

# Named Data Networking

Introduction and hands on tutorial

by

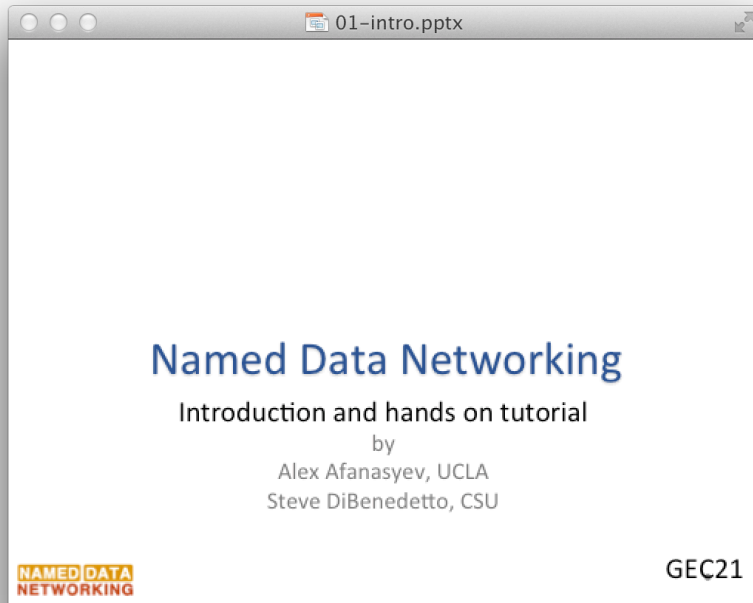
Alex Afanasyev, UCLA

Steve DiBenedetto, CSU

# Goals for today

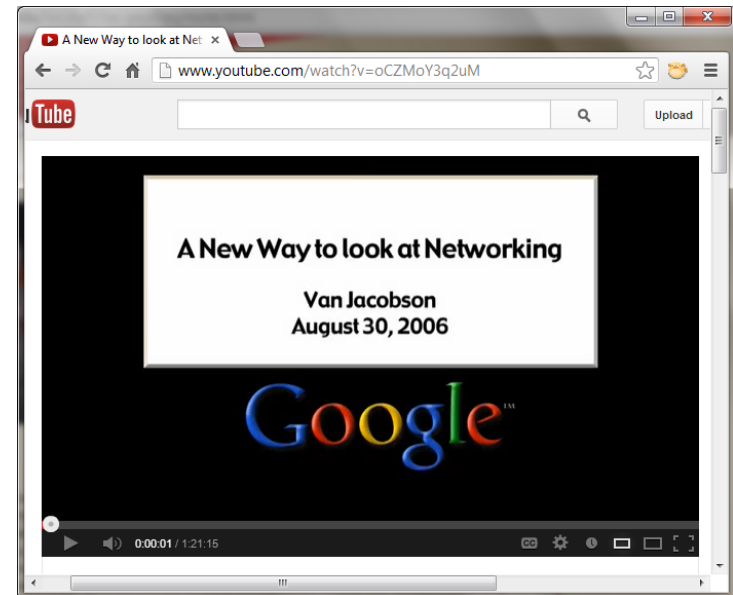
- Introduce Named Data Networking (NDN)
- Describe the project and its goals
- Illustrate NDN concepts
- Show how to write simple applications and how to experiment with NFD forwarder

# Share This Presentation?



What is the best way for me to share these slides with you right now?

What about video?  
What would happen if it became popular?



# Sending This Message?

From: C. D. (Dan) Mote, Jr. <dmote@email.edu>  
Date: Mon, May 13, 2013 at 7:39 PM  
Subject: Congratulations!  
To: Alex Afanasyev <alex@email.edu>

Dear Dr. Afanasyev,

I write to inform you that you have been elected a Fellow to the National Academy of Engineering. As you may understand, this designation follows a process of nomination and subsequent vote by existing Fellows. Congratulations.

Sincerely,  
C.D. Mote, Jr.  
President-Elect, National Academy of Engineering

# Use Connected Environment/IoT?



# Challenges Caused By a Single Problem



## Telephony/Internet Process

1. Find the **number/address** for the one you want to talk to.
2. Use that number to establish a **point-to-point connection**.
3. **Communicate!**

**Sharing**

*Must know address*

**Trust**

*Place all trust in address*

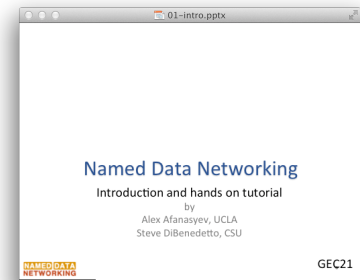
**IoT**

*Know & trust all addresses*

# A Simpler Way

Suppose your device could ask for what it wanted?

**`/this_room/alex/talks/GEC21.pptx`**



**`/youtube.com/video/ndn/van2006`**



**`/ucla/boelter_hall/4th_flor/room412/thermostat/1/status`**



# The Web Has Named World's Data!

<http://www.youtube.com/watch?v=oCZMoY3q2uM>



[http://www.youtube.com/watch?feature=player\\_detailpage&v=oCZMoY3q2uM#t=1736s](http://www.youtube.com/watch?feature=player_detailpage&v=oCZMoY3q2uM#t=1736s)

