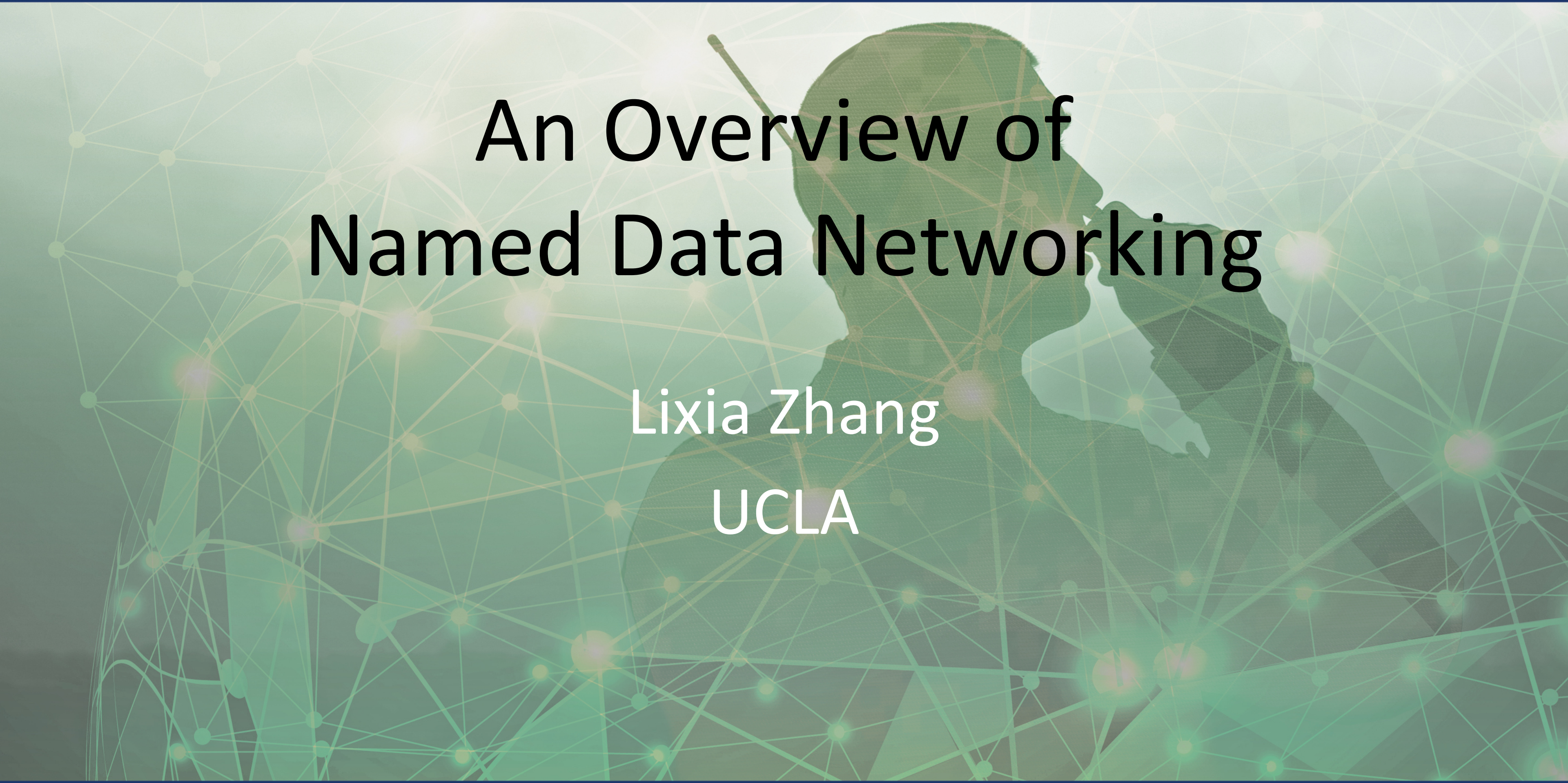
MILCOM2017

MILITARY COMMUNICATIONS AND INNOVATION - PRIORITIES FOR THE MODERN WARFIGHT



An Overview of Named Data Networking

Lixia Zhang

UCLA

BALTIMORE, MD • OCTOBER 23–25, 2017

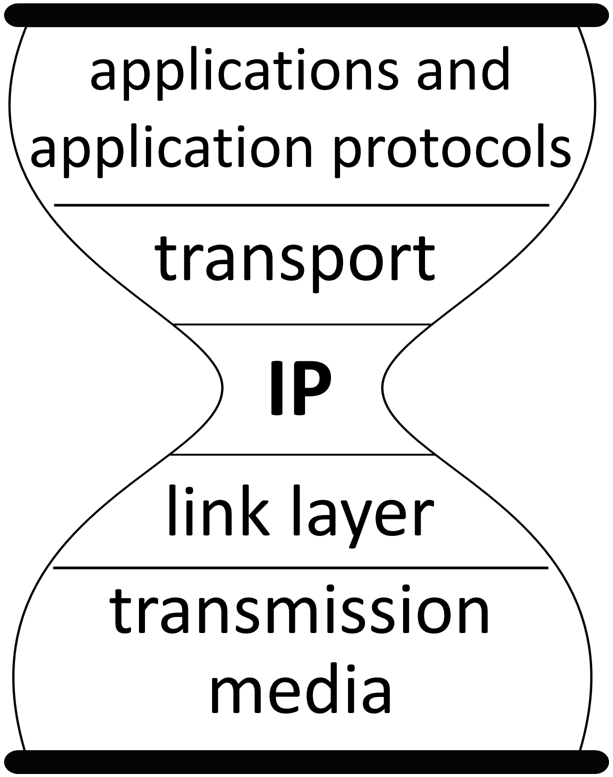
- Named Data Networking: http://www.named-data.net
  - Started 2010 under NSF "Future Internet Architecture" program, continuing
- My talk today:
  - Internet protocol architecture today
  - Where is, and what is, NDN in the big picture
  - How NDN works (brief intro to the basic concepts)

Why NDN is particularly suited for battlefield networking

later talks:

- NDN codebase, testbed, and other experimentation /evaluation tools
- NDN security (briefly)

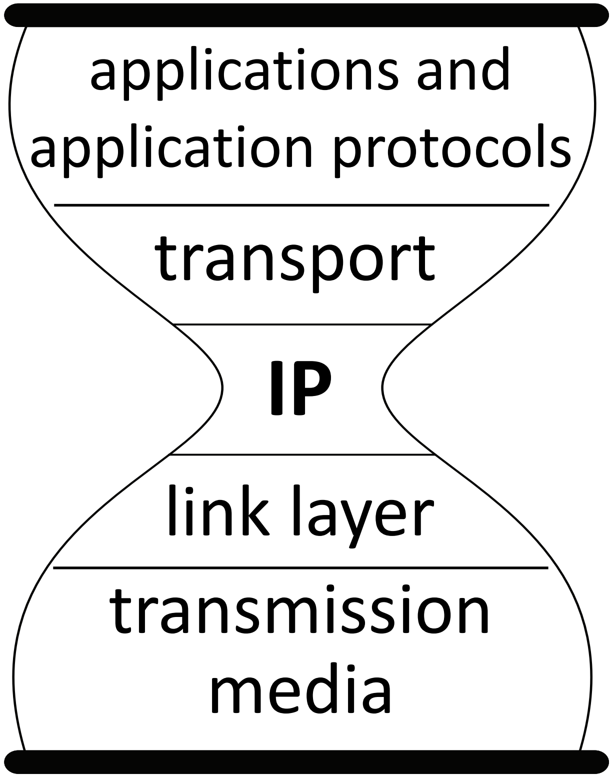
Hourglass-shaped Internet Protocol Architecture



```
+------+ +-----+  +-----+     +-----+
|Telnet| | FTP |  | TFTP| ... | ... |
+------+ +-----+  +-----+     +-----+
      |   |          |           |
     +-----+      +-----+     +-----+
     | TCP |      | UDP | ... | ... |
     +-----+      +-----+     +-----+
        |            |           |
      +------------------------+
      |Internet Protocol & ICMP|
      +------------------------+
        |          |           |
  +-----+  +--------+ +---------------+
  |Local|  |Network|  |Protocols(many)|
  +-----+  +--------+ +---------------+
```

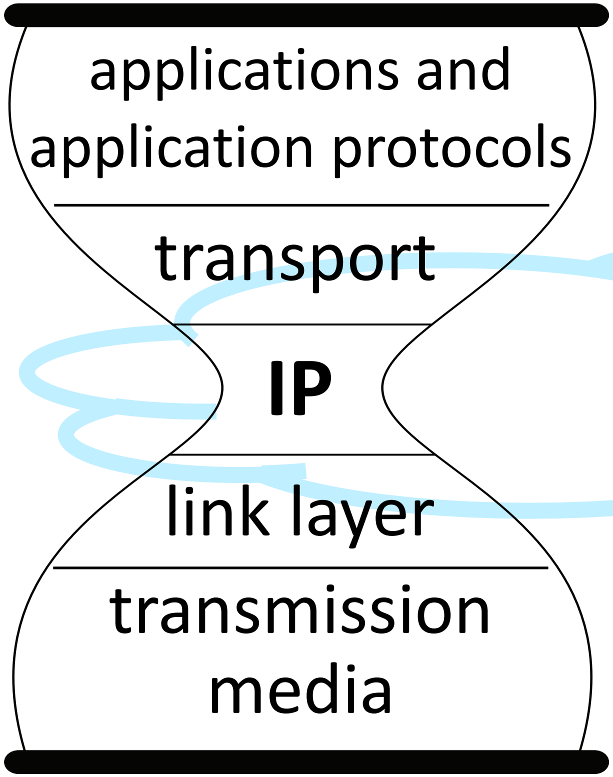
RFC791: Internet Protocol Specification
(September 1981)

Hourglass-shaped Internet Protocol Architecture



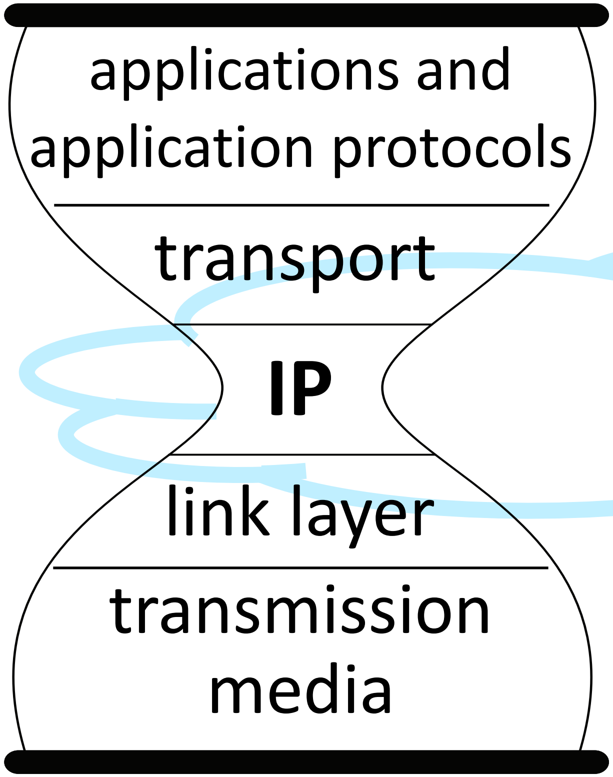
- Transport layer: only sees the two ends
  - Network as a black box
- IP: Delivering packets from any host to any other host
- Link layer: from one node to the next
  - across one-hop

