THE UNIVERSITY OF MEMPHIS

Dreamers. Thinkers. Doers.

NDN Architectural Development and Routing Design

Lan Wang
University of Memphis
Nov. 21, 2014
www.named-data.net

Outline

- Introduce basic concepts of Named Data Networking (NDN)
- Present an overview on NDN development
- Show an example of name-based routing
- Summarize potential solutions in scaling NDN routing

NDN Project

NDN: Named Data Networking

- Van Jacobson's Google talk August 2006, "A New Way to Look at Networking"
- NDN project started on 9/1/2010
- Part of NSF Future Internet Architecture Program

NDN-NP (NDN-Next Phase)

- Started in May 2014
- 8 collaborating institutions

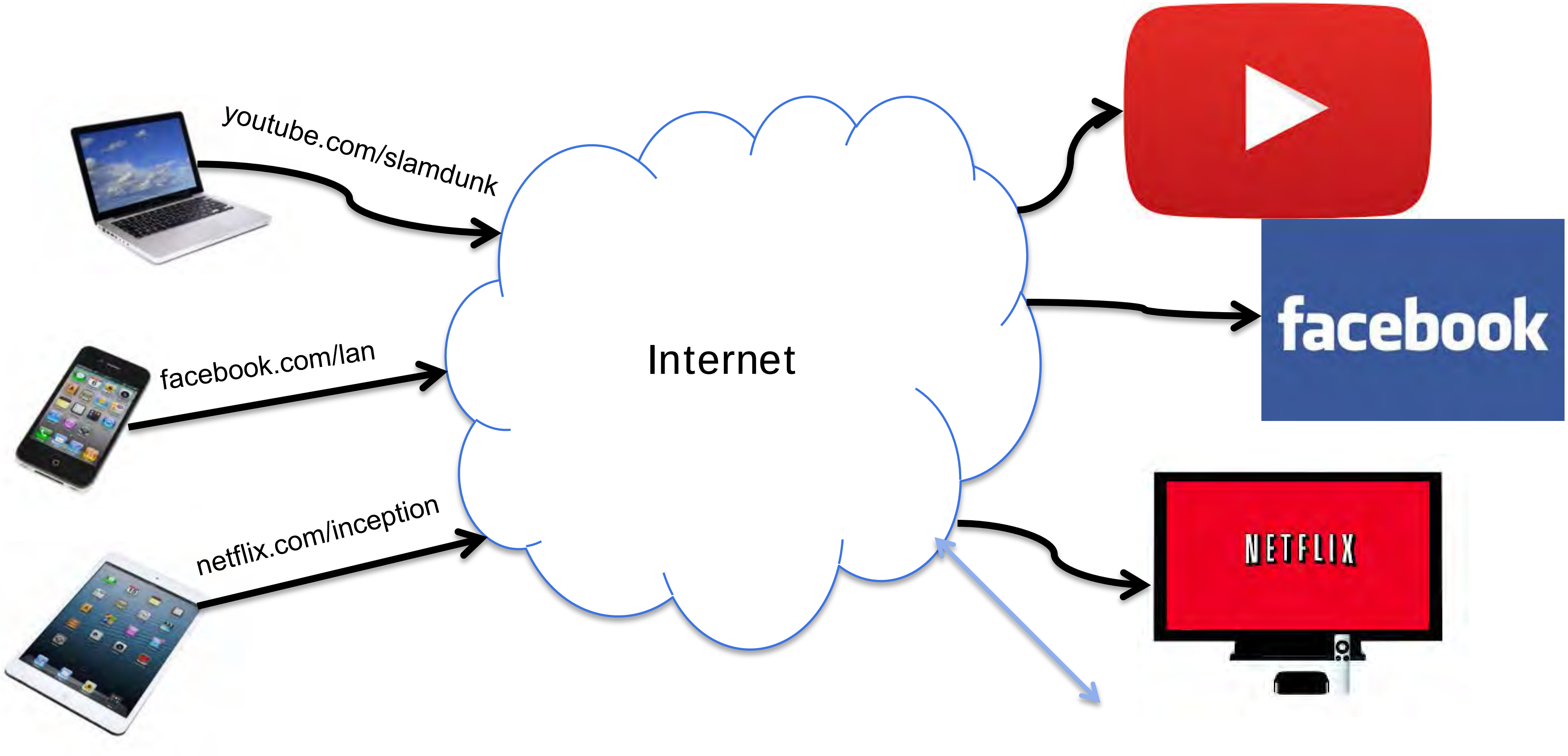
http://www.named-data.net/

NDN-NP PROJECT PIs

UCLA: Van Jacobson, Jeff Burke, Lixia Zhang
University of Arizona: Beichuan Zhang
UCSD: Kim Claffy
Colorado State University: Christos Papadopoulos
UIUC: Tarek Abdelzaher
University of Memphis: Lan Wang
University of Michigan: J. Alex Halderman
Washington University: Patrick Crowley

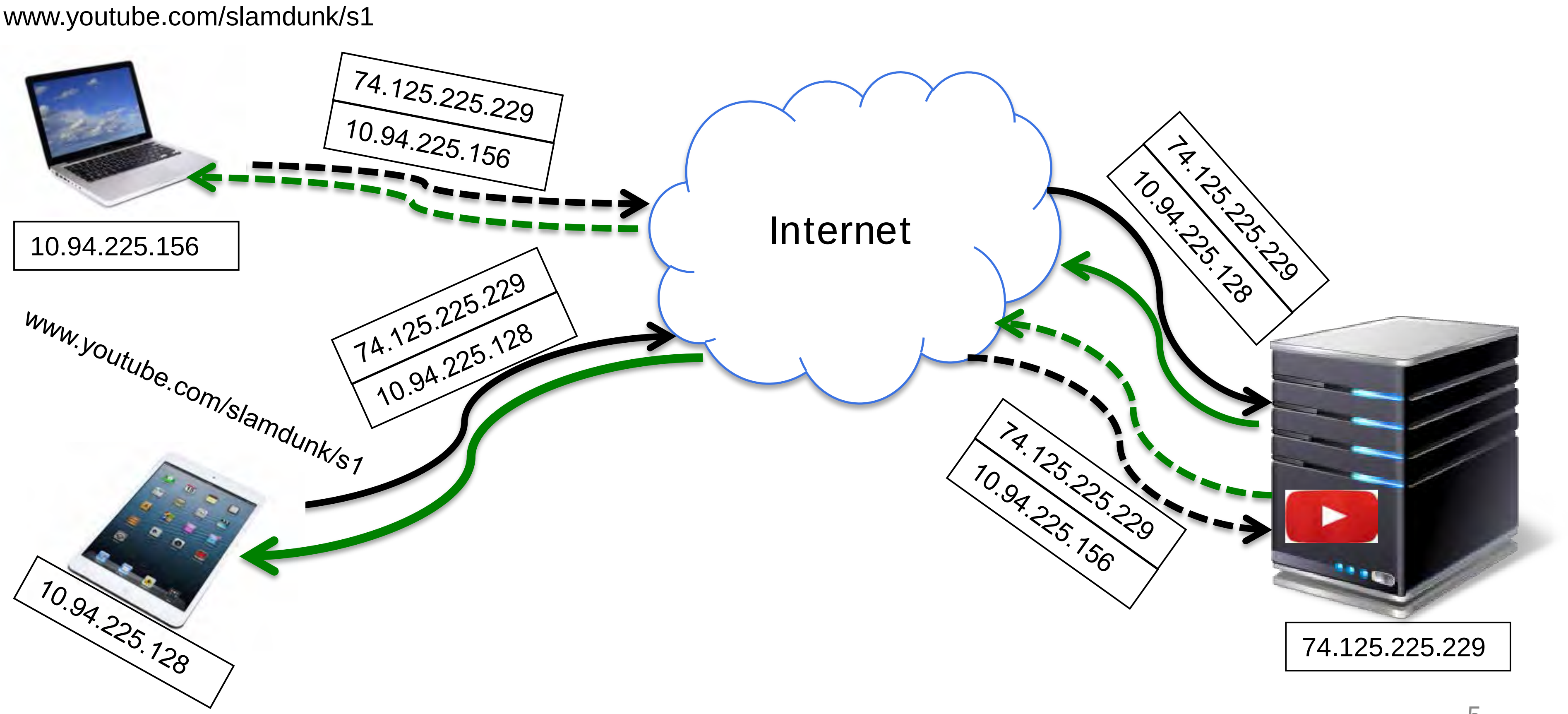
Today's Internet Traffic

- Communication is "*Content*" driven.

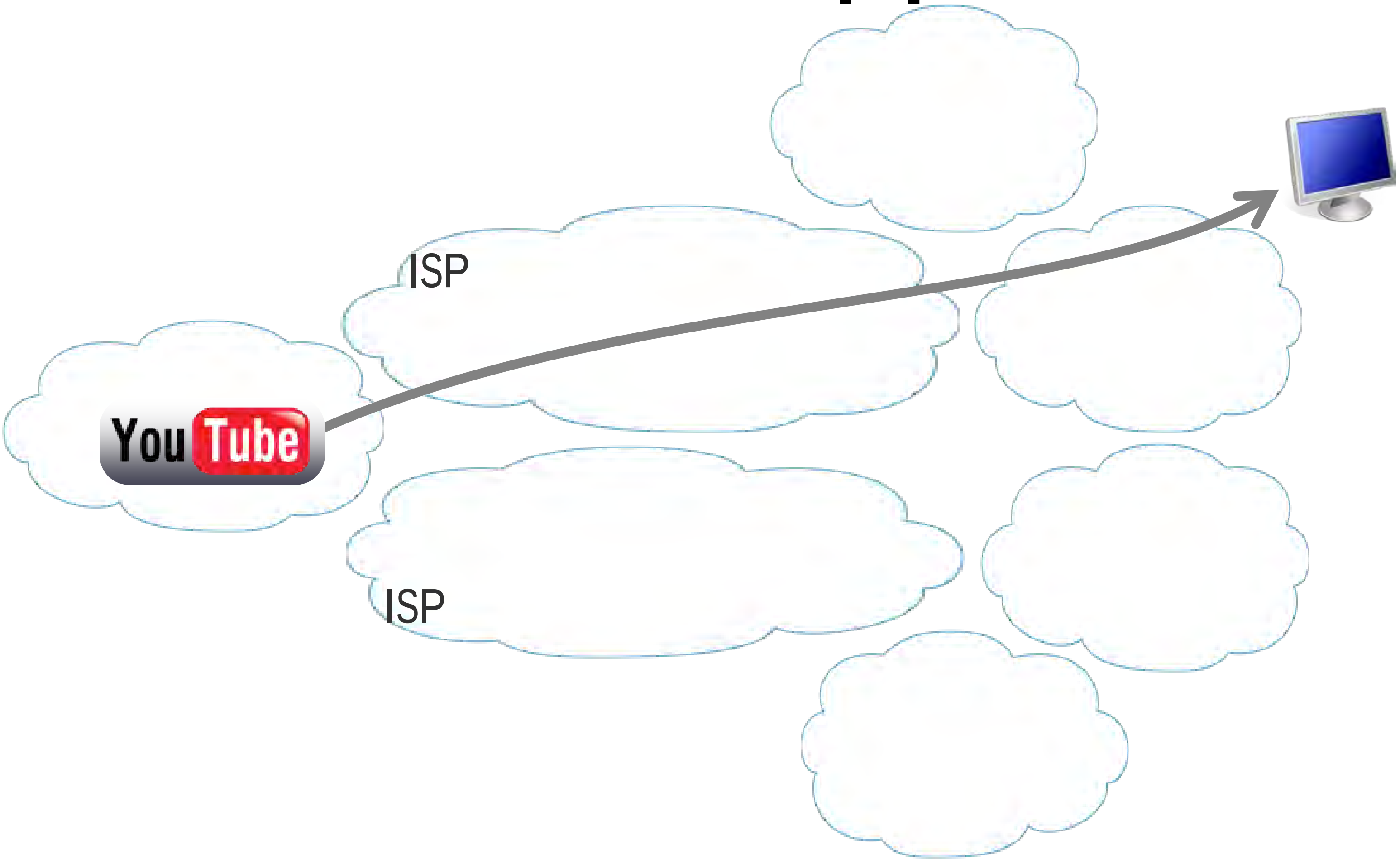


Internet Protocol (IP)

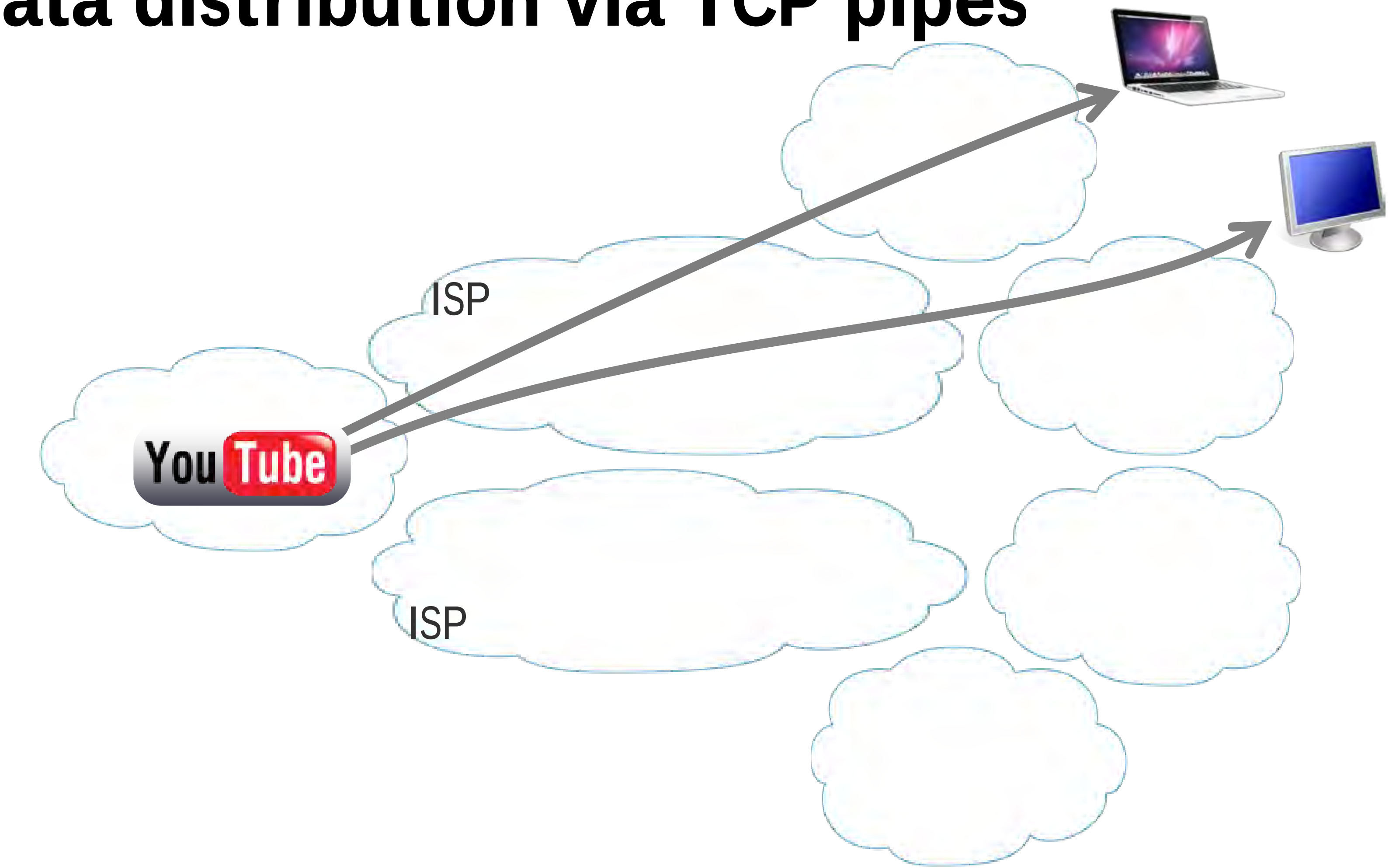
- Underlying communication is “*destination*” driven.



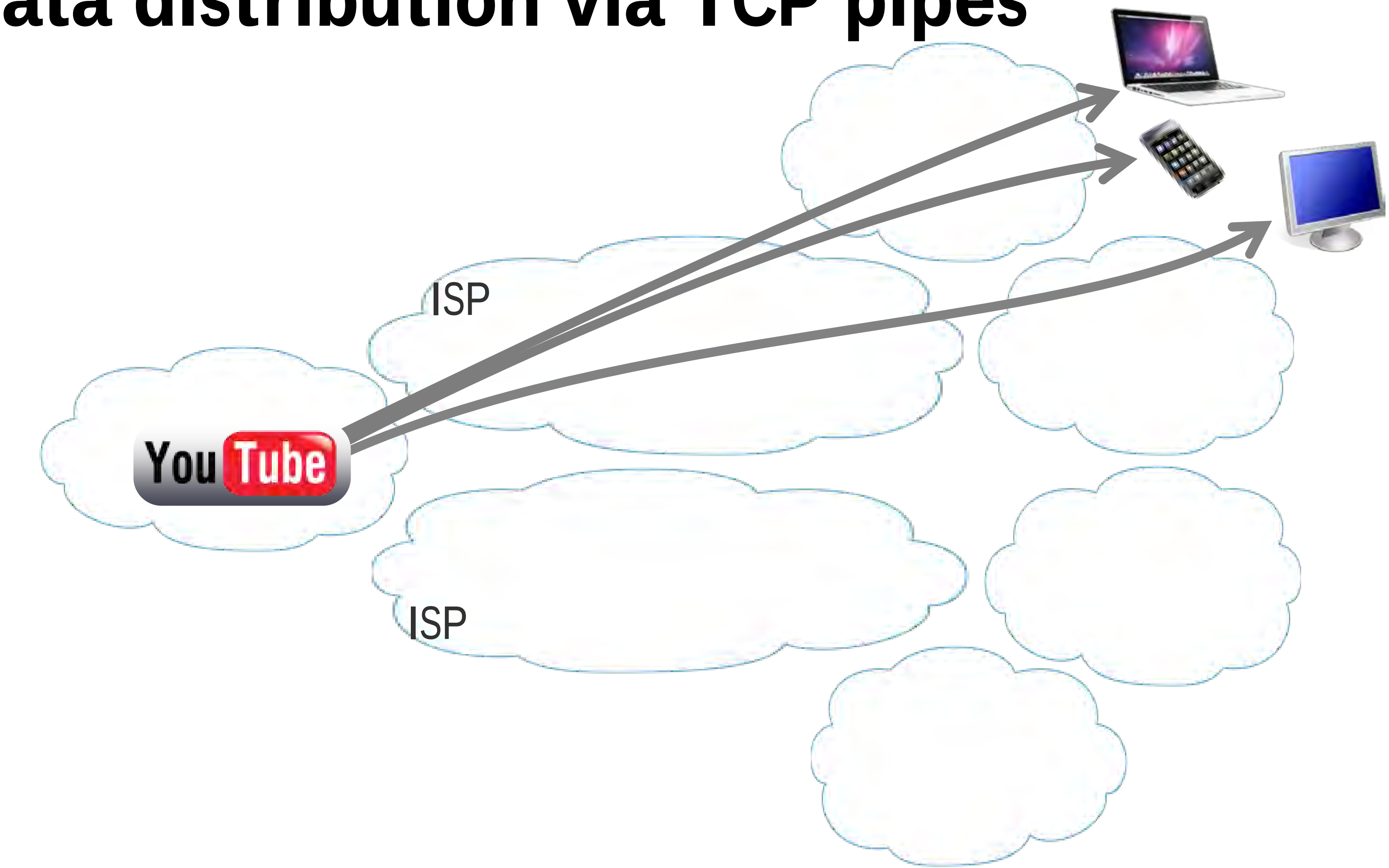
Data distribution via TCP pipes



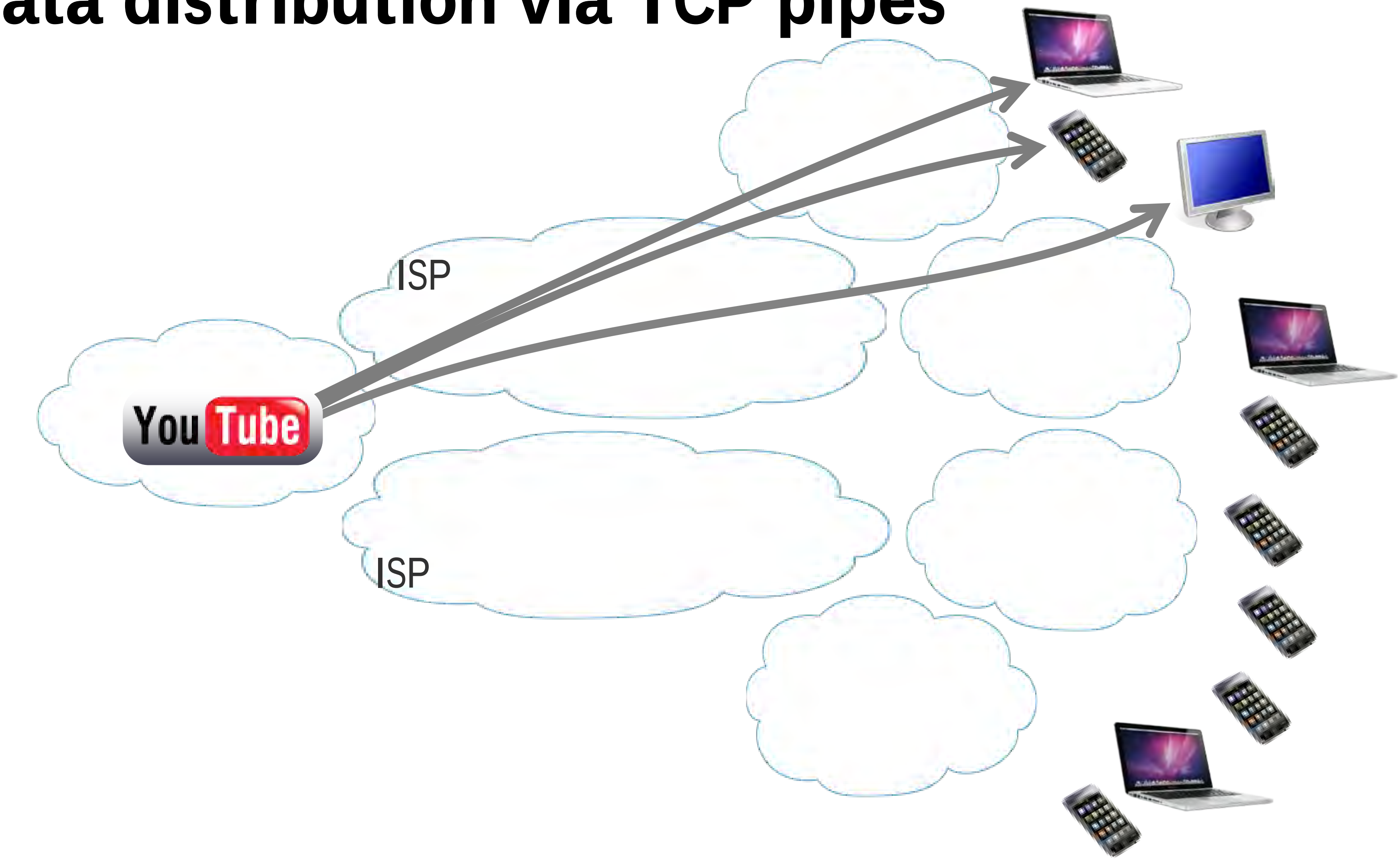
Data distribution via TCP pipes



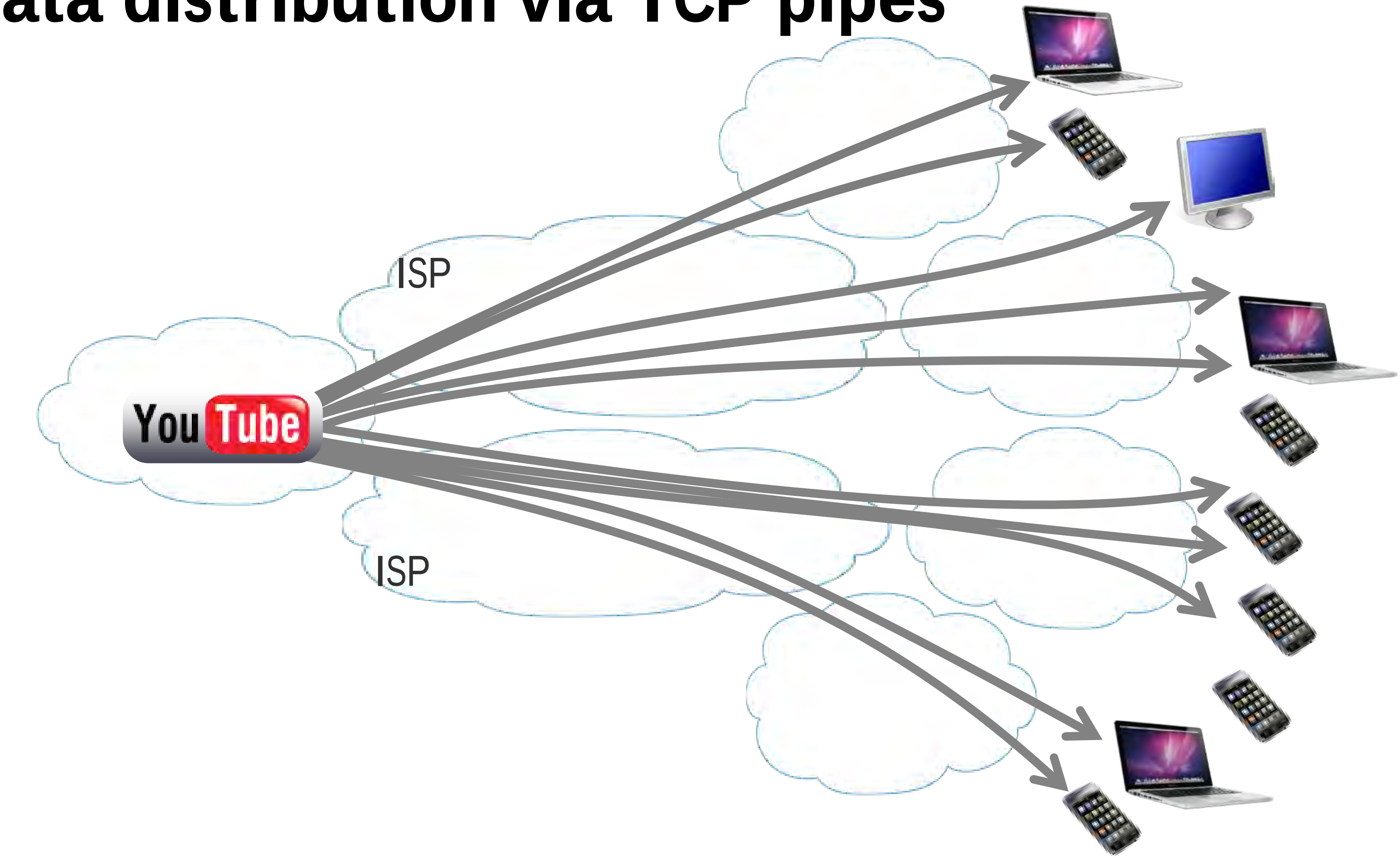
Data distribution via TCP pipes



Data distribution via TCP pipes



Data distribution via TCP pipes



Challenges Caused By Point-to-Point Communication

Difficult to

- *disseminate data to a large group of users.*
- *handle mobile users whose addresses change.*
- *secure data as it moves from device to device.*

A Simpler Way

Suppose you could ask for what you wanted?

/memphis.edu/lanwang/talks/UALR14.pdf

THE UNIVERSITY OF MEMPHIS

Dreamers. Thinkers. Doers.

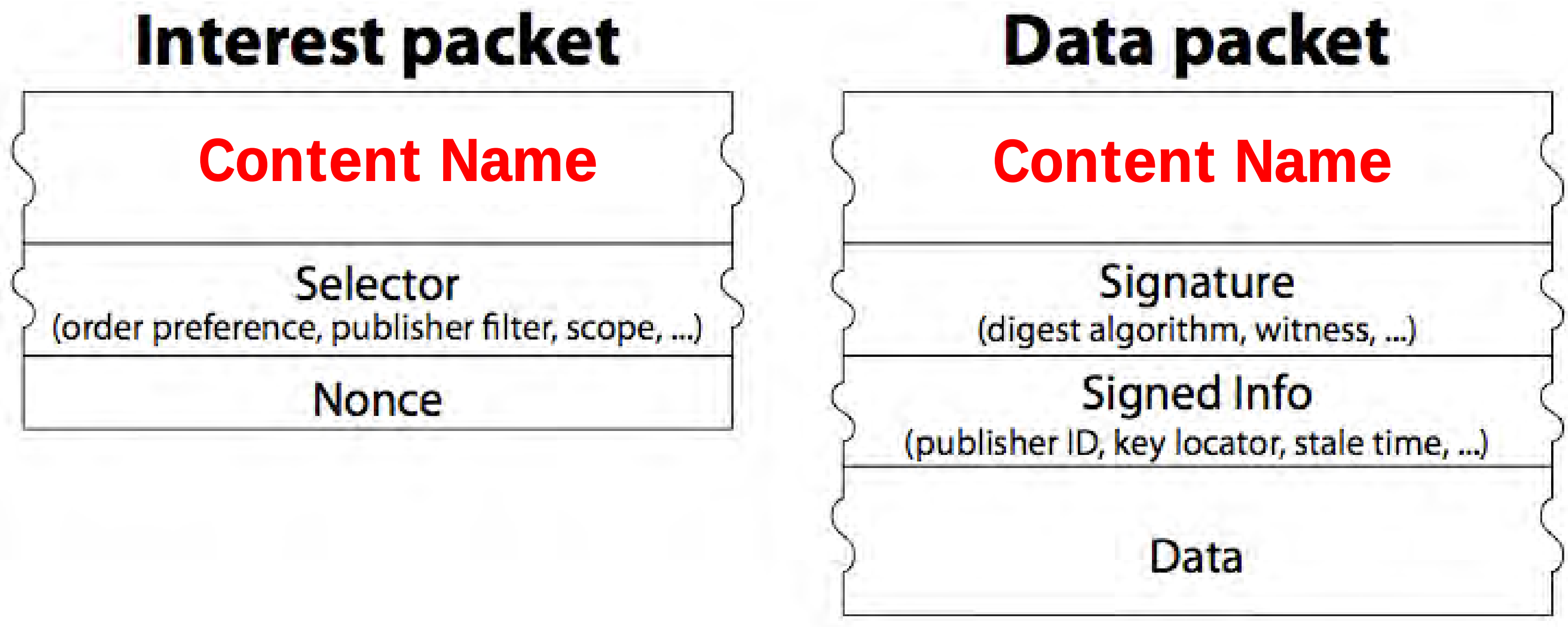
NDN Architectural Development and Routing Design

Lan Wang
University of Memphis
Nov. 21, 2014
www.named-data.net

/named-data.net/video/van-ccnx

/room/thermostat/1/status

NDN: Retrieving Named Data



Interest packet

Content Name

Selector
(order preference, publisher filter, scope, ...)