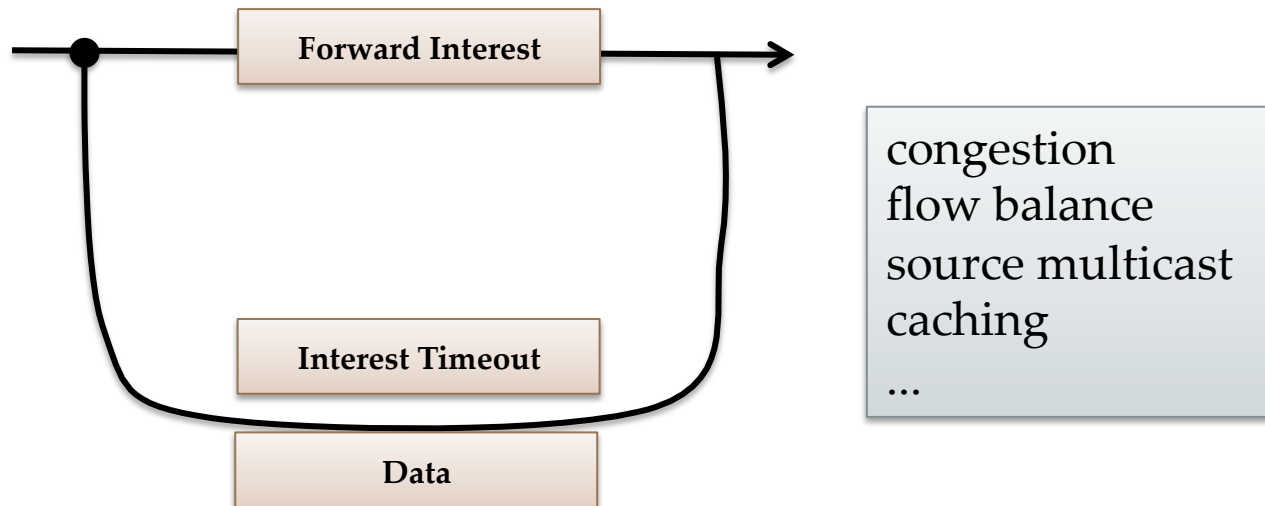


# Core Idea

Use names directly at the networking level

Focus on data, not host-to-host connections

Closed-loop communication



# Named Data Networking

- Leverages the strengths of the Internet, addresses weaknesses
  - Layers efficiently atop Ethernet, UDP, TCP, ...
- Naturally accommodates
  - Mobile devices
  - Wireless and other broadcast-based link types
  - Data authentication and security, privacy, anonymity
  - Policy-based forwarding, routing with loops
- With NDN, we aim to show that
  - Communication is more secure
  - Infrastructure is more efficiently utilized
  - Applications are simpler

# NDN Project

- Project launch:  
9/2010, part of NSF  
FIA Program  
5/2014, part of NSF  
FIA-NP Program
- Research Areas:  
Architecture, Routing,  
Security,  
**Applications**, Scalable  
Forwarding

UCLA: Van Jacobson, Jeff Burke, Deborah Estrin, **Lixia Zhang**

University of Arizona: Beichuan Zhang

University of California, San Diego: Kim Claffy, Dmitri Krioukov

Colorado State University: Christos Papadopoulos

University of Illinois, Urbana-Champaign: Tarek Abdelzaher

University of Memphis: Lan Wang

University of Michigan: Alex Halderman

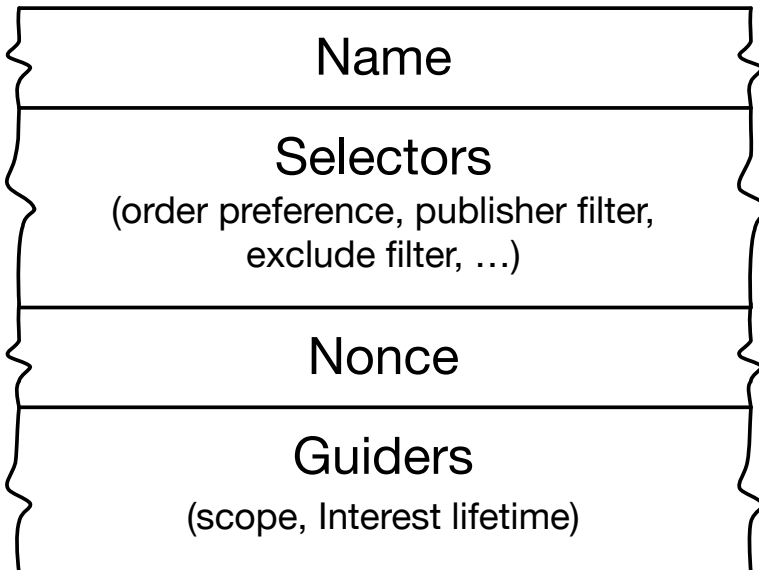
Washington University: Patrick Crowley

Northeastern University: Edmund Yeh

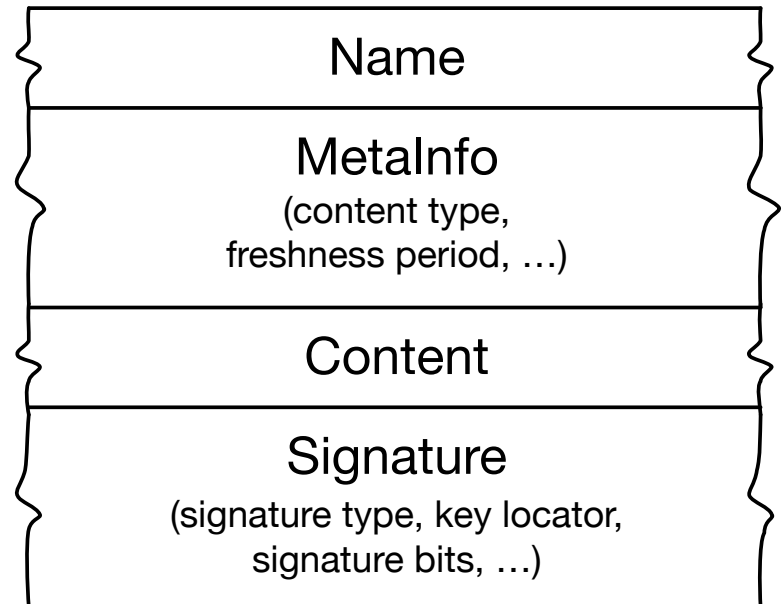
University of Maryland: Katie Shilton

# Two Packet Types

## Interest Packet



## Data Packet



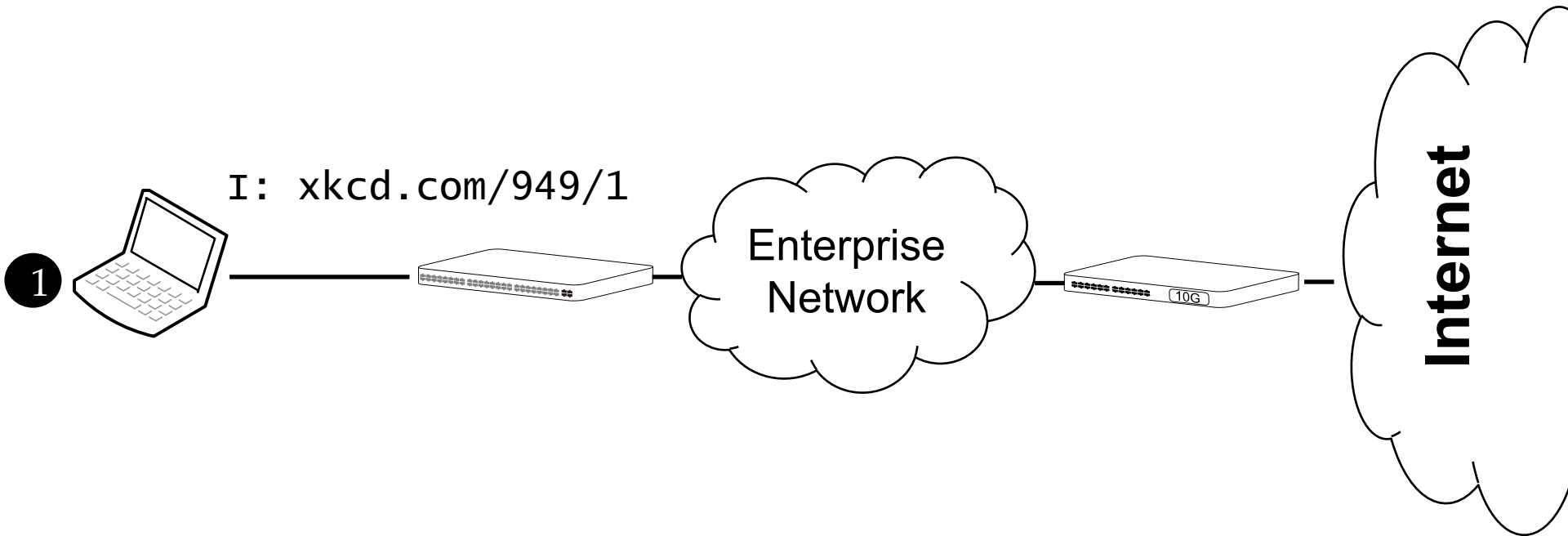
- No addresses
- **Publishers bind names to data; receivers verify**