IP: connecting any host to any other host



- Transport layer: only sees the two ends
  - Network as a black box
- IP: Delivering packets from any host to any other host
- Link layer: from one node to the next
  - across one-hop

Internet applications have changed over time



applications and application protocols

transport

IP

link layer

transmission media

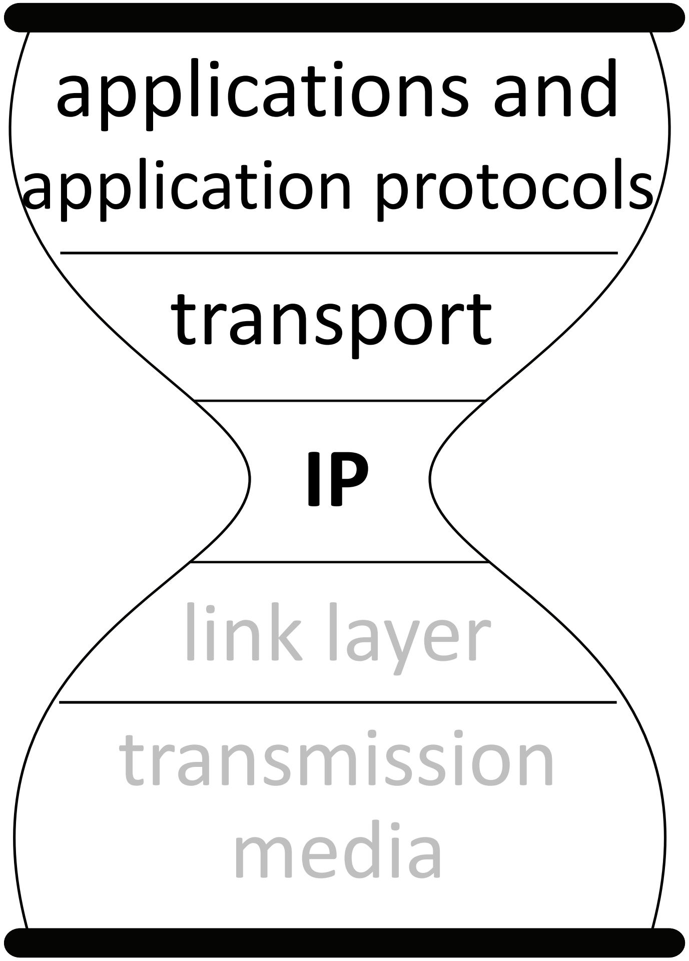
- Tr
  e e
  – Network as a bla

and many more

- IP: Delivering packets from any host to any other host

- Link layer: from one node to the next
  – across one-hop

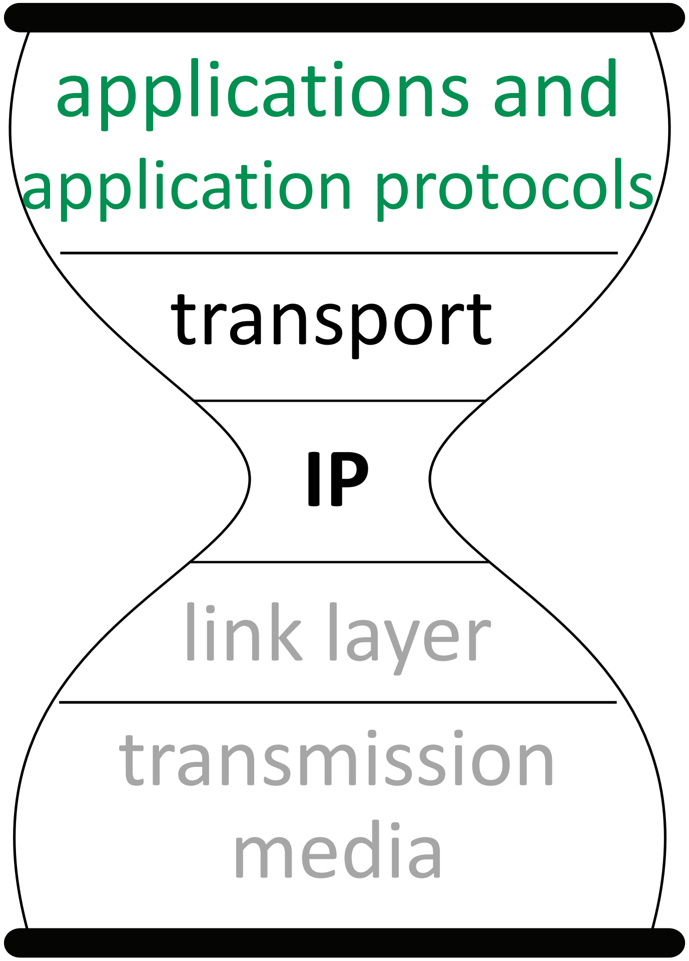
The problem this hourglass picture doesn’t show



Use of different namespaces

- Applications/application protocols use **names** for *data exchange*
- IP + Transport: a virtual *pipe* between **a pair of IP addresses**
- Link layer: deliver based on MAC addresses
  - – ignoring here for simplicity

