Named Data Networking
Architecture Recap

(Pieces available to start exploring NDN applications!)
New Communication Paradigm

• Users today care about content, not the servers

• Accessing the server is a by-product of the need to retrieve the desired content
  o If the server is down, no access to the content

• But what if the content was available from other places (e.g., my neighbor)?
NDN: Focus on Data

Abstracting away the notion of “host”
*Superset of host-to-host communication model*
Two Focal Points in NDN

• Focus on the what not the where
• Secure the data not the container
NDN Packets

Interest Packet

- **Content Name:** Identifies the data I want to receive
- **Selector:** identifier publisher, etc
- **Nonce**

Data Packet

- **Content Name:** Identifies the data in this packet
- **Signature:** Required for all packets
- **Data**
NDN Forwarding: Interests

- Interest: Name = X
- Forward interest towards Publisher (X)
- Mark incoming faces as wanting X (lay down breadcrumbs)
- Merge same interests for X
NDN Forwarding: Data

- Data: Name = X
  - Forward Data back to where interests came from
    - Follow the breadcrumbs back to requestors
    - Delete breadcrumbs
  - Duplicate at appropriate routers
  - Cache data at each router
NDN Interest Forwarding
NDN Data Forwarding
An Example Data Request

- Want data for Jan 30 – Feb 02
- Client expresses interests, one for each day
- Interests for Jan 30-31 go to server1
- Interests for Feb 01-02 go to server2
Named Data Networking

Application Support

(Pieces available to start exploring NDN applications!)
NDN Components

- Apps
- Routing
- Repo

Libraries

NFD

Links and Tunnels
Motivation

“The NDN project's approach is to design and build a variety of applications on NDN to drive the development and deployment of the architecture and its supporting modules, to test prototype implementations, and to encourage community use, experimentation, and feedback into the design.

“Application-driven development also allows verification and validation of performance and functional advantages of NDN, such as how routing on names promotes efficient authoring of sophisticated distributed applications, by reducing complexity, opportunities for error, and time and expense of design and deployment.”

Outline of Talk

- Application library introduction
- Naming conventions
- Repository
Outline of Talk

• Application library introduction

• Naming conventions

• Repository
Evolution of the Libraries

• All libraries now reflect fundamental architectural abstractions directly in objects, and wire format manipulation is abstracted.
  o Name
  o Interest
  o Data
  o Face
  o KeyChain
  o Validator

• Two library efforts available to community
  o NDN-CXX: “C++ for eXtended eXperimentation”
  o NDN-CCL: “Common Client Libraries”
  o Enables diversity of coding choice
  o Drives us towards specification (and not just implementation)
ndn-cxx: Applications

- **NFD**
  - NDN Forwarding Daemon

- **NLSR**
  - Named-data Link-State Routing protocol

- **repo-ng**
  - Next generation of NDN repository

- **ChronoChat**
  - Multi-user NDN chat application

- **ChronoSync**
  - Sync library for multiuser realtime applications for NDN

- **ndn-tlv-ping**
  - Ping Application For NDN

- **ndn-traffic-generator**
  - Traffic Generator For NDN
NDN-CCL Applications

- **CCNx Federated Wiki**, an NDN port of the Smallest Federated wiki (NDN-JS)
- **ChronoChat-js**, a javascript implementation of the ChronoChat demonstration application (NDN-JS)
- **Matryoshka**, an experimental multi-player online game using NDN and the Unity3D game engine. (jndn as the basis of the .NET port of CCL used in this project.)
- **ndn-bms**, a building management system prototype being developed as part of the NDN-NP project (PyNDN, NDN-JS)
- **ndn-lighting**, lighting control application using NDN (PyNDN, NDN-JS)
- **ndn-protocol**, a firefox browser plug-in supporting an ndn:/ retrieval scheme (NDN-JS)
- **ndnfs** and **ChronoShare**, NDN file sharing platforms (PyNDN, NDN-JS – with ndn-cxx)
- **NDNoT**, the Named Data Network of Things toolkit for the Raspberry PI (PyNDN, NDN-JS)
- **ndnrs**, a javascript implementation of an NDN repository (NDN-JS)
- **ndnrtc**, a peer-to-peer multiparty audio, video, and chat application over NDN. (NDN-CPP, NDN-JS)
- **ndnstatus**, the NDN routing status web page (PyNDN, NDN-JS)
- **NDNVideo**, a video playout application for NDN (PyNDN)