# NDN Interest Forwarding

1. Do I have this data?
2. Is a request already pending?
3. Which next hop might lead to the source?

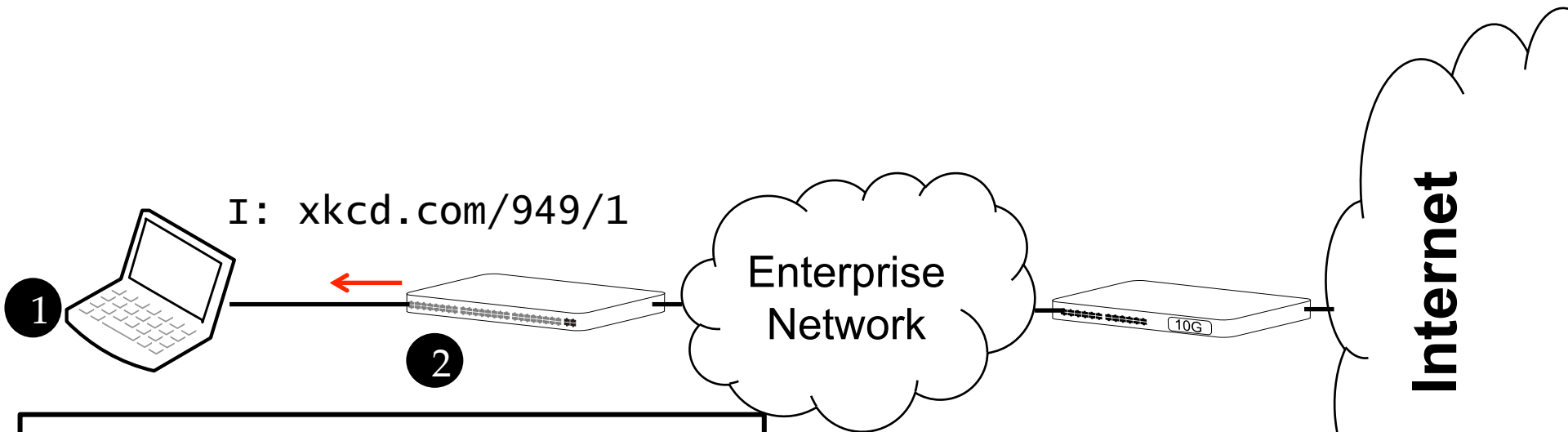
# NDN Forwarding Illustrated

1 Emit Interest: [xkcd.com/949/1](http://xkcd.com/949/1)



# NDN Forwarding Illustrated

- 1 Emit Interest: `xkcd.com/949/1`
- 2 Interest arrives at switch

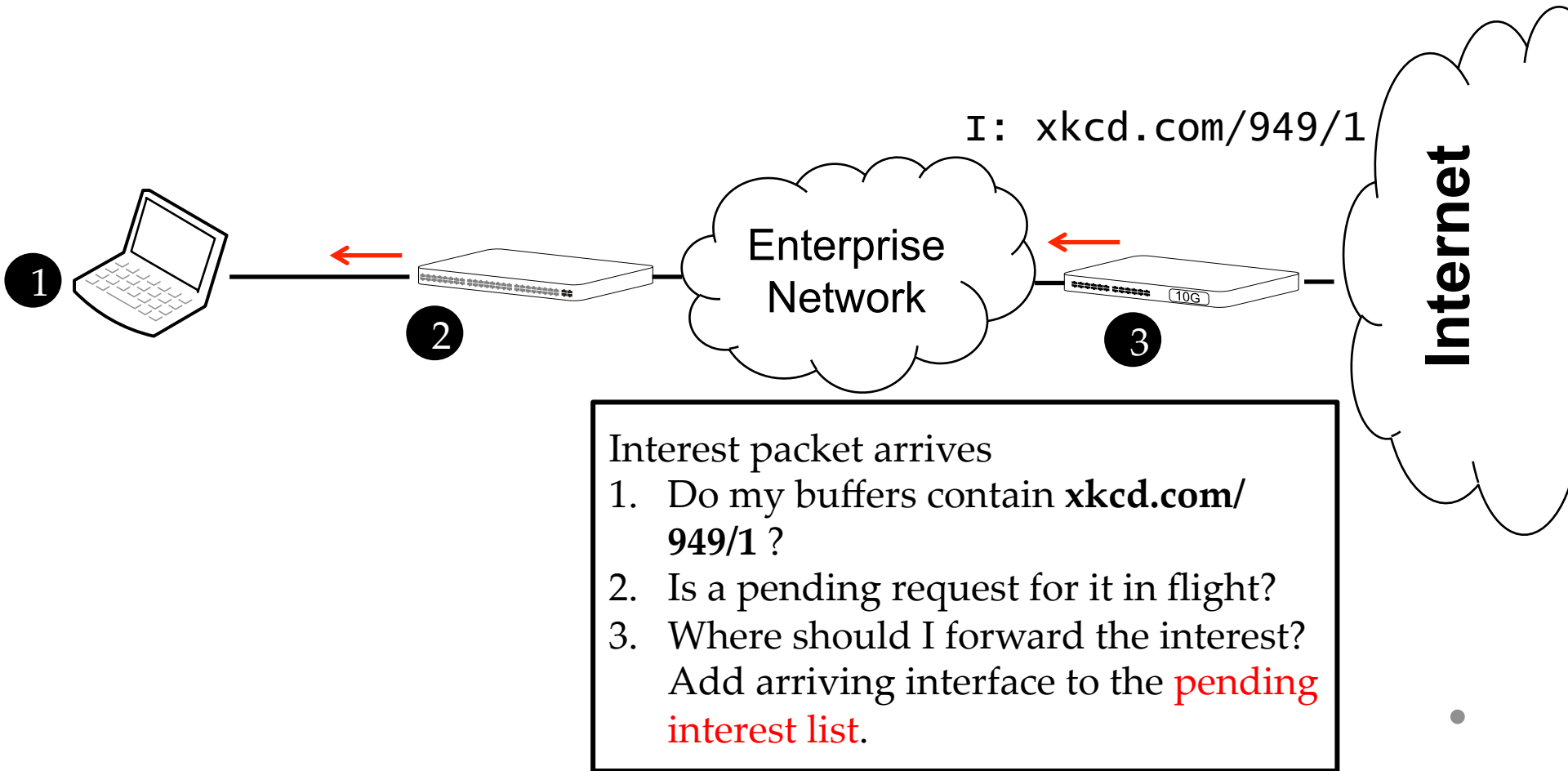


Interest packet arrives

1. Do my buffers contain `xkcd.com/949/1` ?
2. Is a pending request for it in flight?
3. Where should I forward the interest?  
Add arriving interface to the **pending interest list**.

# NDN Forwarding Illustrated

- 1 Emit Interest: xkcd.com/949/1
- 2 Interest arrives at switch
- 3 Interest arrives at gateway



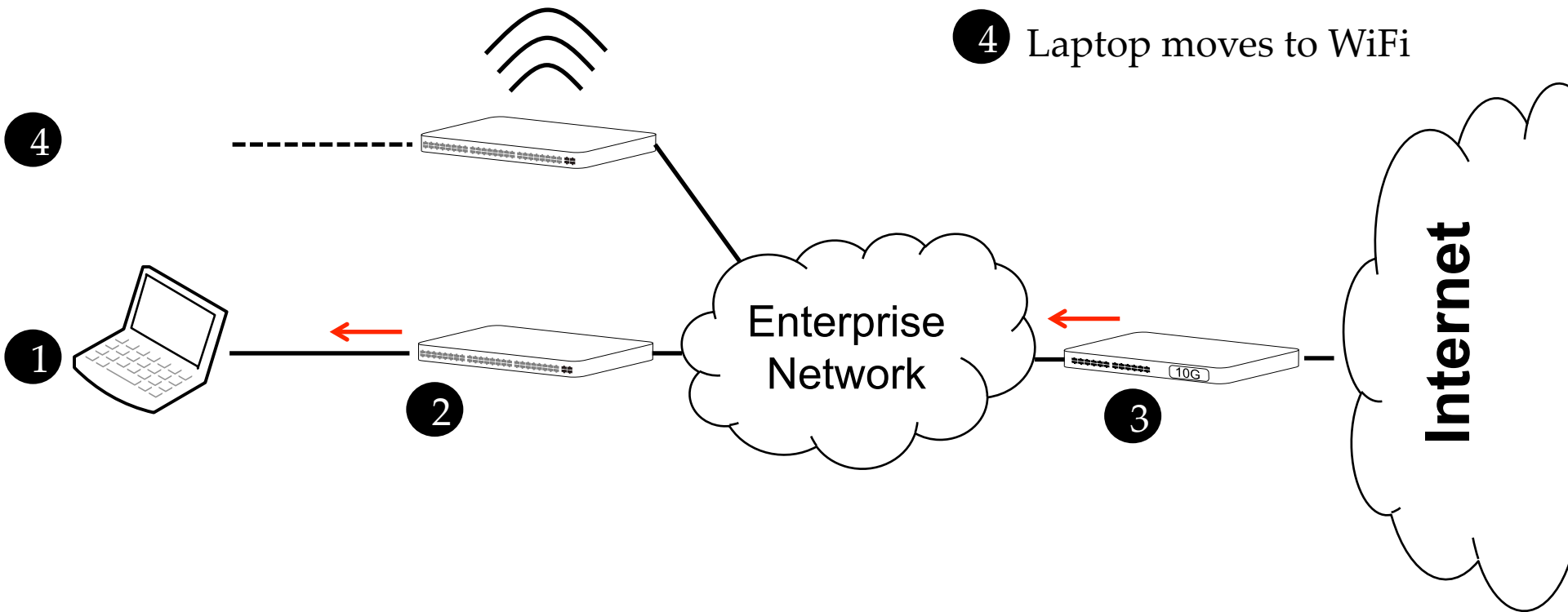
Interest packet arrives

1. Do my buffers contain `xkcd.com/949/1` ?
2. Is a pending request for it in flight?
3. Where should I forward the interest?  
Add arriving interface to the **pending interest list**.