Nonce

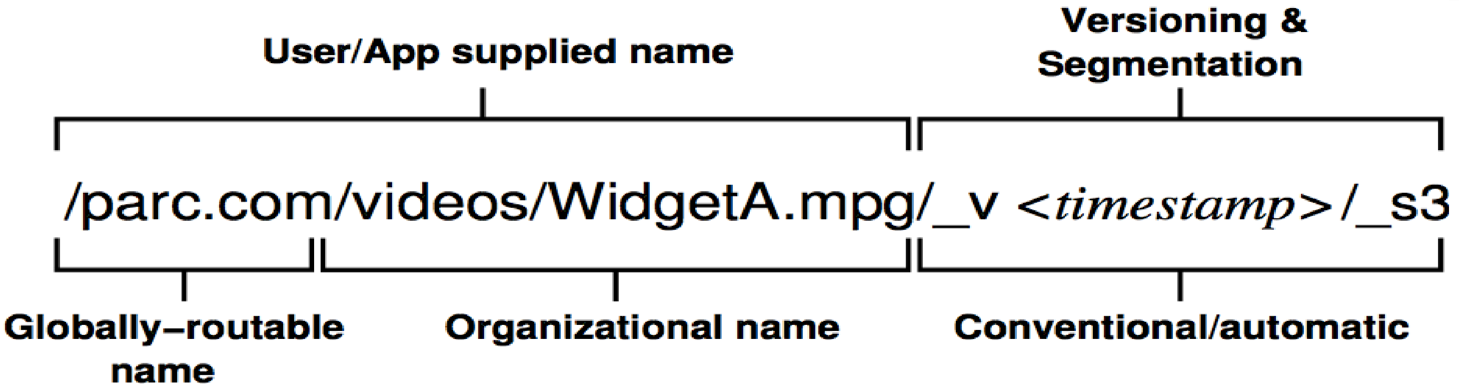
Data packet

Content Name

Signature
(digest algorithm, witness, ...)

Signed Info
(publisher ID, key locator, stale time, ...)

Data



User/App supplied name

Versioning & Segmentation

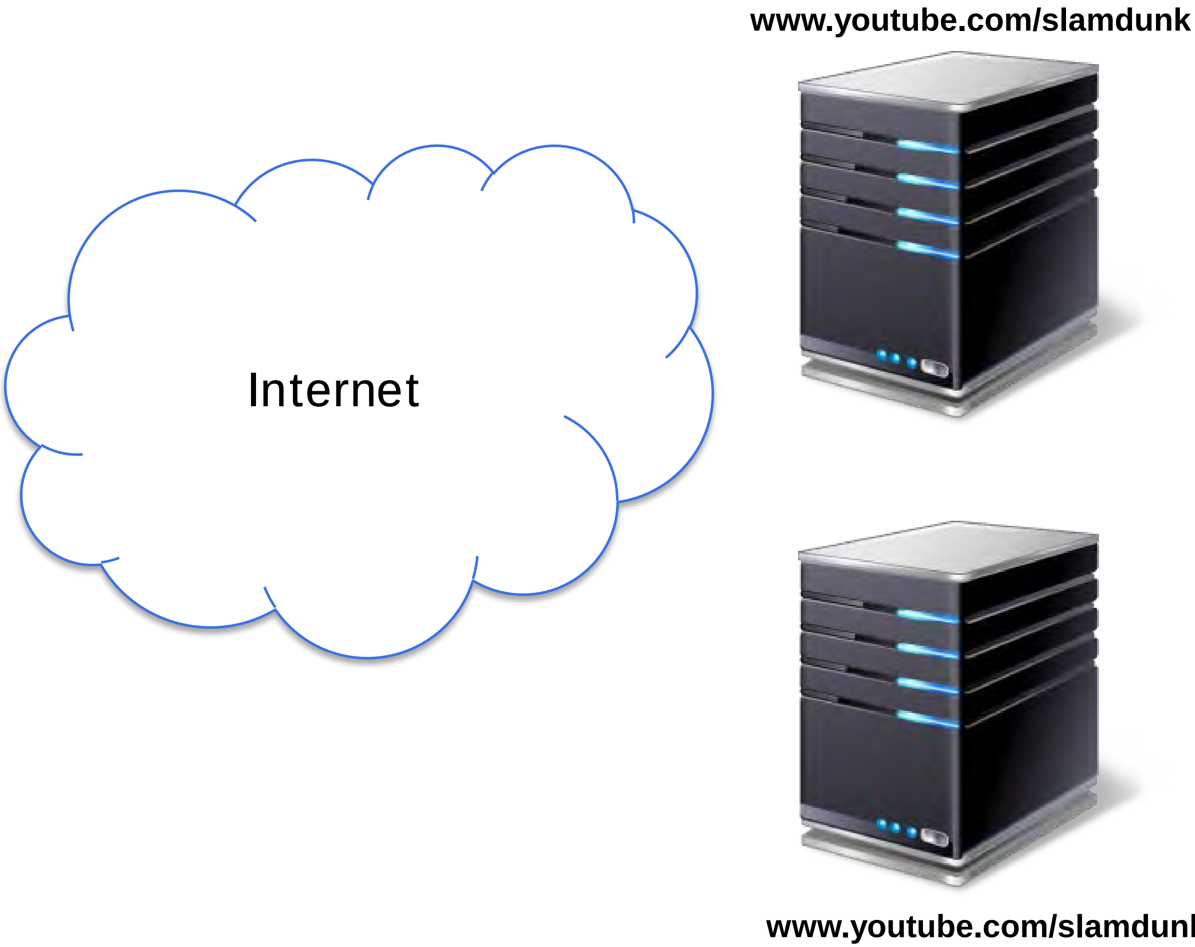
/parc.com/videos/WidgetA.mpg/_v <*timestamp*>/_s3

Globally–routable name

Organizational name

Conventional/automatic

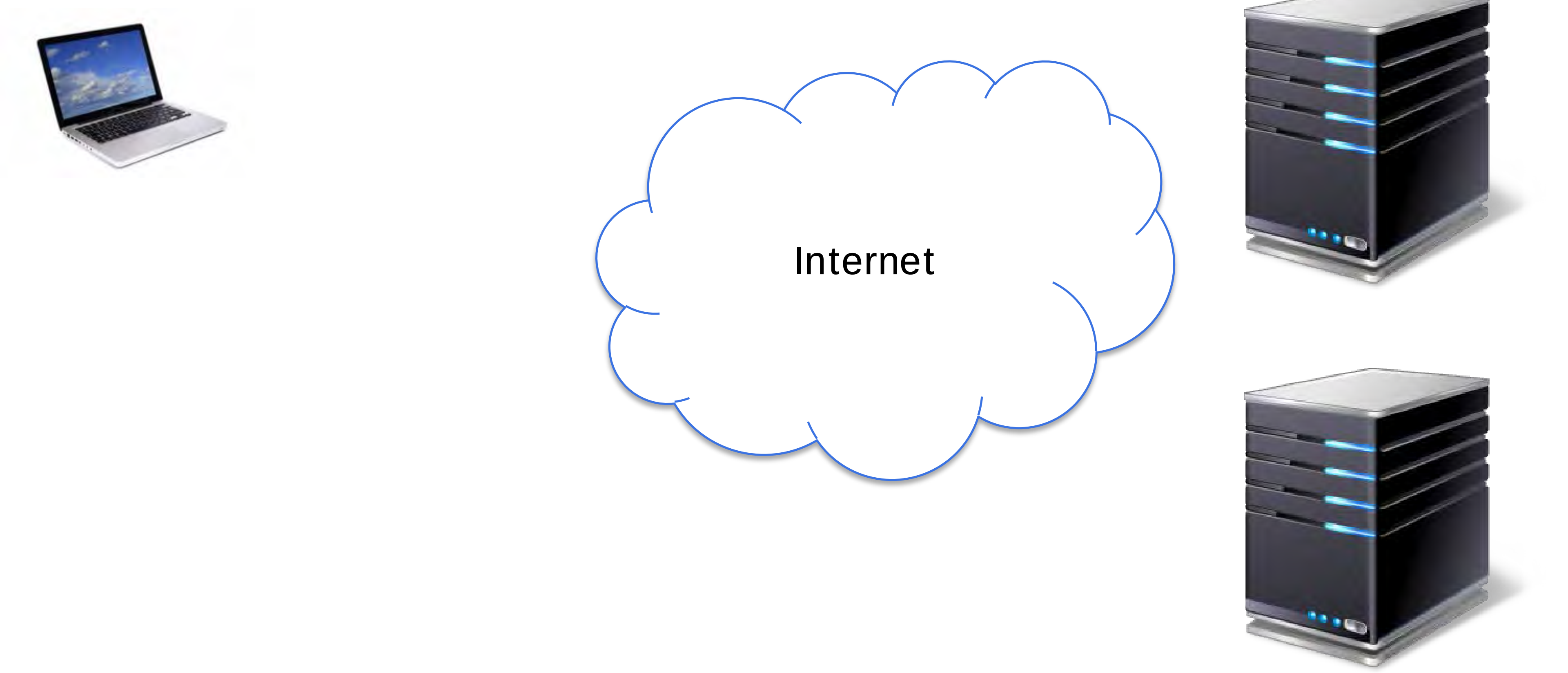
Named Data Networking (NDN)



Named Data Networking (NDN)

www.youtube.com/slamdunk

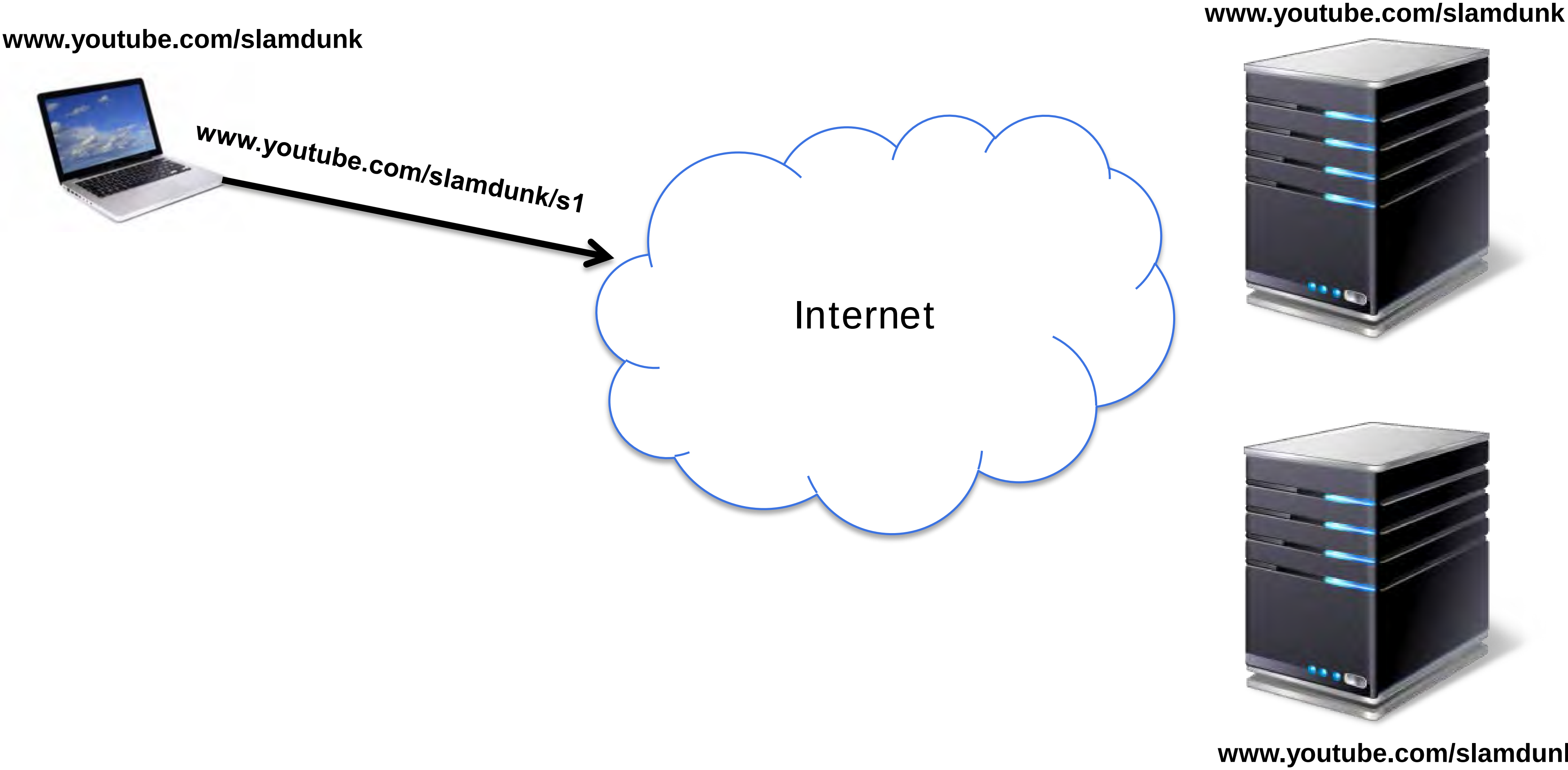
www.youtube.com/slamdunk



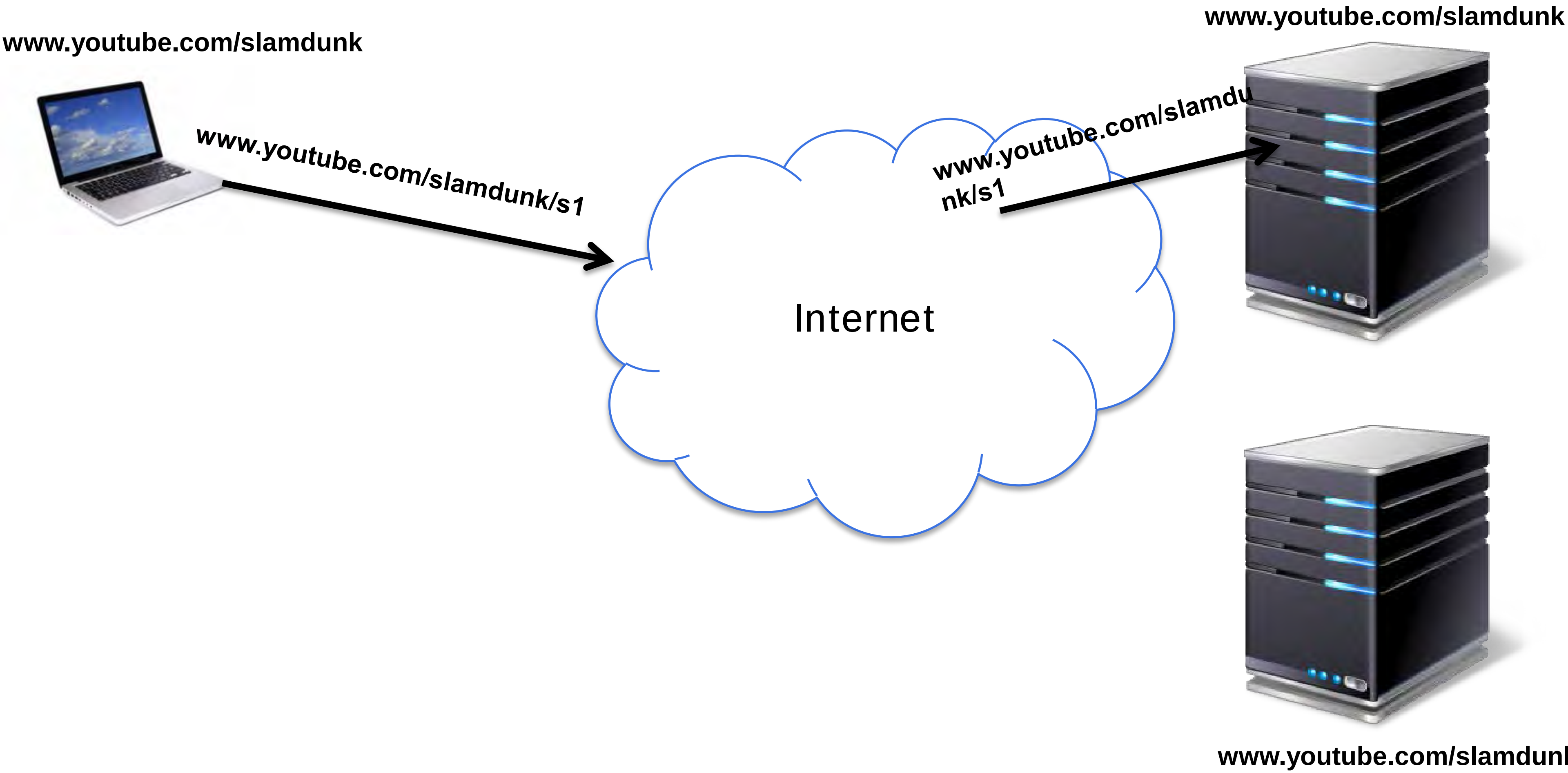
Internet

www.youtube.com/slamdunl

Named Data Networking (NDN)



Named Data Networking (NDN)



www.youtube.com/slamdunk

www.youtube.com/slamdunk/s1

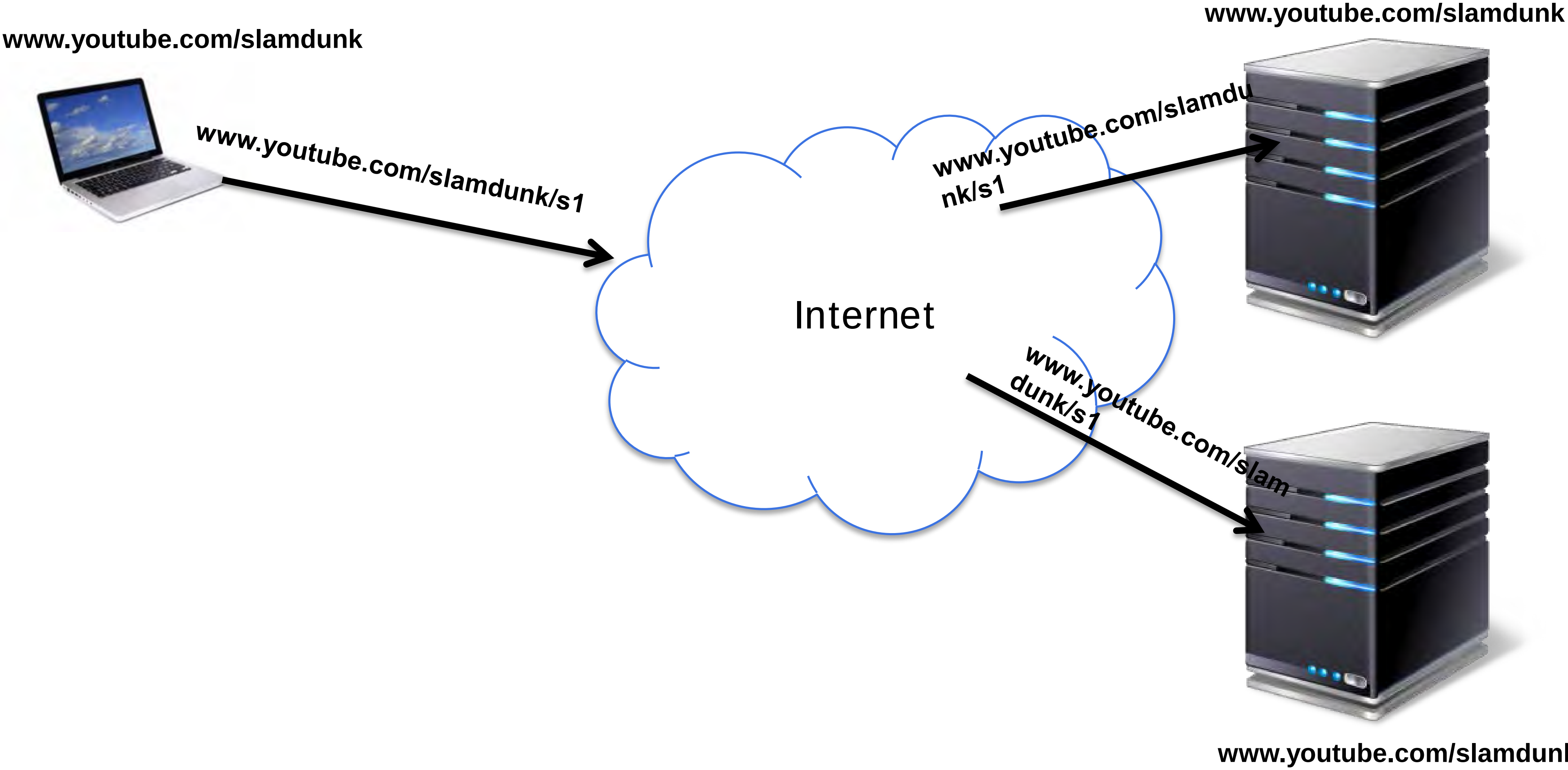
Internet

www.youtube.com/slamdunk/s1

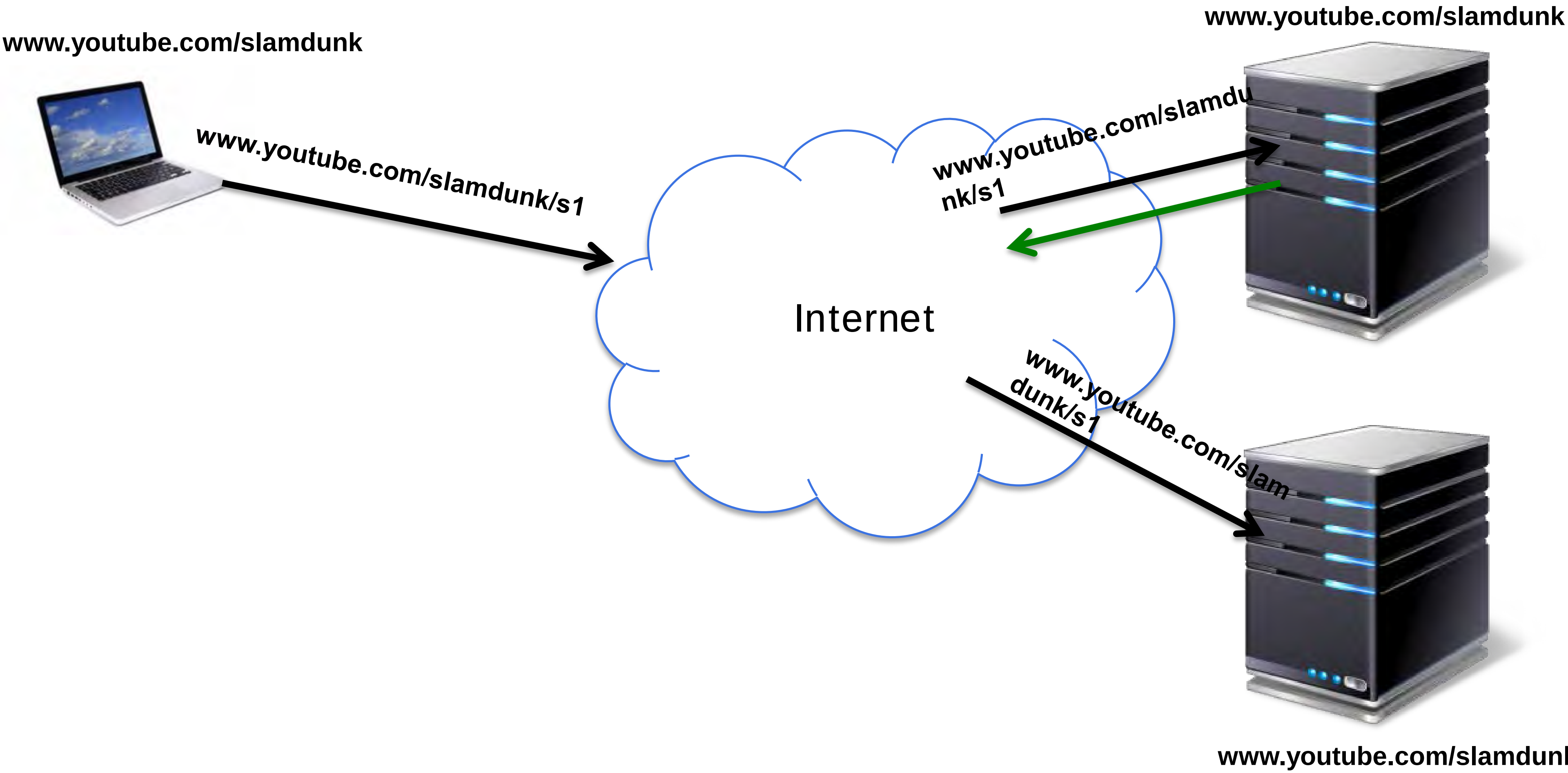
www.youtube.com/slamdunk

www.youtube.com/slamdunk

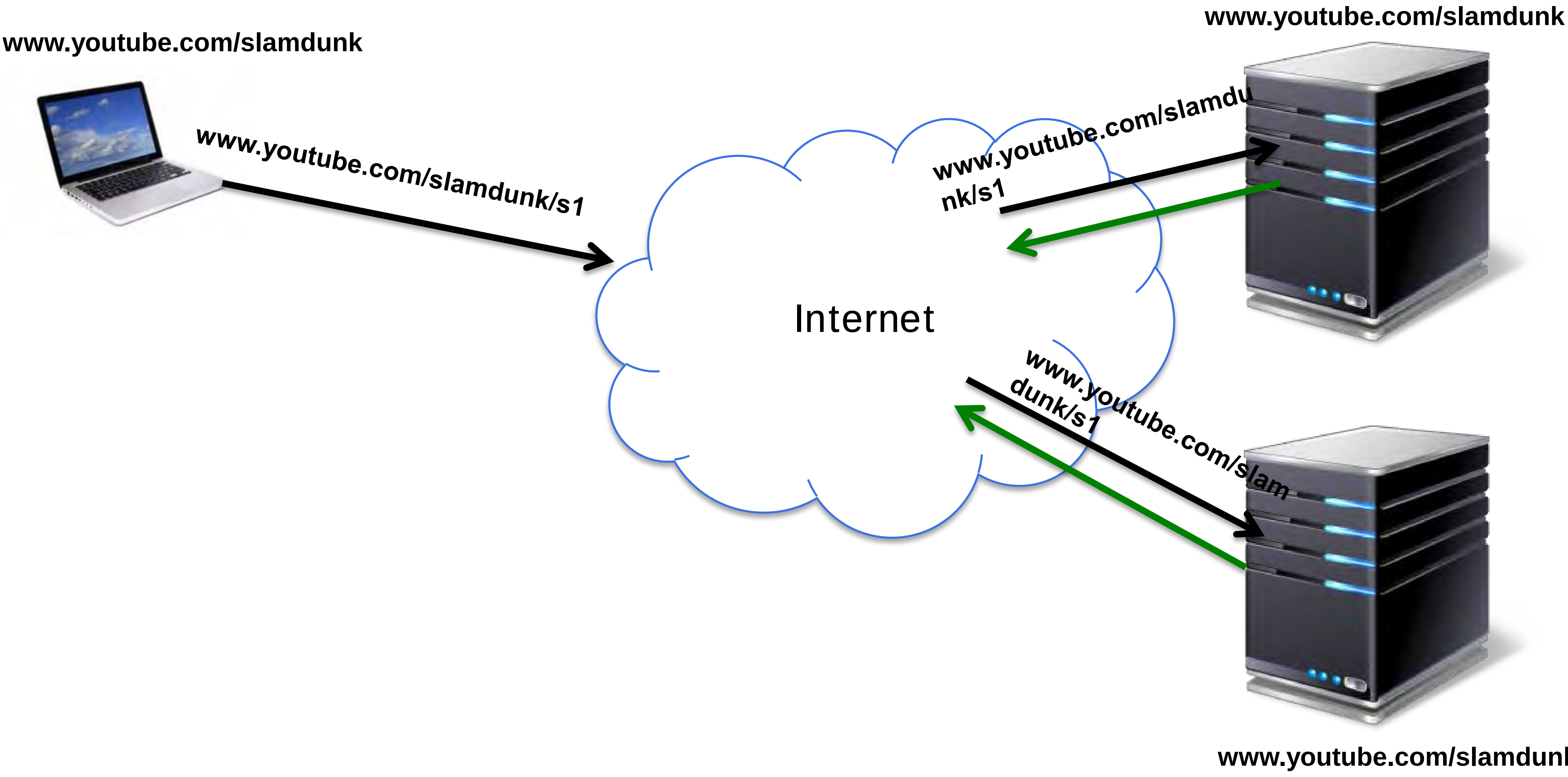
Named Data Networking (NDN)


www.youtube.com/slamdunk
www.youtube.com/slamdunk
www.youtube.com/slamdunk/s1
www.youtube.com/slamdunk/s1
Internet
www.youtube.com/slamdunk/s1
www.youtube.com/slamdunl

