NLSR: Named-data Link State Routing Protocol

A K M Mahmudul Hoque, Syed Obaid Amin, Adam Alyyan, Lan Wnag
(University of Memphis)
Beichuan Zhang (University of Arizona)
Lixia Zhang (UCLA)

Motivation

- Need a routing protocol for NDN
  - Populate FIB so routers can forward interests.
  - Compute next-hops for name prefixes
  - Not necessarily point to the nearest cache.
- Be NDN "native"
  - Carry routing information in Interest/Data.
  - Provide better support for NDN's adaptive forwarding
  - Current OSPFN is a hack of IP-based OSPF implementation.

Approach

- Reuse a mature routing algorithm: link state
  - Each router advertises local links and prefixes
  - Compute best paths based on entire topology.
- Design a native NDN protocol to realize it.
  - Naming
  - Trust
  - Syncing
  - Multipath

Naming

- 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.edu/rtr1
- Updates
  - /<network>/NLSR/LSA/<site>/<router>/<type>/<version>
- Keys
  - /<network>/keys/<site>/.....

LSA messages

- Generated and signed by the NLSR process at a particular router
- Adjacency LSA
  - The content contains all links of a router.
- Prefix LSA
  - The content contains a name prefix registered at the router.

Link State Database (LSDB)

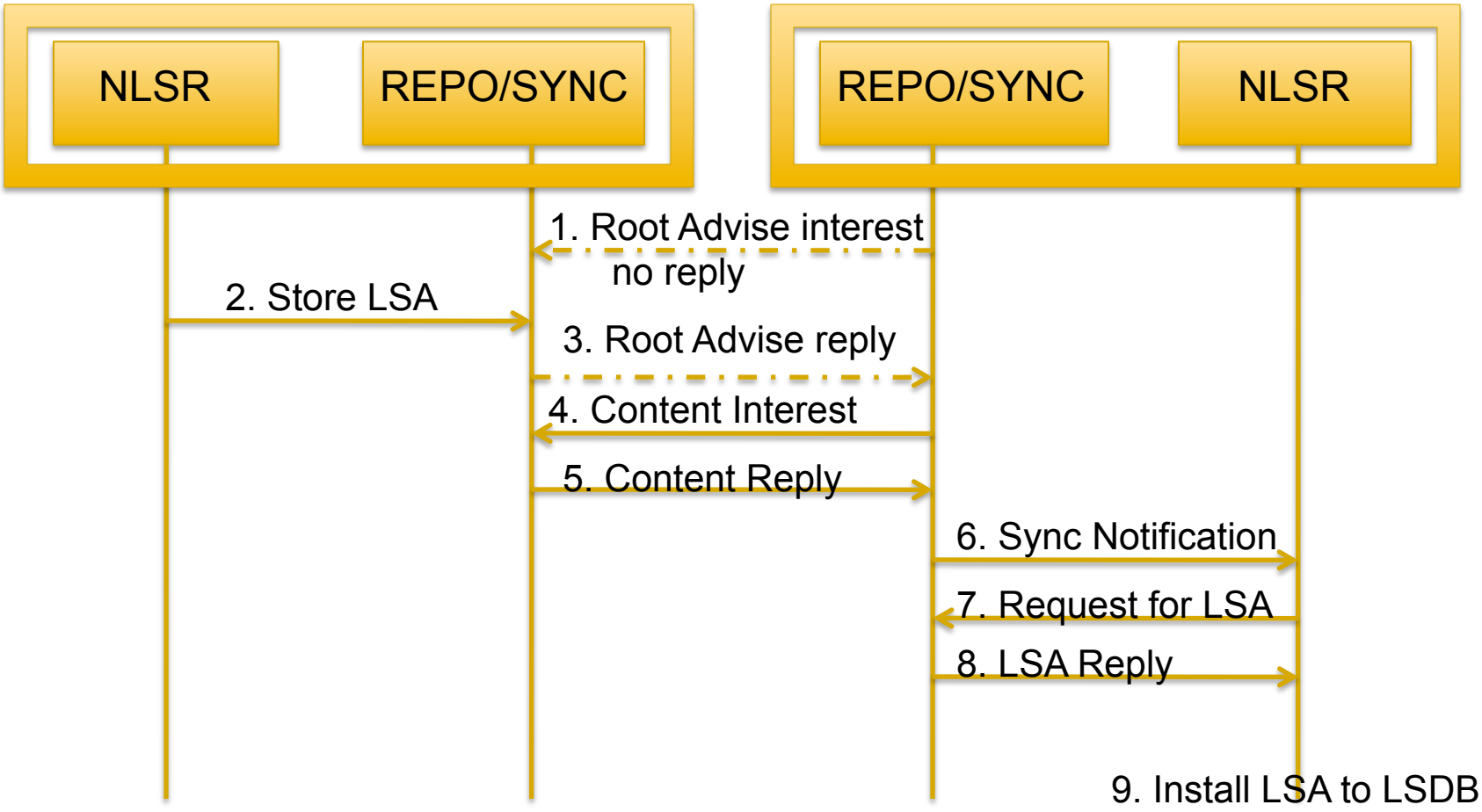
- Traditional OSPF operations
  - LSDB synchronization at the start of a session
  - Reliable flooding of new LSAs with per-hop ack
  - Periodic flooding of current LSAs, i.e., refresh