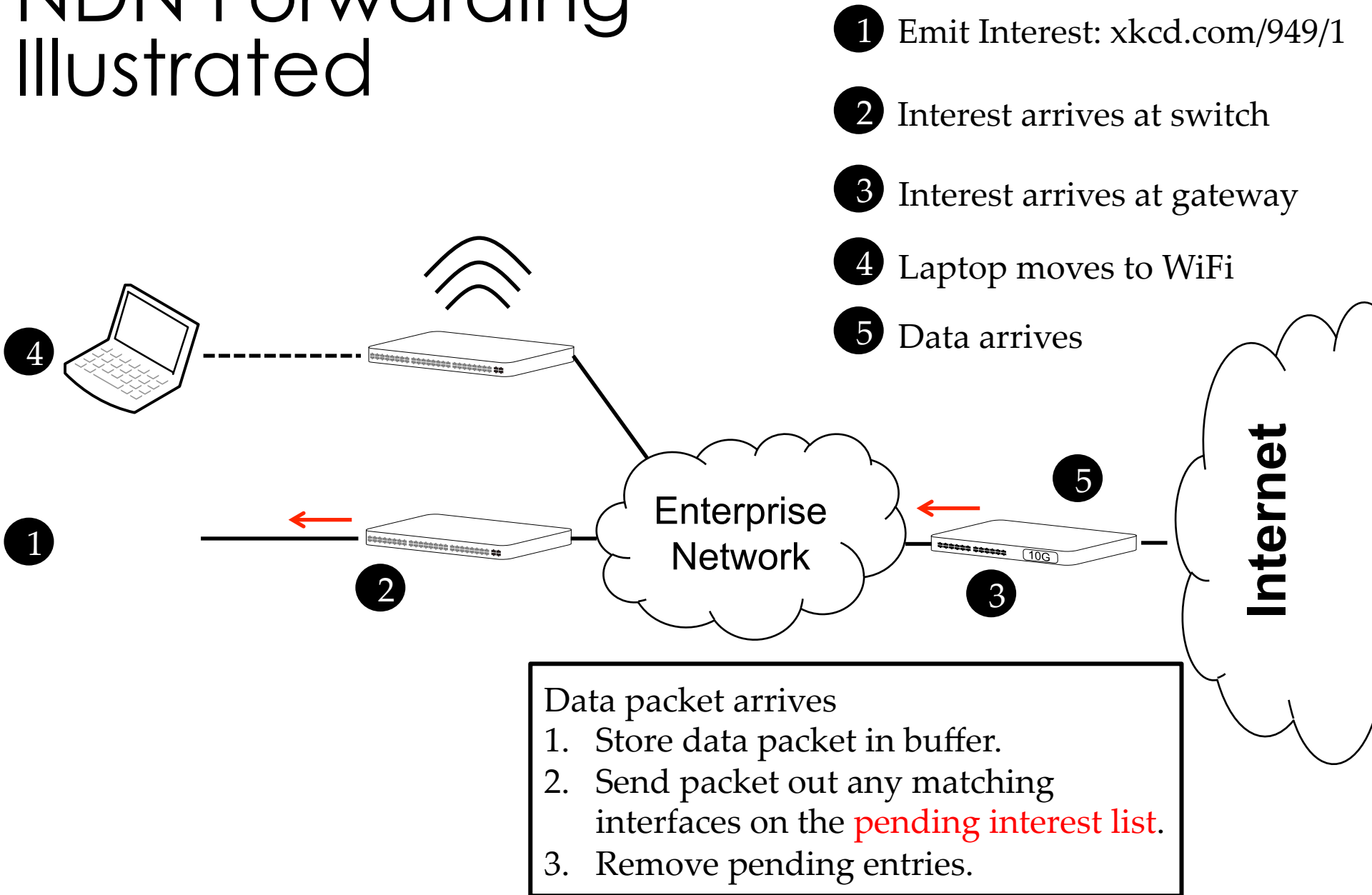


# NDN Forwarding Illustrated

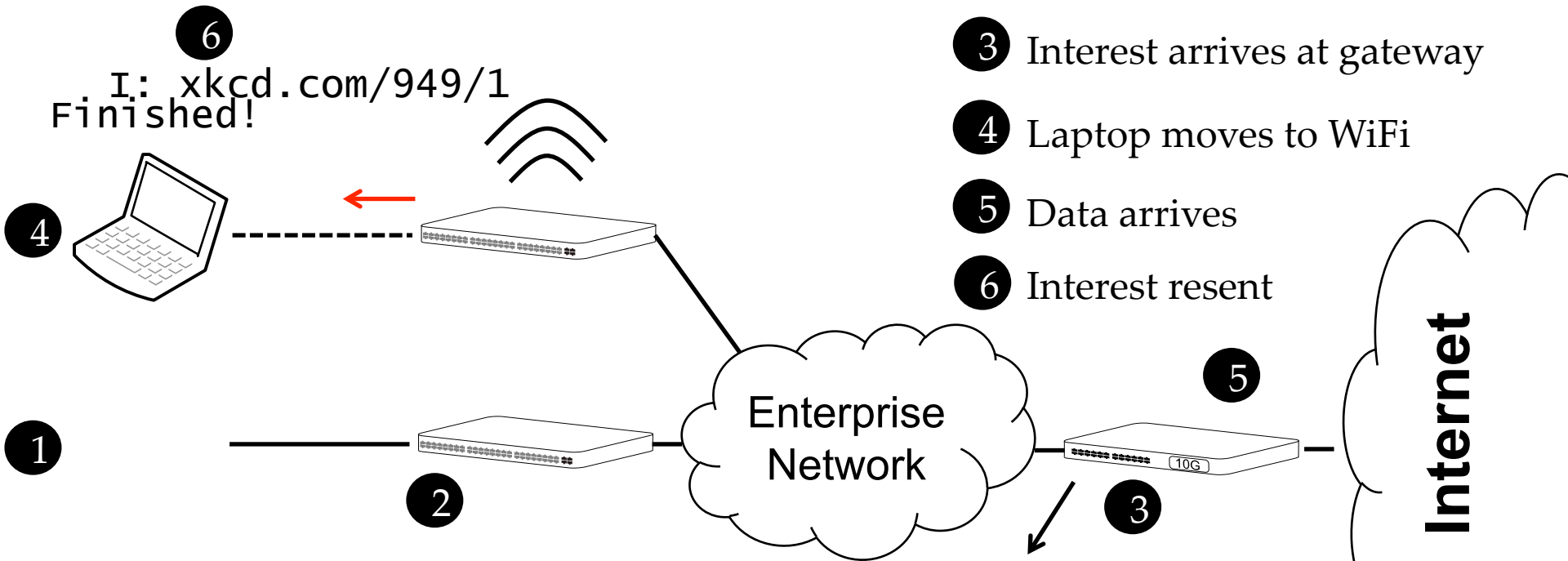
- 1 Emit Interest: `xkcd.com/949/1`
- 2 Interest arrives at switch
- 3 Interest arrives at gateway
- 4 Laptop moves to WiFi



# NDN Forwarding Illustrated



# NDN Forwarding Illustrated



- 1 Emit Interest: xkcd.com/949/1
- 2 Interest arrives at switch
- 3 Interest arrives at gateway
- 4 Laptop moves to WiFi
- 5 Data arrives
- 6 Interest resent

Interest packet arrives

1. Do my buffers contain **xkcd.com/949/1**? **Yes, send it.**
2. Is a pending request for it in flight?
3. Where should I forward the interest? Add arriving interface to the **pending interest list.**

