Fast Forwarding for NDN

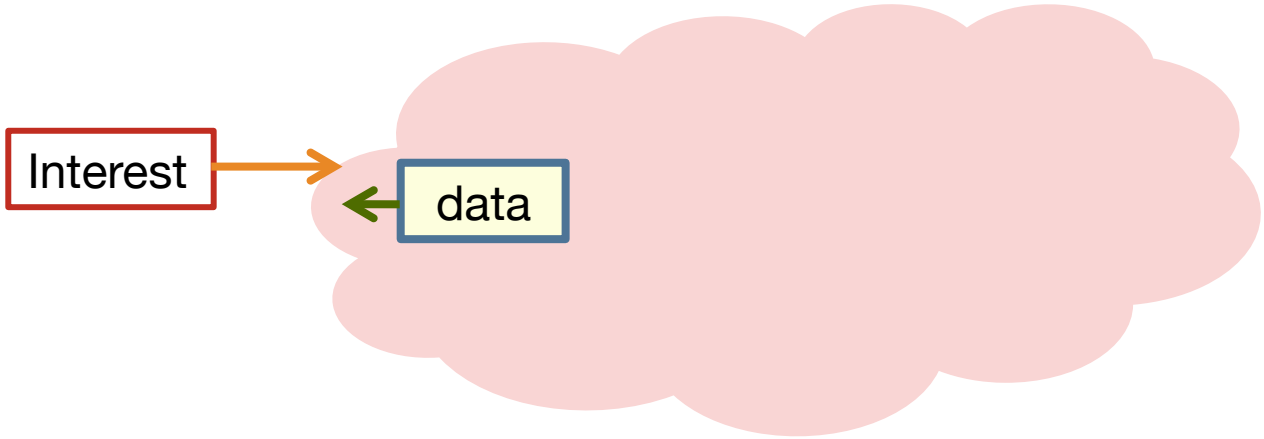
Won So
Ashok Narayanan
Mark Stapp
mjs@cisco.com

Cisco Systems

IETF ICNRG, 31/7/2013, Berlin

CCN/NDN in Two Slides (1)

- Network forwards and caches named data 'objects'; no host addresses
- Pull-based communication model
  - Interest message: Client asks for a content object by name
  - Data message: Any node having data responds with a content object
  - Client is able to validate data because the Content message is signed



NDN in Two Slides (2)

- Routers have some new components
- Content Store (CS) – cache used for local repair; may be persistent/non-volatile
- Pending-Interest Table (PIT)
  - Every Interest message creates state in the router
  - The PIT state is used to direct Content messages to clients
- FIB
  - Holds name prefixes; there are no “addresses”
  - Must be prepared to hold >1 entry per prefix
  - Natural support for multi-path forwarding and mcast

NDN in Two Slides (3)