Why Different Namespaces Matter



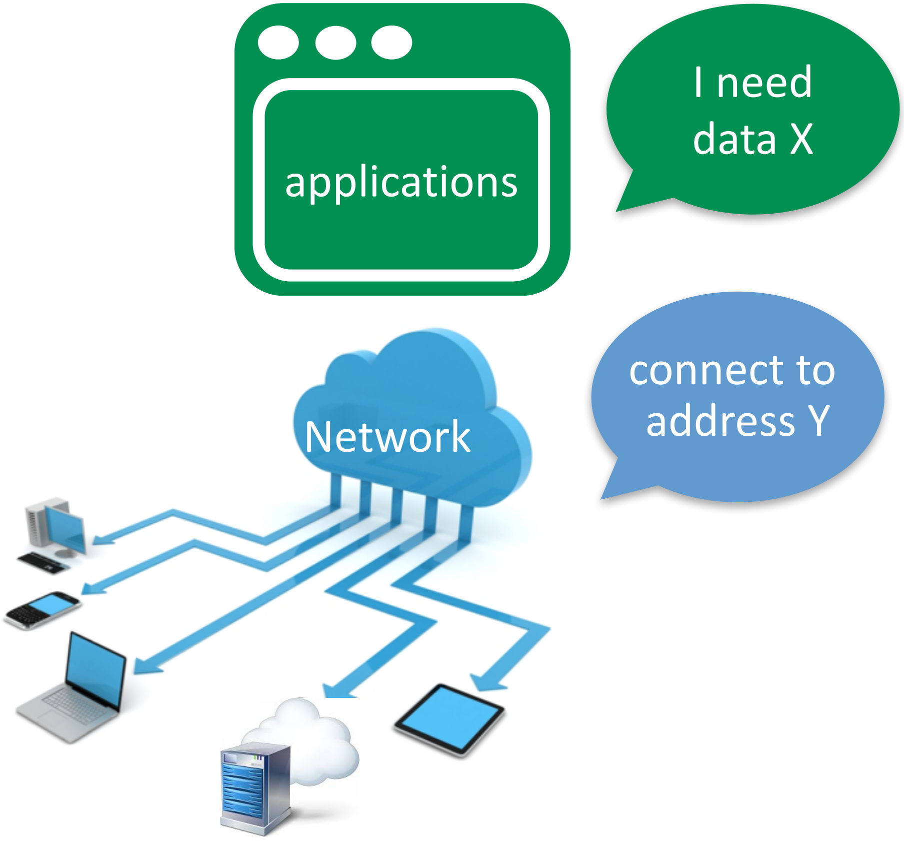
applications and application protocols

transport

IP

link layer

transmission media



applications

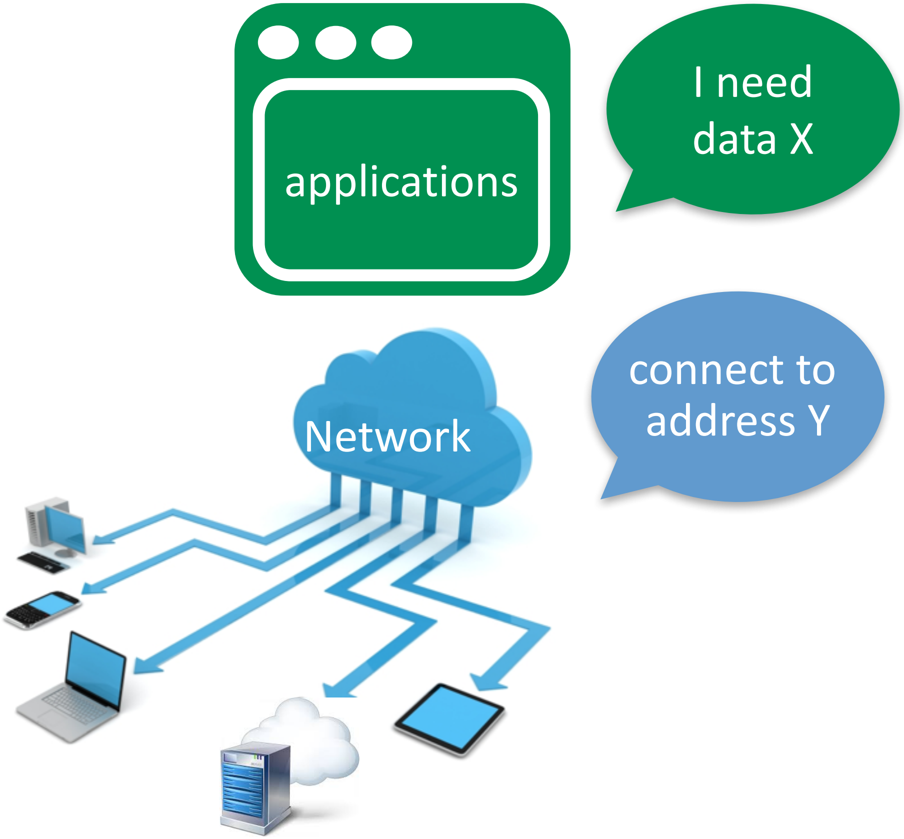
I need data X

Network

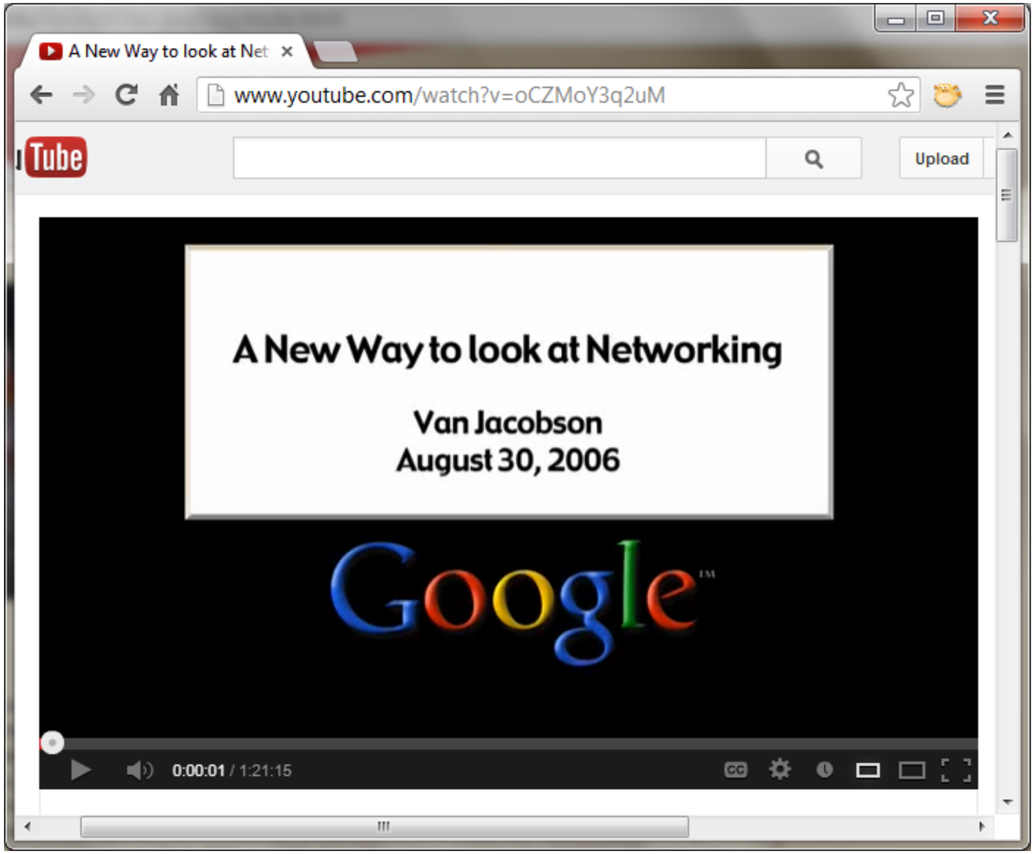
connect to address Y

Dependency on infrastructure service/stability

- A node must get an IP address before it can communicate
  - Dependency on DHCP service
- Application name → IP addresses
  - Dependency on DNS service
- Transport: dependency on stable E2E connectivity



The new way: name & fetch data directly, at network layer

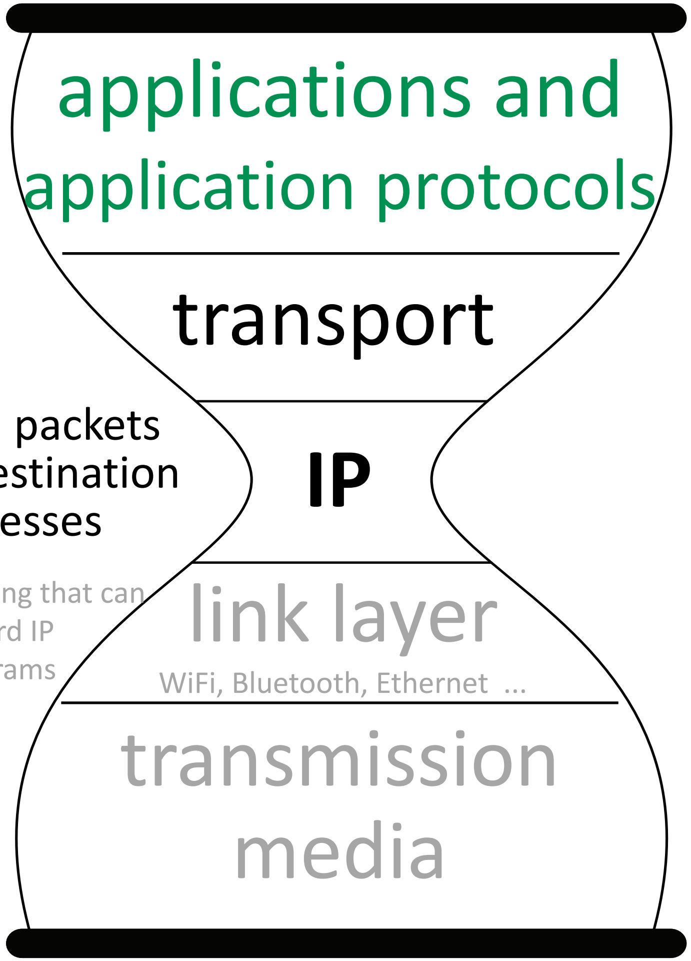


A New Way to look at Networking

Van Jacobson
August 30, 2006

https://www.youtube.com/watch?v=oCZMoY3q2uM

From IP to NDN: a *conceptually* simple change



applications and application protocols

transport

IP

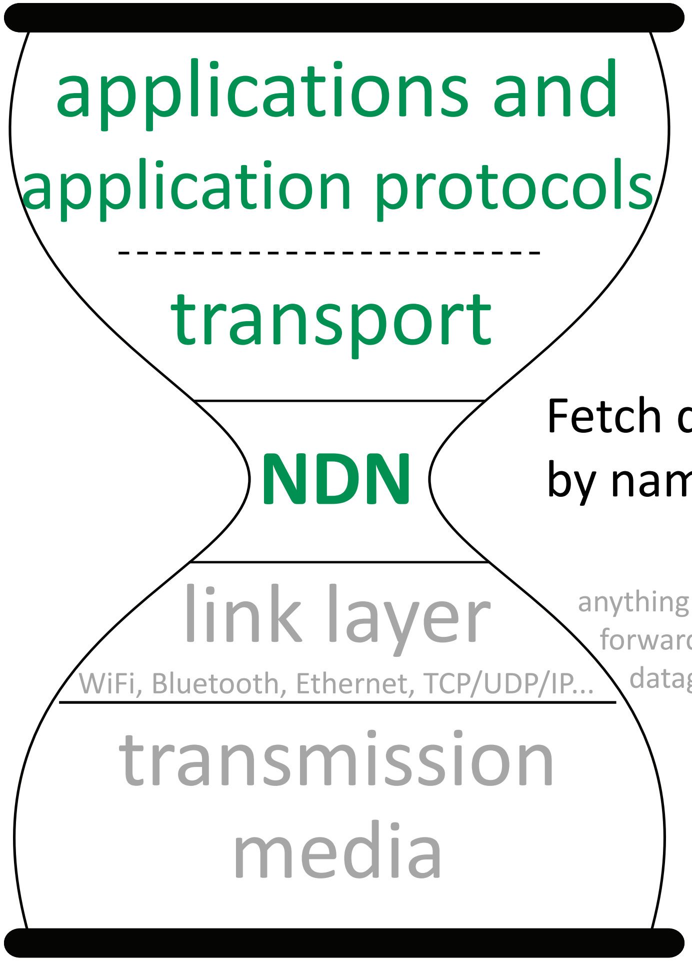
link layer

WiFi, Bluetooth, Ethernet ...

transmission media

Push packets to destination addresses

anything that can forward IP datagrams



applications and application protocols

transport

NDN

link layer

WiFi, Bluetooth, Ethernet, TCP/UDP/IP...

transmission media

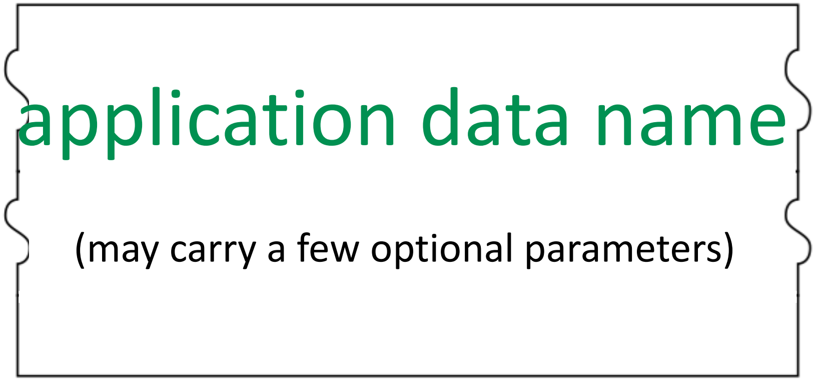
Fetch data by names

anything that can forward NDN datagrams

Application and network layer share the same ***data*** namespace

NDN: 2 types network-layer packets

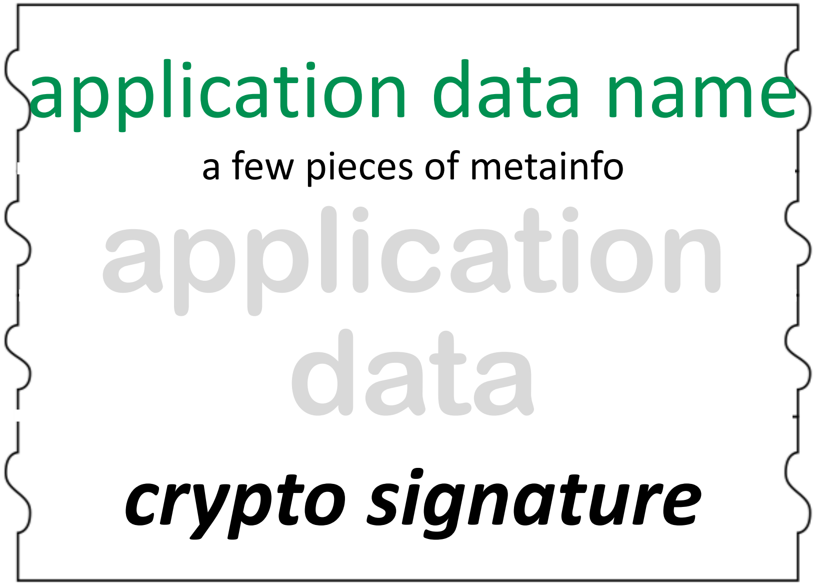
Interest packet



application data name

(may carry a few optional parameters)