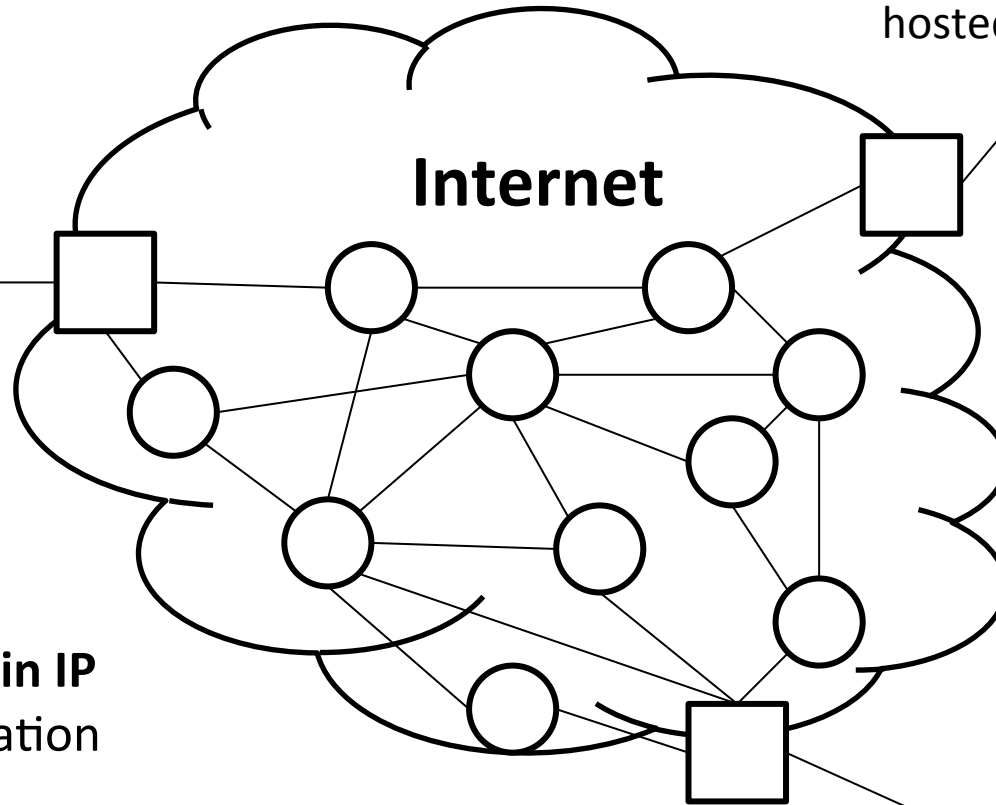
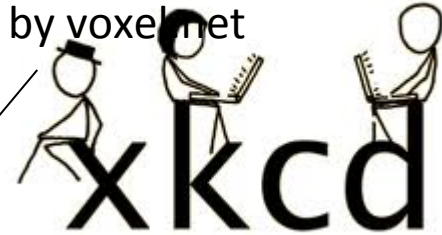
# IP Nodes and Routes

