requested, and not *who* is requesting it (Section 2.2.5). However, NDN poses three important privacy challenges: **(1) Cache privacy**, because as with current web proxies, network neighbors may learn about each others' content accesses using timing information to identify cache hits; **(2) Name privacy**, since the more meaningful NDN content names are, the more sensitive they may be; and **(3) Signature privacy**, because the identity of a content signer and its revocation status may leak sensitive information about individuals and organizations. Various methods of addressing these challenges, *e.g.*, VPN-like tunnels for Name Privacy, offer different trade-offs between privacy and cacheability.

We can only scratch the surface of the privacy models possible in the NDN architecture, but to explore the limits of attainable privacy, we will design and implement NDN analogs to Tor [24] and/or Mixnets [14] to maximize data path diversity and cache dispersal. We also plan to explore practical applications of *group signature* schemes, [13] where any member of a group can sign on behalf of the group, and anyone can verify a group signature, but the identity of the signer remains private and no connection can be made between multiple group signatures.

3.5 Fundamental theory for NDN

Information theory [67] provided the architectural basis for both the digital telephone network and the IP data network. However, classical information theory is connection-based, limiting its utility even for IP, much less for an name-based architecture such as NDN. In a connection-based setting with a fixed number of source-destination pairs, the multi-dimensional generalization of classical point-to-point Shannon capacity is the *network capacity region*, consisting of all the source-destination communication rates achievable using any feasible coding scheme [18]. The corresponding *coding* problem is to find low-complexity error correcting codes which allow for reliable communication at the rates given by the capacity region. To realize the true potential of the NDN paradigm, we need a new fundamental theory of networked communication, which poses some key challenges.

In classical information theory, information streams sent over different source-destination pairs are usually assumed to be independent. In NDN, interest packets for a popular network application, and the data traffic generated in response, are likely to be highly statistically dependent, suggesting a natural role for *multicast*. For multicast with a single source, optimal network coding can minimize both the bandwidth and computational complexity required to send a fixed number of packets [40]. But NDN networks will typically have multiple data streams of interest to different sets of receivers, so we need to develop optimal coding theory and practice for multiple-source multicast. The critical role of caching data in NDN also raises the importance of optimal asynchronous or delay-tolerant multicast. A theory for NDN must capture the essential tradeoff between wires and storage in optimizing communication performance, so that for a given set of link capacities and buffer sizes, we can analyze combinations of transmission and caching strategy to optimize network performance. Two additional issues not prominent in classical theory are the *time value of information* (information cached too long may be useless once reaching receivers) and the central role of mobility in assessing the tradeoff between storage and transmission [34].

In developing appropriate performance measures for NDN, we must let go of the concept of source-destination transmission rate, and focus on the more meaningful *destination reception rate*. To discount for redundant content, we could use a metric such as *total consumed entropy rate* in bits per second, *i.e.*, the total rate of statistically independent information arriving at destinations requesting content. This definition accounts for transmission of information over time, and appropriately corrects for redundant content, but does not capture the impact of storage and mobility on transmission of information across space. In cases where spatial transmission is important, we could measure NDN capacity in bit-meters (one bit of information transported 1 meter toward a destination) per second [35], which would also account for redundant information requested at far-away destinations.

Once we have an appropriate performance metric, the fundamental questions are the same as in classical information theory: for a given set of link capacities, buffer sizes, and data generation, what are the achievable capacity regions and complexity costs for feasible joint routing, scheduling, coding, and buffer management schemes? PARC's CCN has provided one possible achievable scheme corresponding to a subset of the capacity region. Eventually we need new schemes that will allow us to explore the entire region. We will first characterize the macroscopic scaling laws governing NDN network capacity, in a manner similar to [35], which will convey a useful starting picture for analyzing the scalability of NDN. We will then incorporate latency, storage, and fairness issues that prevent all points in the network capacity region from being equally desirable, and will likely reveal optimal policies that differ significantly from those established in classical network theory literature.