- How to "flood" updates in an NDN network?

From flooding to synchronization

- NLSR
  - Keep synchronizing LSDB with neighbors
- Sync
  - Two neighbors periodically send summary digest of LSDB to each other in Interest packets.
  - If the digests are different, figure out the difference and fetch new LSAs.
  - More resilient/scalable, fits NDN model.

Example of Sync



Multipath

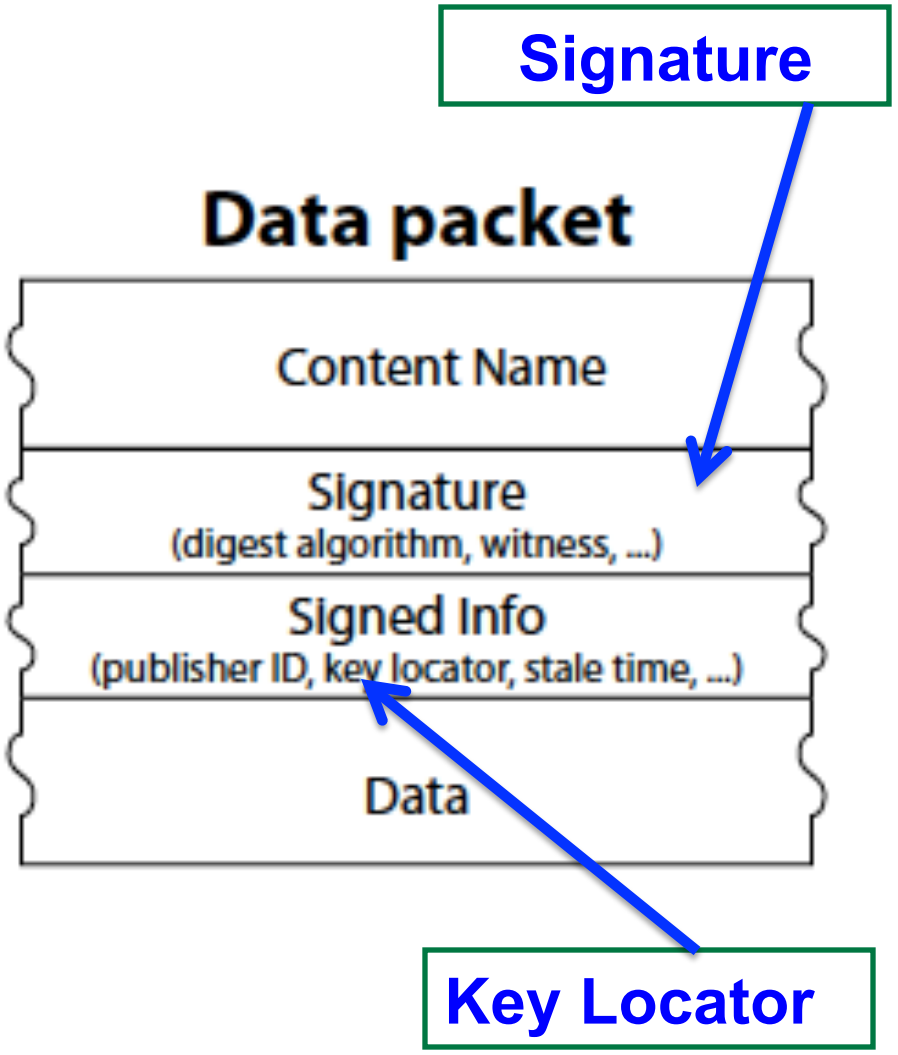
- Traditional OSPF
  - Single best next-hop or ECMP.
- NDN makes multipath easy and natural.
  - Built-in loop detection
  - Returning Data indicates it worked.
  - Interest NACK or PIT timeout indicates it didn't work.
- NLSR provides an ordered list of interfaces for each prefix, so that NDN's forwarding strategy can explore them efficiently.