wustl.edu  
128.252.0.0/16



Washington  
University  
in St. Louis

xkcd.com  
72.26.192.0/19  
hosted by voxelnet



## Forwarding logic in IP

1. Extract destination address
2. Find longest matching prefix in route table
3. Forward packet out matching interface

google.com  
74.125.0.0/16



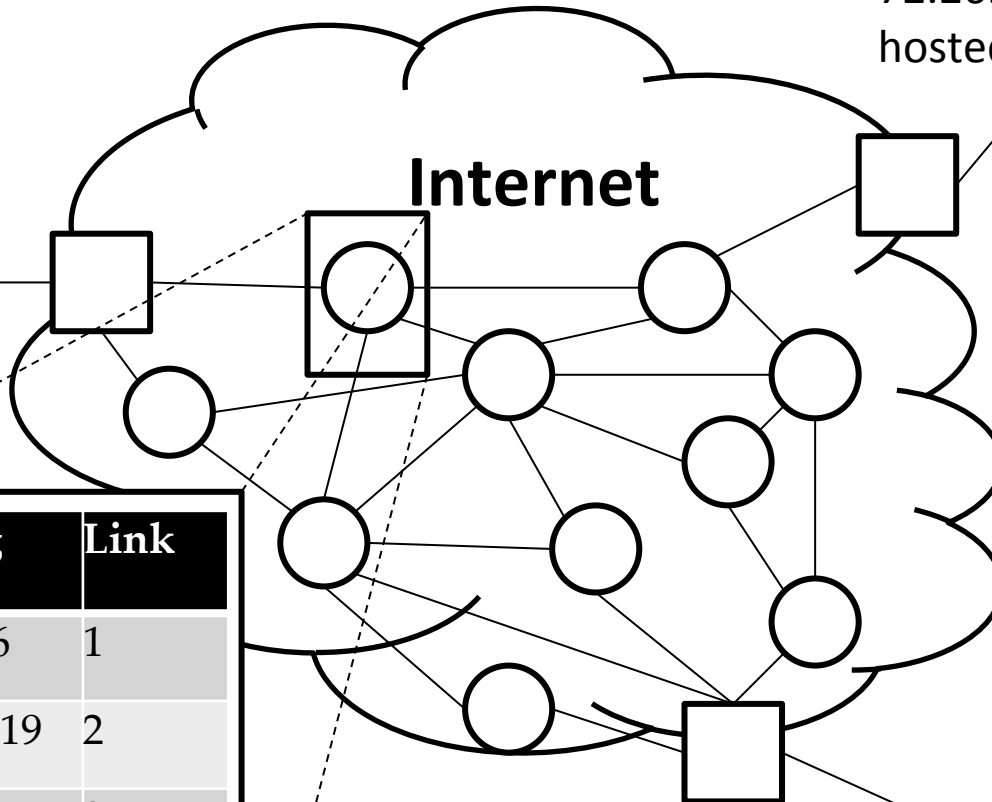
# IP Nodes and Routes

wustl.edu  
128.252.0.0/16



Washington  
University  
in St. Louis

xkcd.com  
72.26.192.0/19  
hosted by voxel.net



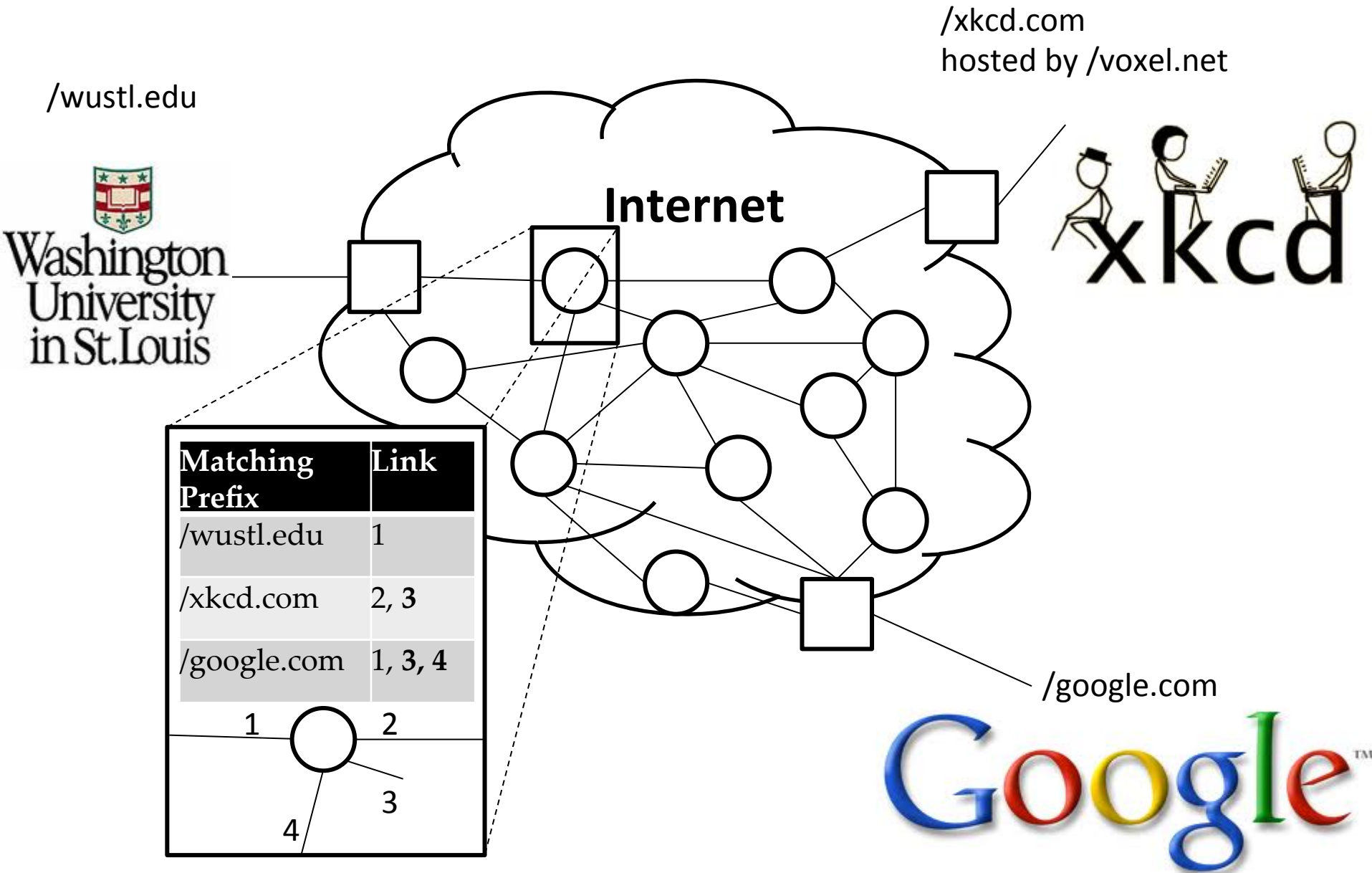
Matching Prefix	Link
128.252/16	1
72.26.192/19	2
74.125/16	3

1 2  
3 4

google.com  
74.125.0.0/16



# NDN Nodes and Routes



# Questions

---

- Can NDN efficiently support host-to-host patterns?
- Can NDN efficiently support user-specific data and services?
- Can you count clicks and ad impressions in NDN?