3.6 Implementation and Deployment

We will prototype the architecture in software, running protocols as an overlay on existing packet transmission networks as discussed in Section 2. There will be four distinct categories of software: 1) applications (Section 3.3), 2) application libraries that support common functions across many applications, 3) generic infrastructure components that implement routing, forwarding, and persistent data packet storage on general purpose computing and mobile phone platforms, and 4) programmable router software. Application and routing software may have special hardware requirements, such as those to perform forwarding described in Section 3.2. Our software prototype will reveal new issues in the initial NDN design and provide practical feedback to guide the evolution of the NDN architecture throughout and beyond the initial project.

Evaluation	Key Metrics	Method
Routing and Data Delivery	FIB size, PIT size and lifetime, routing & Interest message overhead, successful path probability and length distribution, delay in finding data, optimal number and diversity of paths,	T, S, A
Hardware	FIB update delay, packet processing delay (including lookup delay)	T
Caching	cache size, hit ratio as a function of content type	T, S
Flow Control	interface throughput, link utilization	T, S
Application support	ease of creating applications (e.g. closeness of mapping between application needs and network support [32]), application-level data throughput	T
Privacy	privacy preservation capability of TORNADO (NDN version of TOR)	T
Data security	speed of generating and verifying signatures	T
DDoS	percentage of requested data delivered to legitimate users	T, S
Capacity and Traffic	total amount of information transported by the network in space and time (i.e. consumed entropy rate), traffic patterns compared with IP	T, A

Table 1: Evaluation Metrics and Methods (T - testbed measurement, S - simulation, A - theoretical analysis)

We will release all NDN software as open source and contribute to the development of an an active open source community. We will use and extend the early version that PARC has already released under GPL[2], and/or release complementary, interoperating software separately, *e.g.*, to showcase particular applications. Some of the instrumentation we release to support performance evaluation (Section 3.7) will also support operations research on scalable management and monitoring of NDN networks, e.g., peering policy configuration management and traffic engineering.

Initially, islands of NDN nodes will be few and sparsely distributed, with some locally dense and highly communicative NDN islands. We plan to develop a Rendezvous Point solution to self-discovery for NDN nodes, and interconnect islands by tunneling over the non-NDN cloud. If NDN applications gain wide reception, ISP deployment of NDN routers will improve performance for themselves as well as their customers, providing a natural incentive for infrastructure growth. Based on our study of the long history of failed architectural innovation in the Internet, we will carefully consider how to minimize the cost and effort involved in transition. Critically, NDN will run fine over existing IP infrastructure, so ISP's can evolve their network slowly by adding individual NDN routers alongside IP gear without disruption. Our initial set of compelling experimental applications will illustrate the advantages of NDN over IP for increasingly desirable uses of the network, providing additional deployment incentives.

3.7 Evaluation and Assessment

To evaluate the NDN architecture, we will use quantitative and qualitative metrics that reflect its ability to support existing and new applications, as well as the performance, scalability, and robustness of NDN networks. Our evaluation methods will include measurements (including user surveys) of NDN-connected testbeds using our implementation (Section 3.6), simulations, and theoretical analysis to identify problems in the design and evaluate performance at different scales. Table 1 summarizes the key metrics and evaluation methods for each project area. In addition to the overlay testbed network composed of NDN nodes (Section 3.6), we will investigate the use of Open Network Lab (ONL) [60] and the GENI-resident Supercharged PlanetLab Platform (SPP) [69, 31] as testbeds. Our simulations will use the best available data on Internet behavior, using traffic data from the DHS PREDICT project, AS-level topology derived from Route-Views data [68] and annotated router-level topology data from CAIDA. Theoretical analysis will allow us to cross-validate against experimental results, e.g., regarding topology structure and network capacity.