Named Data Networking (NDN)


www.youtube.com/slamdunk
www.youtube.com/slamdunk
www.youtube.com/slamdunk/s1
www.youtube.com/slamdunk/s1
Internet
www.youtube.com/slamdunk/s1
www.youtube.com/slamdunl

Named Data Networking (NDN)



www.youtube.com/slamdunk

www.youtube.com/slamdunk/s1

Internet

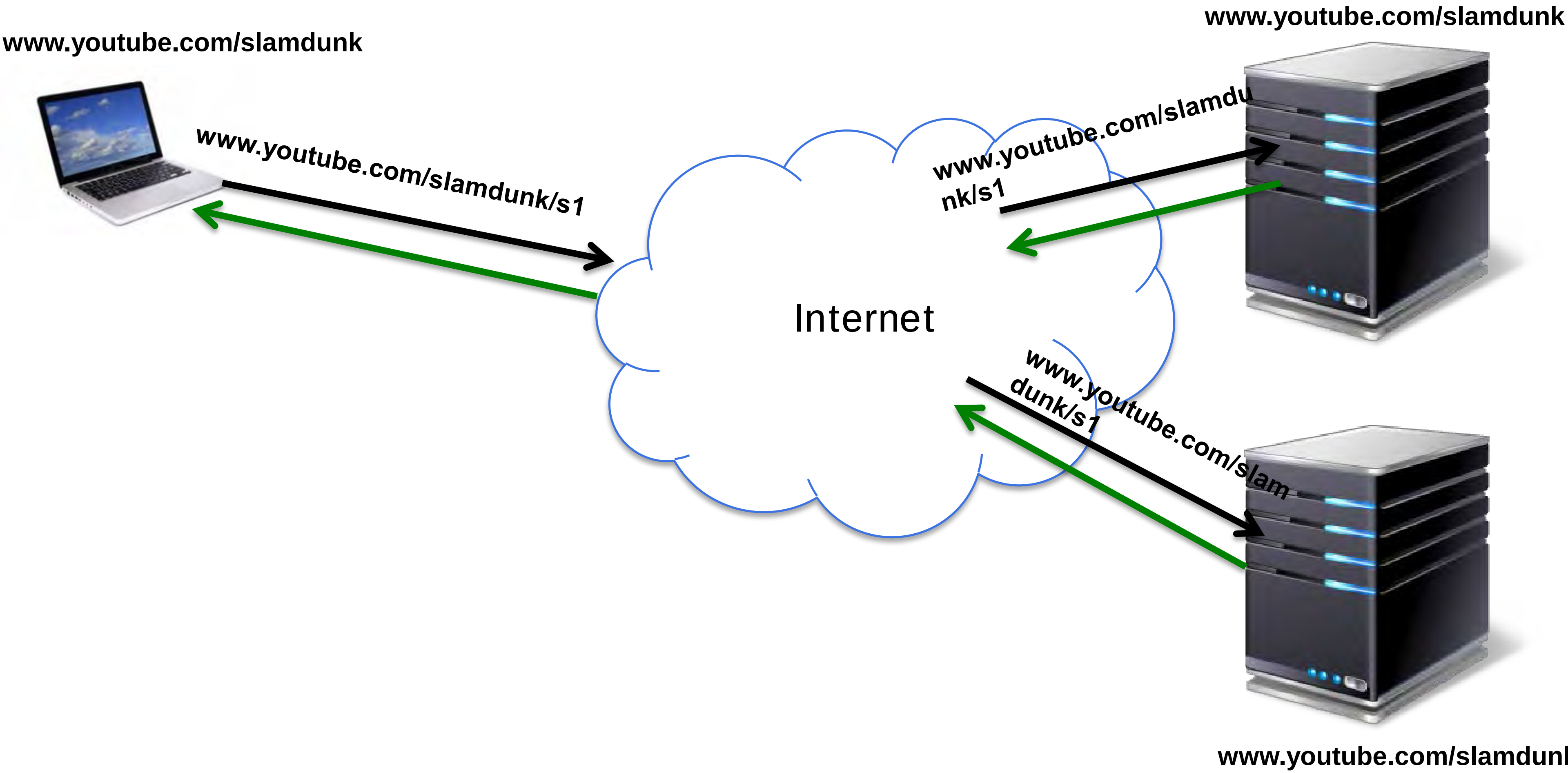
www.youtube.com/slamdunk

www.youtube.com/slamdunk/s1

www.youtube.com/slamdunk/s1

www.youtube.com/slamdunk

Named Data Networking (NDN)



www.youtube.com/slamdunk

www.youtube.com/slamdunk

www.youtube.com/slamdunk/s1

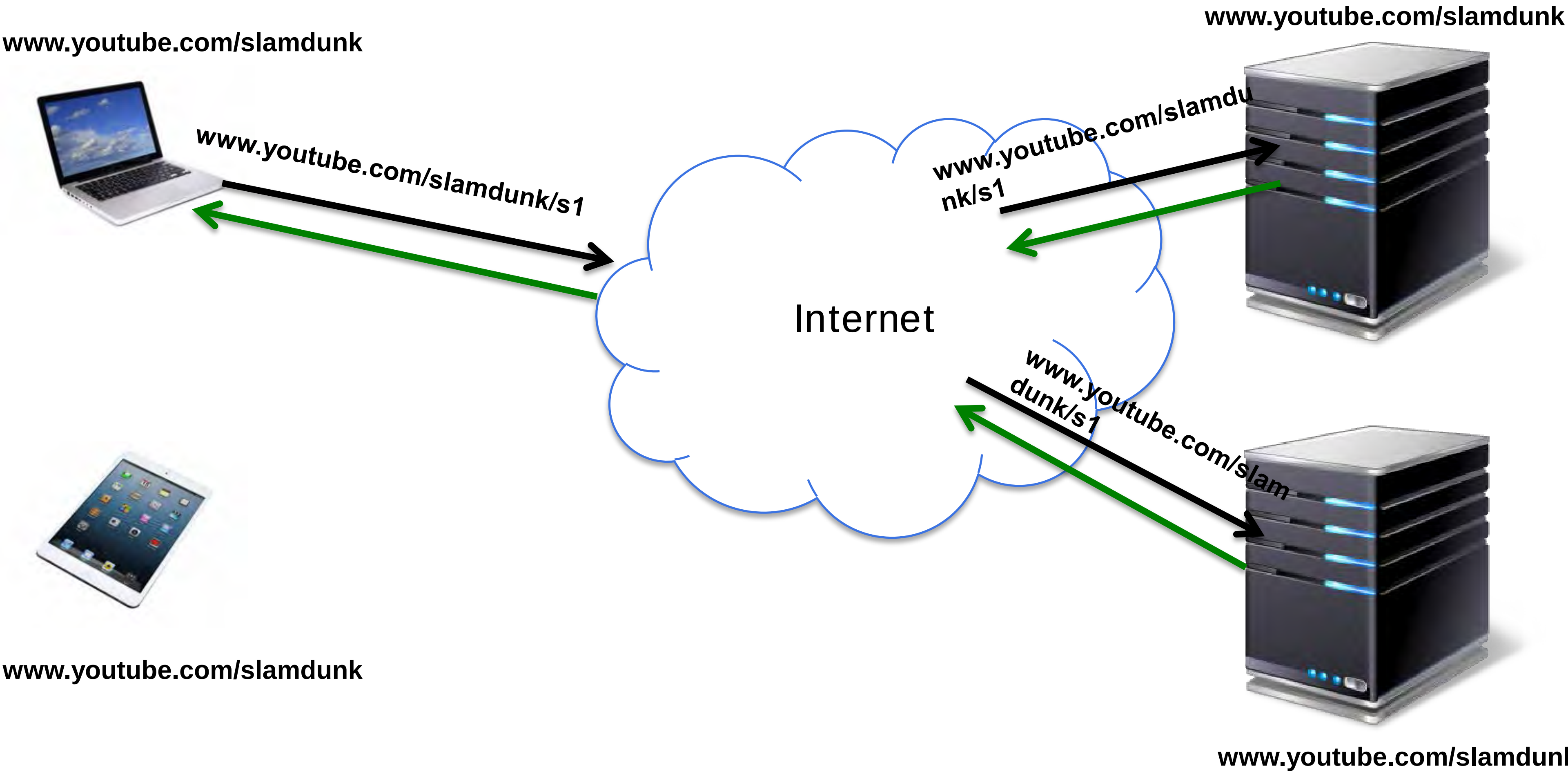
Internet

www.youtube.com/slamdunk/s1

www.youtube.com/slamdunk/s1

www.youtube.com/slamdunl

Named Data Networking (NDN)



www.youtube.com/slamdunk

www.youtube.com/slamdunk

www.youtube.com/slamdunk/s1

www.youtube.com/slamdu nk/s1

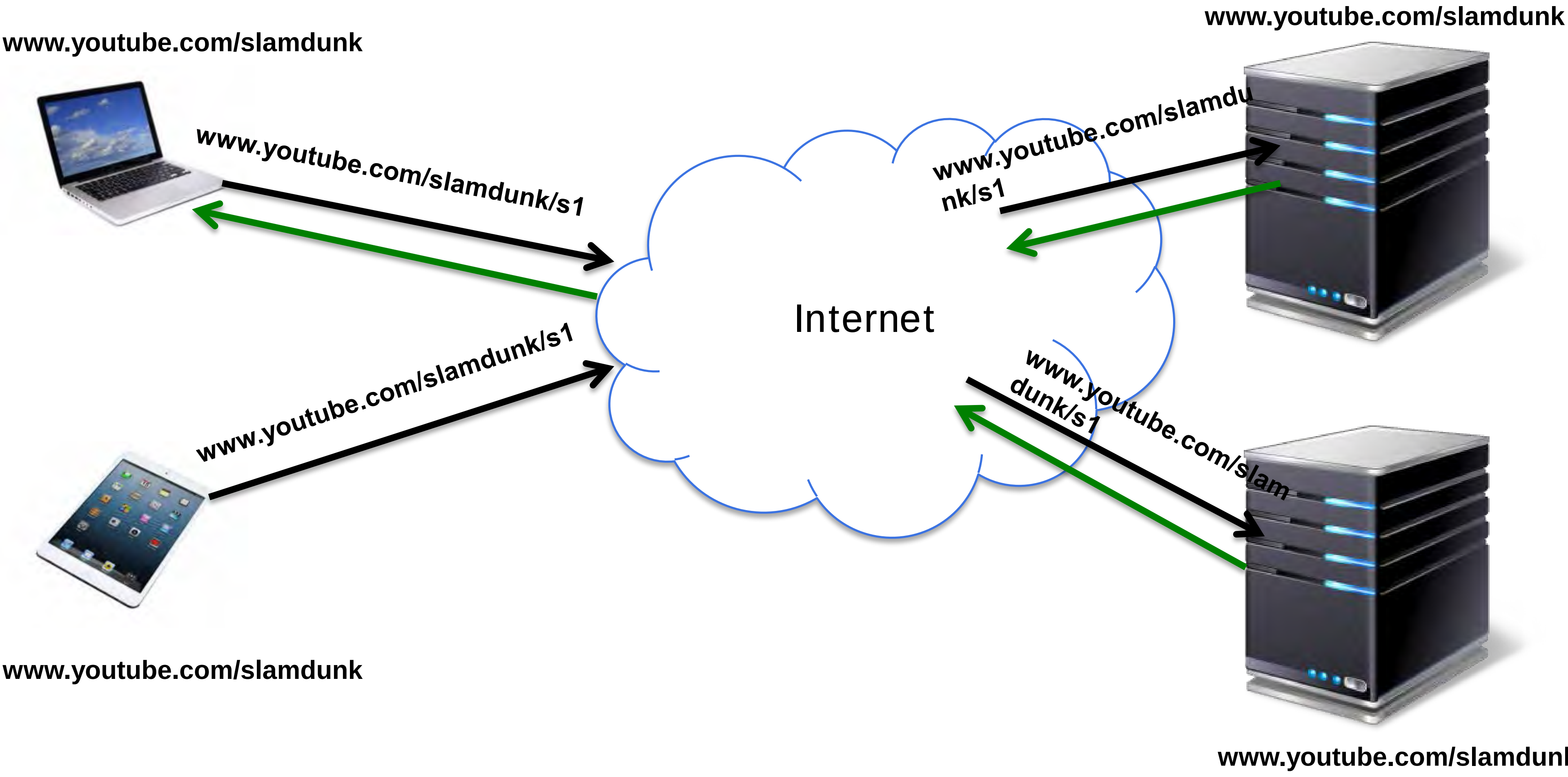
Internet

www.youtube.com/slam dunk/s1

www.youtube.com/slamdunk

www.youtube.com/slamdunk

Named Data Networking (NDN)



www.youtube.com/slamdunk

www.youtube.com/slamdunk

www.youtube.com/slamdunk/s1

www.youtube.com/slamdunk/s1

Internet

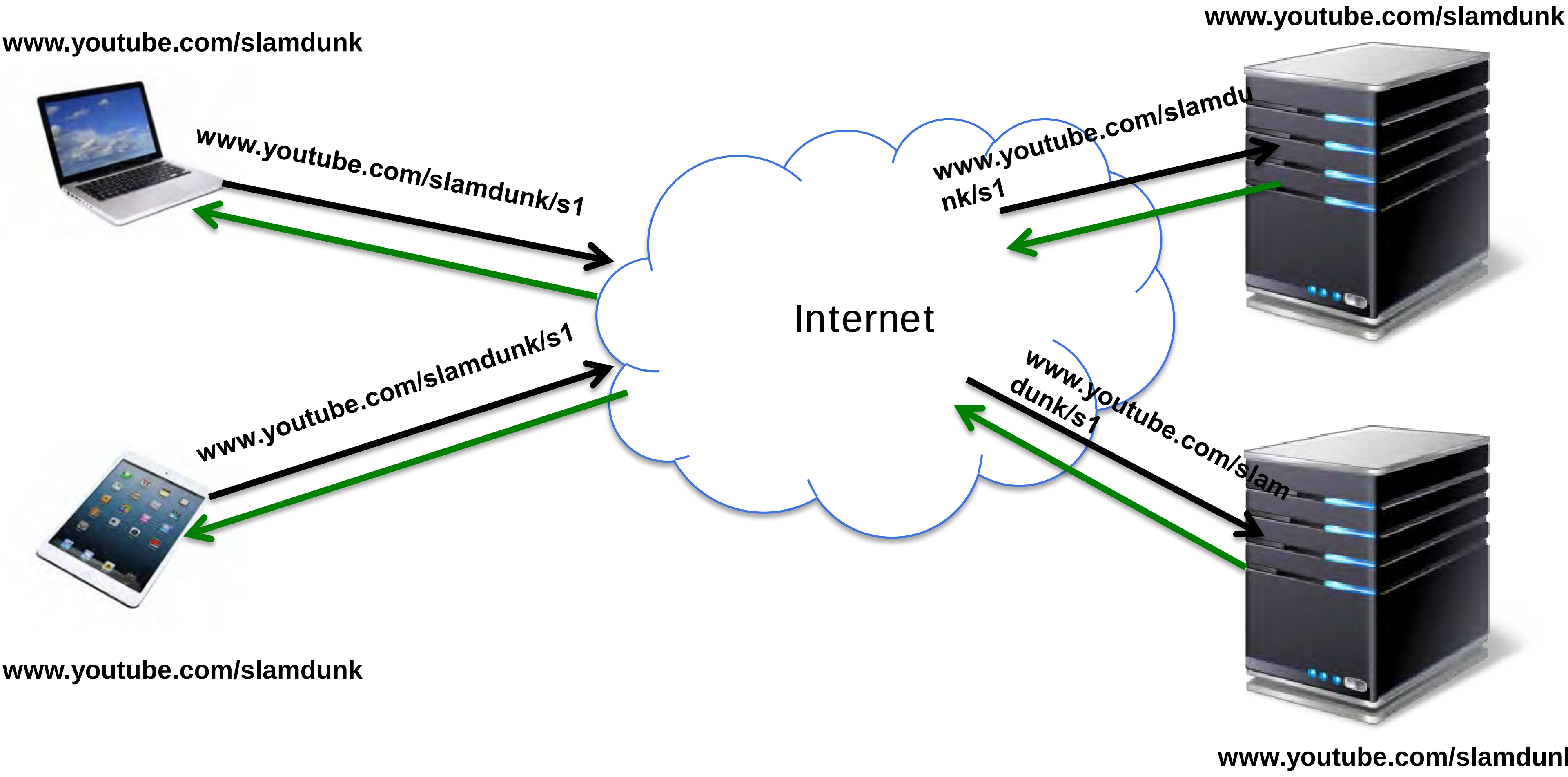
www.youtube.com/slamdu
nk/s1

www.youtube.com/slam
dunk/s1

www.youtube.com/slamdunk

www.youtube.com/slamdunk

Named Data Networking (NDN)



www.youtube.com/slamdunk

www.youtube.com/slamdunk

www.youtube.com/slamdunk/s1

www.youtube.com/slamdunk/s1

Internet

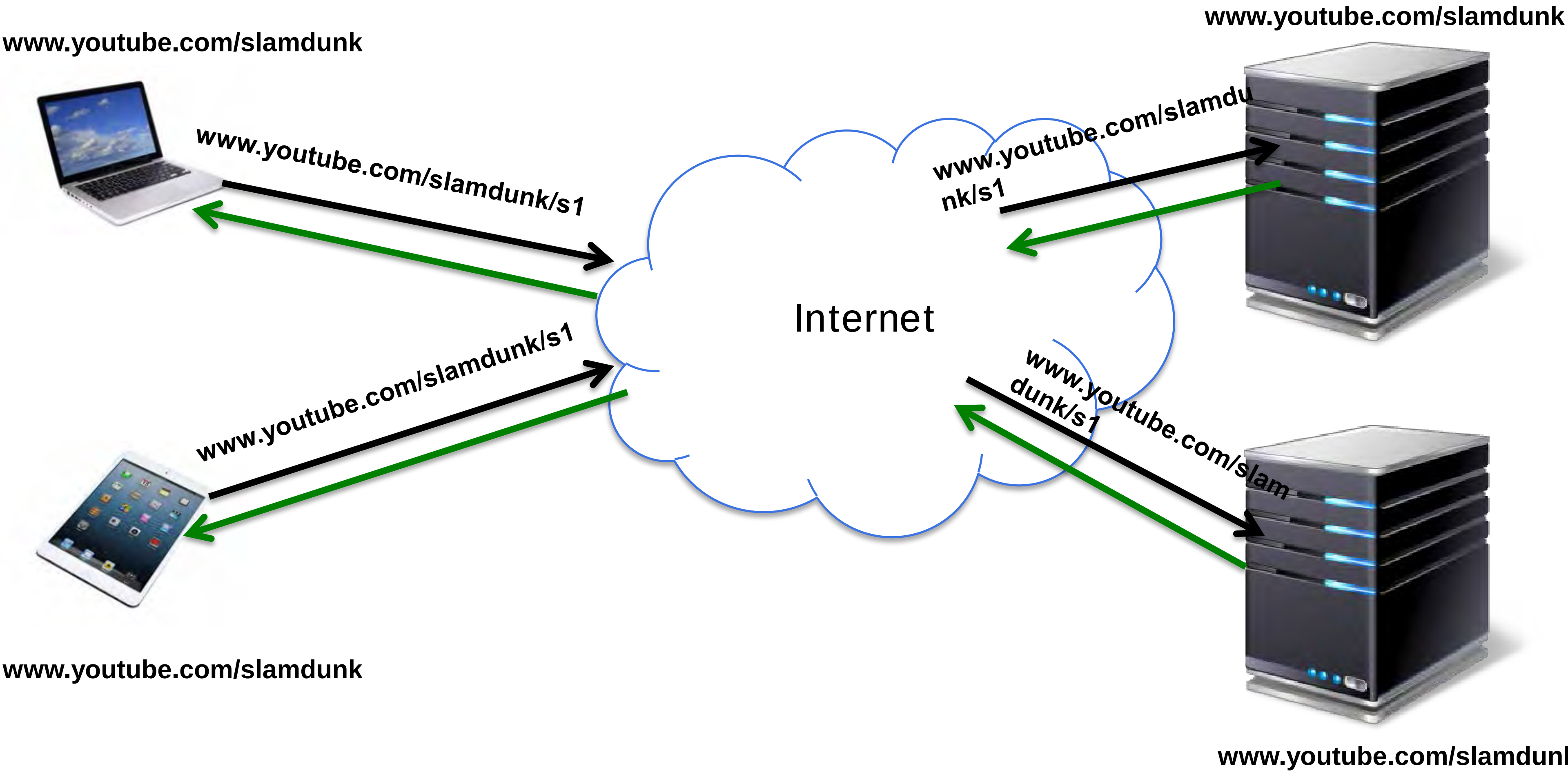
www.youtube.com/slamdunk/s1

www.youtube.com/slamdunk/s1

www.youtube.com/slamdunk

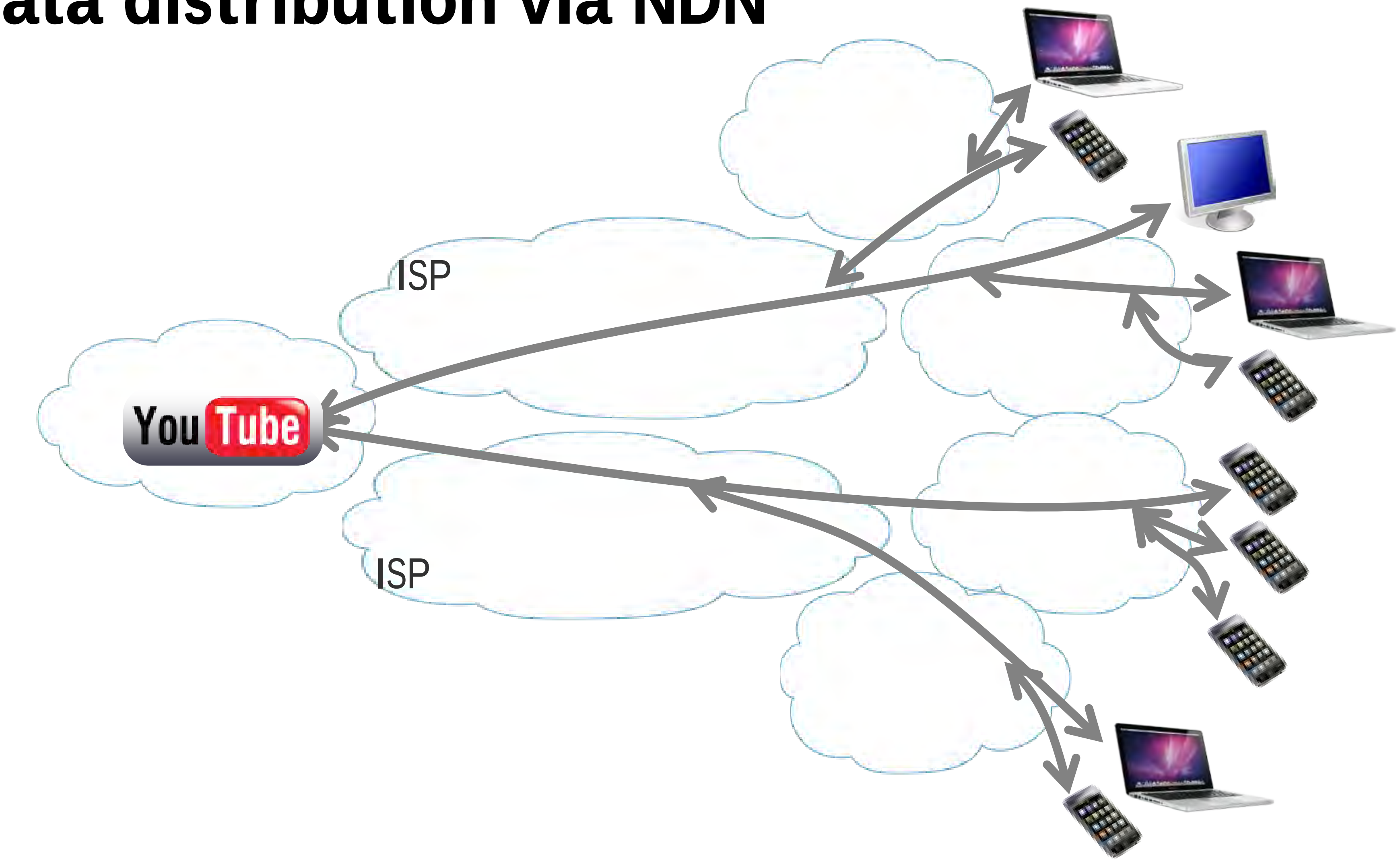
www.youtube.com/slamdunl

Named Data Networking (NDN)

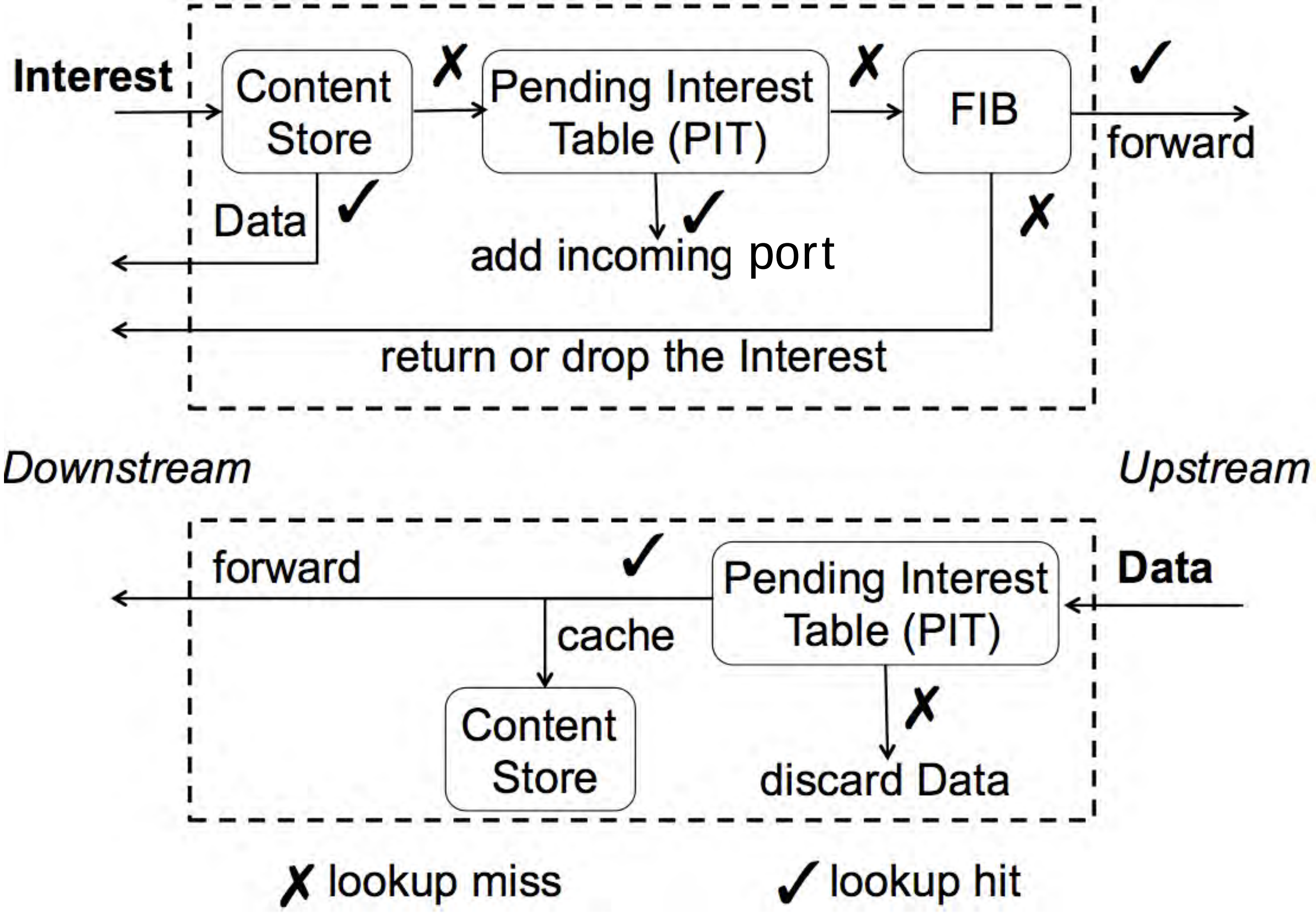

www.youtube.com/slamdunk
www.youtube.com/slamdunk
www.youtube.com/slamdunk/s1
www.youtube.com/slamdunk/s1
Internet
www.youtube.com/slamdunk/s1
www.youtube.com/slamdunk/s1
www.youtube.com/slamdunk
www.youtube.com/slamdunl

- NDN uses “Interest” and “Data” packets to get “content”

Data distribution via NDN



Packet forwarding in NDN



Why network security is difficult today?

Why network security is difficult today?

"the 'trust' the user gets in the content they depend upon is tied inextricably to the connection over which it was retrieved, that trust is *transient* – leaving no reliable traces on the content after the connection ends"

Why network security is difficult today?