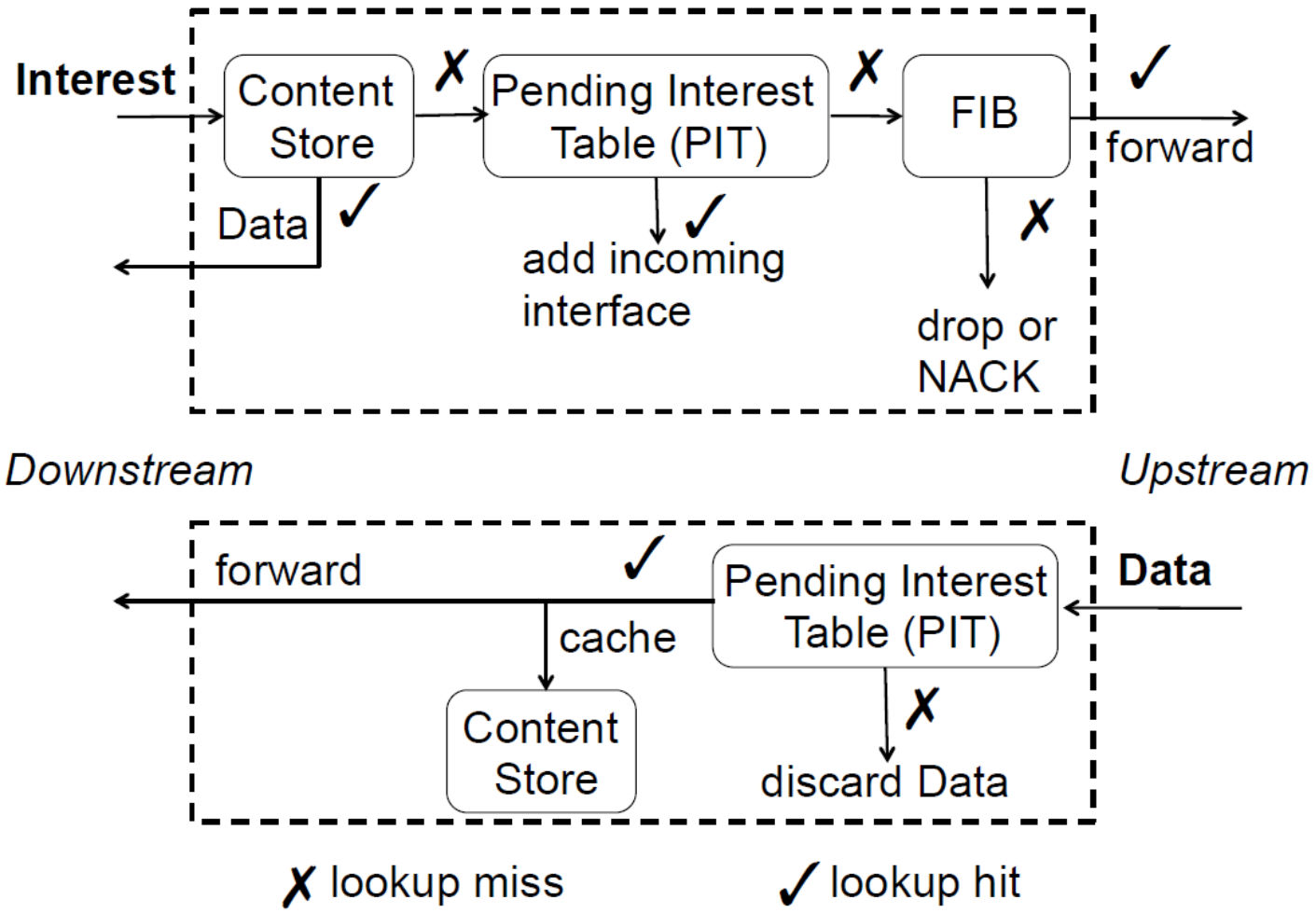


Figure source: Yi et al., Adaptive forwarding in named data networking, ACM SIGCOMM CCR, 42(3), 2012.

Hash-based Forwarding (1)

- Making line-rate decisions using variable-length names in messages is … an opportunity
- We're working on a hash-based approach
  - Using siphash (Aumasson and djb), a 'crypto' hash
  - Locate 'name' in the message
  - Make one pass through name
    - Produce complete hash of the name
    - Collect partial hash result at each name-component boundary
  - Hash of name is used to distribute work to a cpu core
  - CPU core owns a partition of the PIT
    - PIT changes do not incur MESI coherence penalty

Hash-based Forwarding (2)

- When messages arrive at a worker cpu core
  - Check the combined CS/PIT for the name hash
  - Return Content from cache if poss
  - De-dup Interests
  - Dupe Content messages if >1 Interest was received
  - Reject Content with no pending Interest
- Probe the FIB with some of the partial name prefix hashes
  - Start at a “likely” prefix
  - FIB entries tell us how many lookups to attempt
  - Relatively easy to reject name-component-count attacks
  - Locate longest match in FIB
  - Gather FIB re-write info and return message to original Dispatcher
- Dispatcher uses re-write info and puts outbound message back on the switch fabric