Since comprehensive evaluation of the NDN architecture may require wider-scale deployment than we can accomplish in this initial project, our implementation plans include instrumentation that facilitates macroscopic performance evaluation, including software support for measurement of metrics in the first five lines of Table 1. Instrumenting NDN to support and leverage collection of data about faults while respecting

[2] PARC's project on the NDN direction is called CCN, hence the PARC open source package is dubbed CCNx.

legitimate privacy concerns will provide incentives to gather and share measurements.

To evaluate the privacy preservation capability supported by NDN, we plan to implement an NDN version of TOR [25], which we call TORNADO. Using the published known vulnerabilities of TOR [62], we will emulate the same attacks on TORNADO. For broader security evaluation, we will build on our experience with penetration testing games [16] to host "red team" competitions where groups not involved in NDN design attempt attacks against participating NDN testbeds. We will emulate and evaluate the impact of different types of Denial of Service attacks using both simulations and testbeds.

4 Education

Architectural thinking is a type of *computational thinking* that encompasses system principles, invariants, and design trade-offs. As networked computing systems such as the Internet grow rapidly in complexity along with our dependence on them, the ability to rigorously understand the fundamentals of network communication architectures becomes only more important. We desire to use the NDN prototype to help teach students how to assess the systemic impacts of design choices, training a future generation of engineers how to make design choices more likely to strengthen a networked system rather than weaken it.

We plan to develop materials to teach architectural thinking integrated with NDN-enabled research: (1) comparative study of the Internet and NDN designs, (2) development and evaluation of NDN applications, and (3) graduate research. Through our applications, we seek to serve and inspire interdisciplinary collaboration across scientific, medical, educational, and cultural research communities.

We will use the NDN design as new curriculum material to encourage students to view networking in new ways. Today instructors strive to teach architectural concepts, but unfortunately students often focus on mechanics, e.g., *what* fields are in an IP header, rather than *why* those fields are in the IP header. We plan to create and distribute an introductory lecture suitable for inclusion in standard undergraduate networking courses, which compares NDN to the traditional IP architecture, and encourages students to challenge established networking models from architectural viewpoints. By presenting a real alternative, with a significant and growing deployed infrastructure base amenable to evaluation, we can challenge students to think deeply about network design questions. Students will be exposed to NDN architectural concepts in the context of real software challenges, and use APIs to the core NDN layer to build software that explores these concepts. Implementing projects under both IP and NDN paradigms will inform theoretical and practical discussion of networking concepts. Running the NDN variants on our testbeds should lead to unexpected network behavior that challenges testbed operators as well as the students! Supplementing the courses will be a multi-campus lecture series named "the Science of Routing" (started in our previous NSF FIND project).

One way to teach architectural thinking is to examine the history, to compare different design choices at the time, to see what design decisions led to what kinds of consequences, both expected and unforeseen, and to revisit those decisions given what we have learned since. In addition to teaching architectural thinking from history and emerging NDN architecture, we plan to record exchanges and discussions from the NDN design and development efforts, capturing the *process* of creating a new architecture to enable such research and teaching in the future.

5 Summary

The Internet has been a huge success, but the world has changed dramatically since it was created. The Internet architecture is no longer a good match to its primary use, so how do we design a new architecture that addresses today's problems and will take us into the future with even greater success than the past? Our answer is conceptually simple: generalize the Internet architecture by replacing the focus on *where* – endpoint addresses of hosts – with *what* – identifiers of the *content* that users and applications care about. In this report, we have proposed a new Internet architecture called "Named Data Networking" (NDN) based on this simple change.

We have sketched an initial blueprint of this new NDN architecture, and identified a core set of research problems that must be investigated in order to develop and validate it. These intellectual challenges include scalability of routing on names, fast forwarding based on variable-length hierarchical names (including per-packet state updates needed for multipath forwarding and caching), efficiency of signatures for finely-granular signing and verification, usable trust models for data-centric security, network security and defense, content protection and privacy, and fundamental communication theory that incorporates memory as an explicit part of the communication model.

