“Second Generation” NDN Applications: Design Patterns, Libraries, and Architectural Support

Jeff Burke, Peter Gusev, Jeff Thompson, Alex Afanasyev, Spyridon Mastorakis, Lan Wang, Beichuan Zhang, Lixia Zhang

ACM ICN 2018, Boston
NDN Video **Rough Cut**
(from before the talk)

https://vimeo.com/265806508
Tutorial objectives

• Share architectural concepts that the NDN team is exploring in recent applications

• Expand on previous tutorials’ overview of NDN open-source tools to cover more application needs (e.g., retrieving real time data, mobility support)

• Discuss interactions between different parts of the platform (e.g., forwarding strategy and library behavior)

• Discuss opportunities for community involvement
Organization

9:00A - Welcome and Introduction
9:45A - NDN Thinking: App Components & Design Considerations
10:15A - Low-Latency Media over NDN
11:00A - Break
11:15A - Publisher Mobility
11:45A - New Library Directions
12:15P - Concluding Discussion
12:30P – Finish, head to lunch
5:00P – Opportunity for extended discussion
Introduction

Lan Wang
Section objectives

• Introduce basic concepts in NDN architecture
• Outline support for application development
• Give an example
Future: Smart and Connected Communities

Mobile Health

Connected & Autonomous Vehicles

Smart Buildings & Homes
Mobility Support: a challenge!

Connection time between neighboring vehicles is on the order of seconds.
Security remains a major challenge!

Smart light bulbs hacked for DDOS attacks

Smart thermostat device got hacked with spyware
## Why Named Data Networking?

<table>
<thead>
<tr>
<th>Problems</th>
<th>Causes</th>
<th>NDN Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Incompatible communication models between application and network layer</td>
<td>• host-centric packet delivery</td>
<td>• Name data directly</td>
</tr>
<tr>
<td>• Difficult to support</td>
<td>• channel-based security</td>
<td>• Use same name and data unit at both application layer and network layer</td>
</tr>
<tr>
<td>• mobility</td>
<td></td>
<td>• Stateful forwarding with opportunistic caching</td>
</tr>
<tr>
<td>• multicast</td>
<td></td>
<td>• Secure data, not channels</td>
</tr>
<tr>
<td>• multipath forwarding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Complex &amp; inadequate security mechanisms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• App vision (e.g., for IoT, AR) outstrips network capabilities</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Name-based Data Retrieval

1. Name data, not the container
2. tell the network what you want..
3. let the network find it for you

/edu/memphis/lanwang/talks/IoT-061818.pdf

/livingroom/thermostat1/status

Note: this is not Google Search!
Named-based Routing and Forwarding

- Publishers advertise **hierarchical** name prefixes
- Users send **Interests** to published prefix
- “Breadcrumbs” direct **data** back to the user
- Data is **cached** into the network

Switch to AR example if time allows.
Data-Centric Security

- Data producer **signs** data with a **digital signature**.
- .. so the consumers know when they get bad data!
- .. so data can come from any node that has a copy.
NDN Architecture: Preserving the Hourglass Shape

- The Narrow waist: common interface, the network layer
  - IP – address format, IP packet forwarding
  - NDN – data name format, NDN Interest/Data forwarding
New capabilities...

- **Interest/Data exchange retrieves named data from anywhere**

- **Naming**: names facilitate key management and trust derivation.

- **End-to-end security**: data is signed by producer and verified by consumers.

- **Sync**: enable multiple parties of a distributed application to keep synchronized state

- **Repo**: persistent storage helps producers store and publish data

... yield new design patterns – discussed in the next section.
NDN Development

Application Design
(name space, trust model, data storage, data distribution, rendezvous, bootstrapping, ...)

Library Design
(messaging, security, storage, sync, ...)

Forwarder Design
(packet format, FIB, PIT, Content Store, forwarding strategy, fast forwarding, ...)

Evaluation

Codebase

Apply NDN thinking:
ChronoChat, NdnCon, nTorrent, NDNFit, Repo-ng, NLSR, NDNoT, AR Browser...

Facilitate design patterns:
ndn-cxx, NDN-CCL, ChronoSync, PSync, NDN-RTC, ...

Embody the architecture:
NFD, NFD-android, NDN-RIOT, ...

ndnSIM, Mini-NDN, NDN Testbed Tools, ...
Edge-supported Mobile Augmented Reality Application

- Intel/NSF ICN-WEN Support
- App: Mobile AR browser
- Context-context exchange between many-to-many parties
- Demonstrate low-latency edge processing support using NDN-RTC as media transport.
- Acceleration-as-a-service providing low-latency semantic scene analysis support
- Data-centric security approach
ndn-cxx Library

- NDN C++ library with eXperimental eXtensions
  - The reference library implementation

- Supports
  - sending interests
  - receiving data
  - fetching certificates
  - validating data
  - encrypting/decrypting data
  - Other functions

- Used in NFD, NLSR, ndn-tools, ChronoChat, repo-ng, etc.

- Code: https://named-data.net/doc/ndn-cxx
NDN CCL libraries provide a common API for client applications to use NDN

- NDN-CPP (C++)
- jNDN (Java)
- PyNDN (Python)
- NDN-JS (JavaScript)
- NDN-DOT-NET (C#)
- NDN-Squirrel (Squirrel)

Follows the ndn-cxx reference implementation

Used in NDN-RTC, NdnCon, NFD-Android, etc.

https://named-data.net/codebase/platform/ndn-ccl/
Sync

- Transport layer mechanism for NDN

- Set reconciliation between participants that sync data in a name space
  - Represent the data set using a compact “digest” over data name
    - Digest can be hash, IBF, or other format
  - Exchange digests to detect missing data names
  - Retrieve data using the data names

- Protocols: ChronoSync, iSync, PSync, Round Sync, VectorSync, DSSN, ...
  - implemented in libraries
Storage

• Application in-memory storage
  • Allocated in memory
  • Managed by applications

• forwarder content store
  • Managed by routers
  • Opportunistic

• Repository
  • Managed by application users or third party
  • Long-term persistent storage
What do we mean by “second generation” app?

Several meanings:

1. Post NSF FIA/FIA-NP support: Research support not only for NDN architectural development.

2. Previous research outcomes now generalized (somewhat), e.g.:
   - AR Browser uses a first generation application (NDN-RTC) as a library.
   - NDN-CNL (Common Name Library) abstracts from Interest-Data exchange.

3. Addressing a number of more complex issues, e.g.:
   - Interaction between configurable aspects of forwarding plane and application design.
   - Cross-layer optimization for wireless (in ICE-AR research).
   - Granularity and usability of security.
How to learn more

• NDN project website: https://named-data.net

• NFD, core libraries, and other general use software: https://github.com/named-data

• NDN papers