> "the 'trust' the user gets in the content they depend upon is tied inextricably to the connection over which it was retrieved, that trust is *transient* – leaving no reliable traces on the content after the connection ends"

Old way of thinking: add boundary for protection

- Securing my IP network → securing its perimeter
- But people punch holes on the barrier to get work done
- Strong security → burden/barrier to communication

Think differently

- The ultimate question from applications: did I get the right data I need?
  - and did others peek into my data (if secrecy required)?
- NDN secures data directly: every Data packet is signed by the producer

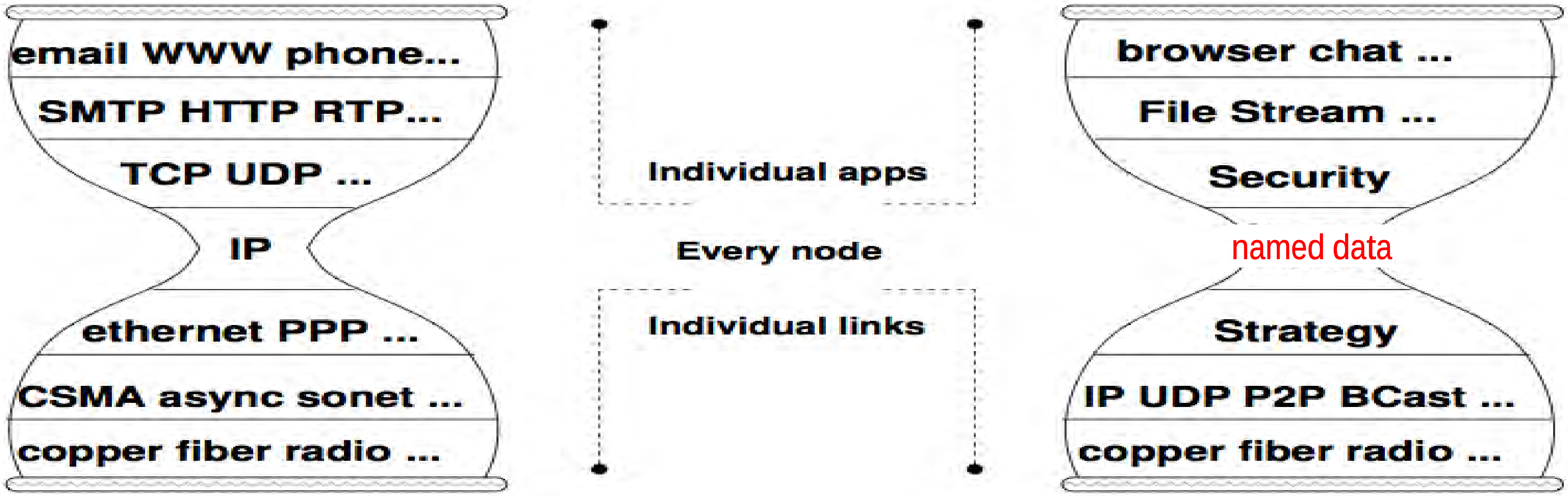
  → *reduces* the reliance on trusting network intermediaries

→This leaves one problem to solve: verify that the data is indeed signed by the producer's key

→Each NDN application has its trust model to verify producer's key.

NDN ARCHITECTURAL DEVELOPMENT

Preserving the Hourglass Shape



- The only layer that requires universal agreement is layer 3, the network layer
- Much of IP’s success is due to the simplicity of its network layer and the weak demands it makes on layer 2

NDN Architecture Development [1]

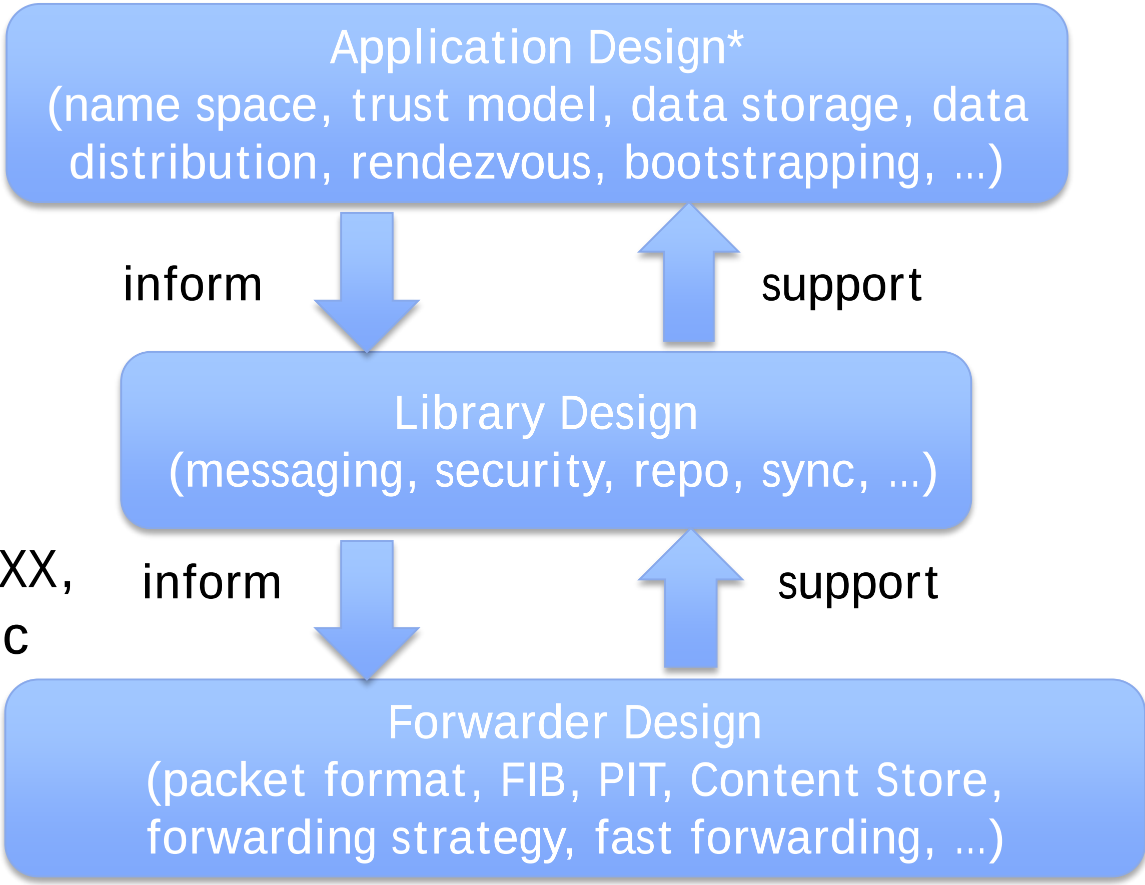
- application-driven
- test and deploy on operational testbed
- conduct real-world demos

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, kc claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, Named Data Networking, in *ACM SIGCOMM CCR,* July 2014 (also *NDN Technical Report 0019*)

Progress

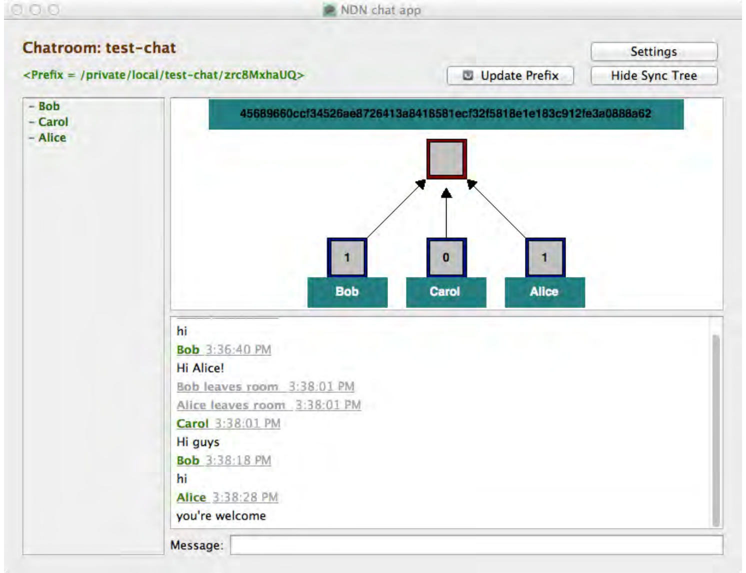
All code is open source at https://github.com/named-data/.

- Multimedia applications
  - NDNVideo [2]
  - ChronoChat [3]
  - NDN-RTC
  - ChronoShare [4]
- IoT applications
  - Building automation and management [5, 6]
  - vehicular net [7]
- Libraries: NDN-CCL, NDN-CXX, pyNDN2, ndn-js, ChronoSync [9], NDN repo
- NDN Forwarding Daemon (NFD)
- Routing protocol [8]


Application Design*
(name space, trust model, data storage, data distribution, rendezvous, bootstrapping, …)
inform
support
Library Design
(messaging, security, repo, sync, …)
inform
support
Forwarder Design
(packet format, FIB, PIT, Content Store, forwarding strategy, fast forwarding, …)

ChronoChat [3]

- **server-less** chat application based on ChronoSync
- chat messages are synchronized among participants
- leverage multicast nature of NDN


NDN chat app
Settings
Update Prefix
Hide Sync Tree
45689660ccf34526ae8726413a8418581ecf32f5818e1e183c912fe3a0888a62
1
Bob
0
Carol
1
Alice

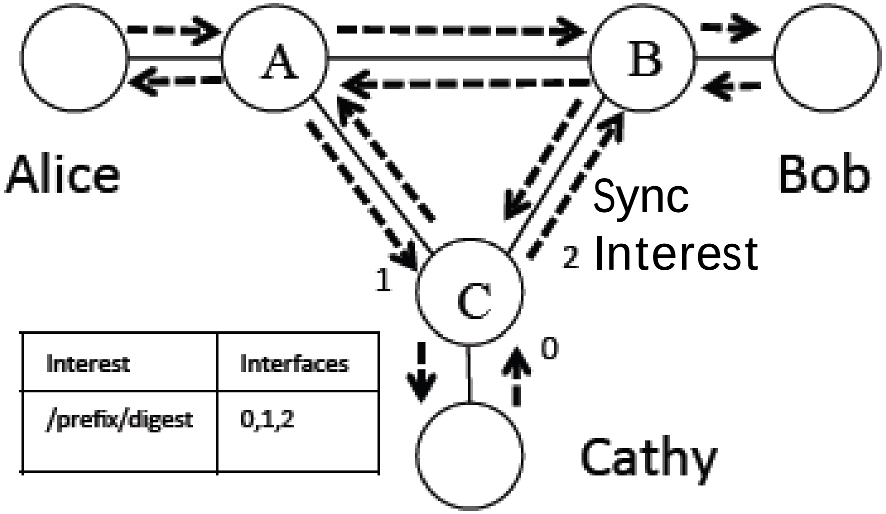
Chatroom: test-chat

<Prefix = /private/local/test-chat/zrc8MxhaUQ>

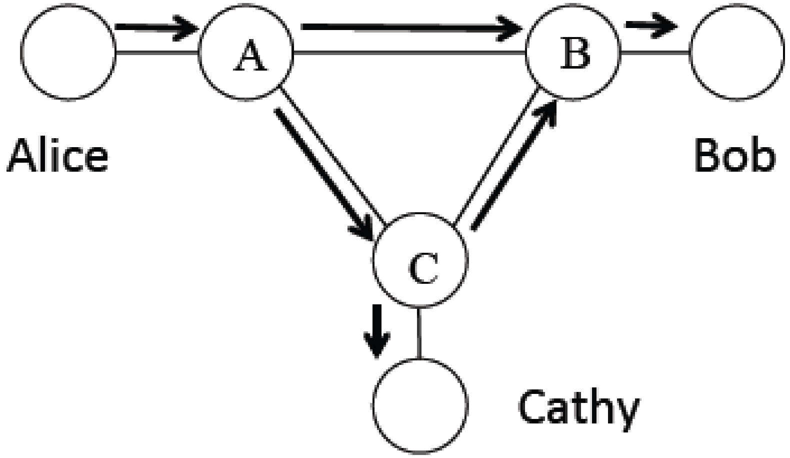
- Bob
- Carol
- Alice

hi
Bob 3:36:40 PM
Hi Alice!
Bob leaves room 3:38:01 PM
Alice leaves room 3:38:01 PM
Carol 3:38:01 PM
Hi guys
Bob 3:38:18 PM
hi
Alice 3:38:28 PM
you're welcome

Message:



Interest	Interfaces
/prefix/digest	0,1,2

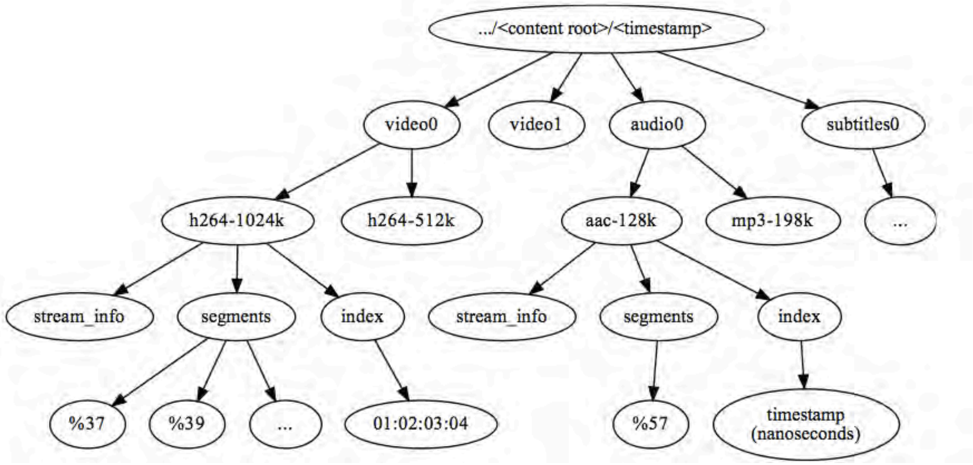


Alice's Sync Data Packet

[3] Z. Zhu, C. Bian, A. Afanasyev, V. Jacobson, and L. Zhang. Chronos: Serverless multi-user chat over NDN. *Technical Report NDN-0008*, NDN Project, October 2012.

NDNVideo [2]

- Live and pre-recorded streaming to multiple consumers.
- No session semantics => scalability. Tested for ~1000 clients from 1 src
- First Interest sent can randomly access a keyframe at any timecode
- Leverages caching.



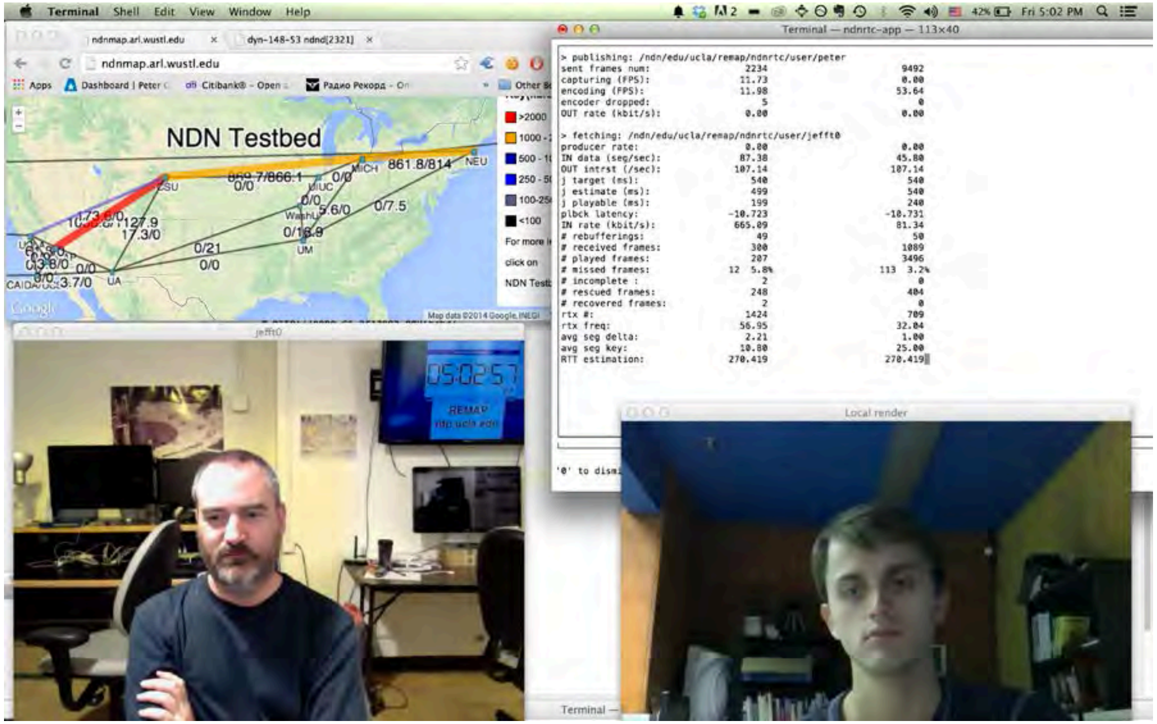
[2] D. Kulinski, J. Burke, and L. Zhang. "Video Streaming over Named Data Networking," *IEEE COMSOC MMTC E-Letter*, 2013.

ndnrtc

- Real-time audio/video/text chat application enabling multi-peer conferencing over NDN.
- Version 0.1.0: Oct 2014.

- Based on WebRTC codebase, using ChronoSync for conference discoveries.

- Explore how to handle packet losses and delays while maintaining a session-less approach.

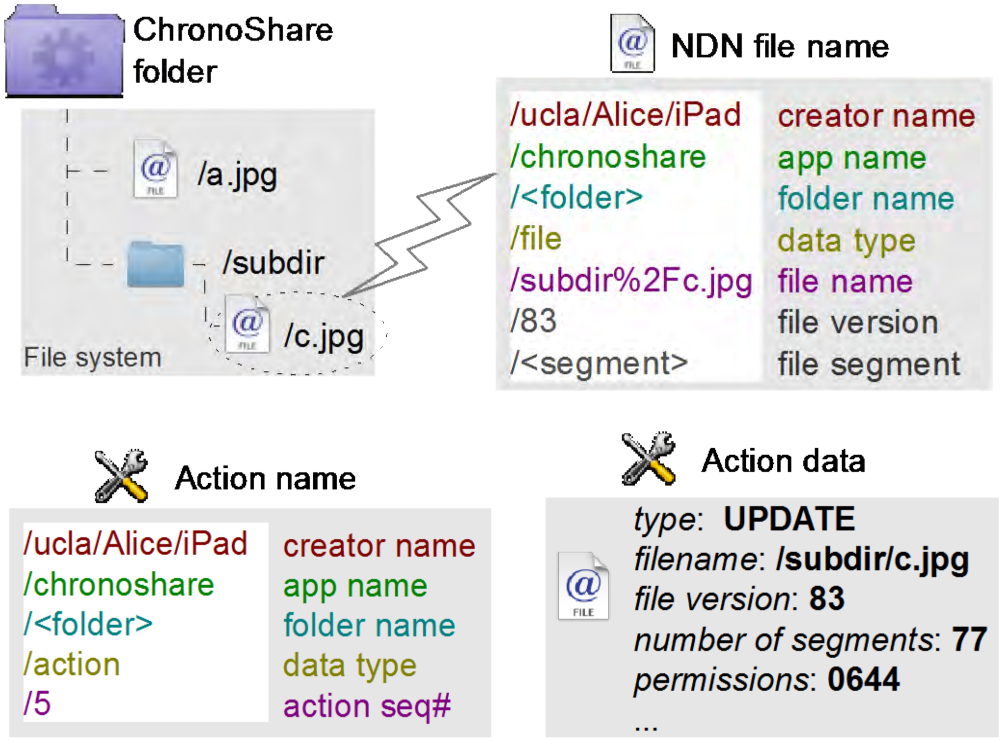


```
> publishing: /ndn/edu/ucla/remap/ndnrtc/user/peter
sent frames num:                  2234                     9492
capturing (FPS):                 11.73                     0.00
encoding (FPS):                  11.98                    53.64
encoder dropped:                     5                        0
OUT rate (kbit/s):                0.00                     0.00

> fetching: /ndn/edu/ucla/remap/ndnrtc/user/jefft0
producer rate:                    0.00                     0.00
IN data (seg/sec):               87.38                    45.80
OUT intrst (/sec):              107.14                   107.14
j target (ms):                     540                      540
j estimate (ms):                   499                      540
j playable (ms):                   199                      240
plbck latency:                 -10.723                  -10.731
IN rate (kbit/s):               665.09                    81.34
# rebufferings:                     49                       50
# received frames:                 300                     1089
# played frames:                   207                     3496
# missed frames:                12  5.8%               113  3.2%
# incomplete :                       2                        0
# rescued frames:                  248                      404
# recovered frames:                  2                        0
rtx #:                            1424                      709
rtx freq:                        56.95                    32.04
avg seg delta:                    2.21                     1.00
avg seg key:                     10.80                    25.00
RTT estimation:                270.419                  270.419
```

ChronoShare: Distributed File Sharing and Editing [4]

- Think Google Drive, but no centralized server
  - Different users can share folders.
  - Each user can sync folders on different devices.
- How?
  - Each user's actions (on file) form a stream of data.
  - Use ChronoSync to distribute knowledge of user actions



/ucla/Alice/iPad	creator name
/chronoshare	app name
/<folder>	folder name
/file	data type
/subdir%2Fc.jpg	file name
/83	file version
/<segment>	file segment

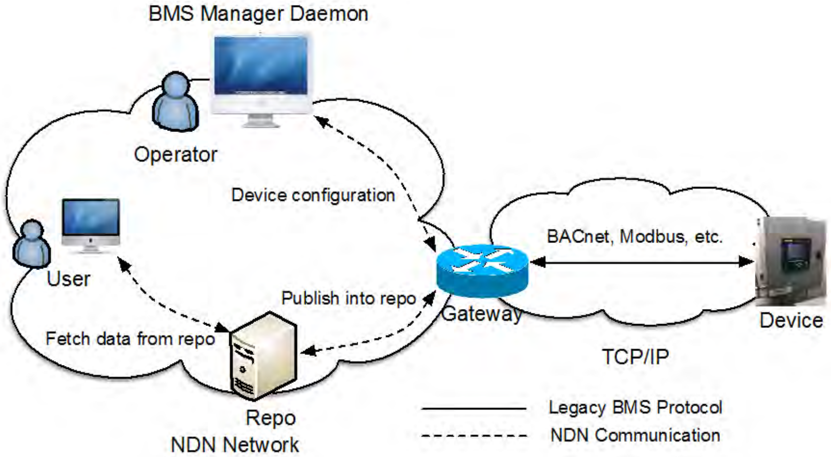
/ucla/Alice/iPad	creator name
/chronoshare	app name
/<folder>	folder name
/action	data type
/5	action seq#

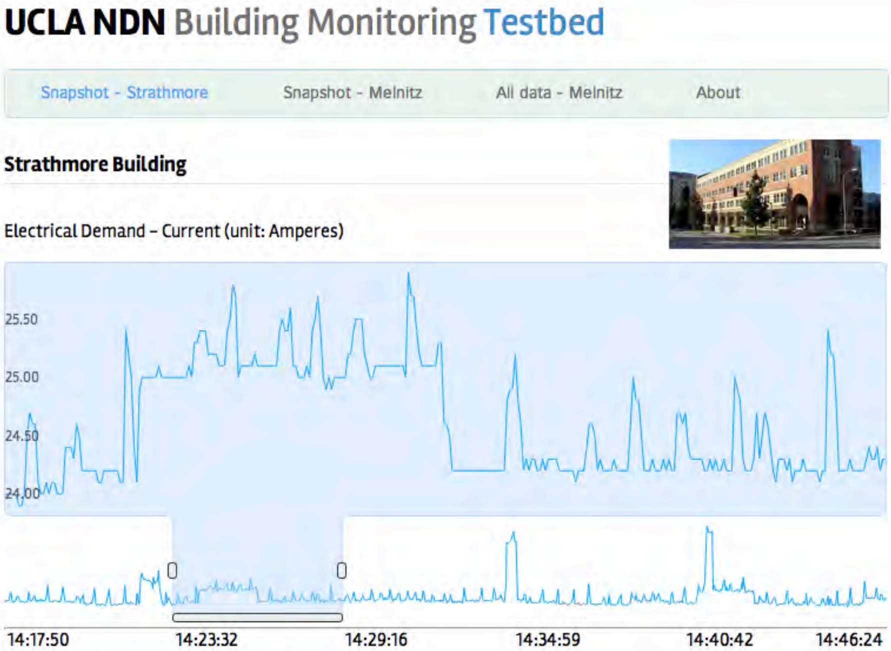
type: **UPDATE**
filename: **/subdir/c.jpg**
file version: **83**
number of segments: **77**
permissions: **0644**
...

[5] A. Afanasyev, Z. Zhu, L. Zhang, The story of ChronoShare, or how NDN brought distributed file sharing back

Building Automation and Management [5,6]

- qImprove application development process, management, interoperability and security.
- Practical work so far: NDN interfaces to BacNET and Modbus sensing, authenticated lighting control.
- Partner: UCLA Facilities Management.





[5] Burke et al. Securing instrumented environments over Content-Centric Networking: the case of lighting control. In *IEEE INFOCOM NOMEN Workshop*, Apr. 2013.

[6] Shang et al., “Securing Building Management Systems Using Named Data Networking,” *IEEE Network*, May/June 2014.

NDN Platform Release

- **http://named-data.net/codebase/platform/**
  - a new release every few months

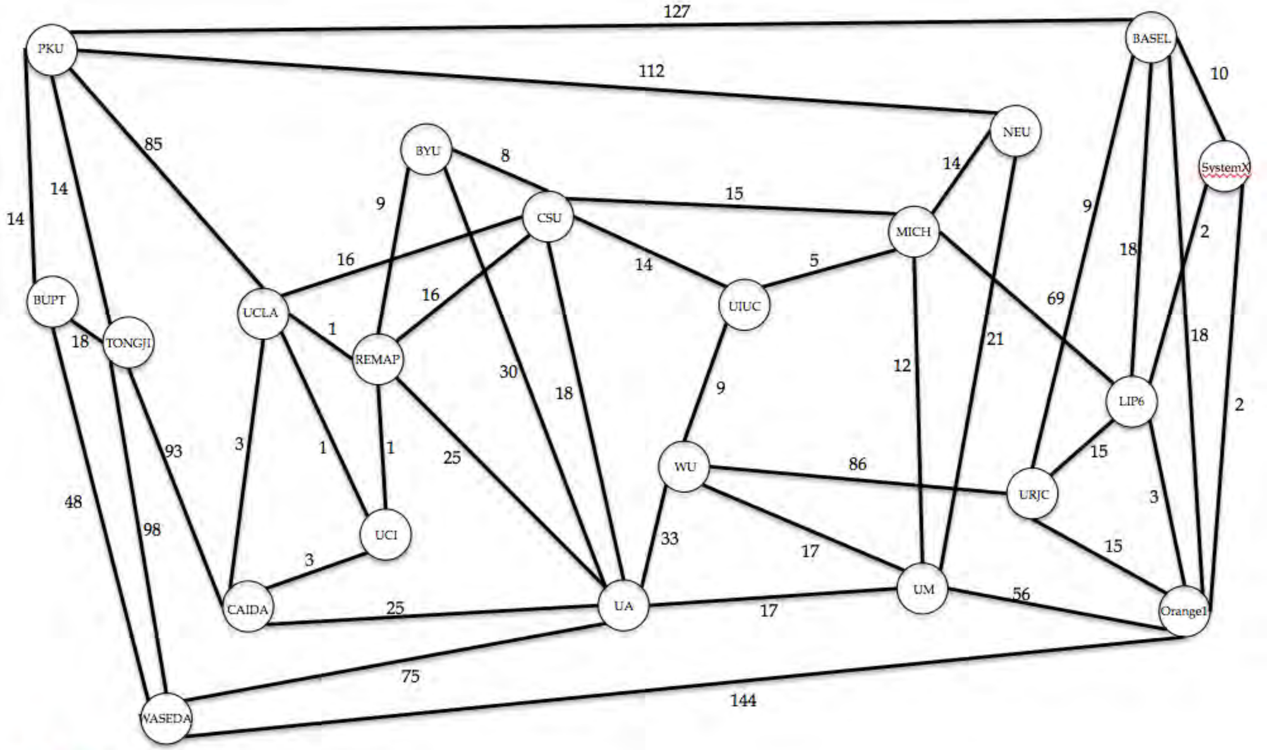
- Latest version: Version 0.3 released on Aug. 25, 2014.
  - The NDN Forwarding Daemon (NFD), version 0.2.0
  - The ndn-cxx library, version 0.2.0
  - The NDN Common Client libraries suite (NDN-CCL), version 0.3
    - C++ – NDN-CPP, Python – PyNDN2, JavaScript – NDN-JS, Java – jNDN
  - The Named Data Link State Routing Protocol (NLSR), version 0.1.0
  - The next generation of NDN repository (repo-ng), version 0.1.0
  - A ping application For NDN (ndn-tlv-ping), version 0.2.0
  - A traffic generator For NDN (ndn-traffic-generator), version 0.2.0
  - A packet capture and analysis tool for NDN (ndndump), version 0.5

Scalable Forwarding Engine Design

- Requirements
  - data structures to store millions to billions of names
  - fast table lookup of variable-length names
  - fast packet processing
- NDN project's progress
  - multi-million entry FIBs stored in less than 10MB [10]
  - FIB lookup speeds on the order of microseconds [10]
  - PIT: 37 to 245 MiB memory for 100 Gbps throughput (small enough to fit in fast memory chips) [11]

[10] H. Yuan, T. Song, and P. Crowley. Scalable NDN forwarding: Concepts, issues and principles. In *ICCCN*, 2012.
[11] H. Yuan and P. Crowley. Scalable pending Interest table design: From principles to practice. *IEEE INFOCOM*, 2014.