Route computation

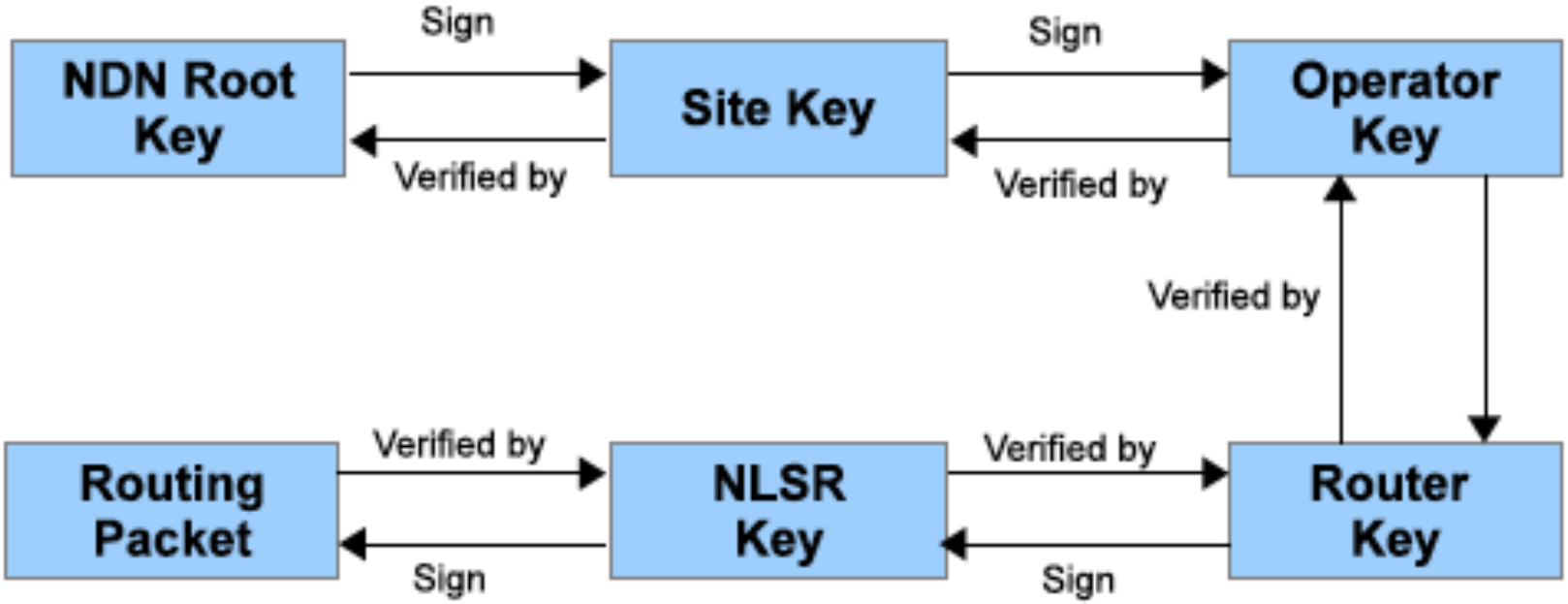
- Currently multiple runs of Dijkstra's algorithm over LSDB
  - Investigating more efficient algorithms.
- Compute the path cost of using each neighbor
  - Keep the link to the neighbor, disable all other neighbor links.
  - Compute shortest path to the name prefix and record the cost.
- Rank all next-hops by their path costs. Install in FIB.

Message authenticity and integrity

- Every NDN Data packet is signed.
- "key locator" includes information about the key.
- Receiver retrieves the key and verifies the signature.



Signing and Verification in NLSR



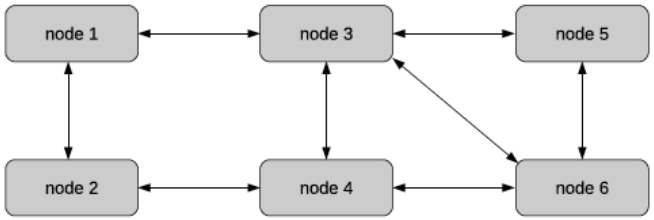
Keys are distributed via Sync, available in Repo, and identified by names carried in the “Key Locator” field.