Data packet



application data name

a few pieces of metainfo

application data

crypto signature

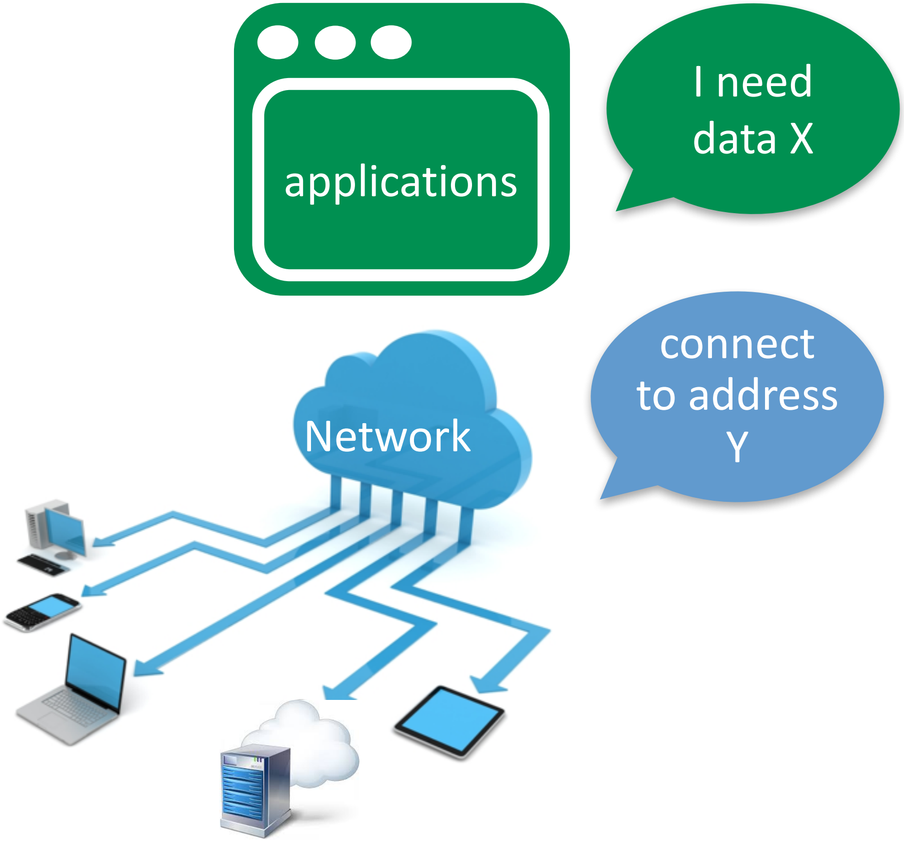
Data consumers send Interest packets

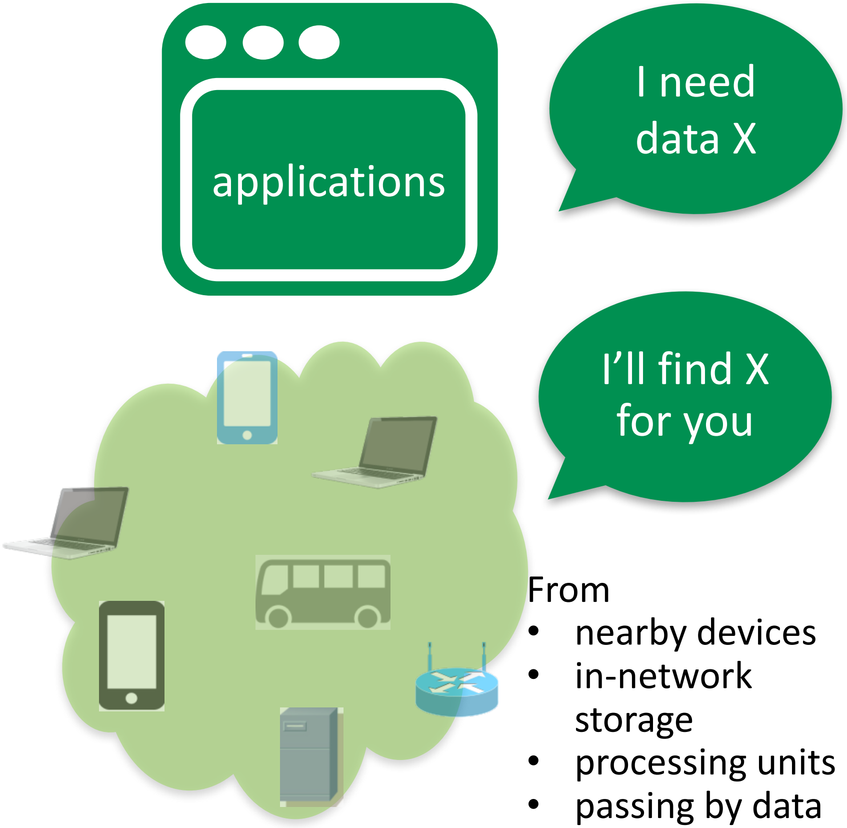
Whoever has the matching Data packet can reply

Publisher binds name to content; receivers verify

All data immutable

NDN: remove incongruity between app and network





From
- nearby devices
- in-network storage
- processing units
- passing by data mules via ad hoc connectivity

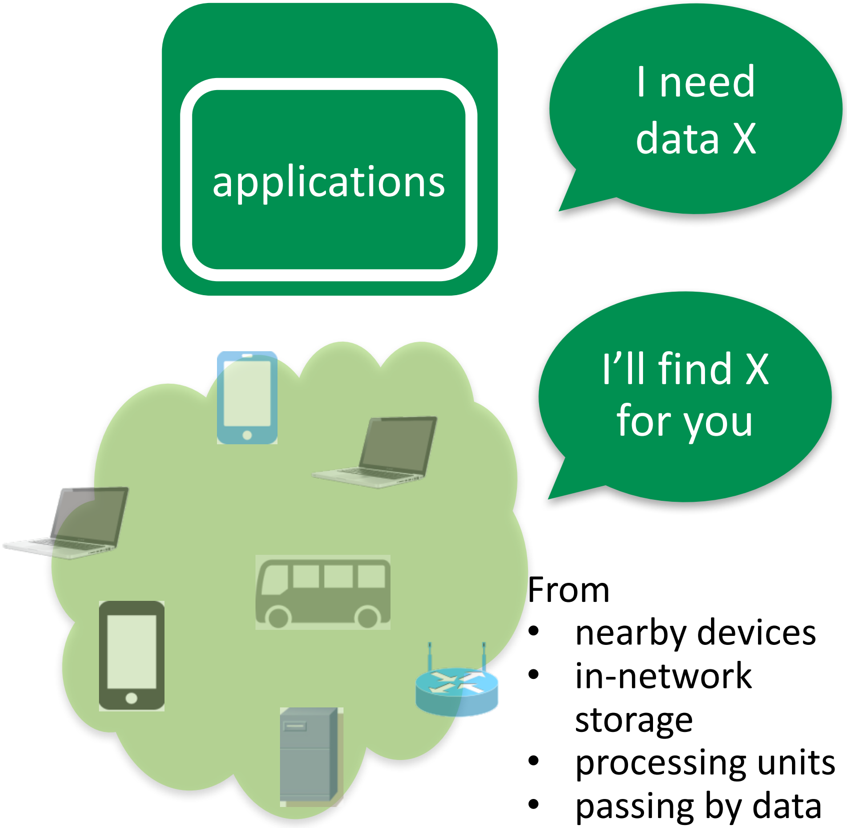
in addition to the original producer

All devices: potential data suppliers

NDN: remove incongruity → simplify overall system

- ~~Dependency on DHCP service~~
- ~~Dependency on DNS service~~
- ~~Dependency on stable end-to-end connectivity~~

When data must is fetched from far away servers: NDN enables resilient hop-by-hop forwarding



applications

I need data X

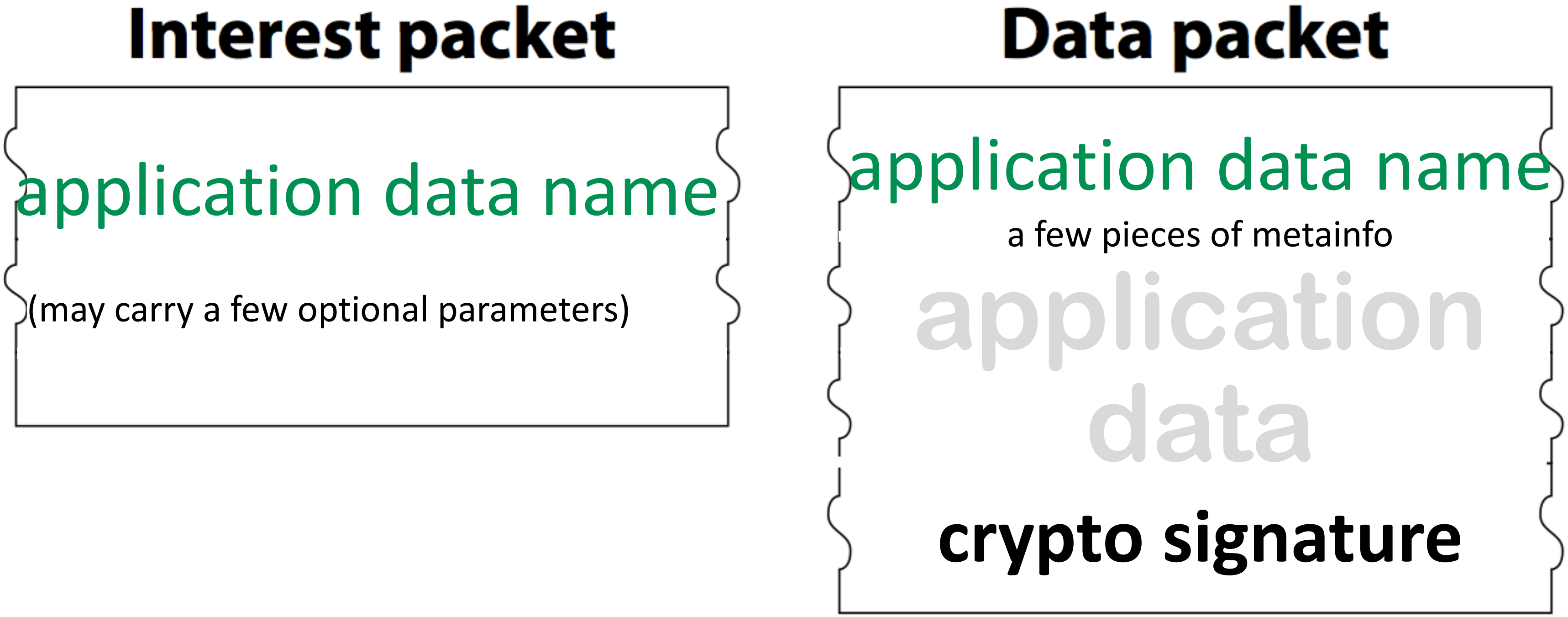
I'll find X for you

From
- nearby devices
- in-network storage
- processing units
- passing by data mules via ad hoc connectivity

in addition to the original producer

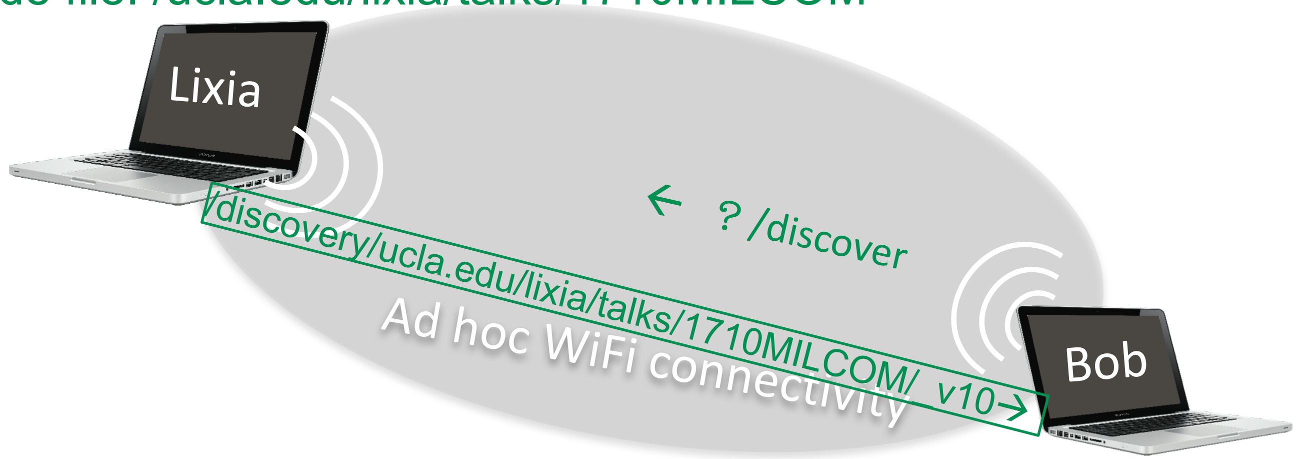
NDN’s secret sauce: It’s the *names*!