NDN Testbed



- **21 Nodes**
- **46 Links (with NLSR routing costs)**

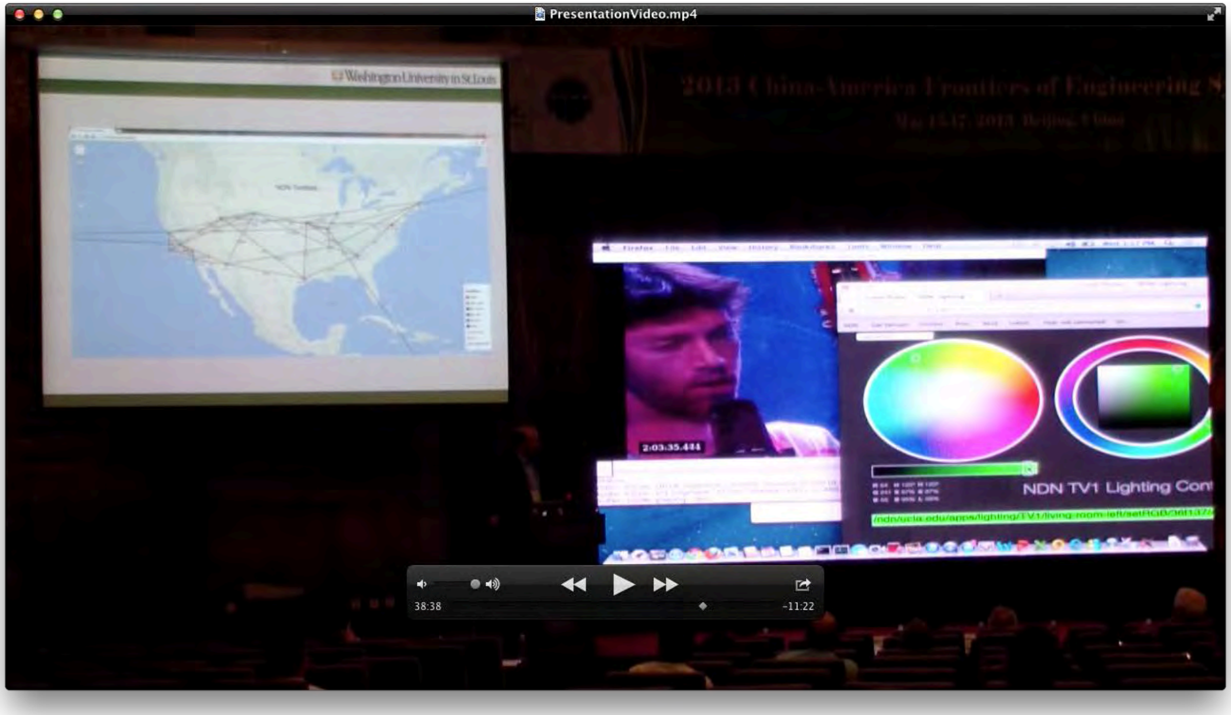
More info at http://named-data.net/ndn-testbed/
Contact us if interested in joining the testbed.

Annual Demonstrations

Demo Feature	2012 Demo	2013 Demo
Large-scale, **wide-area operation**	All 4 US time zones, ~300 machines	5 continents, ~1000 machines
Mix of content distribution and **interactive apps**	4 distinct services	Multiple services
Visualization of both app-level and net-level activity	NDN map	NDN map
Demonstrate both steady-state and **react-to-change** modes	Drop links during app sessions	Forwarding strategy
Something IP+HTTP cannot do	Scalable video streaming*, multi-path routing	Scalable video streaming*, multi-path routing
Integrated PKI, better security		Show key auth
NDN-based device monitoring		Stage lighting ctrl

Live bluegrass band performance, NDN-based control of stage lights

- Delivery of live audio and video from performance studio at UCLA
  - Jeff Burke’s Center for Research in Engineering, Media and Performance (REMAP)
- Lighting control application is NDN-based
- Server at studio homed off REMAP gateway
- Laptop on-site homed off Tokyo gateway



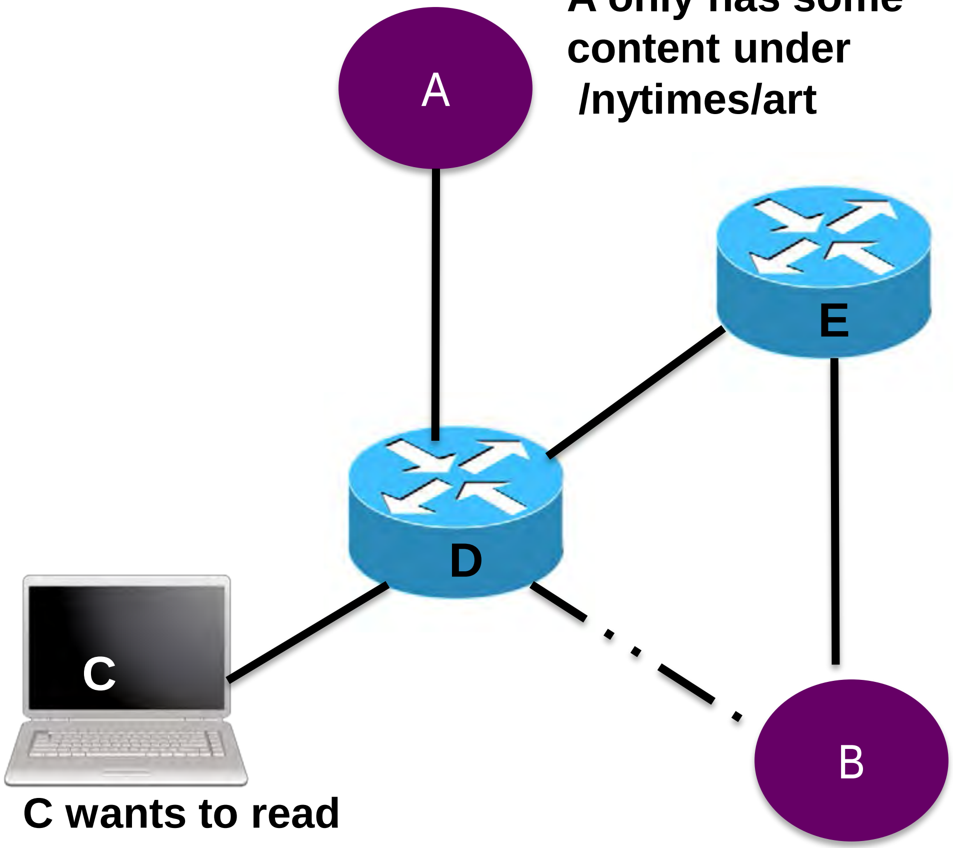
NDN Next Phase

- NSF FIA-NP (2014-2016)
- Applications
  - Open M-Health
  - E-BAMS
  - Mobile Media Application Cluster
- Forwarding and routing: Interdomain routing, forwarding strategy
- Security: privacy, trust management
- NDN Consortium (Sept. 2014): industry and academic members

A NAME-BASED ROUTING PROTOCOL

Routing in NDN

- Requirement: Routing based on "name"
  - Guides each "interest" packet to all potential providers (all paths)
  - Some providers may not have all content in a "name"
- Non-requirement: Fast routing convergence
  - Stateful forwarding plane can adapt to changes



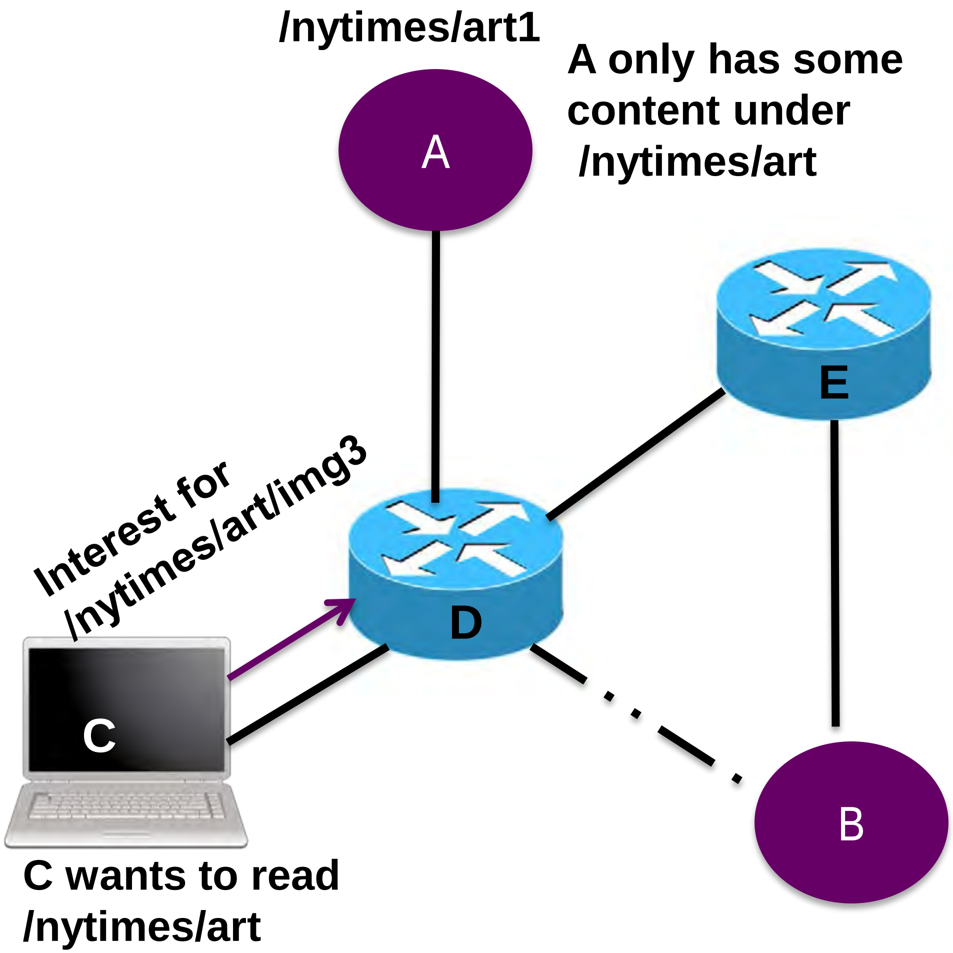
/nytimes/art1

A only has some content under /nytimes/art

C wants to read /nytimes/art

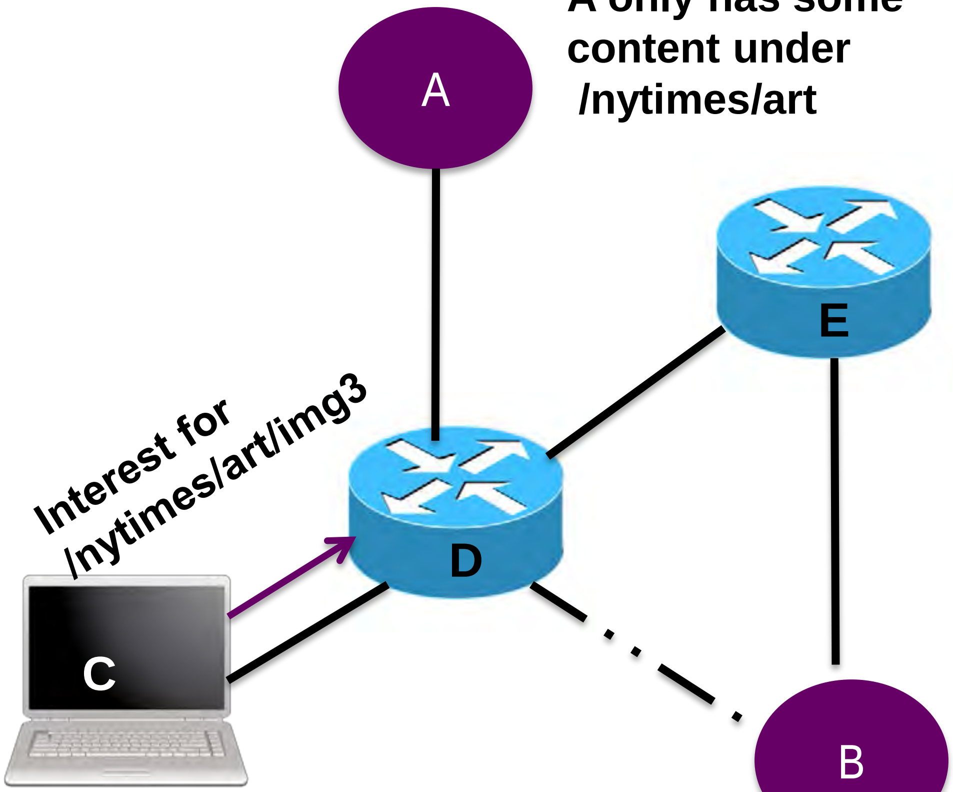
Routing in NDN

- Requirement: Routing based on "name"
  - Guides each "interest" packet to all potential providers (all paths)
  - Some providers may not have all content in a "name"
- Non-requirement: Fast routing convergence
  - Stateful forwarding plane can adapt to changes


/nytimes/art1
A only has some content under /nytimes/art
A
E
Interest for /nytimes/art/img3
D
C
B
C wants to read /nytimes/art

Routing in NDN

- Requirement: Routing based on “name”
  - Guides each “interest” packet to all potential providers (all paths)
  - Some providers may not have all content in a “name”
- Non-requirement: Fast routing convergence
  - Stateful forwarding plane can adapt to changes



/nytimes/art1

A only has some content under /nytimes/art

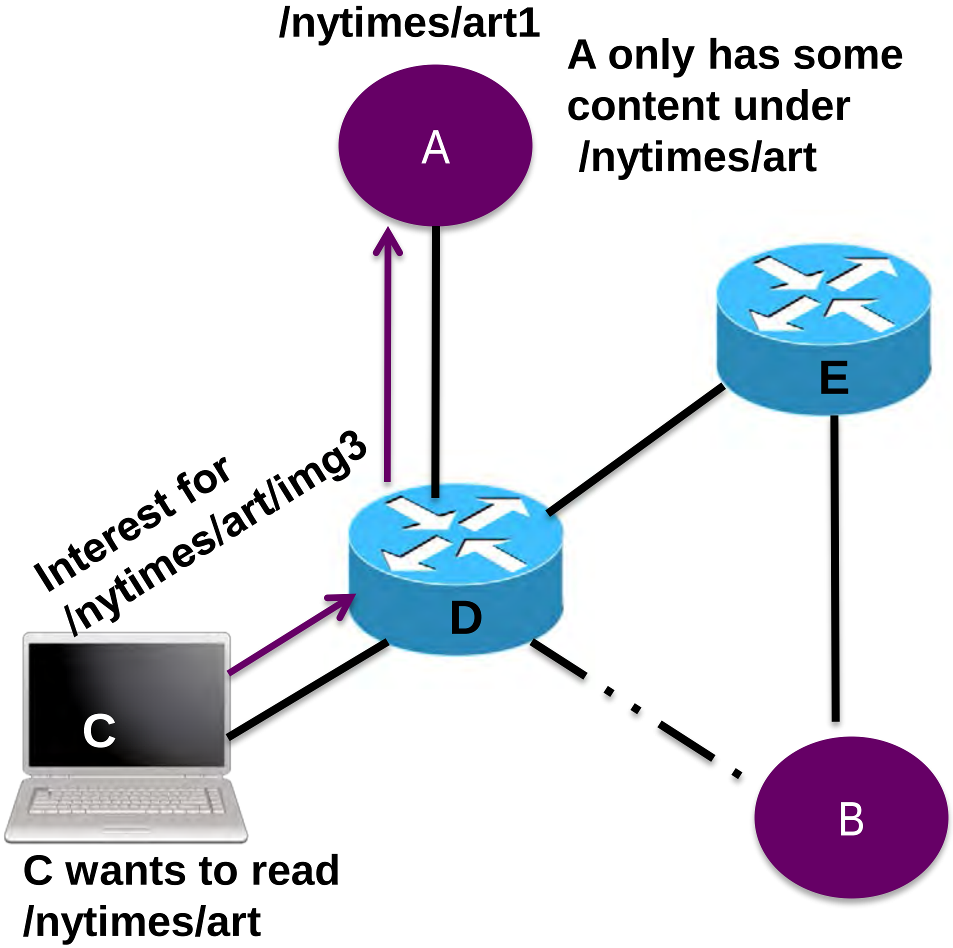
Interest for /nytimes/art/img3

C wants to read /nytimes/art

D’s FIB:	/nytimes/art	A, B, E

Routing in NDN

- Requirement: Routing based on “name”
  - Guides each “interest” packet to all potential providers (all paths)
  - Some providers may not have all content in a “name”
- Non-requirement: Fast routing convergence
  - Stateful forwarding plane can adapt to changes



A only has some content under /nytimes/art

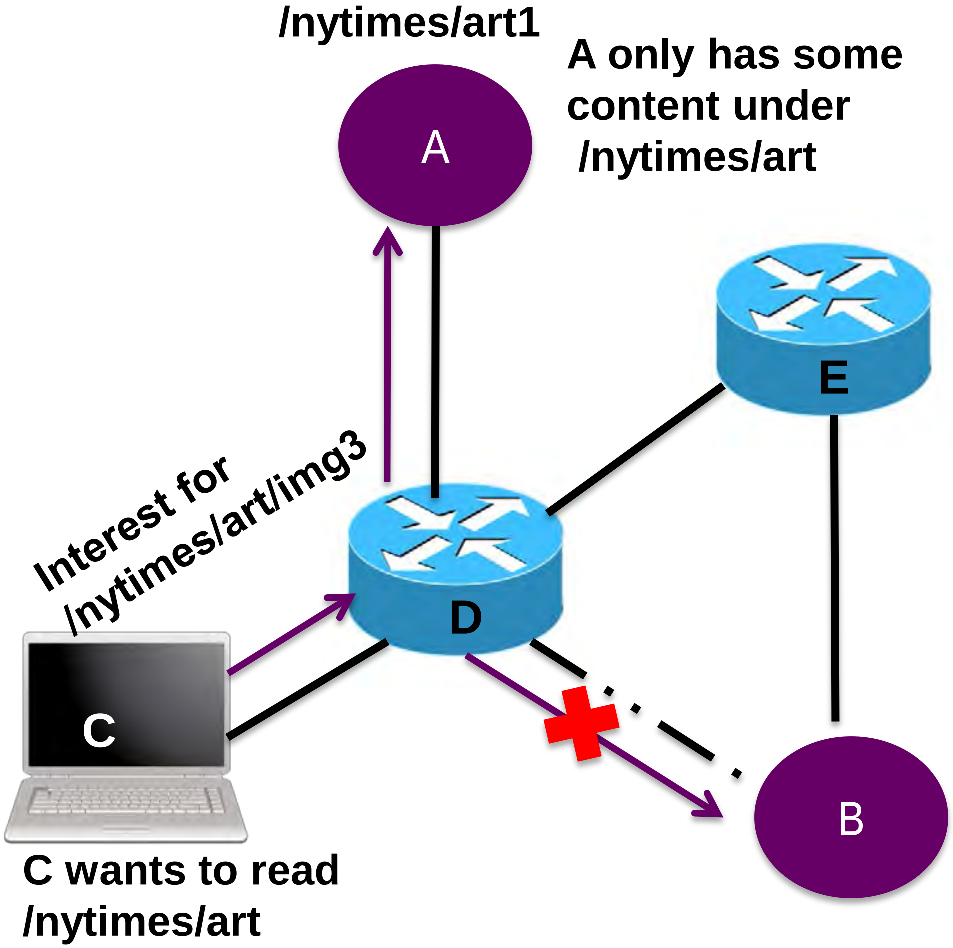
C wants to read /nytimes/art

D’s FIB:

/nytimes/art	A, B, E

Routing in NDN

- Requirement: Routing based on "name"
  - Guides each "interest" packet to all potential providers (all paths)
  - Some providers may not have all content in a "name"
- Non-requirement: Fast routing convergence
  - Stateful forwarding plane can adapt to changes



A only has some content under /nytimes/art

Interest for /nytimes/art/img3

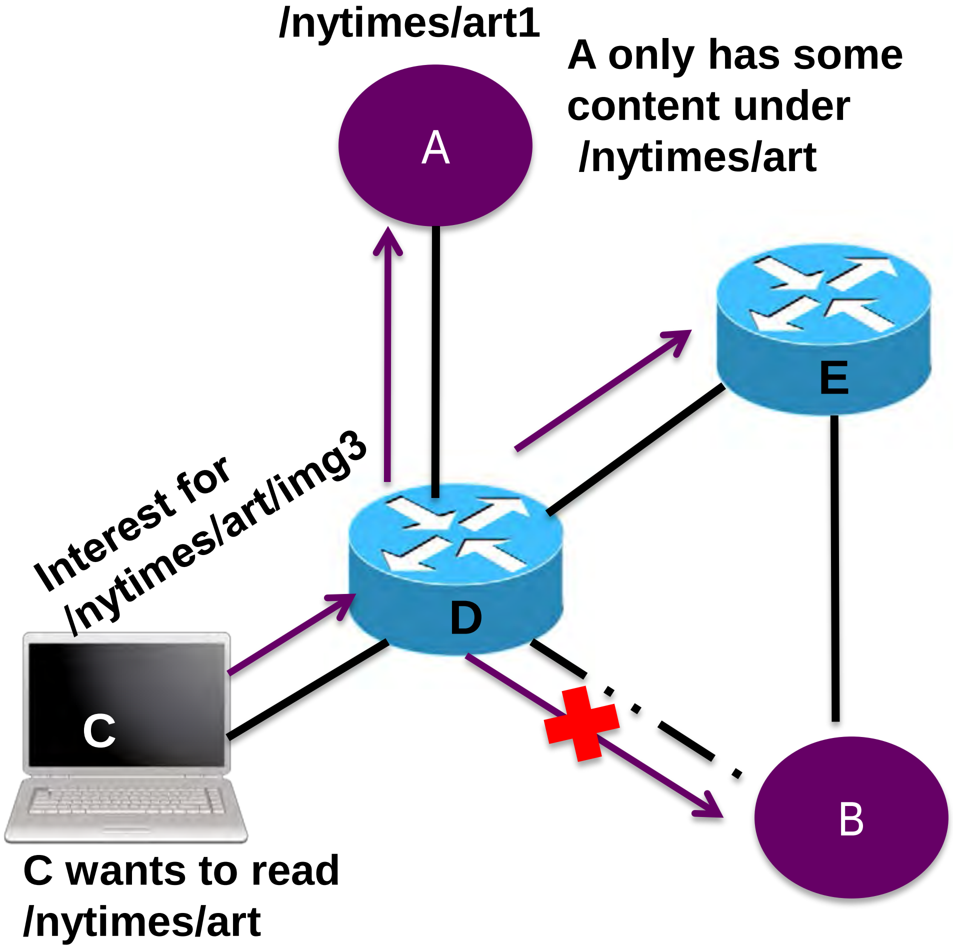
C wants to read /nytimes/art

D's FIB:

/nytimes/art	A, B, E

Routing in NDN

- Requirement: Routing based on “name”
  - Guides each “interest” packet to all potential providers (all paths)
  - Some providers may not have all content in a “name”
- Non-requirement: Fast routing convergence
  - Stateful forwarding plane can adapt to changes



A only has some content under /nytimes/art

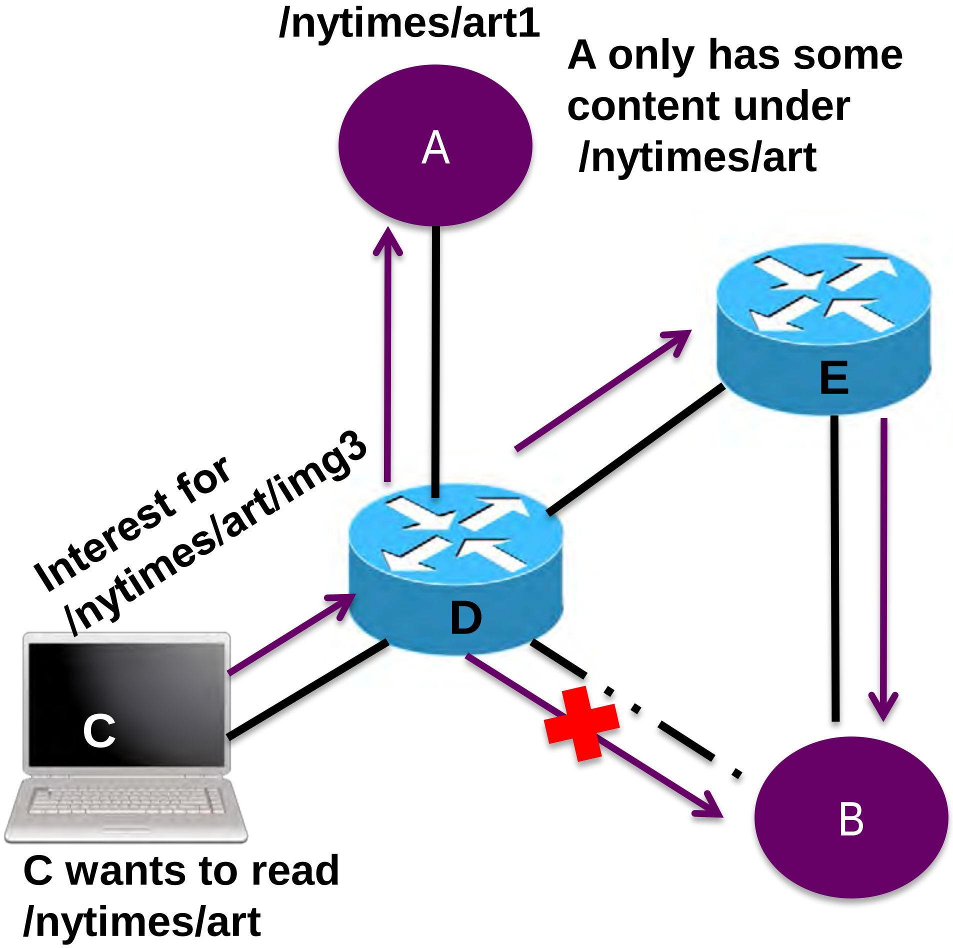
C wants to read /nytimes/art

D’s FIB:

/nytimes/art	A, B, E

Routing in NDN

- Requirement: Routing based on “name”
  - Guides each “interest” packet to all potential providers (all paths)
  - Some providers may not have all content in a “name”
- Non-requirement: Fast routing convergence
  - Stateful forwarding plane can adapt to changes



A only has some content under /nytimes/art

C wants to read /nytimes/art

D’s FIB:

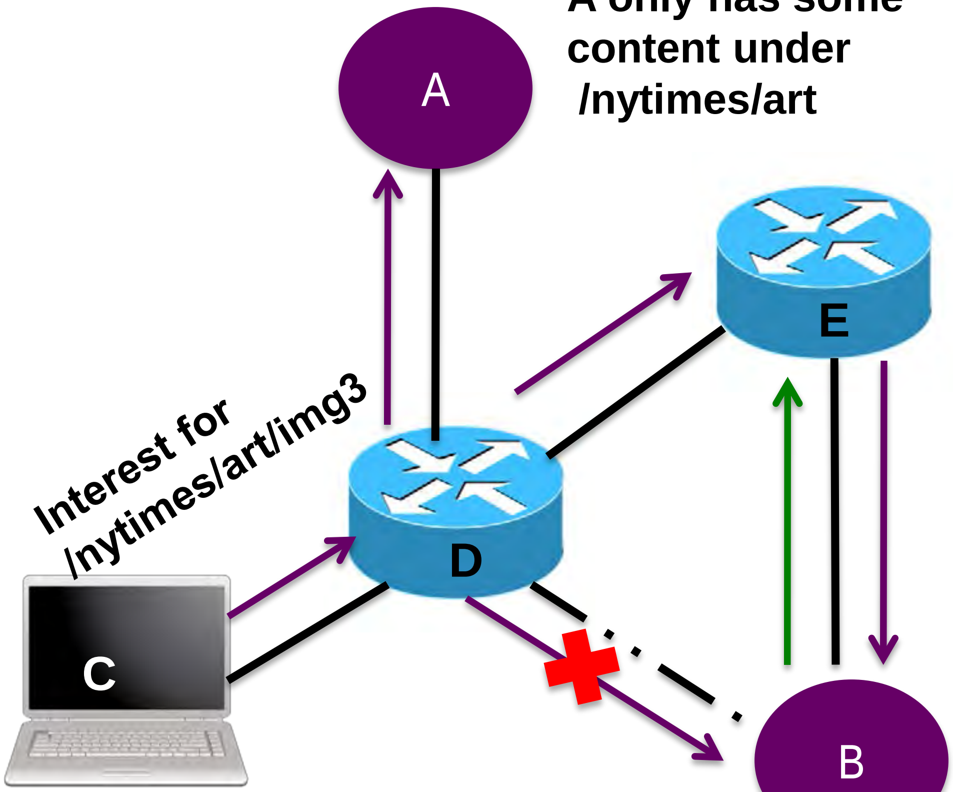
/nytimes/art	A, B, E

Routing in NDN

- Requirement: Routing based on "name"
  - Guides each "interest" packet to all potential providers (all paths)
  - Some providers may not have all content in a "name"
- Non-requirement: Fast routing convergence
  - Stateful forwarding plane can adapt to changes

/nytimes/art1

A only has some content under /nytimes/art



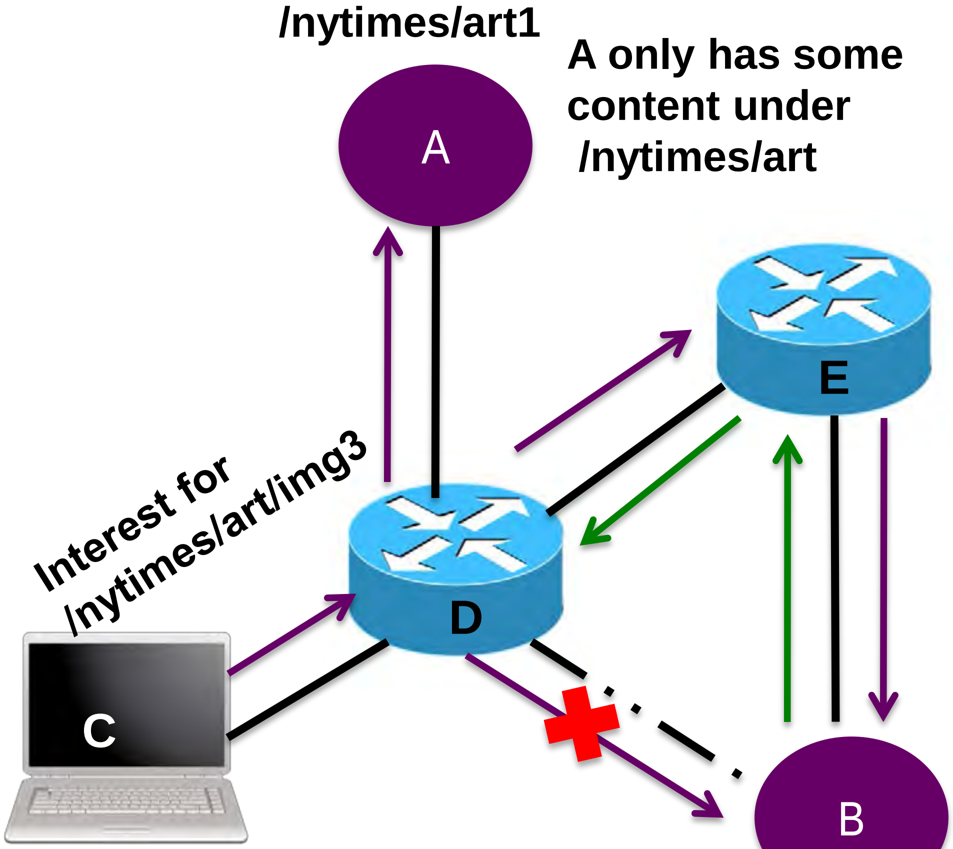
C wants to read /nytimes/art

D's FIB:

/nytimes/art	A, B, E

Routing in NDN

- Requirement: Routing based on "name"
  - Guides each "interest" packet to all potential providers (all paths)
  - Some providers may not have all content in a "name"
- Non-requirement: Fast routing convergence
  - Stateful forwarding plane can adapt to changes



A only has some content under /nytimes/art

Interest for /nytimes/art/img3

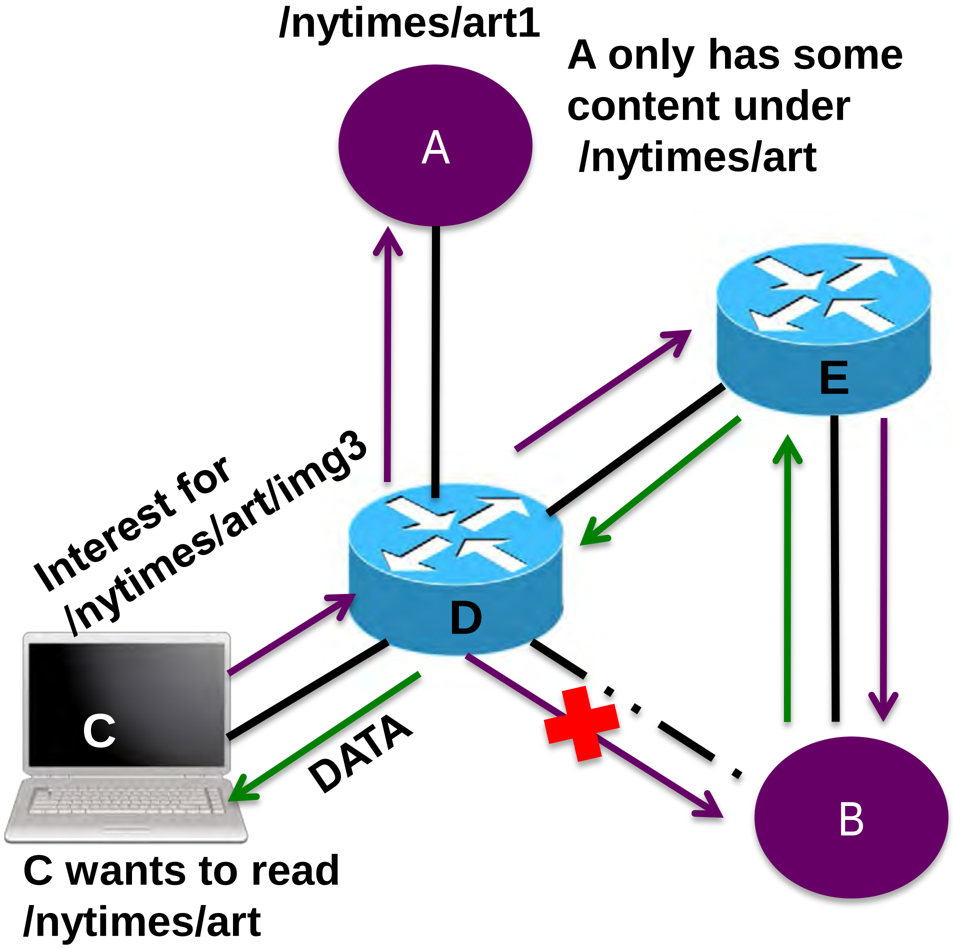
C wants to read /nytimes/art

D's FIB:

/nytimes/art	A, B, E

Routing in NDN

- Requirement: Routing based on “name”
  - Guides each “interest” packet to all potential providers (all paths)
  - Some providers may not have all content in a “name”
- Non-requirement: Fast routing convergence
  - Stateful forwarding plane can adapt to changes


/nytimes/art1
A only has some content under /nytimes/art
A
E
Interest for /nytimes/art/img3
D
C
DATA
B
C wants to read /nytimes/art

D’s FIB:

/nytimes/art	A, B, E

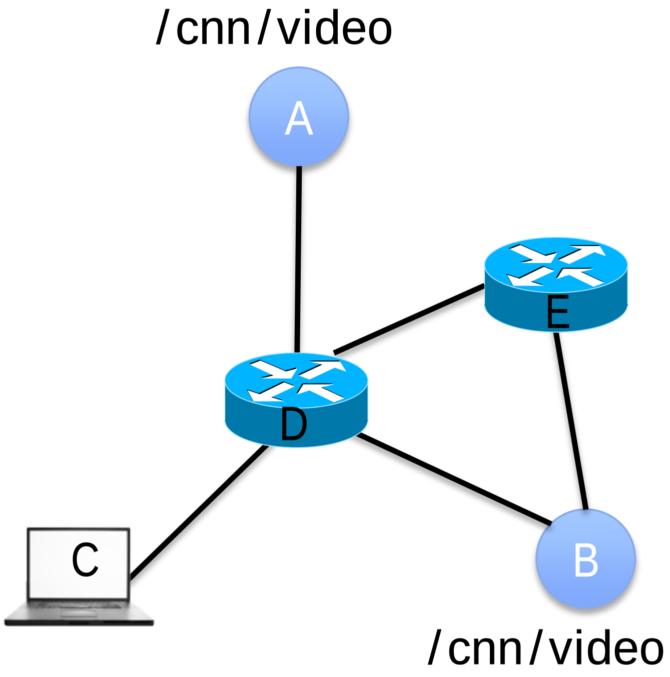
Routing Mechanism in NDN

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
- Propagate routing updates using Interest/Data packets



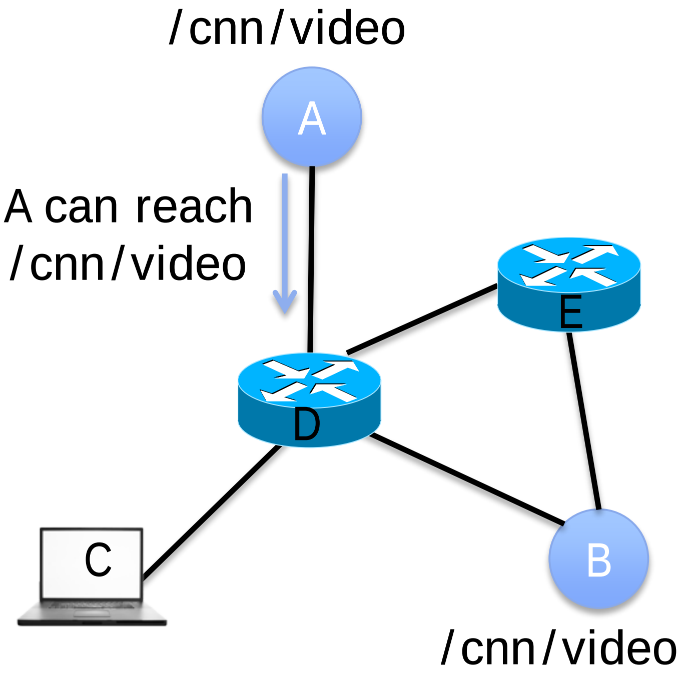
Routing Mechanism in NDN

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
- Propagate routing updates using Interest/Data packets



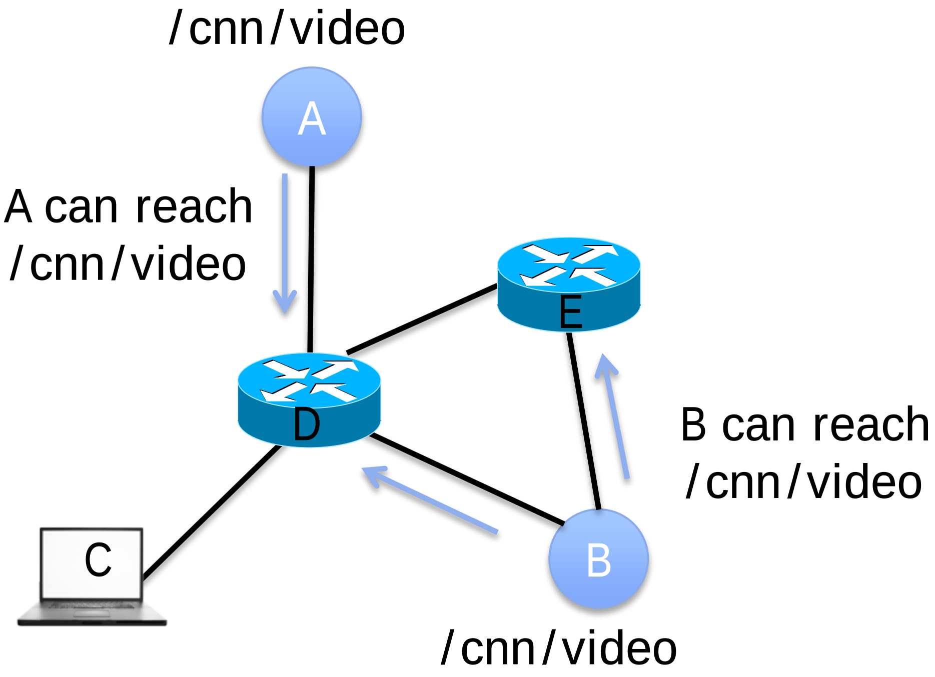
Routing Mechanism in NDN

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
- Propagate routing updates using Interest/Data packets



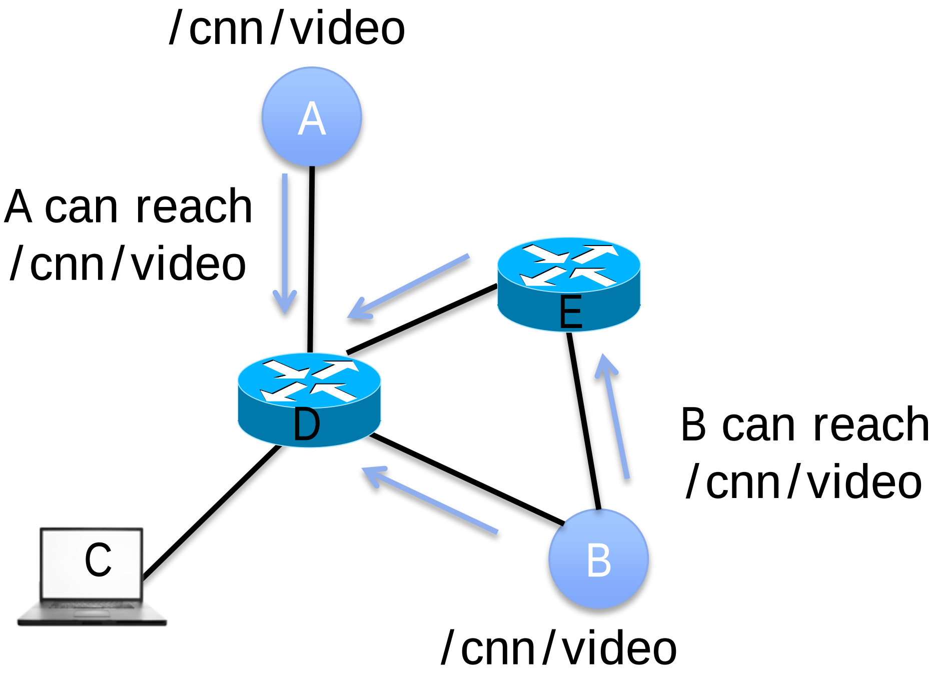
Routing Mechanism in NDN

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
- Propagate routing updates using Interest/Data packets



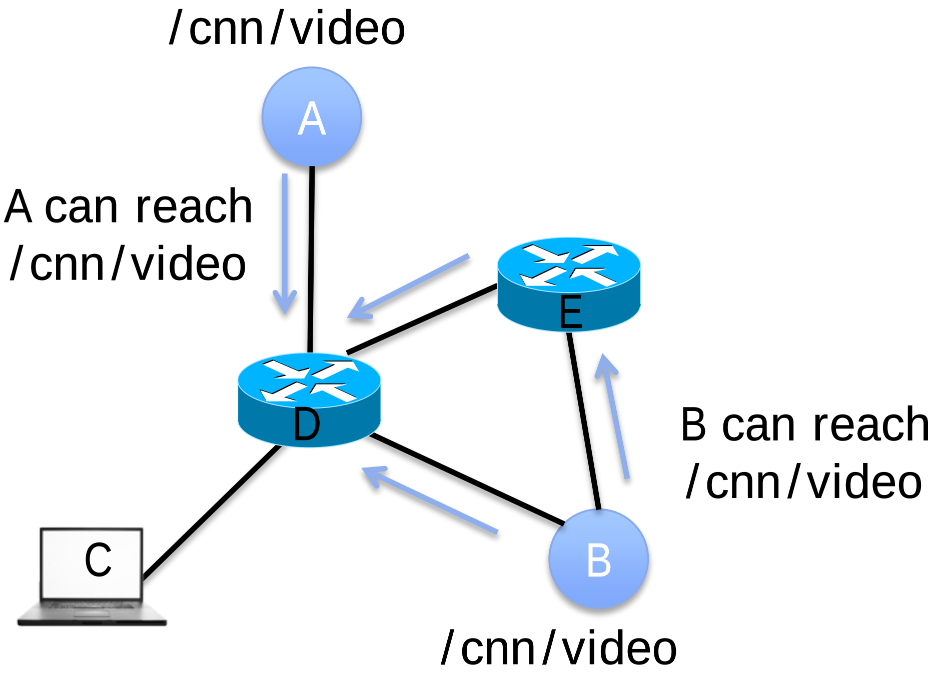
Routing Mechanism in NDN

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
- Propagate routing updates using Interest/Data packets



D’s FIB:

/cnn/video	A, B, E

Motivation for a Native Name-based Routing Protocol

- Need a routing protocol for NDN networks
  - Populate RIB/FIB so routers can forward interests
  - Not necessarily point to the nearest cache.
- Previous work: OSPFN – simple extension of IP OSPF
  - Advertises *"name"* by OSPF Opaque LSA
  - Difficult to manage IP addresses and GRE tunnels
  - Only single path in most cases
  - Security relies on password or secret shared among all routers on a subnet.
- Build a routing protocol purely on top of NDN

Named-data Link State Routing (NLSR)[1]

- Use 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
- Security
  - Routing data is signed by originating router and verified by receivers based on a trust model.
  - a trust model for intra-domain routing

[1] AKM M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang. *NLSR: Named-data link state routing protocol*. In ACM SIGCOMM ICN Workshop, 2013.

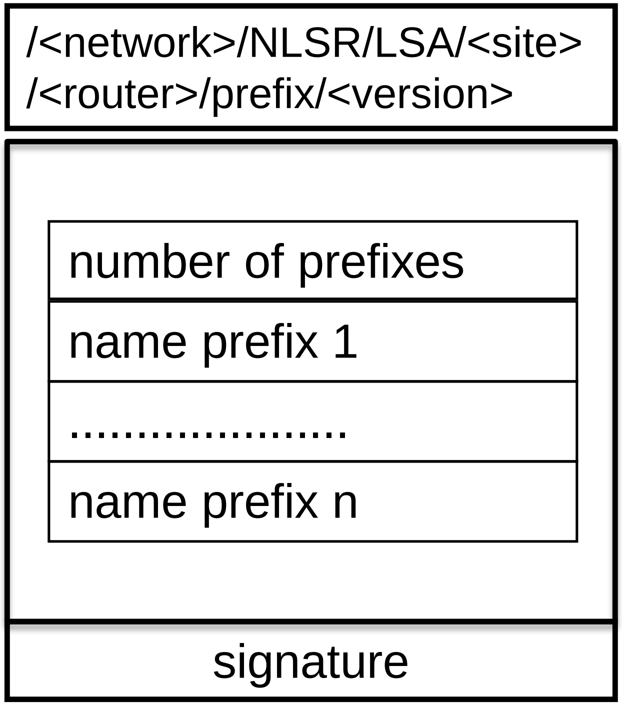
Naming in NLSR

- Follow the hierarchy within a network
  - Easy to identify the relationship among entities
  - Easy to associate keys with key owners
- Router
  - /<network>/<site>/<router>: e.g., /ndn/memphis/rtr1
- Updates
  - /<network>/NLSR/LSA/<site>/<router>/<process>/<type>/<version>
- Keys
  - NLSR key: /<network>/<site>/<router>/<process>/key
  - Router key, operator key, …

Link State Advertisement (LSA)

Link State Advertisement (LSA)

Prefix LSA



/<network>/NLSR/LSA/<site>/<router>/prefix/<version>

number of prefixes

name prefix 1

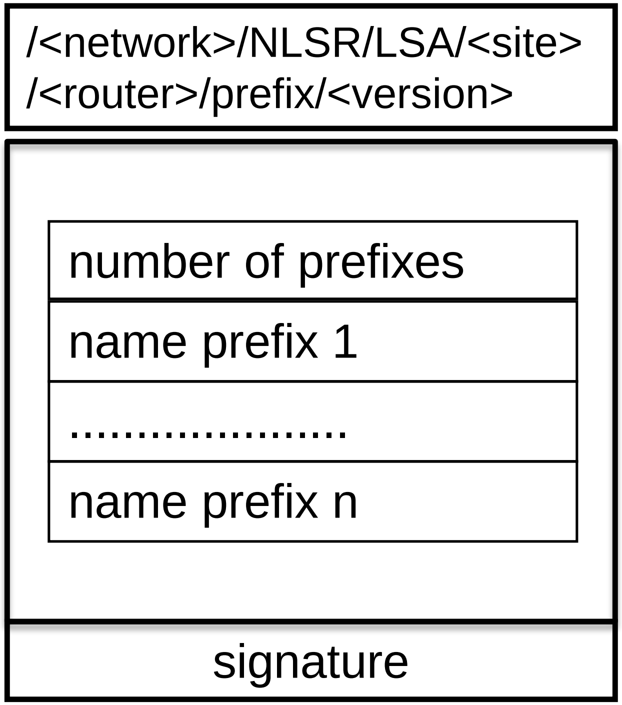
....

name prefix n

signature

Link State Advertisement (LSA)

Prefix LSA



/<network>/NLSR/LSA/<site>
/<router>/prefix/<version>

number of prefixes

name prefix 1

....

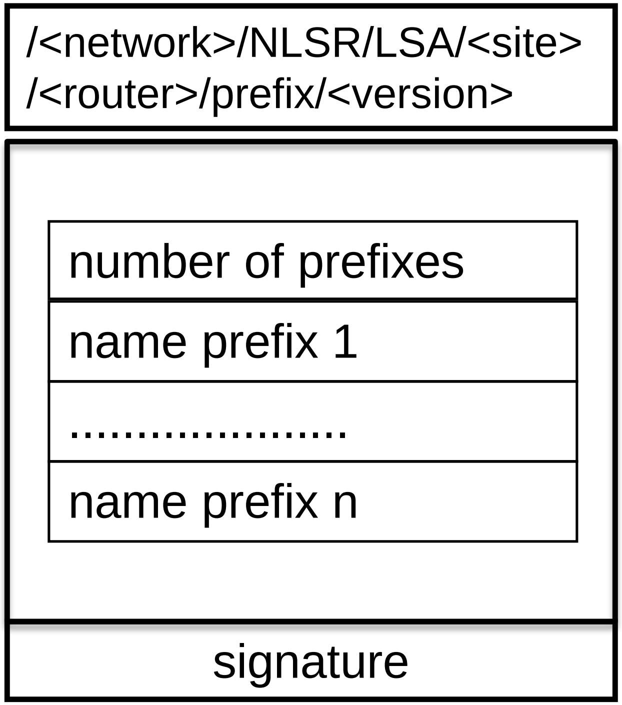
name prefix n

signature

Advertises reachable name prefixes by router

Link State Advertisement (LSA)

Prefix LSA



/<network>/NLSR/LSA/<site>
/<router>/prefix/<version>

number of prefixes

name prefix 1

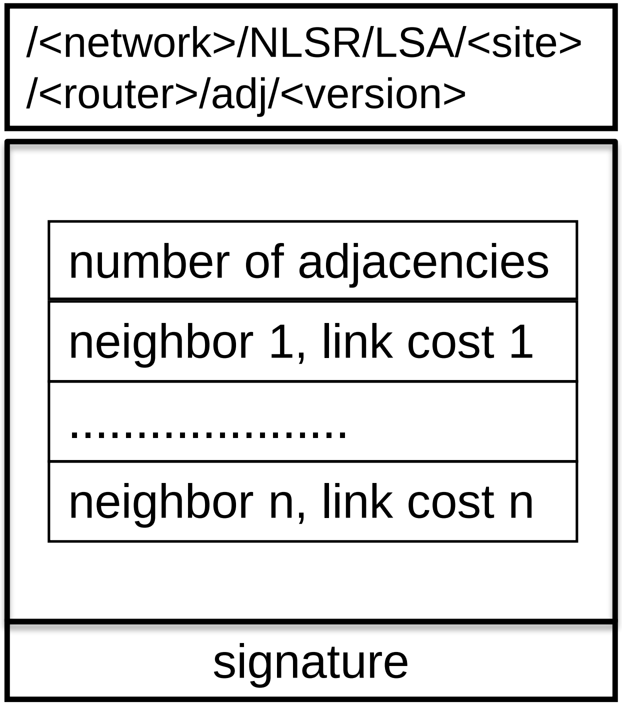
....

name prefix n

signature

Advertises reachable name prefixes by router

Adjacency LSA



/<network>/NLSR/LSA/<site>
/<router>/adj/<version>

number of adjacencies

neighbor 1, link cost 1

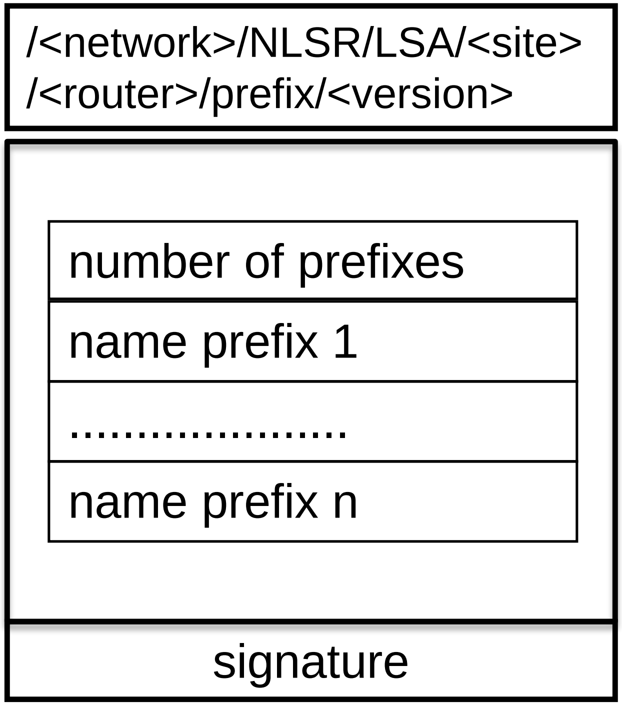
....

neighbor n, link cost n

signature

Link State Advertisement (LSA)

Prefix LSA



/<network>/NLSR/LSA/<site>/<router>/prefix/<version>

number of prefixes

name prefix 1

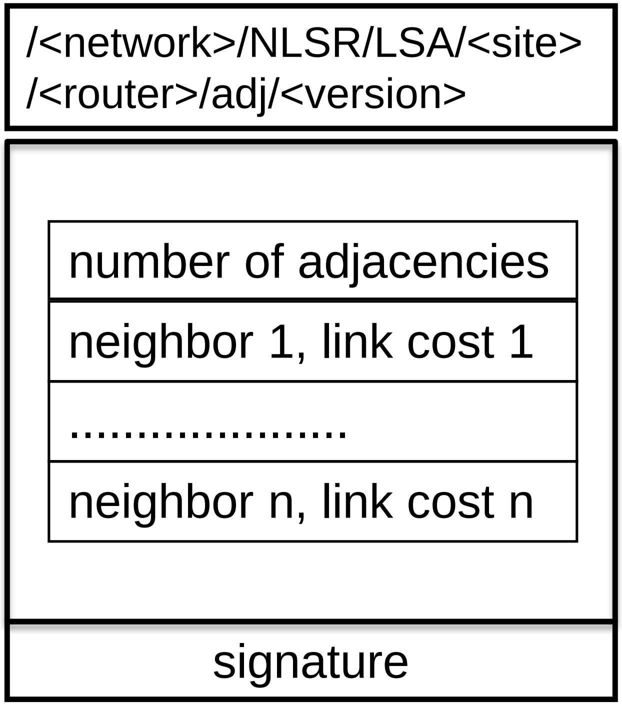
....

name prefix n

signature

Advertises reachable name prefixes by router

Adjacency LSA



/<network>/NLSR/LSA/<site>/<router>/adj/<version>

number of adjacencies

neighbor 1, link cost 1

....

neighbor n, link cost n

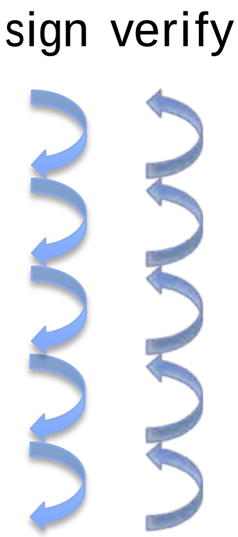
signature

Advertises adjacency state: topology information

Routing Security and Trust Model

- Every NLSR Data packet is signed by the NLSR process key.
- “Key locator” includes information about the key.
- Receiver retrieves the NLSR key and verifies the Data signature.
- Then receiver verifies the NLSR key, router key, operator key, and site key (the root key is preconfigured).

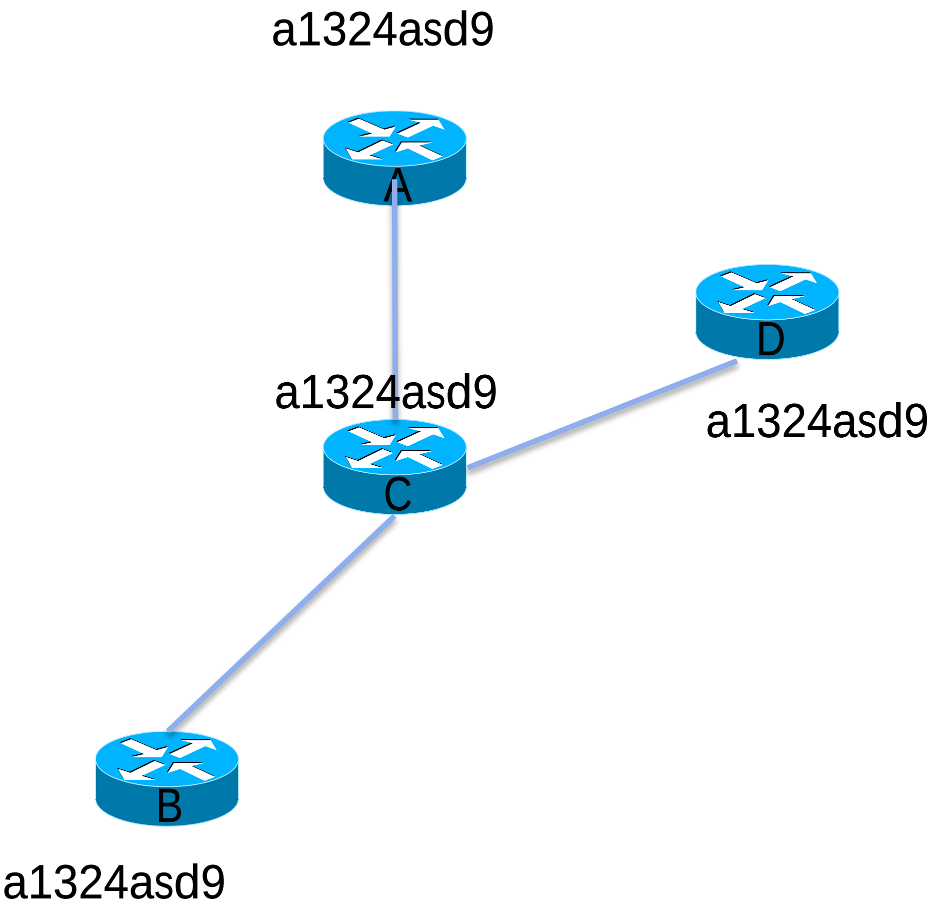
Entity	Name
Root key	/<network>/key
Site key	/<network>/<site>/key
Operator key	/<network>/<site>/<operator>/key
Router key	/<network>/<site>/<router>/key
NLSR key	/<network>/<site>/<router>/NLSR/key
Data	/<network>/NLSR/LSA/<site>/<router>/<type>/<ver>



From Flooding to Synchronization

NLSR uses ChronoSync [2] to synchronize LSDB.

- 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 LSA.

