Routing in NDN

Lan Wang (University of Memphis) & the NDN Team
FIA PI Meeting
11/14/2013
Roadmap

- What does NDN require from a routing protocol?

- How does NDN support in-network storage, anycast, and mobility?

- What does NDN provide as foundations for routing?

- Example: Named-data Link State Routing (NLSR)
NDN’s Requirement on Routing

**Requirement: routing to information**

- guide each Interest packet to all potential providers (via all feasible paths)
- Some providers may not have all content in a name prefix.

**Non-requirement: fast routing convergence:** stateful forwarding plane can adapt to changes/failures quickly.

---

A wants to retrieve /cnn/video/a.mpg

A has some content under /cnn/video

C wants to retrieve /cnn/video/a.mpg
NDN’s Requirement on Routing

**Requirement: routing to information**

- guide each Interest packet to all potential providers (via all feasible paths)

- Some providers may not have all content in a name prefix.

**Non-requirement: fast routing convergence:** stateful forwarding plane can adapt to changes/failures quickly.

---

A, B, E’s FIB:

| /cnn/video | A, B, E |

C wants to retrieve /cnn/video/a.mpg

A has some content under /cnn/video

---

11/14/13
NDN’s Requirement on Routing

**Requirement: routing to information**

- guide each Interest packet to all potential providers (via all feasible paths)
- Some providers may not have all content in a name prefix.

**Non-requirement: fast routing convergence:** stateful forwarding plane can adapt to changes/failures quickly.

---

*Example Diagram:*

- **A** has some content under `/cnn/video`
- **C** wants to retrieve `/cnn/video/a.mpg`
- **D**'s FIB: `/cnn/video` and **A, B, E**
NDN’s Requirement on Routing

**Requirement: routing to information**

- guide each Interest packet to all potential providers (via all feasible paths)
- Some providers may not have all content in a name prefix.

**Non-requirement: fast routing convergence**: stateful forwarding plane can adapt to changes/failures quickly.

---

**Diagram:**

- **A** has some content under `/cnn/video`
- **C** wants to retrieve `/cnn/video/a.mpg`
- **D**'s FIB: `A, B, E`
- **E**
NDN’s Requirement on Routing

**Requirement: routing to information**

- guide each Interest packet to all potential providers (via all feasible paths)
- Some providers may not have all content in a name prefix.

**Non-requirement: fast routing convergence:** stateful forwarding plane can adapt to changes/failures quickly.
NDN’s Requirement on Routing

**Requirement: routing to information**
- Guide each Interest packet to all potential providers (via all feasible paths)
- Some providers may not have all content in a name prefix.

**Non-requirement: fast routing convergence**: stateful forwarding plane can adapt to changes/failures quickly.

---

A, B, E

D’s FIB: /cnn/video A, B, E

A has some content under /cnn/video

C wants to retrieve /cnn/video/a.mpg

C

D

E

A

/cnn/video

Interest for /cnn/video/a.mpg/s1

11/14/13
NDN’s Requirement on Routing

**Requirement: routing to information**

- guide each Interest packet to all potential providers (via all feasible paths)
- Some providers may not have all content in a name prefix.

**Non-requirement: fast routing convergence:** stateful forwarding plane can adapt to changes/failures quickly.
NDN’s Requirement on Routing

**Requirement: routing to information**

- guide each Interest packet to all potential providers (via all feasible paths)
- Some providers may not have all content in a name prefix.

**Non-requirement: fast routing convergence:** stateful forwarding plane can adapt to changes/failures quickly.

---

**Diagram:**

- **A** has some content under `/cnn/video`
- **C** wants to retrieve `/cnn/video/a.mpg`
- **D**'s FIB: `/cnn/video` A, B, E

---

11/14/13
**NDN’s Requirement on Routing**

**Requirement: routing to information**
- guide each Interest packet to all potential providers (via all feasible paths)
- Some providers may not have all content in a name prefix.

**Non-requirement: fast routing convergence:** stateful forwarding plane can adapt to changes/failures quickly.
NDN’s Requirement on Routing

**Requirement: routing to information**

- guide each Interest packet to all potential providers (via all feasible paths)
- Some providers may not have all content in a name prefix.

**Non-requirement: fast routing convergence**: stateful forwarding plane can adapt to changes/failures quickly.

---

A has some content under /cnn/video

C wants to retrieve /cnn/video/a.mpg

D’s FIB: /cnn/video A, B, E

11/14/13
Routing Mechanism in NDN

Any routing algorithm that works for IP (e.g., link-state) can be used in NDN.

- NDN’s forwarding semantics is a superset of the IP model.