- Can you efficiently route all those names?
- Can you scale the forwarding plane?
- Can you prove security and privacy properties?

**Yes!**

**Yes, mostly!**

# Conclusion (1/2)

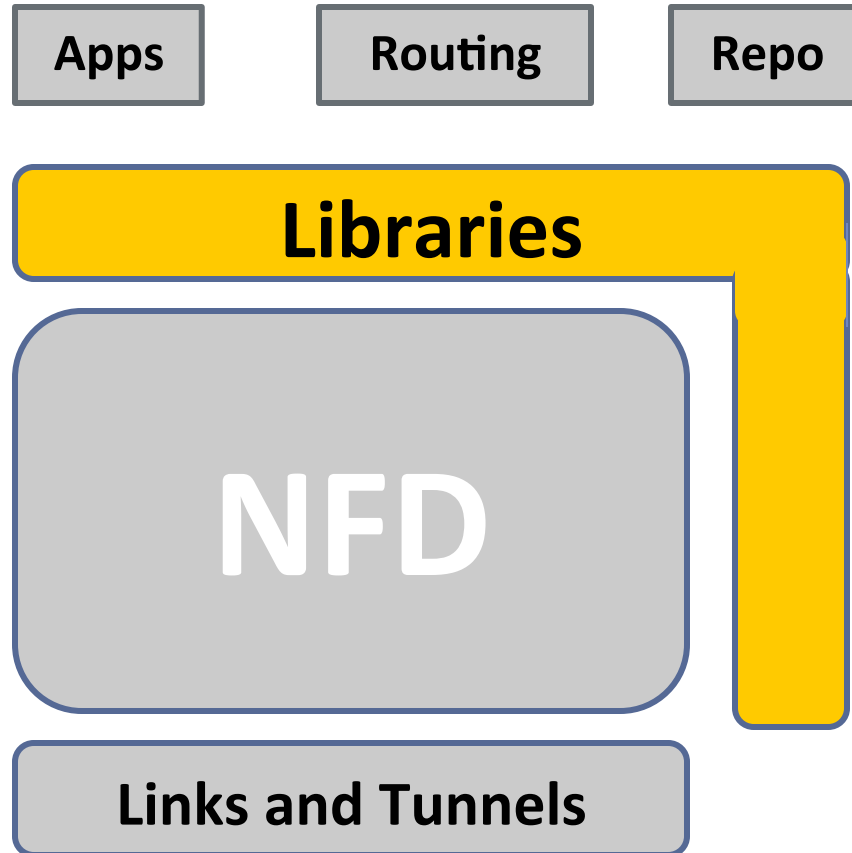
- Growing evidence that with NDN
  - Communication is more secure
  - Infrastructure is more efficiently utilized
  - Applications are simpler
  - New things are possible



## Conclusion (2/2)

- In coming years
  - Growing commercial interest and experimentation
  - Deployments in greenfields / IP trouble spots
    - IoT, building automation, healthcare, vehicular
- Research community is growing
  - We share an open-source code base with related projects and groups moving forwarding in Europe and Asia
  - NDN Consortium, launched this month, already includes 14 universities, 5 for-profit corporations, and 1 non-profit.