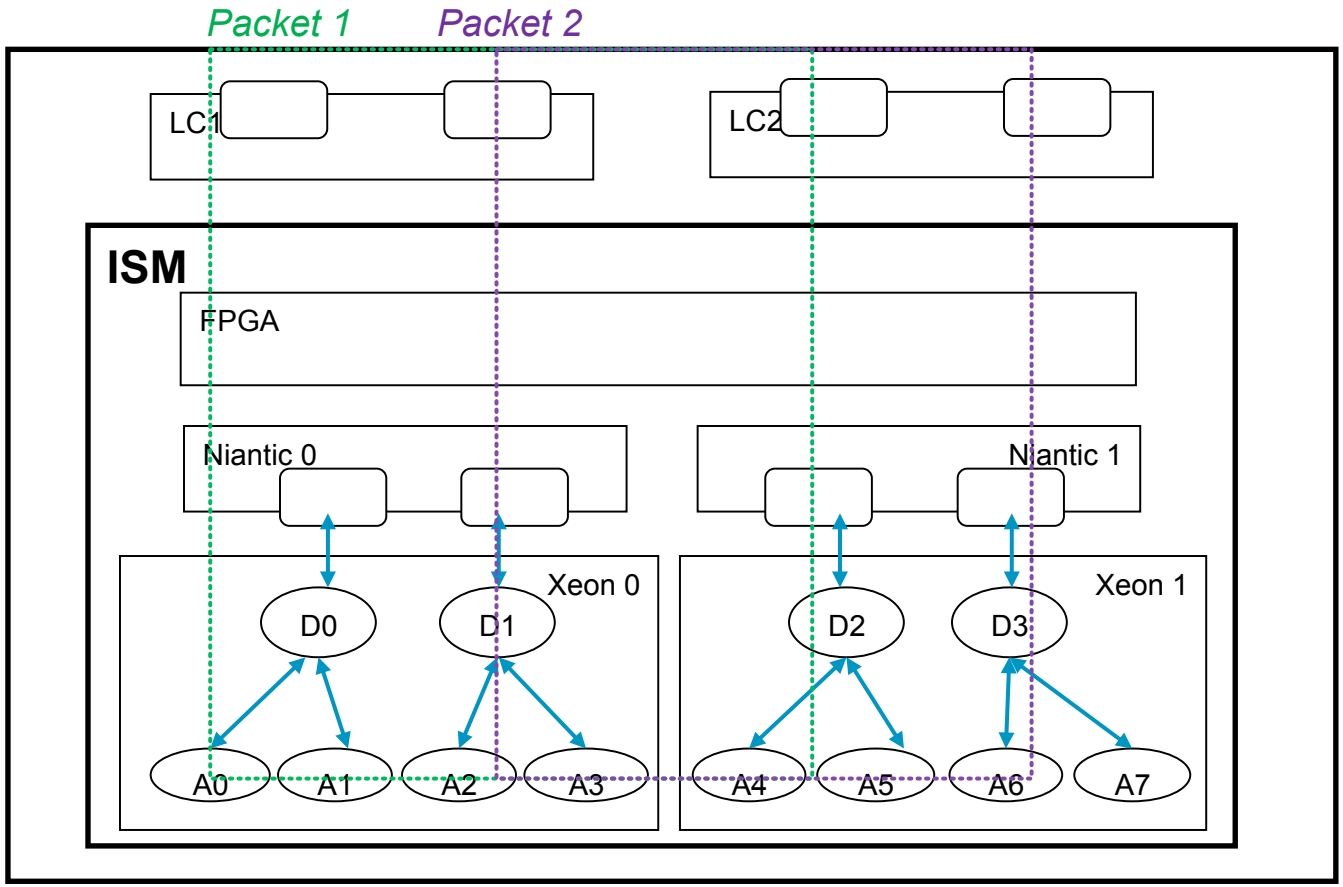
Hash-based Forwarding (3)

- A Real Router: Cisco ASR9000
- Integrated Service Module (ISM) blade
  - Linux
  - 4 x Intel 10G ports
  - 2 x Xeon CPUs
  - 6 cores per CPU package
  - 48GB SDRAM
  - 2 x 1.6 TB modular flash SSD

Hash-based Forwarding (4)

Flow through the ASR9000 ISM Blade:

D == Dispatcher user-space process
A == Worker 'App' user-space process



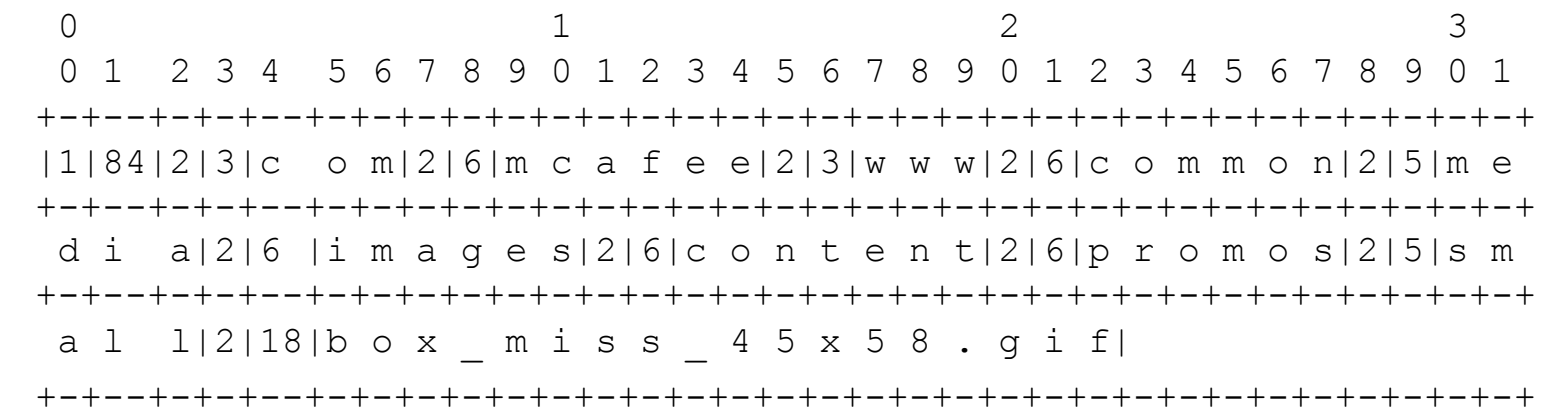
Input Dataset (1)

- IRCache traces to supply 13M +/- urls
  - Generate Interest Names from the urls
- URL blacklist to supply FIB prefix length profile
  - Generate FIB by applying length profile to selected input urls

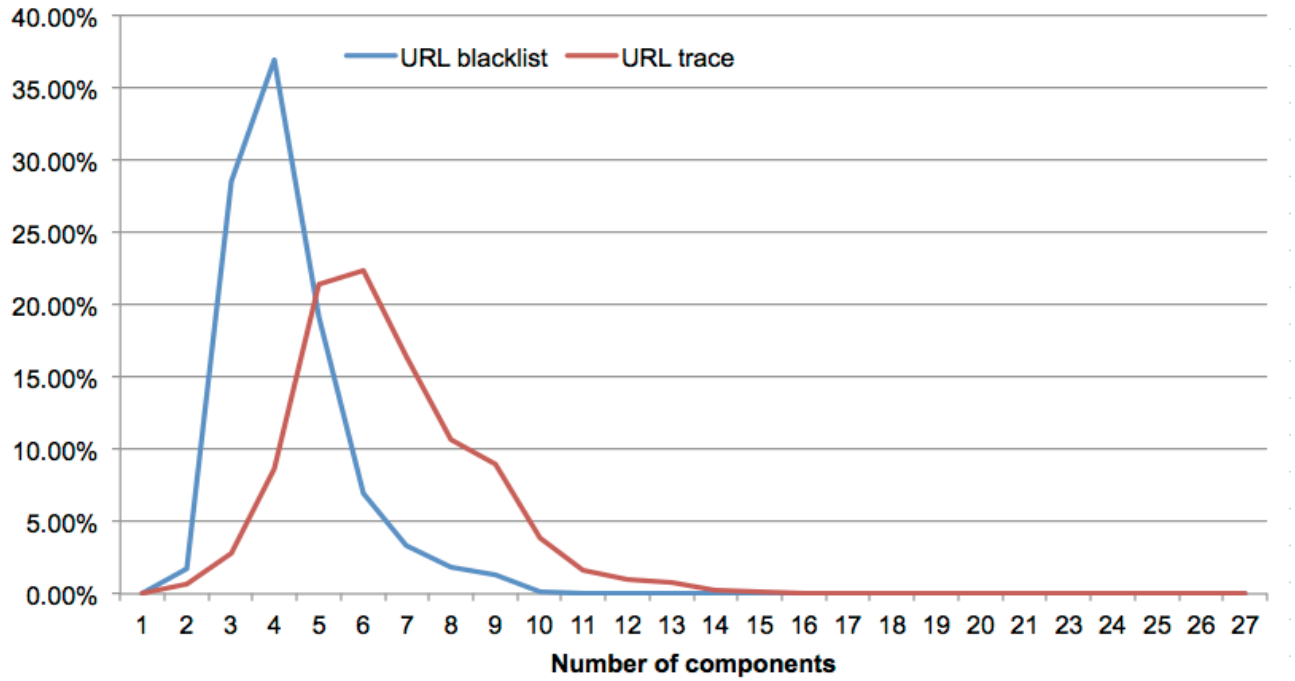
Input Dataset (2)