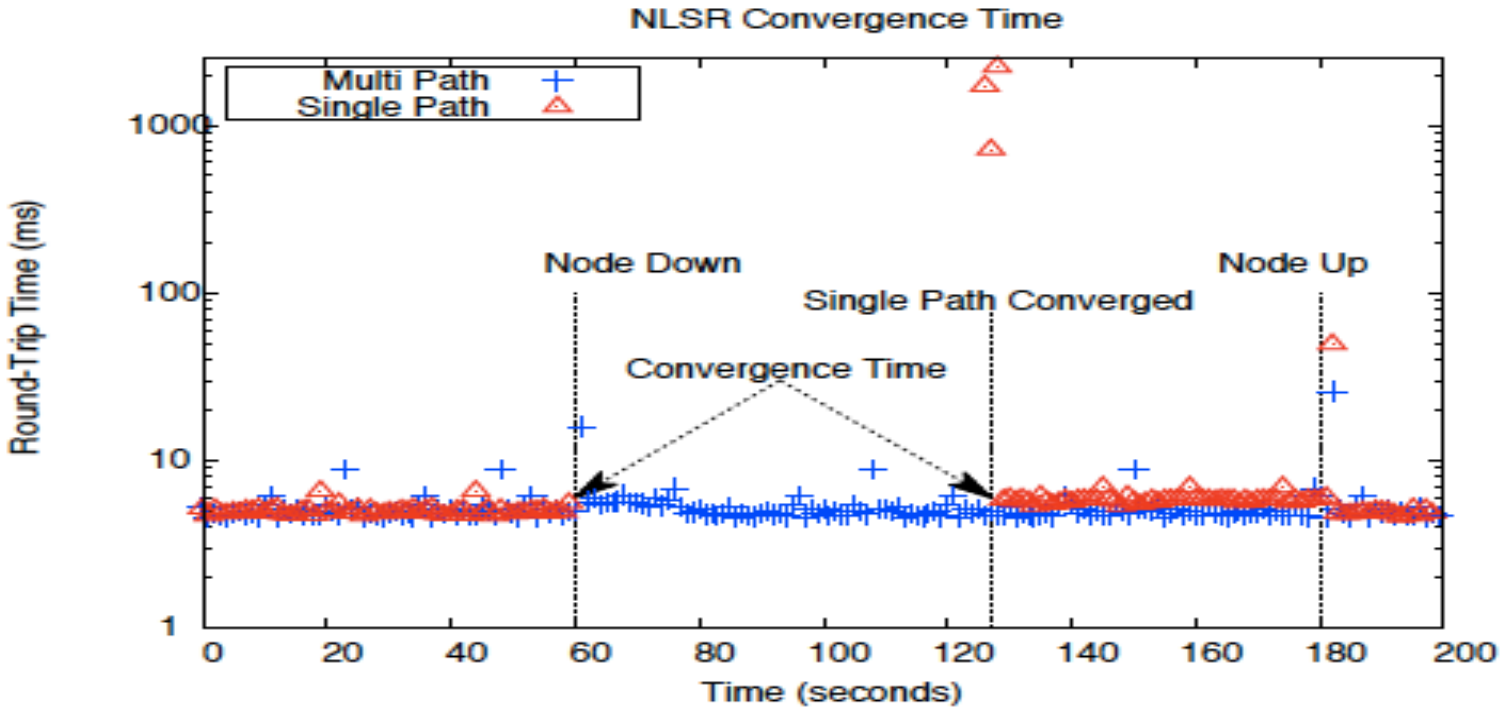
OSPF vs. NLSR

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

	OSPF	**NLSR**
Naming	addresses	hierarchical names
Updates	network flooding	neighbor syncing
Next-hop	single	multiple
Security	Passwords	Public keys

Lab Experiment





Development Status

- Implemented in CCNx with Sync/Repo
  - Lab experiments revealed several bugs and limitations
  - Plan to try ChronoSync
- Ongoing improvement of the design
  - Adjacency discovery and maintenance
  - Faster multipath computation
- Plan to deploy on NDN testbed and make code available on github.

  http://www.github.com/named-data

Questions?