In order to develop NDN from this initial blueprint into a proven and deployed architecture, we must grow a community to research and experiment with this new architecture at scale. We offer this report to the research community as a statement of the NDN architectural direction and research agenda, and invite others to join the effort to advance a named-data approach to some of the Internet's most pressing problems in security, scalability, sustainability, and stewardship.

References

[1] Fast batch verification for modular exponentiation and digital signatures. In *Advances in Cryptology - Eurocrypt 1998*, pages 236–250, 1998.

[2] Martin Abadi. On SDSI's Linked Local Name Spaces. *Journal of Computer Security*, 6(1-2):3–21, October 1998.

[3] Selim G. Akl and Peter D. Taylor. Cryptographic solution to a problem of access control in a hierarchy. *ACM Trans. Comput. Syst.*, 1(3):239–248, 1983.

[4] Mikhail J. Atallah, Marina Blanton, Nelly Fazio, and Keith B. Frikken. Dynamic and efficient key management for access hierarchies. *ACM Trans. Inf. Syst. Secur.*, 12(3):1–43, 2009.

[5] Dirk Balfanz, Glenn Durfee, and D.K. Smetters. Making the impossible easy: Usable PKI. In Lorrie Faith Cranor and Simpson Garfinkel, editors, *Security and Usability – Designing Secure Systems that People Can Use*, chapter 16, pages 319–334. O'Reilly Media, Inc., 2005.

[6] Matt Blaze. A cryptographic file system for unix. In *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*, pages 9–16, New York, NY, USA, 1993. ACM.

[7] M. Boguñá, D. Krioukov, and kc claffy. Navigability of complex networks. *Nature Physics*, 5:74–80, 2009.

[8] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Advances in Cryptology - Eurocrypt 2003*, pages 416–432, 2003.

[9] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO '05: Proceedings of the 25th Annual International Cryptology Conference on Advances in Cryptology*, pages 258–275. Springer-Verlag, 2005.

[10] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing.

[11] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, and I. Stoica. ROFL: routing on flat labels. *ACM SIGCOMM Computer Communication Review*, 36(4):374, 2006.

[12] A. Chaintreau, P. Fraigniaud, and E. Lebhar. Opportunistic spatial gossip over mobile social networks. In *WOSN*, 2008.

[13] David Chaum and E. van Heijst. Group signatures. In *EUROCRYPT*, pages 257–265. Springer-Verlag, 1991. LNCS 547.

[14] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, 1981.

[15] David D. Clark, John Wroclawski, Karen R. Sollins, and Robert Braden. Tussle in cyberspace: defining tomorrow's internet. In *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 347–356, New York, NY, USA, 2002. ACM.

[16] M Collins, D Schweitzer, and D Massey. Canvas: a regional assessment exercise for teaching security concepts. *Proceedings from the 12th Colloquium for Information Systems Security Education*, 2008.

[17] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. IETF, May 2008. RFC 5280.

[18] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, 1991.

[19] E. M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant MANETs. In *MobiHoc*, 2007.

[20] S. Dharmapurikar, P. Krishnamurthy, and D. E. Taylor. Longest prefix matching using bloom filters. 2003.

[21] Xuhua Ding, Daniele Mazzocchi, and Gene Tsudik. Equipping smart devices with public key signatures. *ACM Trans. Internet Techn.*, 7(1), 2007.

[22] Xuhua Ding and Gene Tsudik. Simple identity-based cryptography with mediated rsa. In *CT-RSA*, pages 193–210, 2003.

[23] Xuhua Ding, Gene Tsudik, and Shouhuai Xu. Leak-free group signatures with immediate revocation. In *IEEE ICDCS*, pages 608–615, 2004.

[24] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[25] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *USENIX*, 2004.

[26] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian M. Thomas, and Tatu Ylonen. *SPKI Certificate Theory*, September 1999. RFC2693.

[27] Ganti et al. Greengps: A participatory sensing fuel-efficient maps application.