Differences:
- replace IP prefixes with name prefixes
- calculate a list of next-hops for each name prefix
- use Interest/Data packets
Routing Mechanism in NDN

Any routing algorithm that works for IP (e.g., link-state) can be used in NDN.

- NDN's forwarding semantics is a superset of the IP model.

Differences:
- replace IP prefixes with name prefixes
- calculate a list of next-hops for each name prefix
- use Interest/Data packets

A can reach /cnn/video
Routing Mechanism in NDN

Any routing algorithm that works for IP (e.g., link-state) can be used in NDN.

- NDN’s forwarding semantics is a superset of the IP model.

Differences:
- replace IP prefixes with name prefixes
- calculate a list of next-hops for each name prefix
- use Interest/Data packets

A can reach /cnn/video
B can reach /cnn/video
C
D
E
Routing Mechanism in NDN

Any routing algorithm that works for IP (e.g., link-state) can be used in NDN.

- NDN’s forwarding semantics is a superset of the IP model.

Differences:
- replace IP prefixes with name prefixes
- calculate a list of next-hops for each name prefix
- use Interest/Data packets
Routing Mechanism in NDN

Any routing algorithm that works for IP (e.g., link-state) can be used in NDN.

- NDN’s forwarding semantics is a superset of the IP model.

Differences:

- replace IP prefixes with name prefixes
- calculate a list of next-hops for each name prefix
- use Interest/Data packets
In-network Storage

- Routing to information unifies all types of storage.
  - data producer: stationary and mobile
  - in-network: persistent storage (e.g., repos) and transient storage (e.g., caches)

- In-network Storage
  - Routing support: advertise data’s name prefix if data is expected to stay for a while.
  - Forwarding support: routers remember which faces data come from, so similar Interests will be forwarded to the storage.
Anycast

- All data providers advertise the same prefix.
- Sessionless: a consumer is not bound to a particular anycast provider.

C wants /cnn/video/a.mpg
Mobility (1)

- NDN properties that facilitate mobility
  - Data names do not change with mobility.
  - Granular data
  - Sessionless
  - Caching

- Consumer mobility
  - Data may go to the old location.
  - Consumer reissues Interest upon timeout.
  - Data may be returned by an intermediate cache.
Mobility (1)

- **NDN properties that facilitate mobility**
  - Data names do not change with mobility.
  - Granular data
  - Sessionless
  - Caching

- **Consumer mobility**
  - Data may go to the old location.
  - Consumer reissues Interest upon timeout.
  - Data may be returned by an intermediate cache.

C wants to retrieve /cnn/video/a.mpg
Mobility (1)

- NDN properties that facilitate mobility
  - Data names do not change with mobility.
  - granular data
  - sessionless
  - caching

- Consumer mobility
  - Data may go to the old location.
  - Consumer reissues Interest upon timeout.
  - Data may be returned by an intermediate cache.

C wants to retrieve /cnn/video/a.mpg
Mobility (1)

- **NDN properties that facilitate mobility**
  - Data names do not change with mobility.
  - granular data
  - sessionless
  - caching

- **Consumer mobility**
  - Data may go to the old location.
  - Consumer reissues Interest upon timeout.
  - Data may be returned by an intermediate cache.

---

C wants to retrieve /cnn/video/a.mpg
Mobility (1)

- NDN properties that facilitate mobility
  - Data names do not change with mobility.
  - granular data
  - sessionless
  - caching

- Consumer mobility
  - Data may go to the old location.
  - Consumer reissues Interest upon timeout.
  - Data may be returned by an intermediate cache.
Mobility (1)

- NDN properties that facilitate mobility
  - Data names do not change with mobility.
  - Granular data
  - Sessionless
  - Caching

- Consumer mobility
  - Data may go to the old location.
  - Consumer reissues Interest upon timeout.
  - Data may be returned by an intermediate cache.

C wants to retrieve /cnn/video/a.mpg
Mobility (1)

- **NDN properties that facilitate mobility**
  - Data names do not change with mobility.
  - Granular data
  - Sessionless
  - Caching

- **Consumer mobility**
  - Data may go to the old location.
  - Consumer reissues Interest upon timeout.
  - Data may be returned by an intermediate cache.

C wants to retrieve /cnn/video/a.mpg
Mobility (1)

- NDN properties that facilitate mobility
  - Data names do not change with mobility.
  - granular data
  - sessionless
  - caching

- Consumer mobility
  - Data may go to the old location.
  - Consumer reissues Interest upon timeout.
  - Data may be returned by an intermediate cache.
Mobility (1)

- NDN properties that facilitate mobility
  - Data names do not change with mobility.
  - granular data
  - sessionless
  - caching

- Consumer mobility
  - Data may go to the old location.
  - Consumer reissues Interest upon timeout.
  - Data may be returned by an intermediate cache.