1. Interest and data packets carry names



Native support for ad hoc networking

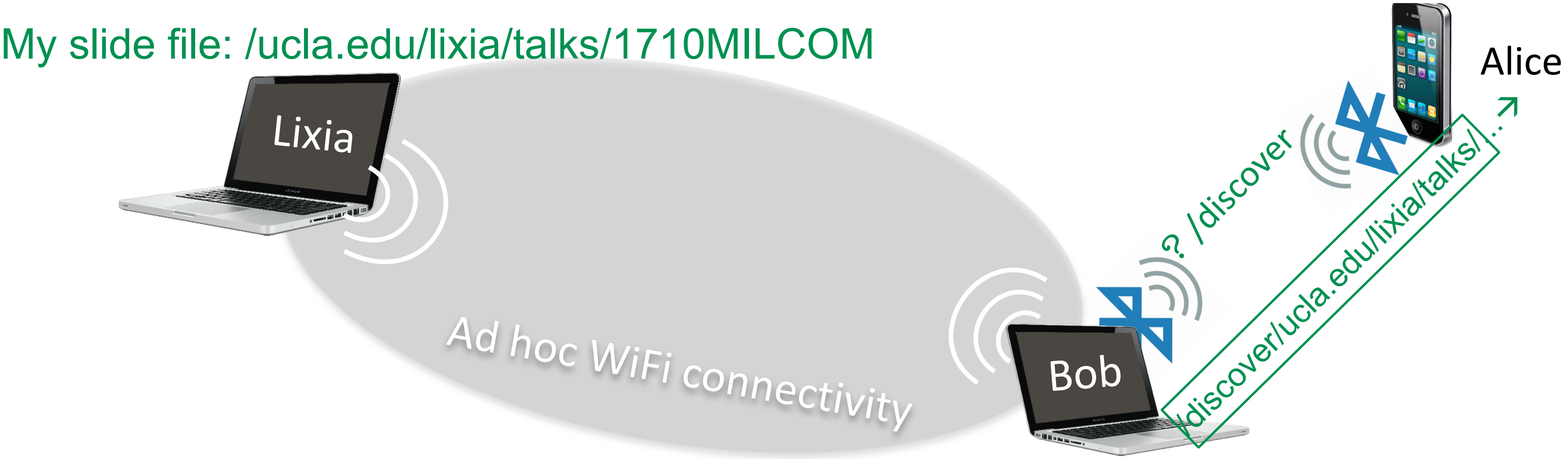
My slide file: /ucla.edu/lixia/talks/1710MILCOM



Example: Bob wants to discover nearby available content

- Bob broadcasts Interest packet: ?/discover/ *naming convention*
- Reply Data packet: /discover/ucla.edu/lixia/talks/1710MILCOM/_v10
  – may contain additional metainfo (e.g. total # of packets in the file)
- Bob saves the reply, remembers its incoming interface
  – A Data packet identified by its name, signed by its producer, cacheable at any node, *at network layer*

Multi-hop Information Discovery



My slide file: /ucla.edu/lixia/talks/1710MILCOM
Lixia
Alice
? /discover
/discover/ucla.edu/lixia/talks/...
Ad hoc WiFi connectivity
Bob

- Alice sends discovery Interest
- Bob's laptop finds a matching reply from cache
- Alice wants to retrieve the file

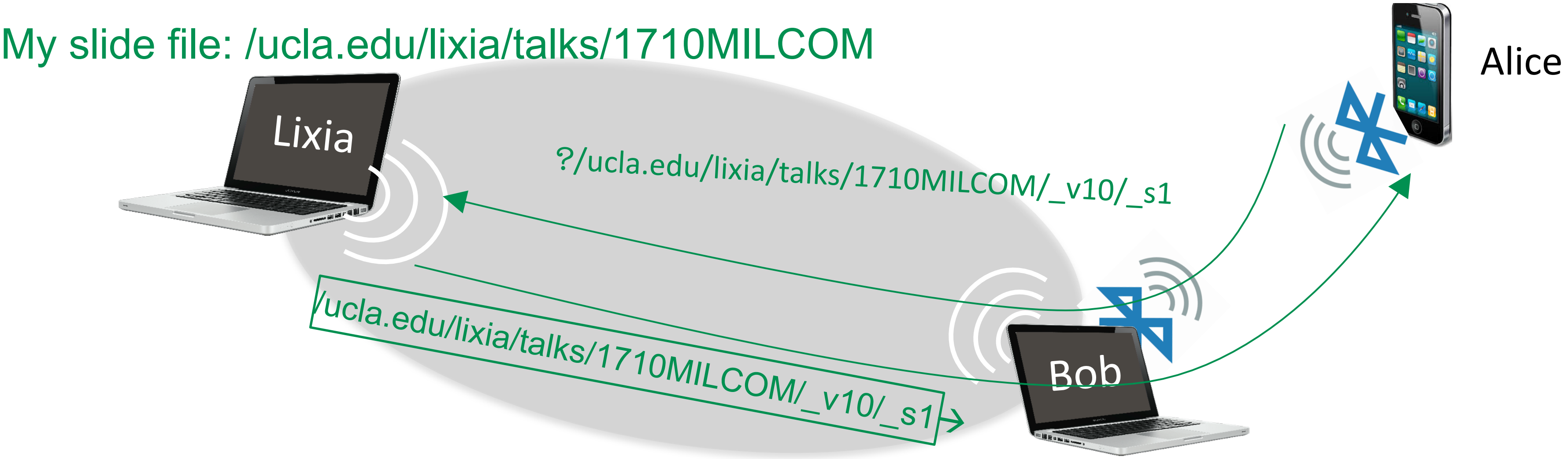
NDN’s secret sauce: It’s the *names*!

1. Interest and data packets carry names
2. Applications assign names to *every* data packet

   e.g.
   /ucla.edu/lixia/talks/1710MILCOM/_V10/_segment=1

   naming convention

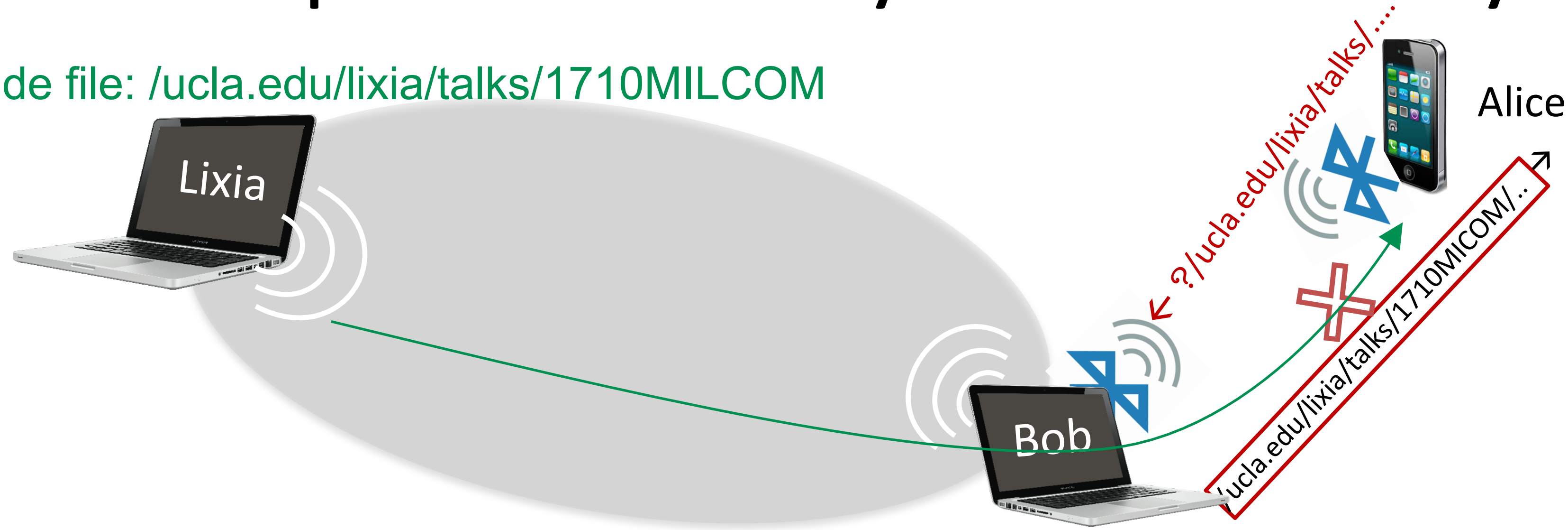
Multi-hop data delivery (without routing)


My slide file: /ucla.edu/lixia/talks/1710MILCOM
Lixia
Alice
?/ucla.edu/lixia/talks/1710MILCOM/_v10/_s1
/ucla.edu/lixia/talks/1710MILCOM/_v10/_s1
Bob

- Alice’s interest → Bob → Lixia: ?/ucla.edu/lixia/talks/1710MILCOM/_v10/_s1
  - Alice may send out multiple Interests (pipelining)
- NDN uses a *stateful forwarding plane*
  - Remember every forwarded, but not replied Interest packet → state to handle data return

Multi-hop data delivery with resiliency

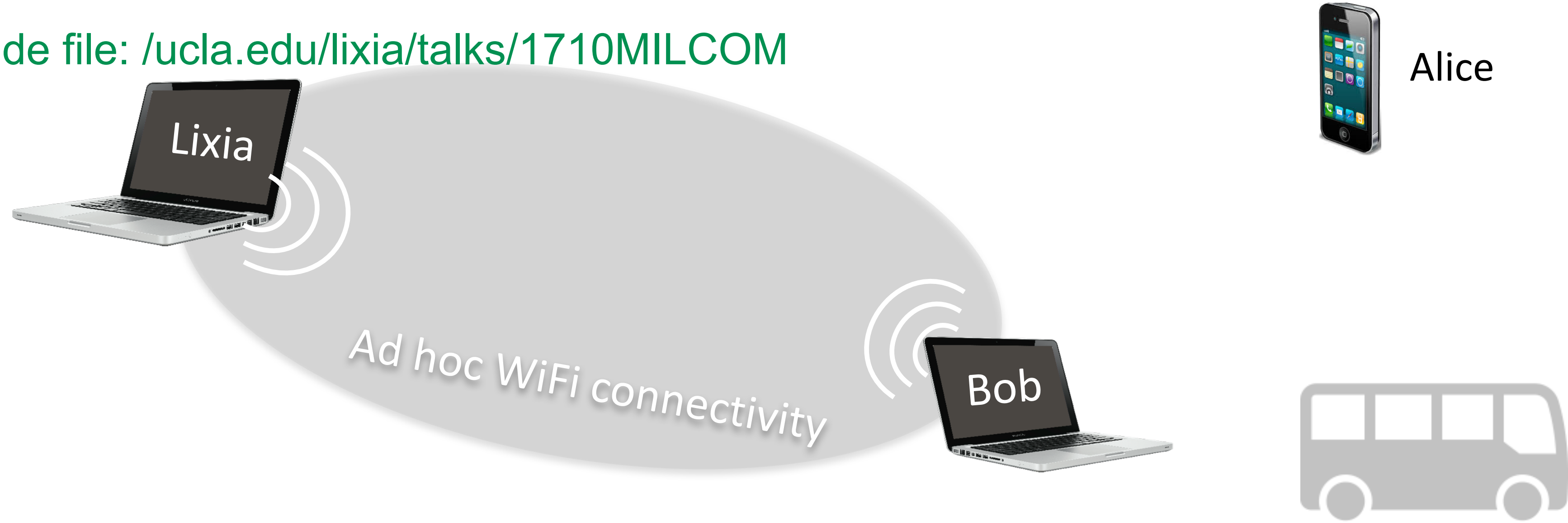
My slide file: /ucla.edu/lixia/talks/1710MILCOM