[28] Hull et al. Cartel: a distributed mobile sensor computing system. 2006.

[29] Mun et al. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. 2009.

[30] Amos Fiat and Moni Naor. Broadcast encryption. In *CRYPTO '93: Proceedings of the 13th annual international cryptology conference on Advances in cryptology*, pages 480–491, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[31] The GENI initiative. `http://www.geni.net/`.

[32] T. R. G. Green. Instructions and descriptions: some cognitive aspects of programming and similar activities., booktitle = Proceedings of Working Conference on Advanced Visual Interfaces (AVI 2000), pages = 21-28, publisher = ACM Press, year = 2000, editor = V. Di Ges and S. Levialdi and L. Tarantino.

[33] M. Gromov. *Metric Structures for Riemannian and Non-Riemannian Spaces*. Birkhäuser, Boston, 2007.

[34] M. Grossglauser and D. N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE Trans. on Networking*, 10:47–86, Aug. 2002.

[35] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Trans. on Information Theory*, 46(2):388–404, Mar. 2000.

[36] Peter Gutmann. Underappreciated security mechanisms. `http://www.cs.auckland.ac.nz/~pgut001/pubs/underapp`

[37] Peter Gutmann. Pki: It's not dead, just resting. *IEEE Computer*, 35(8):41–49, 2002.

[38] Peter Gutmann. Plug-and-play PKI: A PKI your mother can use. In *Proceedings of the 12th USENIX Security Symposium*, pages 45–58, Washington, D.C., August 2003.

[39] L. Harn. Batch verifying multiple DSA-type digital signatures. *Electronics Letters*, 34:870–871, 1998.

[40] T. Ho, R. Koetter, M. Médard, M. Effros, J. Shi, and D. Karger. A random linear network coding approach to multicast. *IEEE Trans. on Information Theory*, 52(10):4413–4430, Oct. 2006.

[41] P. Hui, J. Crowcroft, and E. Yoneki. BUBBLE rap: Social-based forwarding in delay tolerant networks. In *MobiHoc*, 2008.

[42] John Ioannidis and Steven M. Bellovin. Router-based defense against ddos attacks. 2002.

[43] Van Jacobson, Diana K. Smetters, Nicholas H. Briggs, Michael F. Plass, Paul Stewart James D. Thornton, and Rebecca L. Braynard. Voccn: Voice-over content centric networks. pages 1–6, 2009.

[44] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. Networking named content. pages 1–12, 2009.

[45] Dan Jen, Michael Meisel, He Yan, Daniel Massey, Lan Wang, Beichuan Zhang, and Lixia Zhang. Towards A Future Internet Architecture: Arguments for Separating Edges from Transit Core. In *ACM Workshop on Hot Topics in Networks*, 2008.

[46] Mike Just, Evangelos Kranakis, Danny Krizanc, and Paul van Oorschot. On key distribution via true broadcasting. In *CCS '94: Proceedings of the 2nd ACM Conference on Computer and communications security*, pages 81–88, New York, NY, USA, 1994. ACM.

[47] D. Krioukov, F. Papadopoulos, M. Boguñá, and A. Vahdat. Greedy forwarding in scale-free networks embedded in hyperbolic metric spaces. *ACM SIGMETRICS Perf E R*, 37(2):15–17, 2009.

[48] D. Krioukov, F. Papadopoulos, A. Vahdat, and M. Boguñá. Curvature and temperature of complex networks. *Phys Rev E*, 80:035101(R), 2009.

[49] S. Kumar and P. Crowley. Segmented hash: An efficient hash table implementation for high-performance networking subsystems. 2005.

[50] Paul Laskowski and John Chuang. Network monitors and contracting systems: competition and innovation. In *SIGCOMM*, pages 183–194, New York, NY, USA, 2006. ACM.