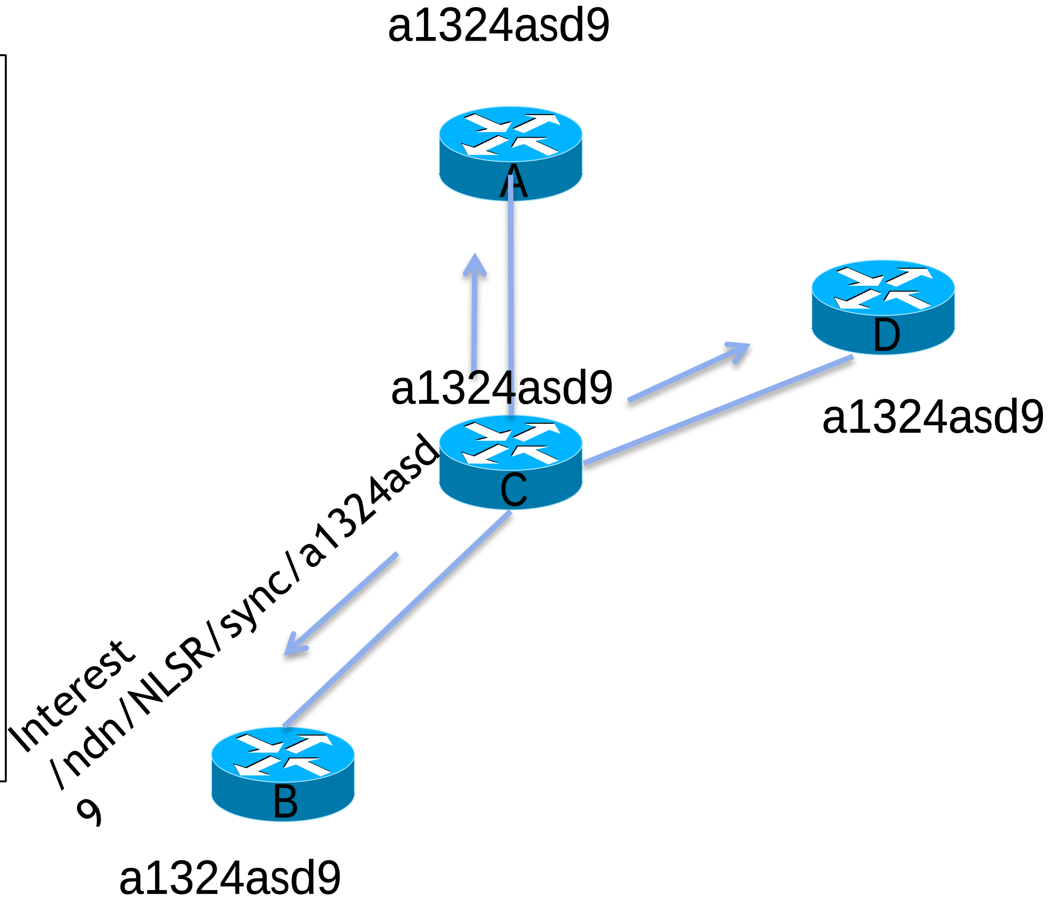


[2] Z. Zhu, A. Afanasyev, and L. Zhang. Let's ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.

From Flooding to Synchronization

NLSR uses ChronoSync [2] to synchronize LSDB.

- 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 LSA.

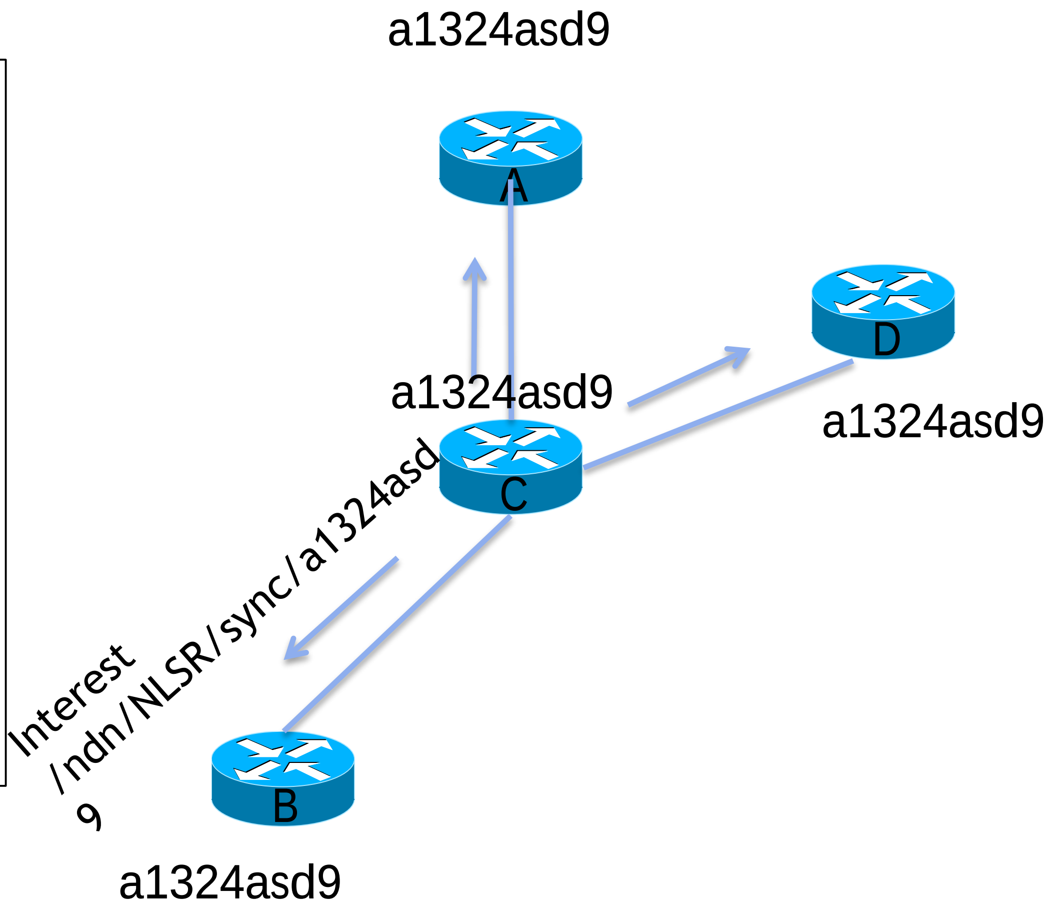


[2] Z. Zhu, A. Afanasyev, and L. Zhang. Let's ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.

From Flooding to Synchronization

NLSR uses ChronoSync [2] to synchronize LSDB.

- 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 LSA.

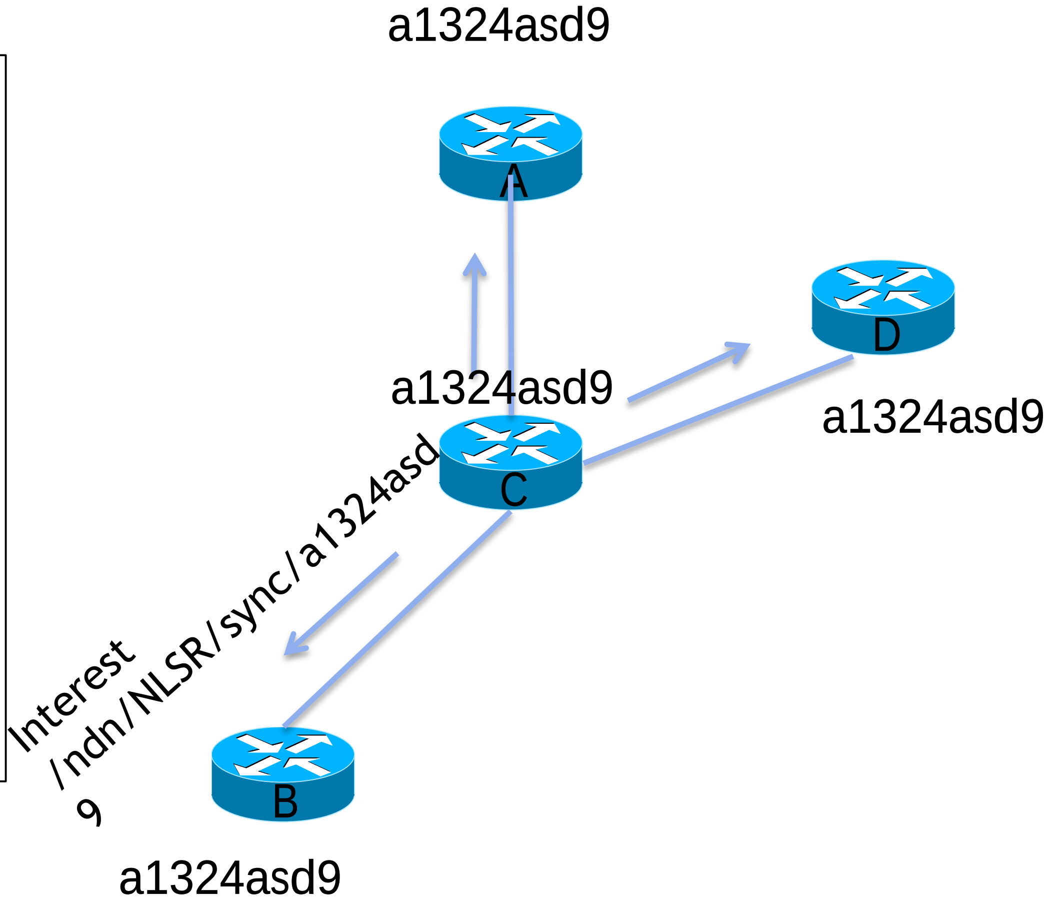


[2] Z. Zhu, A. Afanasyev, and L. Zhang. Let's ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.

From Flooding to Synchronization

NLSR uses ChronoSync [2] to synchronize LSDB.

- 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 LSA.

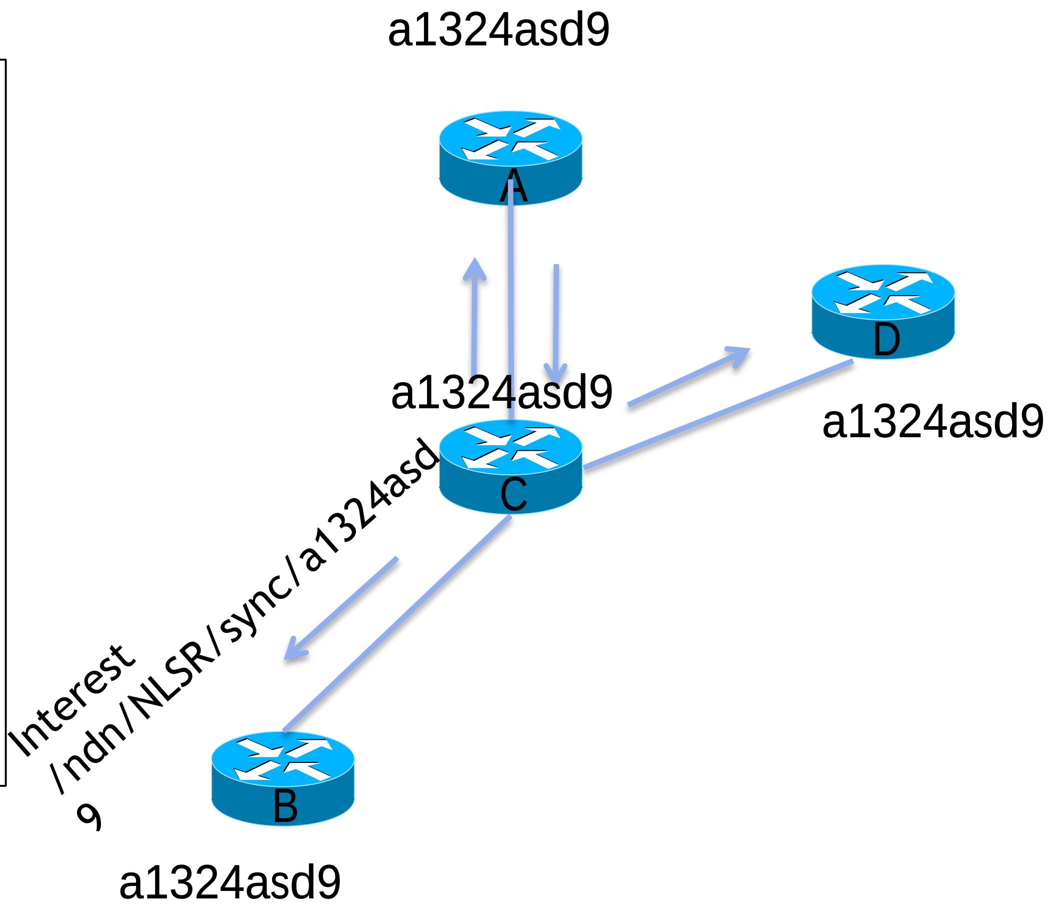


[2] Z. Zhu, A. Afanasyev, and L. Zhang. Let's ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.

From Flooding to Synchronization

NLSR uses ChronoSync [2] to synchronize LSDB.

- 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 LSA.

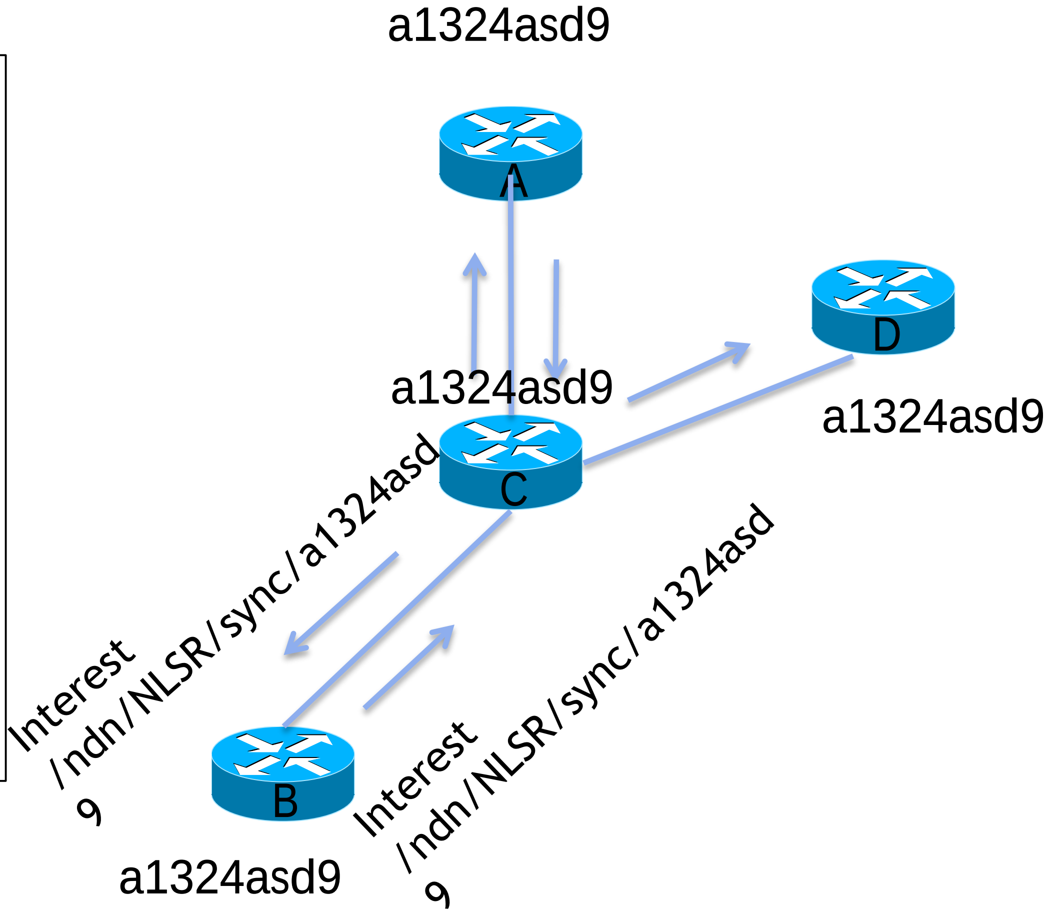


[2] Z. Zhu, A. Afanasyev, and L. Zhang. Let's ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.

From Flooding to Synchronization

NLSR uses ChronoSync [2] to synchronize LSDB.

- 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 LSA.

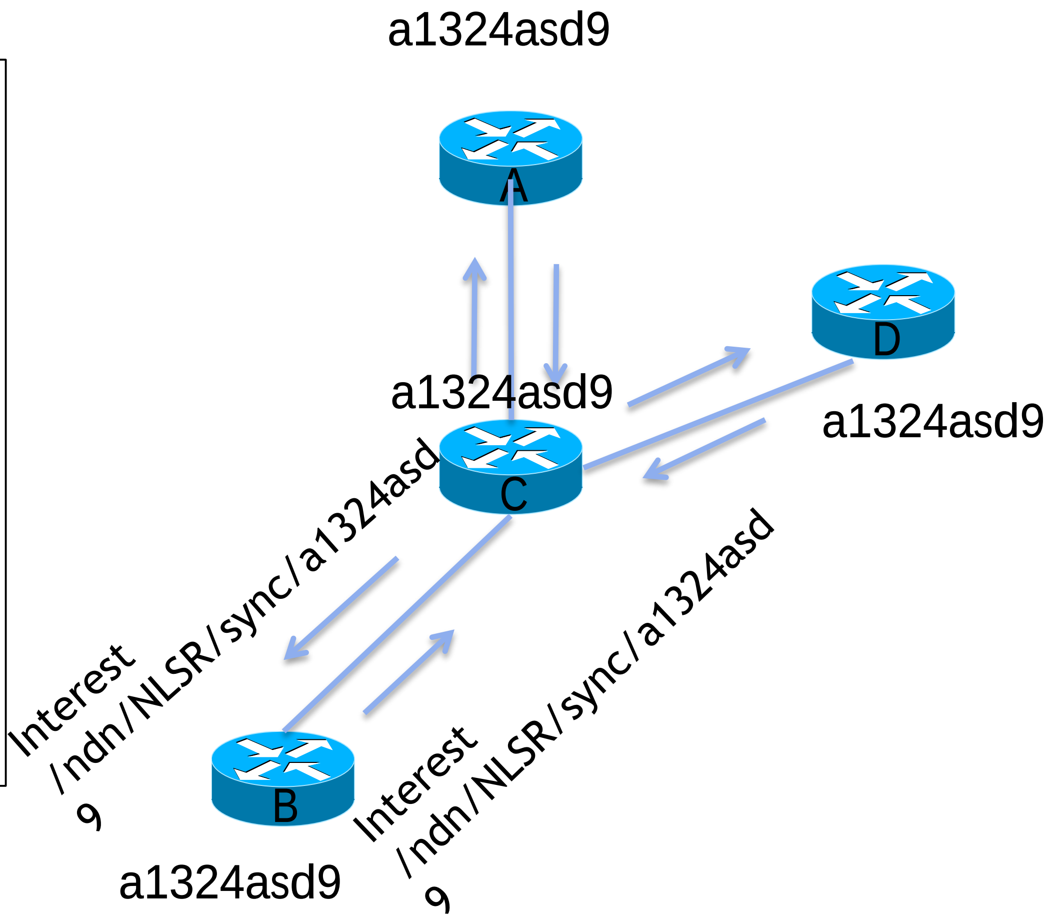

a1324asd9
A
D
a1324asd9
a1324asd9
C
Interest /ndn/NLSR/sync/a1324asd9
B
Interest /ndn/NLSR/sync/a1324asd9
a1324asd9

[2] Z. Zhu, A. Afanasyev, and L. Zhang. Let's ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.

From Flooding to Synchronization

NLSR uses ChronoSync [2] to synchronize LSDB.

- 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 LSA.


a1324asd9
A
D
a1324asd9
C
a1324asd9
Interest /ndn/NLSR/sync/a1324asd9
B
a1324asd9
Interest /ndn/NLSR/sync/a1324asd9

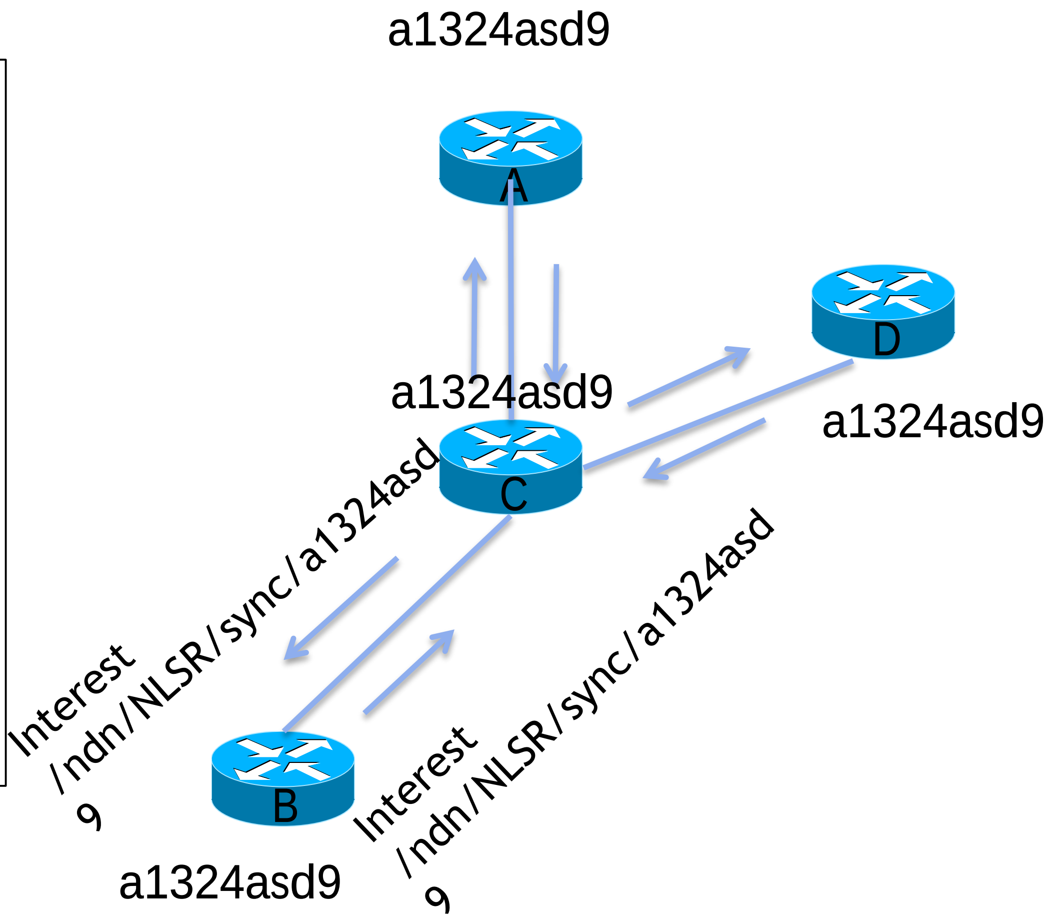
[2] Z. Zhu, A. Afanasyev, and L. Zhang. Let's ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.

From Flooding to Synchronization

A has a new LSA

NLSR uses ChronoSync [2] to synchronize LSDB.

- 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 LSA.

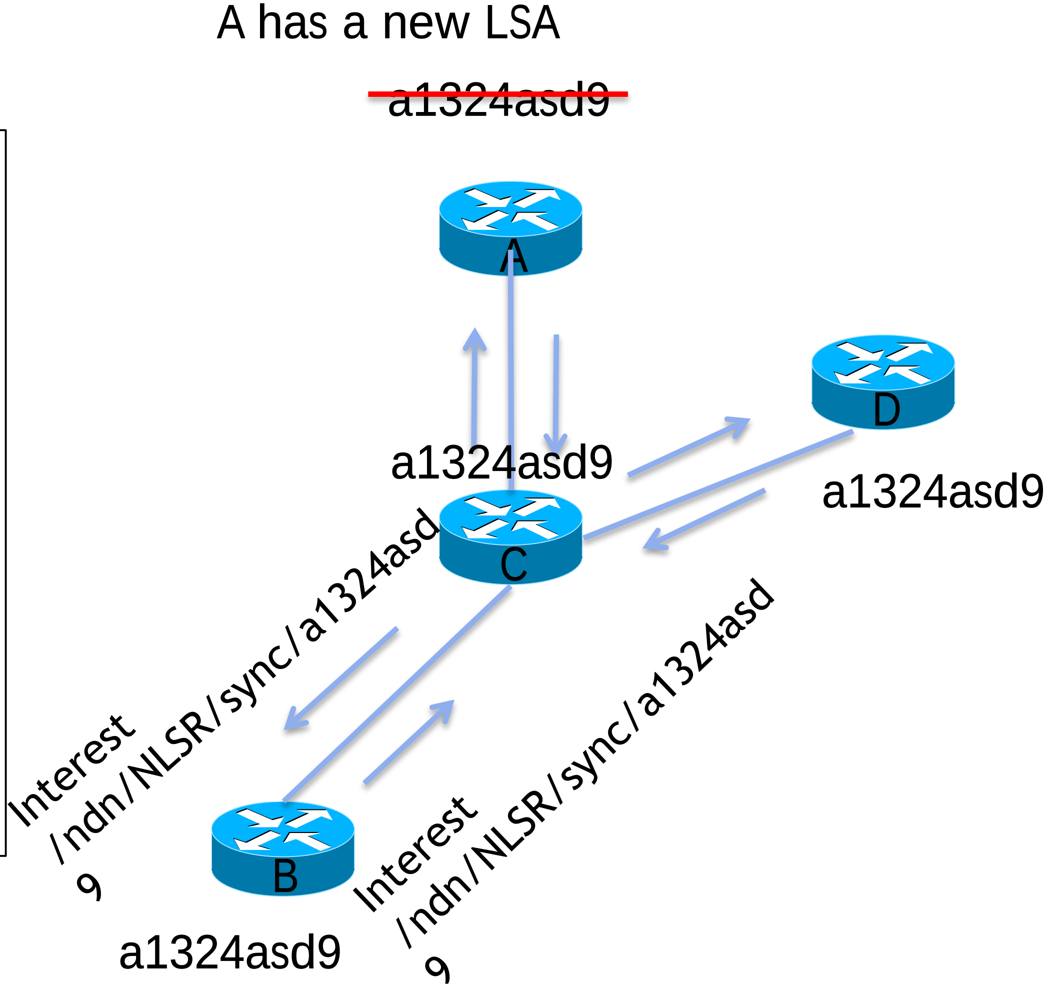

a1324asd9
A
D
a1324asd9
a1324asd9
C
Interest /ndn/NLSR/sync/a1324asd9
Interest /ndn/NLSR/sync/a1324asd9
B
a1324asd9

[2] Z. Zhu, A. Afanasyev, and L. Zhang. Let’s ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.

From Flooding to Synchronization

NLSR uses ChronoSync [2] to synchronize LSDB.

- 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 LSA.


A has a new LSA
~~a1324asd9~~
A
D
a1324asd9
a1324asd9
C
Interest /ndn/NLSR/sync/a1324asd9
B
Interest /ndn/NLSR/sync/a1324asd9
a1324asd9

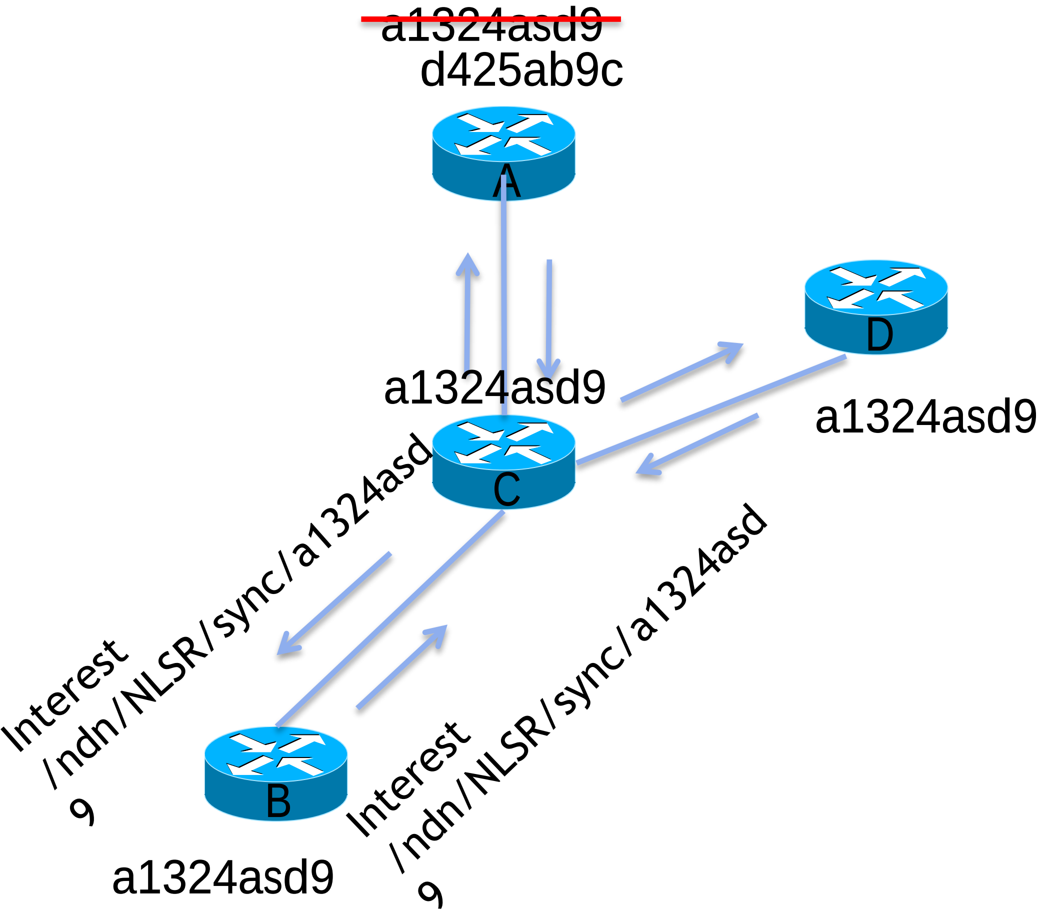
[2] Z. Zhu, A. Afanasyev, and L. Zhang. Let's ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.