C wants to retrieve /cnn/video/a.mpg
Mobility (1)

- NDN properties that facilitate mobility
  - Data names do not change with mobility.
  - granular data
  - sessionless
  - caching

- Consumer mobility
  - Data may go to the old location.
  - Consumer reissues Interest upon timeout.
  - Data may be returned by an intermediate cache.
Mobility (1)

- **NDN properties that facilitate mobility**
  - Data names do not change with mobility.
  - Granular data
  - Sessionless
  - Caching

- **Consumer mobility**
  - Data may go to the old location.
  - Consumer reissues Interest upon timeout.
  - Data may be returned by an intermediate cache.

C wants to retrieve /cnn/video/a.mpg
Mobility (2)

Producer mobility: multiple complimentary mechanisms

- **Caching**: Interest for previously requested data may be satisfied by intermediate caches.

- **Routing advertisements**: advertise same name prefix at new location → subsequent Interests will go to new location.

- **Forwarding Hint**: use name of the new attachment point as a hint to guide forwarding
  - Name = /lan/video/a.mpg/s1
  - Hint = /att/atlanta/rtrB

- **Rendezvous Points**: producer may publish data to the same repo regardless of location (or one of a set of sync’ed repos).
NDN’s Support for Routing

- **Built-in security**: routing data is signed by originator and can be verified by receivers.

- **Naming**: names facilitate management and trust.
  - Names identify routing components and relationship.
    - `/att/atlanta/rtr1` → `rtr1` in Atlanta PoP of AT&T
  - Naming of data and keys reflect trust relationship. Given a piece of data, you can derive the name of the signing key based on the trust model.

- **Sync mechanism**: a new notion of transport to ensure multiple parties have the same information.
  - efficient way of set reconciliation
  - Routing protocol uses Sync to distribute routing information.
Named-data Link State Routing (NLSR)*

- Reuse a mature routing algorithm: link state
- NDN native
  - Names, not addresses (networks, routers, processes, data, keys)
  - Interest/Data are used to distribute routing info.
- Multipath support: modified Dijkstra’s algorithm to produce a ranked list of next-hops for each name prefix.
- Security
  - a trust model for intra-domain routing
  - Routing data is signed by originating router and verified by receivers based on trust model.

Naming in NLSR

- Follow the hierarchy within a network
  - Easy to identify the relationship among entities
  - Easy to associate keys with key owners

- Topology
  - /<network>/<site>/<router>
    - E.g., /ndn/memphis/rtr1

- Updates
  - /<network>/NLSR/LSA/<site>/<router>/<type>/<version>

- Keys
  - NLSR key: /<network>/keys/<site>/<router>/NLSR
  - Router key, operator key, …
Message Authenticity and Integrity

- Every NLSR Data packet is signed.
- “key locator” includes information about the key.
- Receiver retrieves the key and verifies the signature.

### Signing and verification in NLSR

<table>
<thead>
<tr>
<th>Key Owner</th>
<th>Key Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>/&lt;network&gt;/keys</td>
</tr>
<tr>
<td>Site</td>
<td>/&lt;network&gt;/keys/&lt;site&gt;</td>
</tr>
<tr>
<td>Operator</td>
<td>/&lt;network&gt;/keys/&lt;site&gt;/&lt;operator&gt;</td>
</tr>
<tr>
<td>Router</td>
<td>/&lt;network&gt;/keys/&lt;site&gt;/&lt;router&gt;</td>
</tr>
<tr>
<td>NLSR</td>
<td>/&lt;network&gt;/keys/&lt;site&gt;/&lt;router&gt;/NLSR</td>
</tr>
</tbody>
</table>
NLSR synchronizes LSDB among routers.

- Every node periodically sends a digest of LSDB to others in Interest packets.
- When a node has a new LSA, its digest changes and it will reply to others’ Interests with name of new LSA.
- Other nodes fetch new LSAs.
- More resilient/scalable, fits NDN model.
NLSR synchronizes LSDB among routers.

- Every node periodically sends a digest of LSDB to others in Interest packets.
- When a node has a new LSA, its digest changes and it will reply to others’ Interests with name of new LSA.
- Other nodes fetch new LSAs.
- More resilient/scalable, fits NDN model.
From Flooding to Synchronization

NLSR synchronizes LSDB among routers.