Lixia
Alice
?/ucla.edu/lixia/talks/....
/ucla.edu/lixia/talks/1710MICOM/..
Bob

- The data packet lost between Bob–Alice
- NDN adheres to E2E reliability: Alice retransmits unsatisfied interest
- The retransmitted Interest finds matching data in Bob’s cache

Support for ad hoc mobility, delay tolerance

My slide file: /ucla.edu/lixia/talks/1710MILCOM

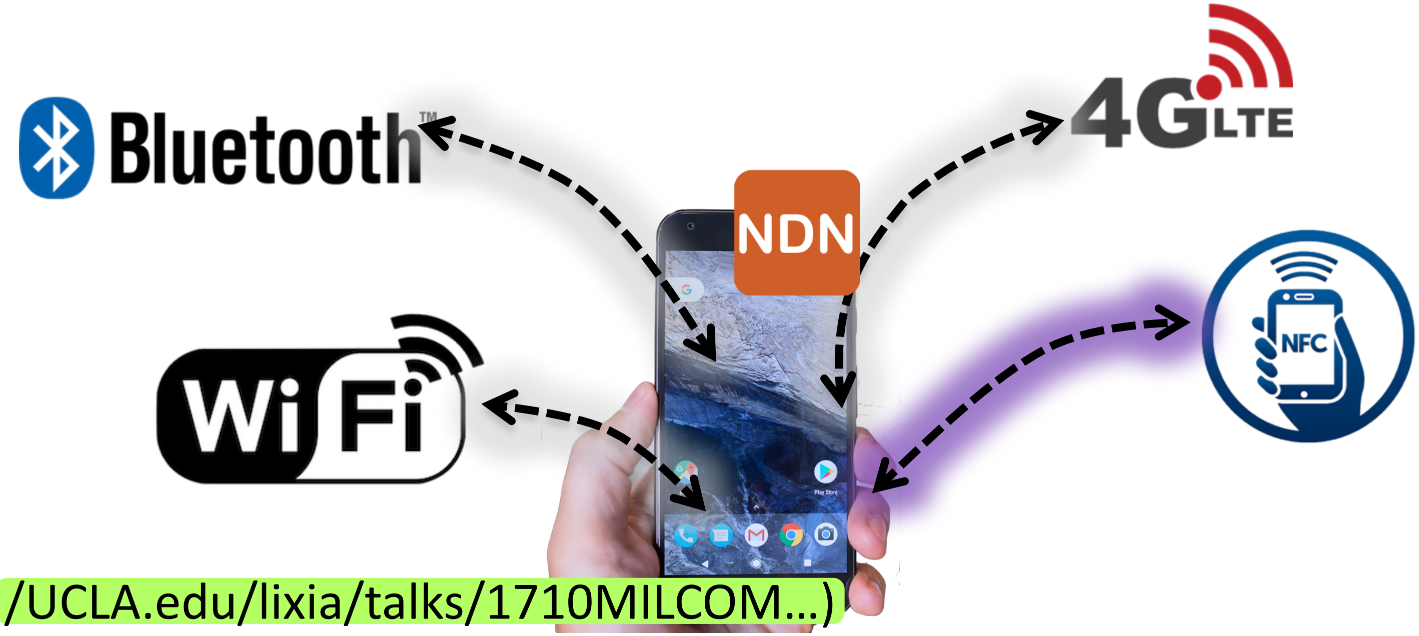


- Alice moves on, carrying the file to other places
  - data muling: may share the file with others on the bus
- Delay tolerant networking: only need 2 things

  a) device has storage, b) data has device-independent name

NDN’s secret sauce: It’s the *names*!

1. Interest and data packets carry names
2. Applications assign names to every data packet
3. Names enable automatic information discovery at *network layer* → ad hoc, delay tolerance networking
4. Naming data enables use of multiple interfaces



/UCLA.edu/lixia/talks/1710MILCOM…)

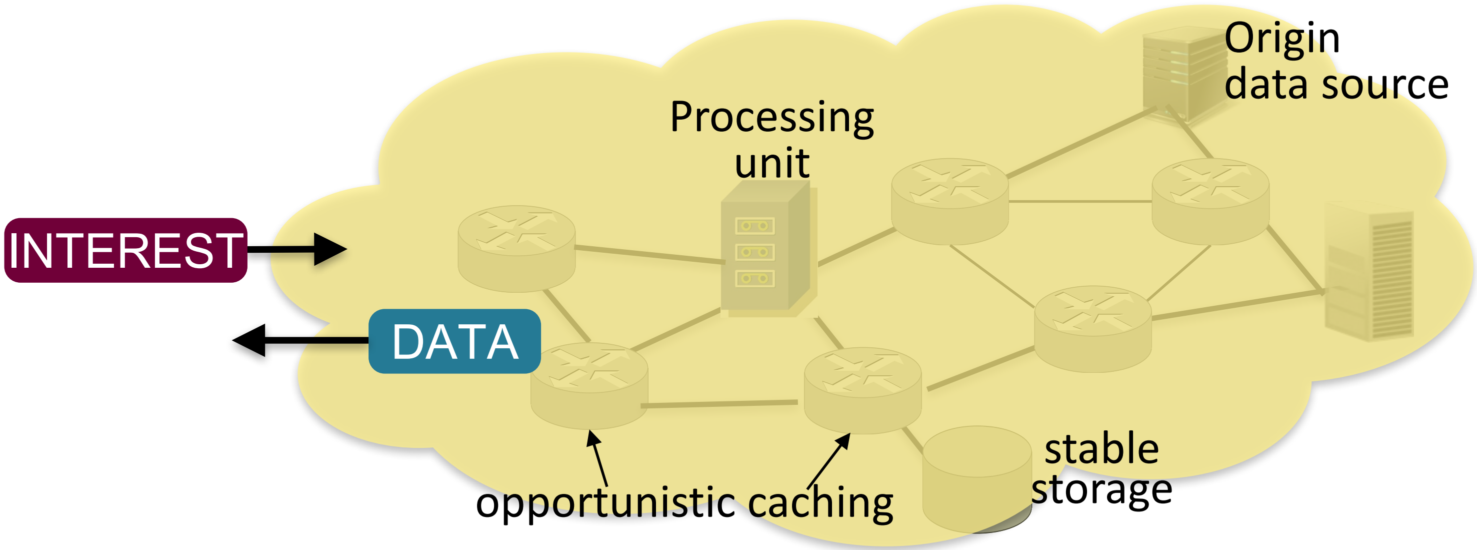
- IP assigns address for each interface, making it difficult to support host multihoming
- NDN data name independent from interface, freely use any or all interfaces
  - How to choose: NDN forwarding strategy (later)

NDN: It's the *names*!

1. Interest and data packets carry names
2. Applications assign names to *every* data packet
3. Names enable automatic information discovery at *network layer* → ad hoc, delay tolerance networking
4. Naming data enables use of multiple interfaces
5. Names are hierarchical
   - Preserve application context for data consumption →facilitate data authentication, confidentiality support (later talk)
   - Facilitate name aggregation

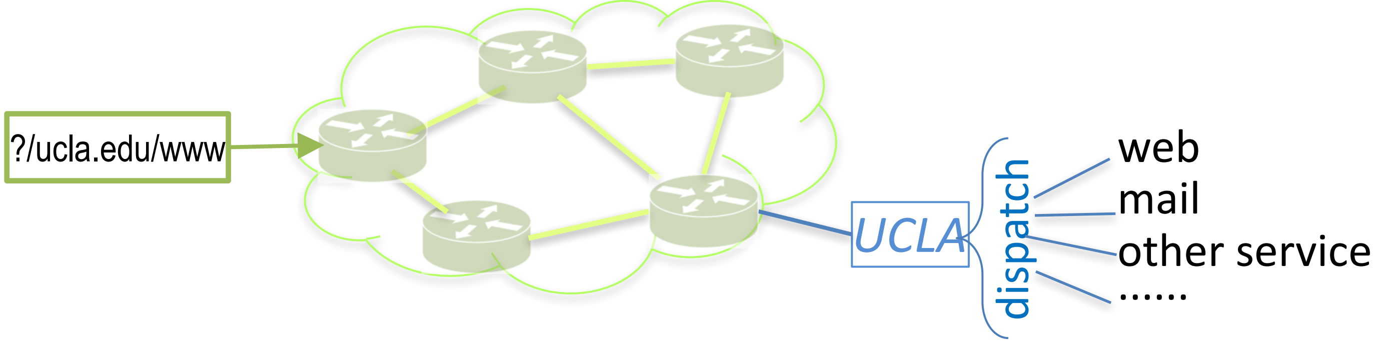
Steering Interest packets toward data in large scale

- Data source, in-network storage, and processing units can all supply requested data
- Building knowledge of directions to reach data in large scale environment: *utilizing routing protocols*



NDN: It’s the *names*!

1. Interest and data packets carry names
2. Applications assign names to every data packet
3. Names enable automatic information discovery at *network layer* → ad hoc, delay tolerance networking
4. Naming data enables use of multiple interfaces
5. Names are hierarchical
6. Names as demultiplexer across protocol layers

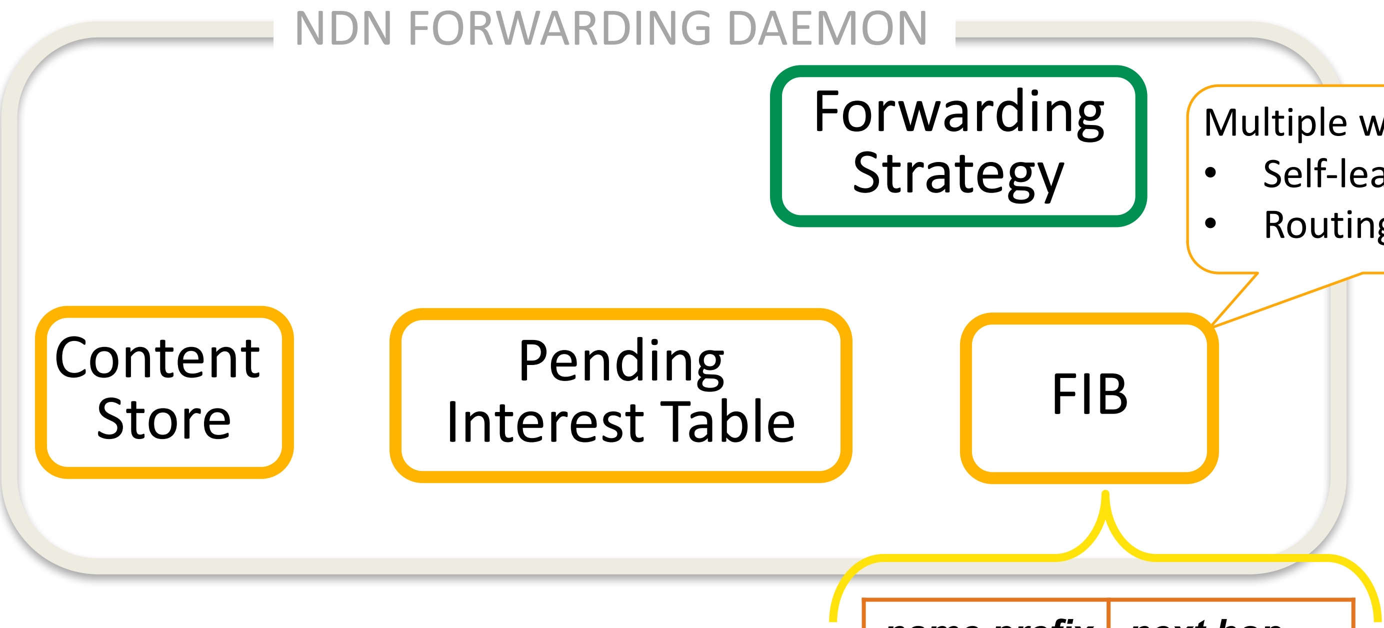


Advantages from naming data

- Networking without addresses
- Utilizing multiple interfaces
- Named, secured data packets can be cached anywhere
- Organic support for ad hoc mobile, DTN networking
- In cyberspace, everything can be treated as named, secured bags of bits
  - Crypto keys, security policies