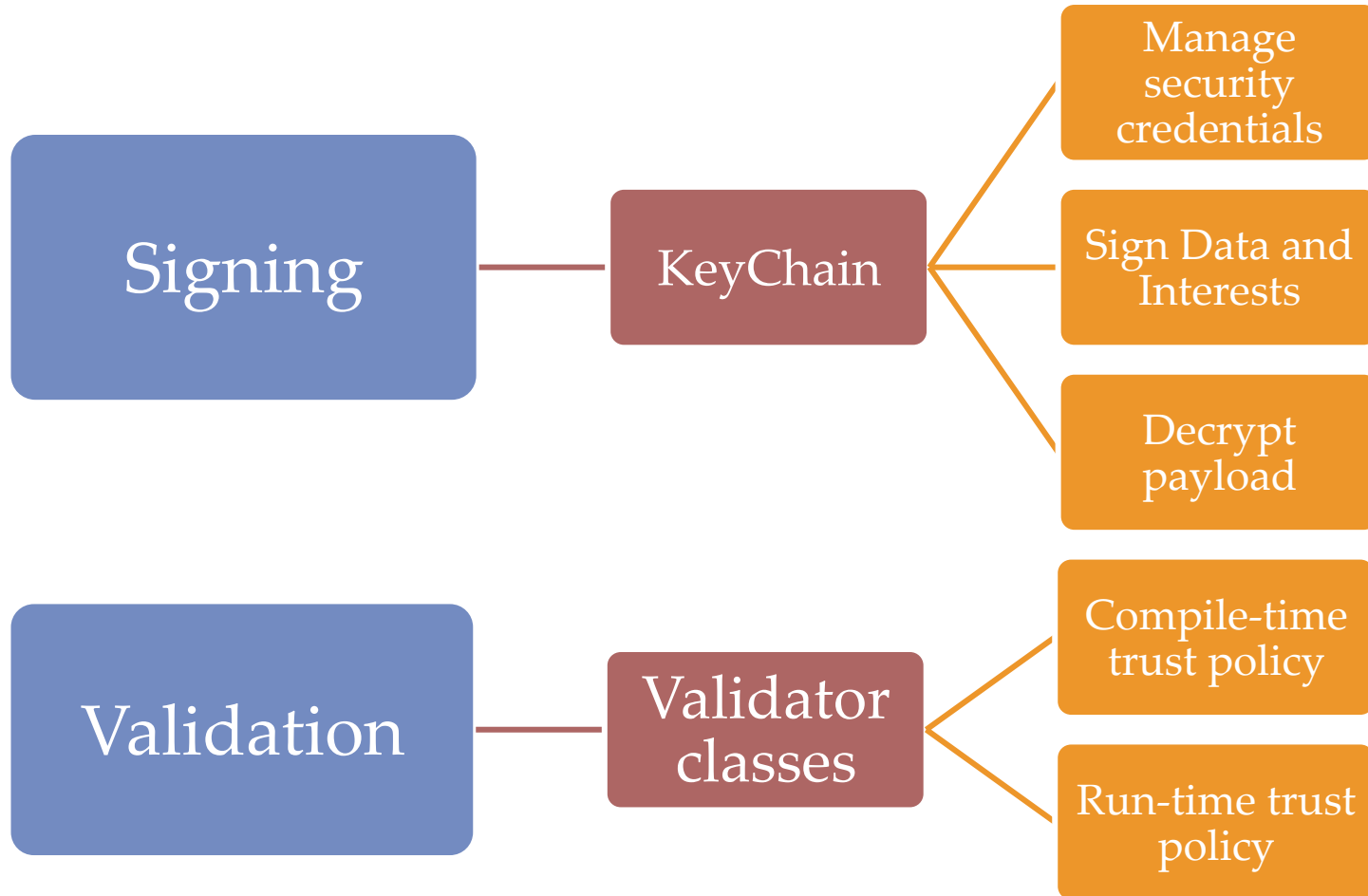
# NDN Components



# NDN Libraries

- All libraries now reflect fundamental architectural abstractions directly in objects, and wire format manipulation is abstracted.
  - Name, Component
  - Interest, Selectors
  - Data, MetaInfo, SignatureInfo, SignatureValue, KeyLocator
  - Face
  - KeyChain, Validator
  
- Multiple library efforts
  - NDN-CXX: “C++ for eXtended eXperimentation”
    - C++ (soon to be C++11), Boost (Asio, Filesystem, ...)
  - NDN-CCL: “Common Client Libraries”
    - C++
    - Python (2 and 3)
    - JavaScript (browser and node.js)
    - Java
  - Enables diversity of coding choice
  - Drives us towards specification (and not just implementation)

# Security Support



<http://named-data.net/doc/ndn-cxx/0.2.0/tutorials/security-library.html#signing>

# Supported Security Features

- Asymmetric cryptography
  - RSA
  - ECDSA
- Symmetric cryptography\*
  - AES
  - HMAC
- Trivial cryptography
  - SHA256 digest
- Signing/verification granularity
  - Data packet
  - Set of Data packets\*

\* work in progress

# NDN Platform

- **Provide a coherent, usable, and well-documented “platform” for exploring NDN in practical applications – for the NDN project team and external users.**
- Use a release “heartbeat” to stimulate interoperability testing and discussion of how the various moving parts work together.
- Along the way, improve access to and consistency of various NDN code projects.
- Open and lightweight process, with no unrealistic centralization or over-management but clear ownership of each component project.
- Managed nodes on the testbed run the Platform.

# NDN Platform 0.3 (August 2014)

- **NFD** NDN Forwarding Daemon, version 0.2.0 (1)
- **ndn-cxx** library, version 0.2.0
  - The NDN C++ library with eXperimental eXtensions (CXX)
  - The ndnsec security tools to manage security identities and certificates
- **NDN-CCL** - NDN Common Client libraries suite, version 0.3
  - NDN-CPP C++ / C library
  - PyNDN2 Python library
  - NDN-JS JavaScript library (with Node.js support)
  - jNDN Java library (preliminary)
- **NLSR** - Named Data Link State Routing Protocol , version 0.1.0
- **repo-ng** - next generation of NDN repository , version 0.1.0
- **ndn-tlv-ping** - ping application for NDN , version 0.2.0
- **ndn-traffic-generator** - traffic generator for NDN , version 0.2.0
- **ndndump** - packet capture and analysis tool for NDN , version 0.5
- Partial binary package support on Ubuntu, MacOS X, others...

# Community Outreach

- One public Github repo for all code
  - <http://github.com/named-data>
- Public Redmine with Wiki documentation for components
  - <http://redmine.named-data.net>
- Components website
  - NFD: <http://named-data.net/doc/NFD/>
  - ndn-cxx: <http://named-data.net/doc/ndn-cxx/>
  - NLSR: <http://named-data.net/doc/NLSR/>
  - NDN-CCL: <http://named-data.net/doc/NDN-CCL/>
- Code review
  - <http://gerrit.named-data.net>
- Technical reports and NDN technical memos
  - <http://named-data.net/publications/techreports/>
- Mailing lists
  - <http://named-data.net/codebase/platform/support/mailing-lists/>

*Open to contributors and collaborators!*



Ready for the Action?