From Flooding to Synchronization

NLSR uses ChronoSync [2] to synchronize LSDB.

- 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 LSA.

A has a new LSA

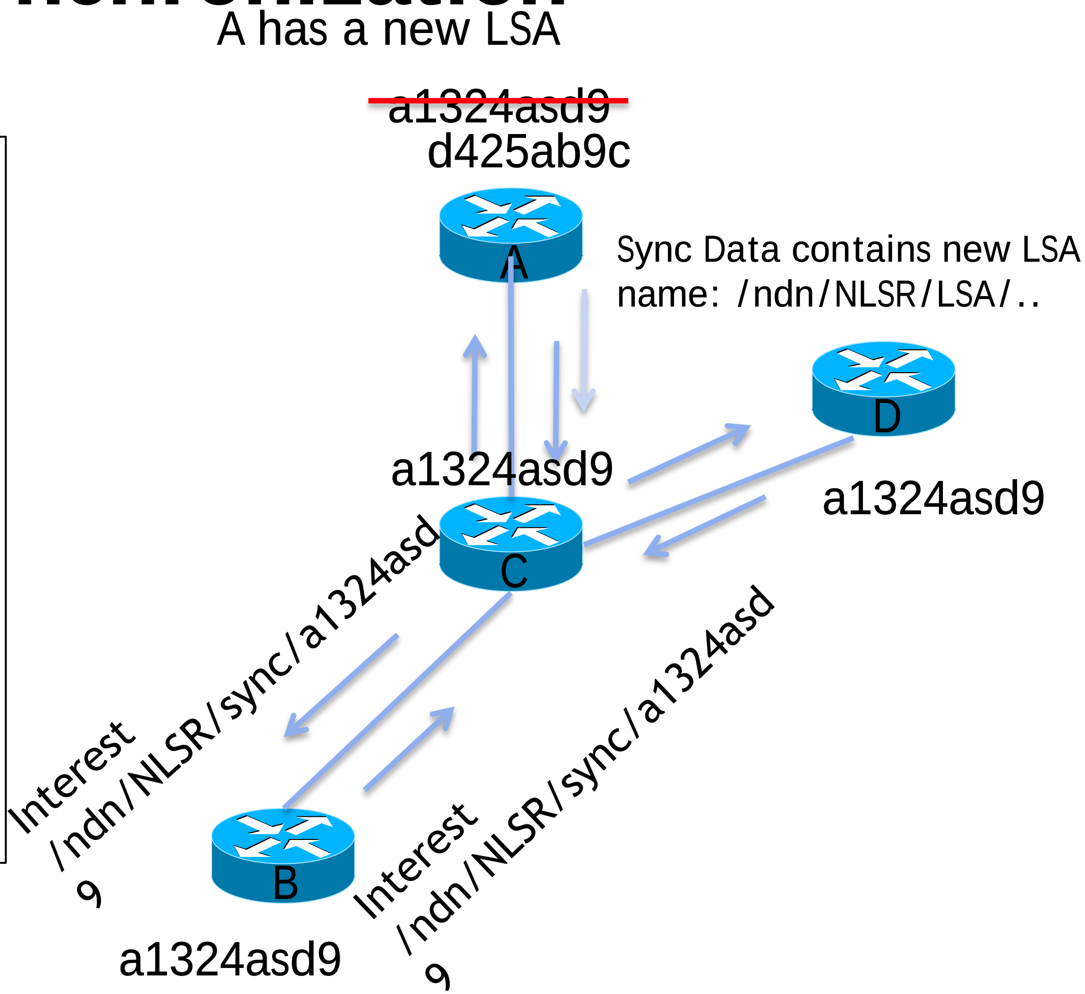

~~a1324asd9~~
d425ab9c
A
D
a1324asd9
a1324asd9
C
Interest /ndn/NLSR/sync/a1324asd9
Interest /ndn/NLSR/sync/a1324asd9
B
a1324asd9

[2] Z. Zhu, A. Afanasyev, and L. Zhang. Let’s ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.

From Flooding to Synchronization

NLSR uses ChronoSync [2] to synchronize LSDB.

- 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 LSA.

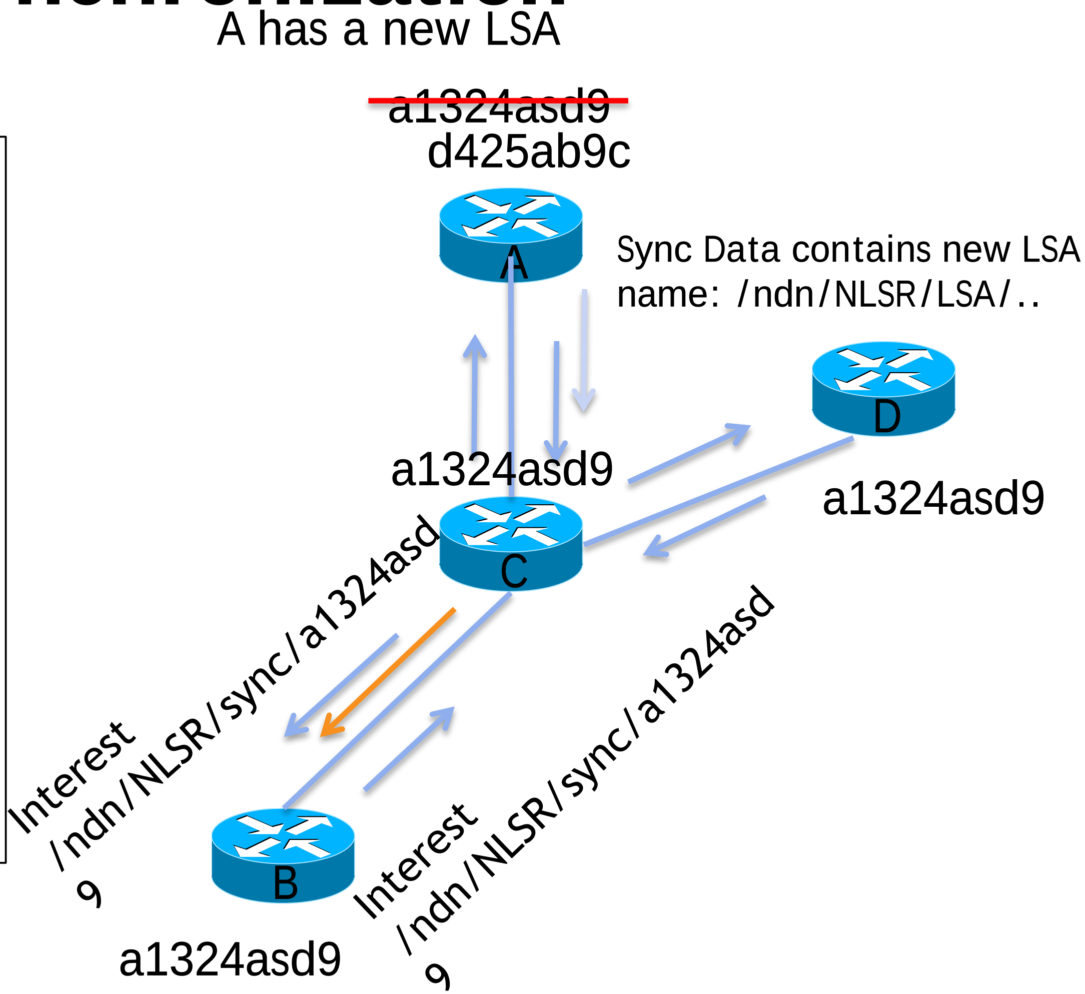


[2] Z. Zhu, A. Afanasyev, and L. Zhang. Let's ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.

From Flooding to Synchronization

NLSR uses ChronoSync [2] to synchronize LSDB.

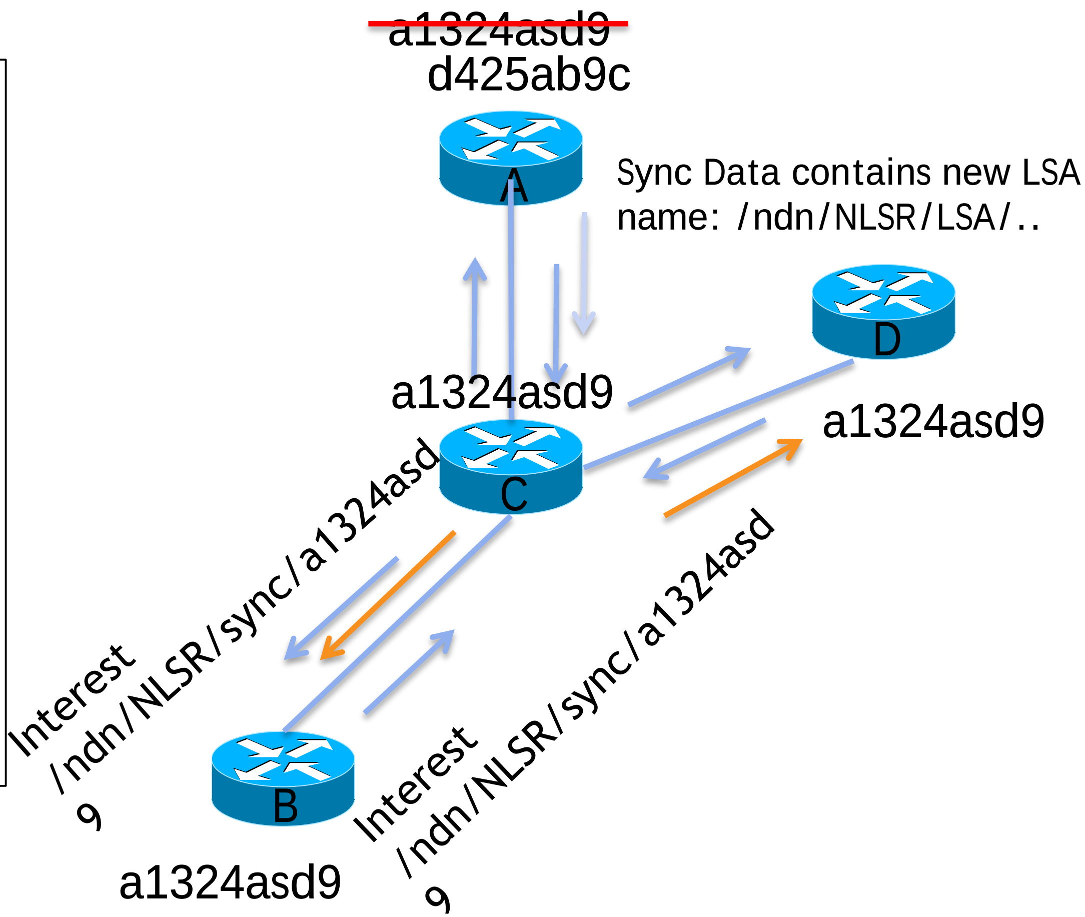
- Every node periodically sends a digest of LSDB to others in Interest packets.
- When a node has a new LSA, its digest changes and it will reply to others' Interests with name of new LSA.
- Other nodes fetch new LSA.


A has a new LSA
~~a1324asd9~~
d425ab9c
A
Sync Data contains new LSA name: /ndn/NLSR/LSA/..
D
a1324asd9
C
a1324asd9
Interest /ndn/NLSR/sync/a1324asd9
B
a1324asd9
Interest /ndn/NLSR/sync/a1324asd9

[2] Z. Zhu, A. Afanasyev, and L. Zhang. Let's ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.

From Flooding to Synchronization

A has a new LSA


~~a1324asd9~~
d425ab9c
A
Sync Data contains new LSA
name: /ndn/NLSR/LSA/..
D
a1324asd9
C
a1324asd9
Interest /ndn/NLSR/sync/a1324asd9
B
a1324asd9
Interest /ndn/NLSR/sync/a1324asd9

NLSR uses ChronoSync [2] to synchronize LSDB.

- 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 LSA.

[2] Z. Zhu, A. Afanasyev, and L. Zhang. Let’s ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.

Development Status

- NLSR 0.1 was released on 8/25/14.
  - Supports both link state and hyperbolic routing
  - Uses ChronoSync to synchronize routing data
  - Uses a hierarchical trust model for routing within a single administrative domain (validation rules are configurable).
- Running on NDN testbed.
- Code and doc:
  - http://named-data.net/doc/NLSR/0.1.0/

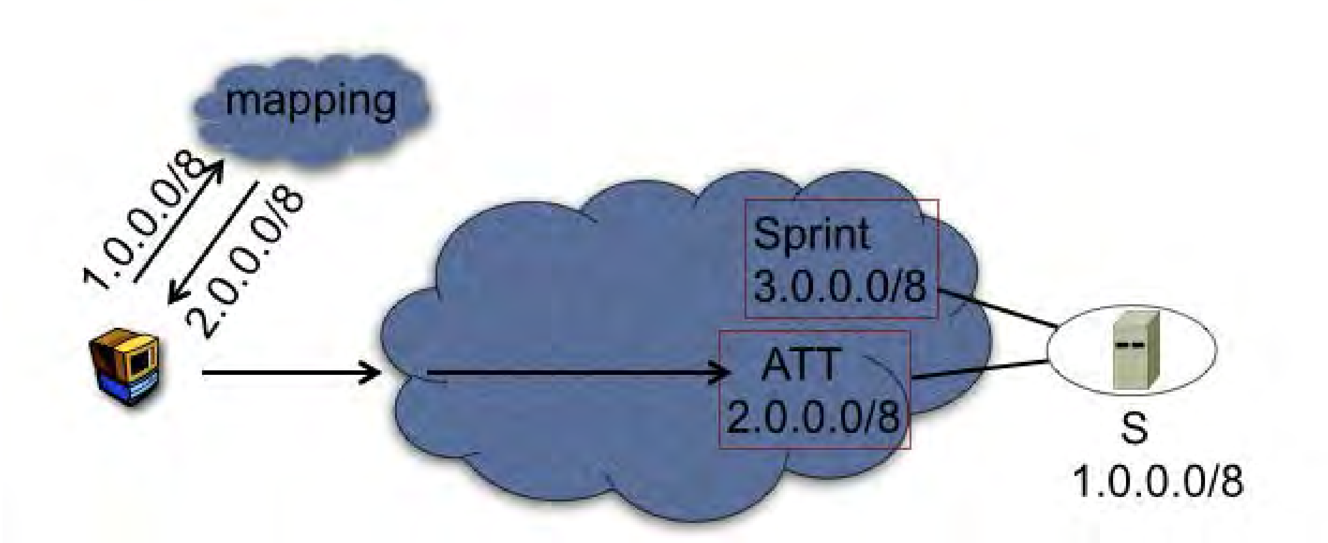
Lessons from practice

- NLSR provides a real use case for several NDN features, and drives the development of these features.
  - Security and trust model in NDN-CXX library
  - RIB management and prefix management in NFD
- Using ChronoSync simplifies protocol design and implementation.
  - ChronoSync fixed a number of problems of previous sync/repo.
- Testbed deployment helped discover problems in NLSR, NFD, NDN-CXX.
  - E.g., clock out-of-sync, flaky UDP tunnels.

ROUTING SCALABILITY

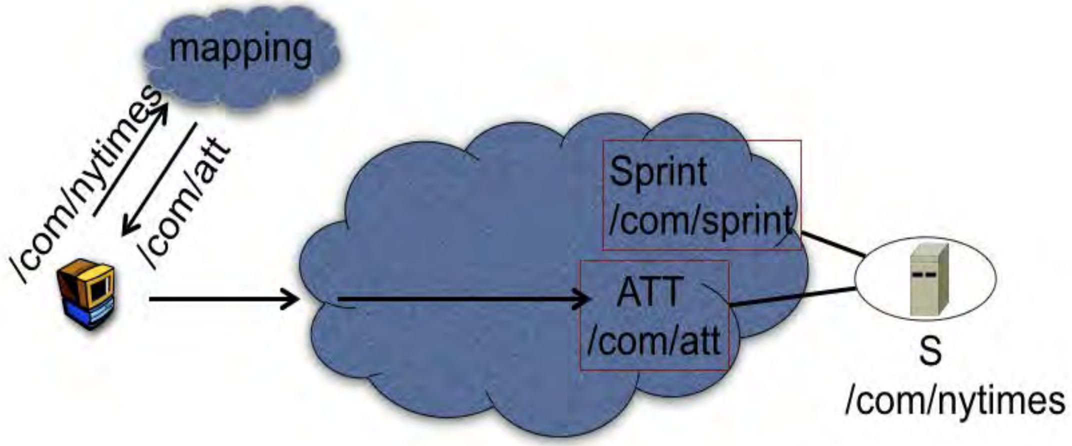
Scaling Solution 1: Map-and-Encap

- Number of application names is unbounded: over 200 million 2nd-level DNS names
- Map-n-Encap: originally proposed to scale IP routing (RFC 1955 in 1996).
  - Map edge network addresses to transit network addresses.



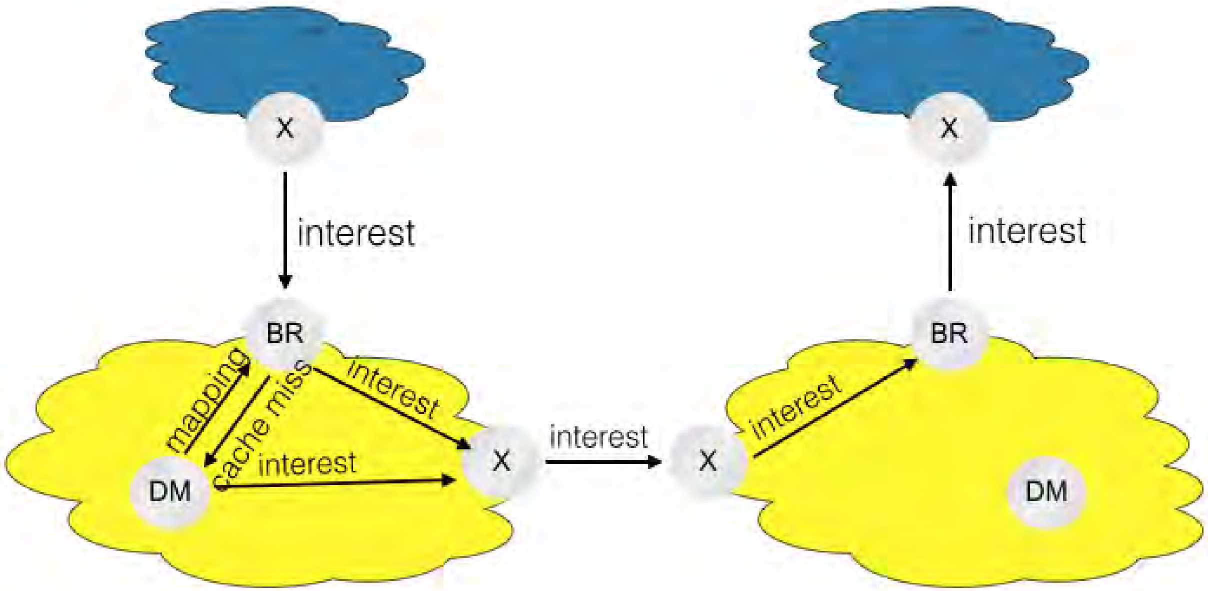
NDN can follow Map-n-Enap approach.

- Map application name prefixes to routable name prefixes, which usually belong to ISPs
- Packets carry the routable name prefix (and application name).
- Routers store only routable (ISP) name prefixes.



APT-NDN

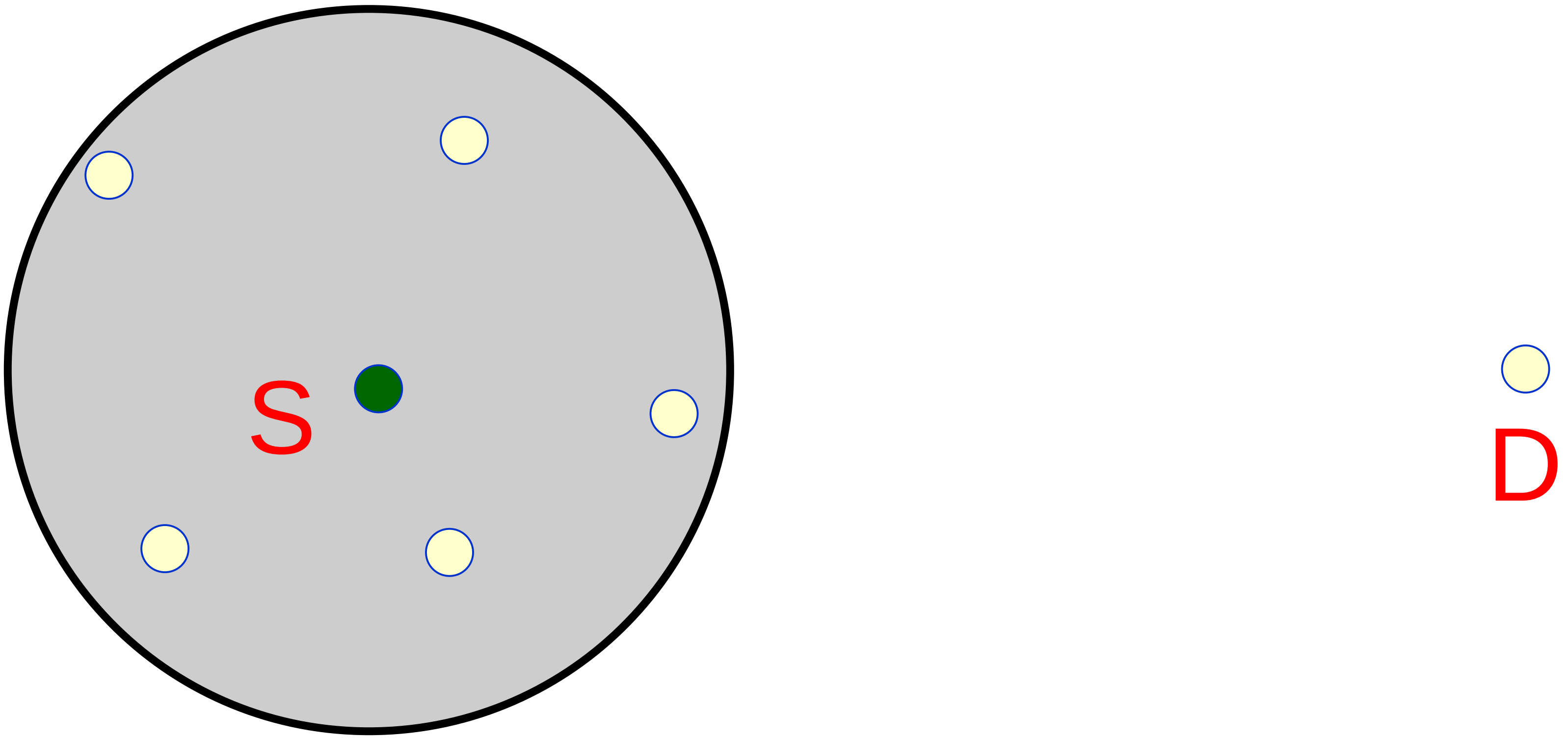
- Concrete realization of Map-and-Encap in NDN
- Status: implemented in ndnSIM.



Scaling Solution 2: Hyperbolic Routing

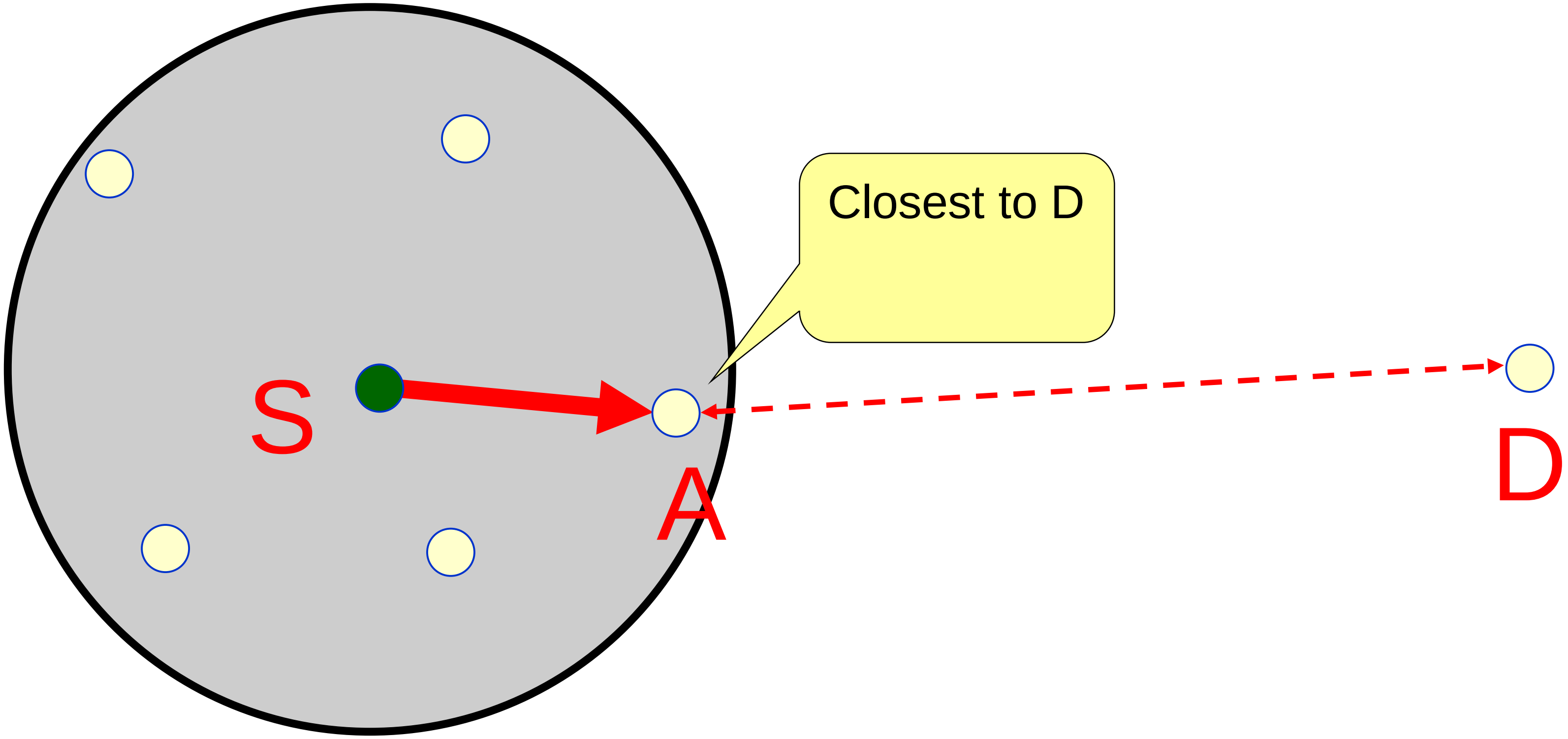
- There are many (emerging) ways to do routing, e.g., Small Worlds, Geographic Hyperbolic, Epidemic percolation.
- In general they're easier to implement and work better for NDN than for IP:
- Hyperbolic routing
  - Each node and name prefix have a set of semi-static hyperbolic coordinates.
  - Calculate next-hops based on each neighbor's distance to the name prefix
  - No need to distribute topology (links) and updates.

Greedy Forwarding in Hyperbolic Routing



- To forward packets:
  - Find neighbor who is closest to the destination
  - Forward the packet to the neighbor

Greedy Forwarding in Hyperbolic Routing



- To forward packets:
  - Find neighbor who is closest to the destination
  - Forward the packet to the neighbor

Current status

- Hyperbolic coordinates distributed in NLSR for comparison with link-state routing
- Preliminary emulation results at http://netwisdom.cs.memphis.edu/pvthome.html
- Plan: run hyperbolic routing on NDN testbed next month

References

1. L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, kc claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, Named Data Networking, to appear in *ACM SIGCOMM CCR* (also *NDN Technical Report 0019*)
2. D. Kulinski, J. Burke, L. Zhang. Video Streaming over Named Data Networking. *IEEE COMSOC MMTC E-Letter*, 2013.
3. Z. Zhu, C. Bian, A. Afanasyev, V. Jacobson, and L. Zhang. Chronos: Serverless multi-user chat over NDN. *Technical Report NDN-0008*, NDN Project, October 2012.
4. A. Afanasyev, Z. Zhu, L. Zhang, The story of ChronoShare, or how NDN brought distributed file sharing back, under review
5. J. Burke, P. Gasti, N. Nathan, and G. Tsudik. Securing instrumented environments over Content-Centric Networking: the case of lighting control. In *IEEE INFOCOM NOMEN Workshop*, Apr. 2013.
6. W. Shang, Q. Ding, A. Marianantoni, J. Burke, and L. Zhang. Securing building management systems using named data networking. *IEEE Network Special Issue on Information-Centric Networking*, April 2014.
7. G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang. VANET via Named Data Networking. In *IEEE INFOCOM NOMEN Workshop*, Apr. 2014.
8. A. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang. Named-data link state routing protocol. In *ACM SIGCOMM ICN Workshop*, 2013.

References (cont’d)

9. Z. Zhu, A. Afanasyev, and L. Zhang. Let’s ChronoSync: Decentralized dataset state synchronization in NDN. In *ICNP*, 2013.
10. H. Yuan, T. Song, and P. Crowley. Scalable NDN forwarding: Concepts, issues and principles. In *ICCCN*, 2012.
11. H. Yuan and P. Crowley. Scalable pending Interest table design: From principles to practice. *IEEE INFOCOM*, 2014.
12. W. So, A. Narayanan, and D. Oran. Named data networking on a router: Fast and DoS-resistant forwarding with hash tables. In *ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, Oct 2013.
13. M. Varvello, D. Perino, and J. Esteban. Caesar: A content router for high speed forwarding. In *ACM SICOMM Workshop on Information-centric Networking*, 2012.