NDN’s Stateful Forwarding Plane

NDN’s node model



Multiple ways to fill FIB

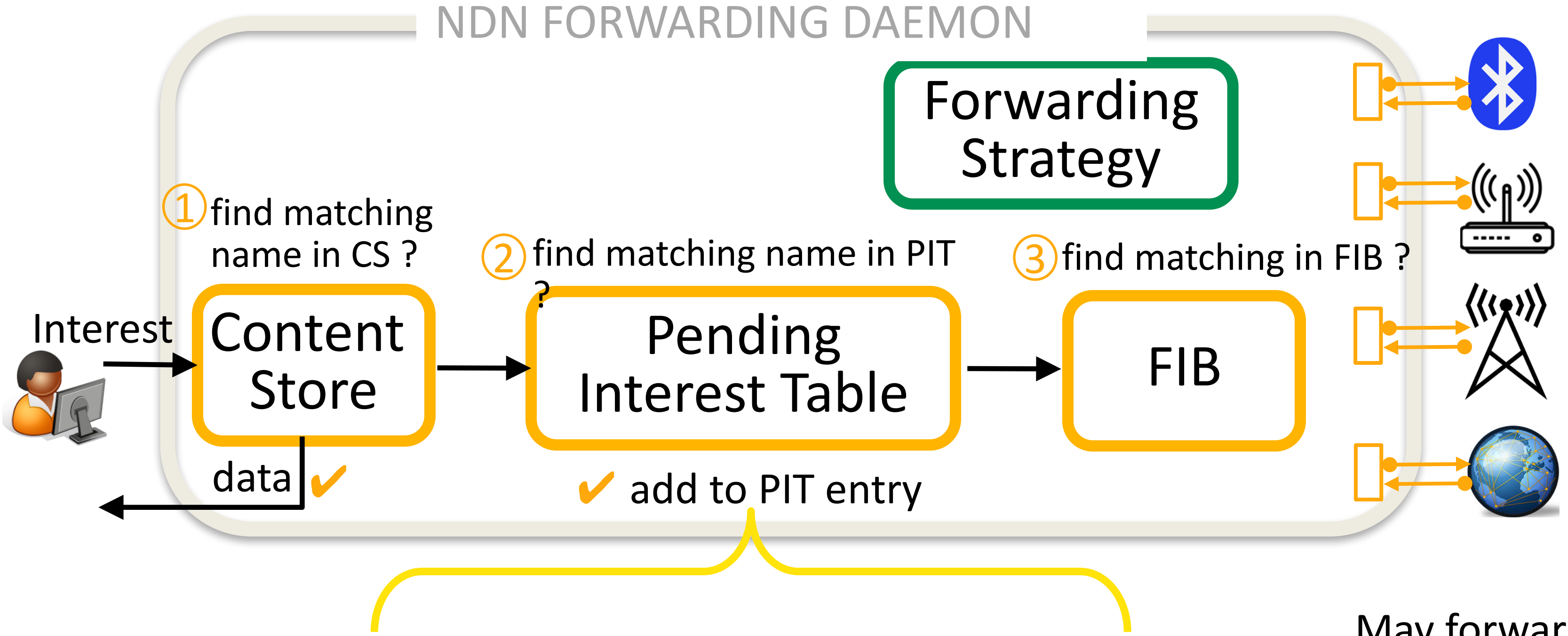
- Self-learning
- Routing protocol

name prefix	***next hop***
google.com	A, B
ucla.edu	B,C, D
ucla.edu/cs/lixia	A, B, D,

NFD module resides in every NDN node

PIT breaks Interest looping, enable NDN to freely use multiple paths

NDN Interest Forwarding: 3 steps

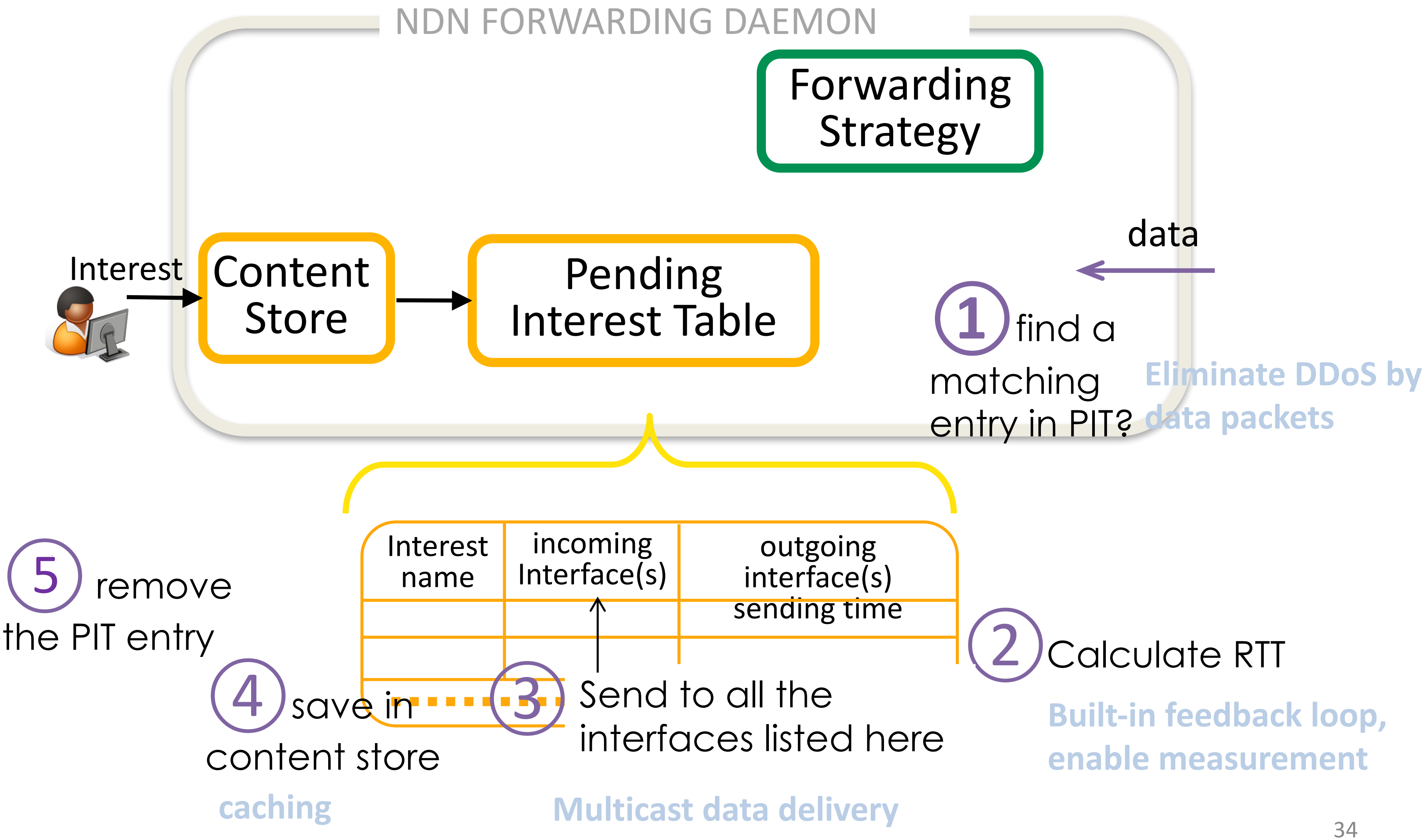


For each PIT entry:

Interest name	incoming Interface(s)	outgoing interface(s) sending time

May forward an interest packet out through one or more interfaces

NDN Data Packet Return



NDN FORWARDING DAEMON

Forwarding Strategy

Interest

Content Store

Pending Interest Table

data

1 find a matching entry in PIT?

Eliminate DDoS by data packets

Interest name	incoming Interface(s)	outgoing interface(s) sending time

2 Calculate RTT

Built-in feedback loop, enable measurement

3 Send to all the interfaces listed here

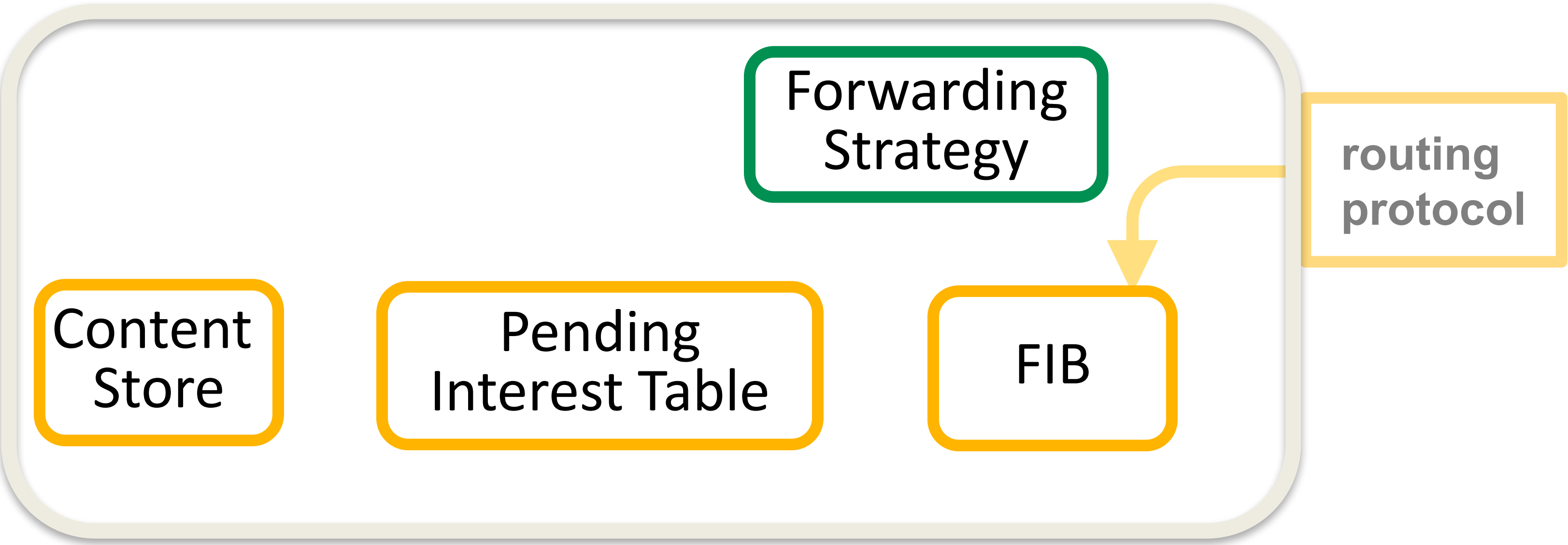
Multicast data delivery

4 save in content store

caching

5 remove the PIT entry

Forwarding Strategy



- Forwarding Strategy makes interest forwarding decisions by taking input from
  - FIB
  - measurement from Interest-data exchange (and any other local resource information)
  - *Per-namespace forwarding policies*

NDN’s secret sauce: It’s the *names*!