Coming:
- **NDNEx**, an NDN-based mobile health application being developed as part of the NDN-NP research project. (jndn)
- **OpenPTrack-NDN** an open source person tracking system that will add NDN support in Fall 2014. (NDN-CPP)
ChronoSync

- Synchronization protocol for multi-user real-time application for NDN

- Users synchronize their knowledge about a data set
  - knowledge: names of data in the set
    - effectively represented by a crypto digest
  - users with different digests: one of them is missing some knowledge
    - one sends its digest in interest
      - “This is what I have known, any new updates?”
    - others, if can tell the difference, may reply some missing knowledge in data packet
      - “I guess this information may help us to get to the same page.”
  - replies effectively utilize NDN multicast forwarding feature
    - in most cases, users express interests with the same digest
    - reply to a interest multicast to the senders with the same digest.

- User fetches data separately when discovering a new name
  - allow flexible data fetching strategy
  - allow flexible trust model to validate actual data
Outline of talk

• Application library introduction

• Naming conventions

• Repository
Outline of talk

- Application library introduction
- Naming conventions
- Repository
Naming Conventions

- Namespace design is a critical component of application development.

- “Where possible, put it in the name” philosophy is expressing some packet requirements / features in the name.

- Three areas covered here
  
  - Scope control obeyed by NFD: /localhost, /localhop
  
  - Signed interest format
  
  - Versioning, segmenting, etc.
Scope Control

• Reflects research into how names can be used for scoping.

• \texttt{/localhost}
  Limits propagation to the applications on the originating host.

• \texttt{/localhop}
  The localhop scope limits propagation to no further than the next node.
Versions, segments, and more...

• Reflects ongoing research in common naming conventions.

• Initial discussion in Technical Report #22, “Naming Conventions”

• Segmenting
  o Cut large data (e.g. video frame) into packet-sized pieces
  o Final segment indicated by MetaInfo FinalBlockID

• Versioning
  o Data packet is immutable: a new version needs a new name
  o Suggest millisecond time stamp but not required

• Time-stamping
  o When data packet was produced
  o Microseconds since January 1, 1970

• Sequencing
  o Sequential items in a collection
  o 0, 1, ... X. Assume item X + 1 may be produced
Signed Interests

- Emerged from protocols for interacting with the forwarder, as well as work in authenticated control for building management systems.

- Primary mechanism for interacting with NFD.

```
/signed/interest/name/<timestamp>/<random-value>/<SignatureInfo>/<SignatureValue>
```

Additional components of Signed Interest

<table>
<thead>
<tr>
<th>Interest Type (0x01)</th>
<th>Interest length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name Type</td>
<td>Name Length</td>
</tr>
<tr>
<td>Component TLV 1</td>
<td>...</td>
</tr>
<tr>
<td>Other TLVs in Interest</td>
<td></td>
</tr>
</tbody>
</table>

Signed portion of Interest

Signature
Outline of talk

• Application library introduction

• Naming conventions

• Repository
Repository

- Storage in the network: a fundamental concept for NDN

- For now, particular app requirements may drive performance needs and features.

- **Basic Repo Insertion Protocol**
  insertion of a single or collection of Data packets.

- **Watched Prefix Insertion Protocol**
  a protocol to insert continuously generated data for a given namespace

- **Bulk Insert Repo Insertion Protocol**
  a simple protocol to insert Data packets in bulk
  (e.g., from a producer on the same host)

- **Repo Deletion Protocol**
  deletion of a single or collection of Data packets under certain prefix

- Signed interest-based.
Outline of talk

• Application library introduction

• Naming conventions

• Repository

• Briefly, what’s next?
What’s next

“A few years of designing and developing prototype applications on NDN has revealed five key areas of application research that map to important features of the architecture:

1. namespaces;
2. trust models;
3. in-network storage;
4. data synchronization;
5. rendezvous, discovery, and bootstrapping.”


Also, see Tech Report #17 for ideas on API evolution.