[51] Dan Massey, Lan Wang, Beichuan Zhang, and Lixia Zhang. A Scalable Routing System Design for Future Internet. In *Proc. of ACM SIGCOMM Workshop on IPv6*, 2007.

[52] Ralph Charles Merkle. *Secrecy, authentication, and public key systems.* PhD thesis, 1979.

[53] A. Miklas, K. Gollu, K. Chan, S. Saroiu, K. Gummadi, and E. de Lara. Exploiting social interactions in mobile systems. In *UbiComp*, 2007.

[54] A. Mtibaa, A. Chaintreau, J. LeBrun, E. Oliver, A.-K. Pietilainen, and C. Diot. Are you moved by your social network application? In *WOSN*, 2008.

[55] A. Mtibaa, M. May, M. Ammar, and C. Diot. PeopleRank: combining social and contact information for opportunistic forwarding. In *INFOCOM*, 2010.

[56] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 2560 (Proposed Standard), June 1999.

[57] Einar Mykletun, Maithili Narasimha, and Gene Tsudik. Authentication and integrity in outsourced databases. In *NDSS*, 2004.

[58] Dalit Naor, Moni Naor, and Jeffrey B. Lotspiech. Revocation and tracing schemes for stateless receivers. In *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 41–62, London, UK, 2001. Springer-Verlag.

[59] T. Okamoto and J. Stern. Almost Uniform Density of Power Residues and the Provable Security of ESIGN. In *ASIACRYPT*, 2003.

[60] Open network lab. `http://onl.wustl.edu`.

[61] Eric Osterweil, Dan Massey, Batsukh Tsendjav, Beichuan Zhang, and Lixia Zhang. Security Through Publicity. In *HOTSEC '06*, 2006.

[62] Lasse Overlier and Paul Syverson. Locating hidden servers. *IEEE Symposium on Security and Privacy*, 100-114, 2006.

[63] F. Papadopoulos, D. Krioukov, M. Boguñá, and A. Vahdat. Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces. In *INFOCOM*, 2010.

[64] Ronald L. Rivest and Butler Lampson. SDSI - A Simple Distributed Security Infrastructure. Technical report, MIT, 1996.

[65] J. Saltzer, D. Reed, and D. Clark. End-to-end arguments in system design.

[66] Vyas Sekar, Nick Duffield, and Oliver Spatscheck. Lads: Large-scale automated ddos detection system.

[67] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[68] The Route Views Project. `http://www.routeviews.org/`.

[69] Jonathan S. Turner, Patrick Crowley, John DeHart, Amy Freestone, Brandon Heller, Fred Kuhns, Sailesh Kumar, John Lockwood, Jing Lu, Michael Wilson, Charles Wiseman, and David Zar. Supercharging planetlab: a high performance, multi-application, overlay network platform. *SIGCOMM Comput. Commun. Rev.*, 37(4):85–96, 2007.

[70] Jia Wang. A survey of web caching schemes for the internet. *ACM Computer Communication Review*, 29:36–46, 1999.

[71] Mark Weiser. The computer for the 21st century.

[72] Dan Wendlandt, David Andersen, and Adrian Perrig. Perspectives: Improving SSH-style host authentication with multi-path probing. In *Proc. USENIX Annual Technical Conference*, Boston, MA, June 2008.

[73] Alma Whitten and J. D. Tygar. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *Proceedings of the 8th USENIX Security Symposium*, pages 169–183, Washington, DC, August 1999.

[74] Xiaowei Yang, David Wetherall, and Thomas Anderson. A DoS-limiting network architecture. In *Proceedings of the ACM SIGCOMM '05*, pages 241–252, 2005.

[75] X. Zhao, Y. Liu, L. Wang, and B. Zhang. On the Aggregatability of Router Forwarding Tables. In *Proc. IEEE INFOCOM*, 2010.

[76] Philip R. Zimmermann. *The Official PGP User's Guide*. MIT Press, Cambridge, MA, USA, 1995. ISBN 0-262-74017-6.