- Every node periodically sends a digest of LSDB to others in Interest packets.
- When a node has a new LSA, its digest changes and it will reply to others’ Interests with name of new LSA.
- Other nodes fetch new LSAs.
- More resilient/scalable, fits NDN model.

Diagram:

A ➔ C ➔ B ➔ D

a1324asd9 ➔ a1324asd9 ➔ a1324asd9 ➔ a1324asd9

Interest
/ndn/broadcast/NLSR/a13
24asd9

11/14/13
NLSR synchronizes LSDB among routers.

- Every node periodically sends a digest of LSDB to others in Interest packets.
- When a node has a new LSA, its digest changes and it will reply to others’ Interests with name of new LSA.
- Other nodes fetch new LSAs.
- More resilient/scalable, fits NDN model.
From Flooding to Synchronization

NLSR synchronizes LSDB among routers.

- Every node periodically sends a digest of LSDB to others in Interest packets.
- When a node has a new LSA, its digest changes and it will reply to others’ Interests with name of new LSA.
- Other nodes fetch new LSAs.
- More resilient/scalable, fits NDN model.
NLSR synchronizes LSDB among routers.

- Every node periodically sends a digest of LSDB to others in Interest packets.
- When a node has a new LSA, its digest changes and it will reply to others’ Interests with name of new LSA.
- Other nodes fetch new LSAs.
- More resilient/scalable, fits NDN model.
NLSR synchronizes LSDB among routers.

- Every node periodically sends a digest of LSDB to others in Interest packets.
- When a node has a new LSA, its digest changes and it will reply to others’ Interests with name of new LSA.
- Other nodes fetch new LSAs.
- More resilient/scalable, fits NDN model.
From Flooding to Synchronization

NLSR synchronizes LSDB among routers.

- Every node periodically sends a digest of LSDB to others in Interest packets.
- When a node has a new LSA, its digest changes and it will reply to others’ Interests with name of new LSA.
- Other nodes fetch new LSAs.
- More resilient/scalable, fits NDN model.

A has a new LSA

```
a1324asd9
d425ab9c
```

```
A
```

```
B
```

```
C
```

```
D
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```

```
/ndn/broadcast/NSR/a13
24asd9
```
NLSR synchronizes LSDB among routers.

- Every node periodically sends a digest of LSDB to others in Interest packets.
- When a node has a new LSA, its digest changes and it will reply to others’ Interests with name of new LSA.
- Other nodes fetch new LSAs.
- More resilient/scalable, fits NDN model.
NLSR synchronizes LSDB among routers.

- Every node periodically sends a digest of LSDB to others in Interest packets.
- When a node has a new LSA, its digest changes and it will reply to others’ Interests with name of new LSA.
- Other nodes fetch new LSAs.
- More resilient/scalable, fits NDN model.

A has a new LSA

Sync Data /ndn/NLSR/LSA/..

A1324asd9
d425ab9c

A1324asd9

A has a new LSA

D
NLSR synchronizes LSDB among routers.

- Every node periodically sends a digest of LSDB to others in Interest packets.
- When a node has a new LSA, its digest changes and it will reply to others’ Interests with name of new LSA.
- Other nodes fetch new LSAs.
- More resilient/scalable, fits NDN model.
Status


- Code available on github
  - [https://github.com/NDN-Routing/NLSR0.0/tree/nlsr-sync](https://github.com/NDN-Routing/NLSR0.0/tree/nlsr-sync)

- Plan to deploy on NDN testbed soon.
Scaling

- Next step: Inter-domain routing
  - routing policy and trust model

- Hyperbolic routing
  - Each node has a set of hyperbolic coordinates.
  - Each name prefix has a set of hyperbolic coordinates.
  - Calculate next-hops based on each neighbor’s distance to the name prefix
  - No need to distribute topological information.

- Map-and-Encap
  - Map application name prefixes to routable name prefixes (typically ISP name prefixes)
  - Orders of magnitude fewer routable name prefixes than application name prefixes
  - How to do mapping? ongoing research, e.g., DNS-like system
**OSPF vs. NLSR**

- Both are link-state intra-domain routing protocols.

<table>
<thead>
<tr>
<th></th>
<th>OSPF</th>
<th>NLSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naming</td>
<td>addresses</td>
<td>hierarchical names</td>
</tr>
<tr>
<td>Updates</td>
<td>network flooding</td>
<td>neighbor syncing</td>
</tr>
<tr>
<td>Next-hop</td>
<td>single</td>
<td>multiple</td>
</tr>
<tr>
<td>Security</td>
<td>password</td>
<td>public keys and trust model</td>
</tr>
</tbody>
</table>