- NDN treats everything as named, secured bags of bits
- which can be fetched by its name, resiliently
  - – assisted by NDN’s stateful forwarding plane

How does a consumer learn about names?

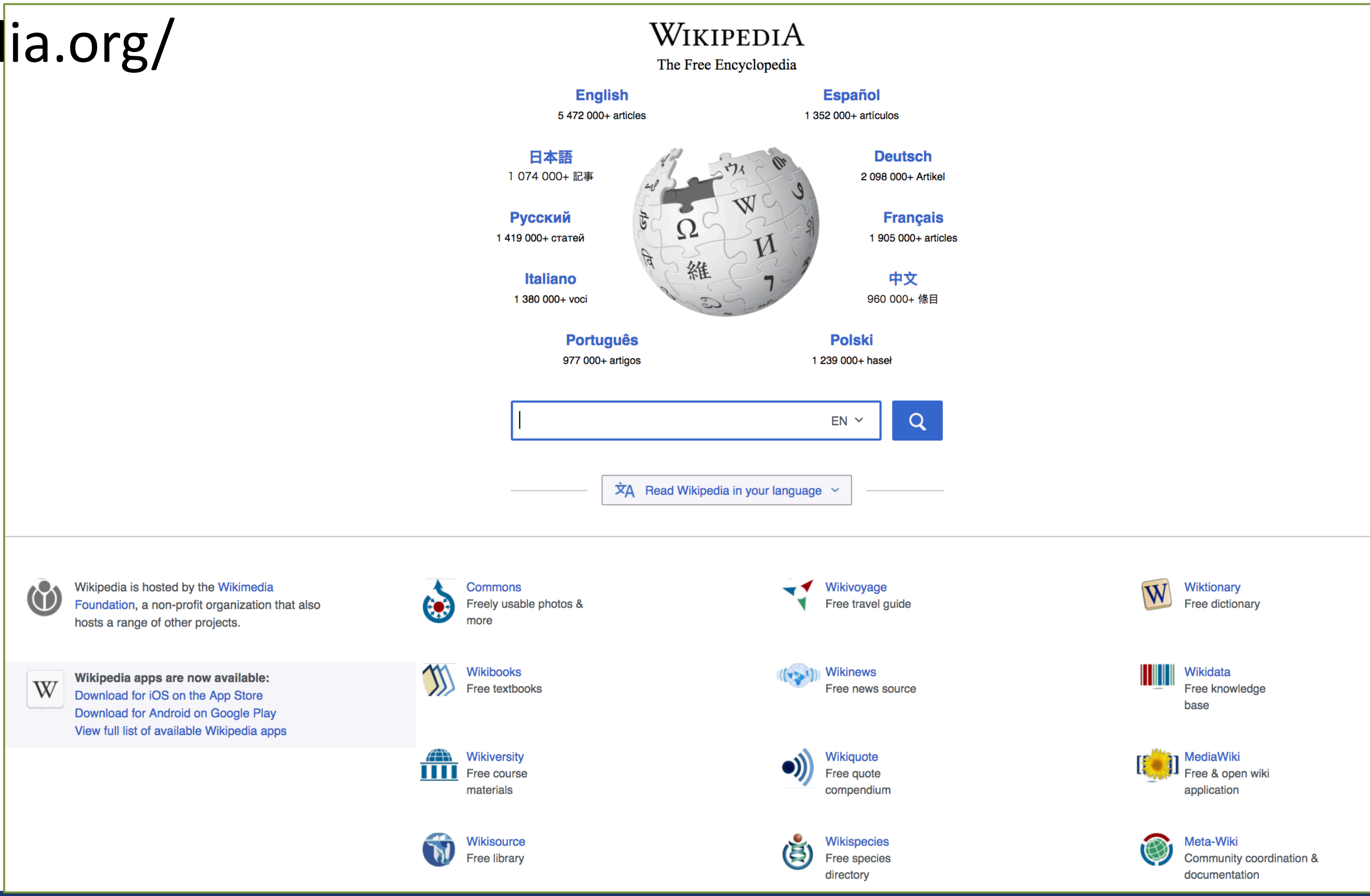
- Defining naming conventions
  - Well known rules about how to construct the name for desired data
- Use of metadata
- Dynamic name discovery
  - Several approaches

Naming convention: a few examples

- Naming conventions are widely used *today*
  - www.google.com, www.ucla.edu, www.cs.ucla.edu
  - mail.google.com, mail.ucla.edu, mail.cs.ucla.edu
- Two examples we saw earlier:
  - Suffix /.../_V10/_S1
  - Prefix /discover
    - Informs forwarding strategy how to handle interests packets with this prefix

Name Discovery by Metadata

https://wikipedia.org/



WIKIPEDIA
The Free Encyclopedia

English
5 472 000+ articles

Español
1 352 000+ artículos

日本語
1 074 000+ 記事

Deutsch
2 098 000+ Artikel

Русский
1 419 000+ статей

Français
1 905 000+ articles

Italiano
1 380 000+ voci

中文
960 000+ 條目

Português
977 000+ artigos

Polski
1 239 000+ haseł

Wikipedia is hosted by the Wikimedia Foundation, a non-profit organization that also hosts a range of other projects.

Wikipedia apps are now available:
Download for iOS on the App Store
Download for Android on Google Play
View full list of available Wikipedia apps

Commons
Freely usable photos & more

Wikivoyage
Free travel guide

Wiktionary
Free dictionary

Wikibooks
Free textbooks

Wikinews
Free news source

Wikidata
Free knowledge base

Wikiversity
Free course materials

Wikiquote
Free quote compendium

MediaWiki
Free & open wiki application

Wikisource
Free library

Wikispecies
Free species directory

Meta-Wiki
Community coordination & documentation

Name Discovery by Metadata

- Request: /wikipedia.org
- Reply: /wikipedia.org/index.html, containing URLs (names) to multiple pieces of content

Name Discovery: another example

- In-network name discovery via Longest prefix match

e.g. SIGCOMM 2017 streaming

– Interest name (a prefix): /acm/sigcomm17/video

– Reply Data name: /acm/sigcomm17/video/monday/_f45/_s23

naming convention defined by the application

- Discovering dynamically created contents from producers unknown a prior
  - Broadcast Interests (using naming conventions)
    - e.g. the earlier example: ?/discovery
  - Interests and Data packets meet at an established rendezvous point
    - i.e. using a well-known rendezvous name
    - Flavors of “pub-sub”

Naming convention & name discovery: active research area

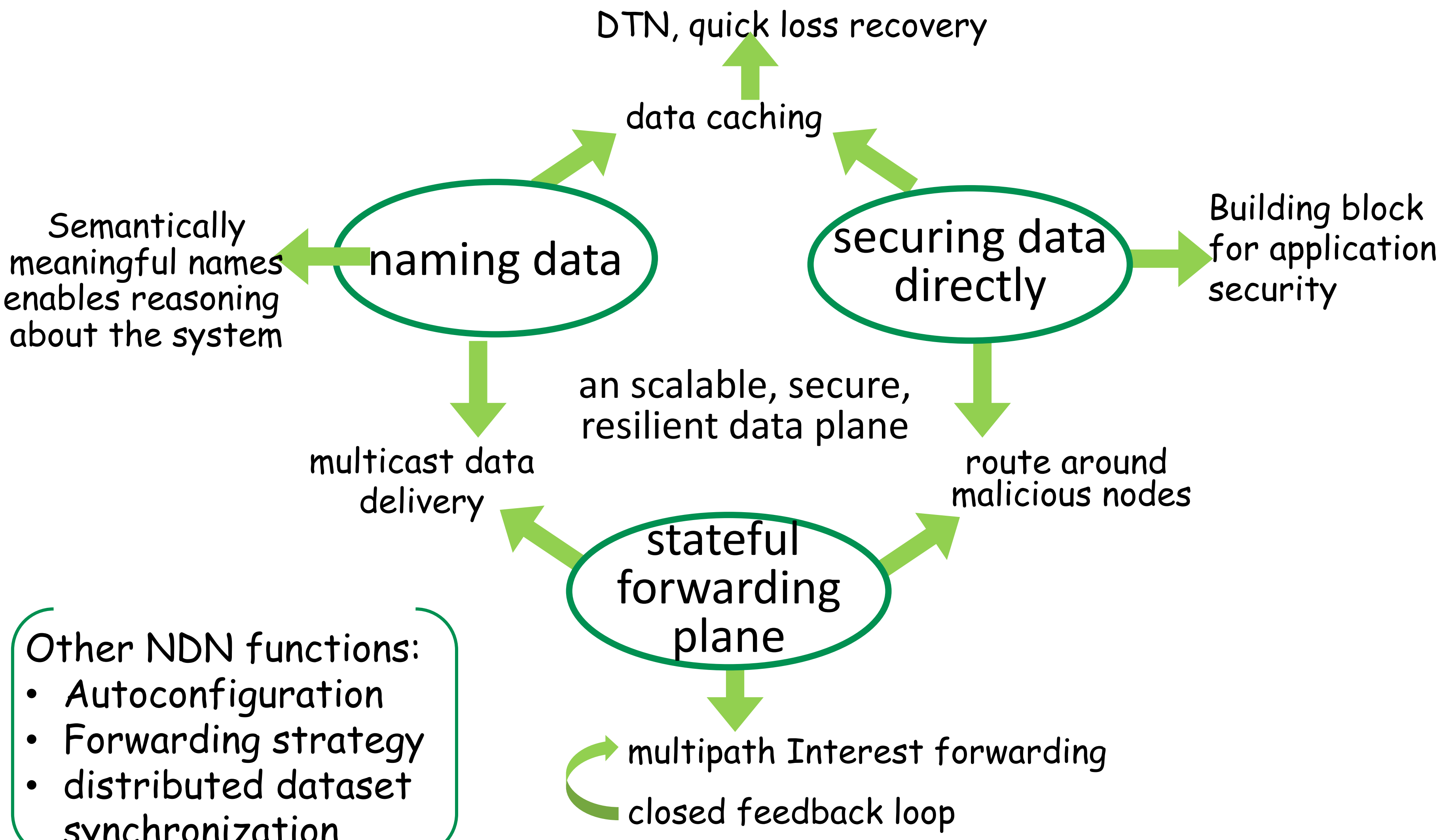
NDN and End-to-End Principle

- NDN Producer signs data right after production
  - cryptographically binds the (semantically meaningful) name to the content
  - Encrypt as needed
- Consumers verify/decrypt
  - Eliminating security dependency on lower layers or intermediaries
- NDN enables true *end-to-end* security, despite
  - scalability challenges (no single server can handle all users)
  - connectivity challenges (ad hoc, intermittent, data muling)

Data Packet

Name
Content
Signature

A Quick Summary on NDN: 3 simple ideas



DTN, quick loss recovery

data caching

Semantically meaningful names enables reasoning about the system

naming data

securing data directly

Building block for application security

an scalable, secure, resilient data plane

multicast data delivery

route around malicious nodes

stateful forwarding plane

multipath Interest forwarding

closed feedback loop

Other NDN functions:

- Autoconfiguration
- Forwarding strategy
- distributed dataset synchronization

Explore NDN as a new protocol architecture to evolve the Internet into next 40 years

For more information

http://www.named-data.net

https://named-data.net/publications/tutorials/