- Example transforming url to NDN name tlv:

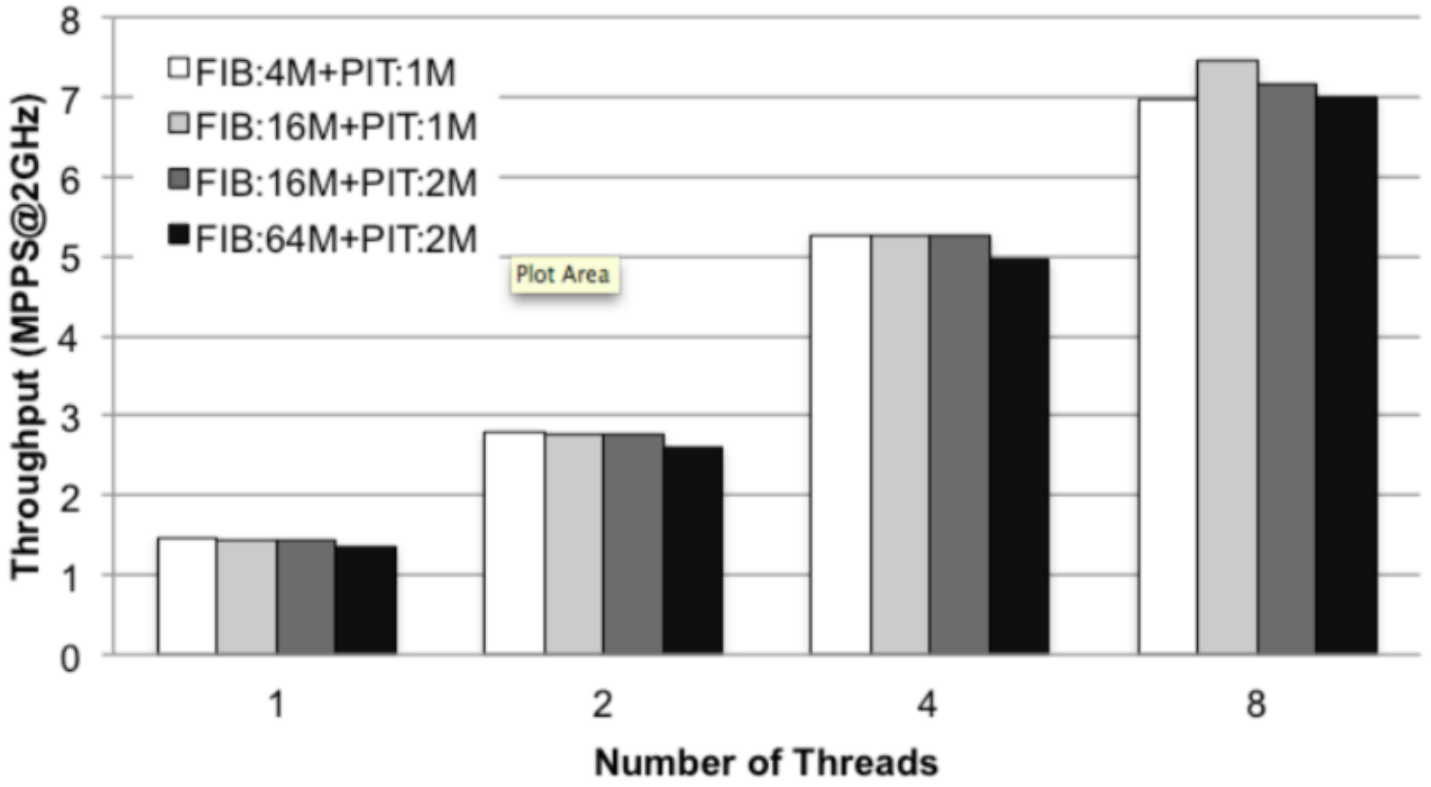
```
www.mcafee.com/common/media/images/content/promos/small/box_miss_45x58.gif
```



Input Dataset (3)



Performance Summary



- Reasonably linear multi-core scalability
- For ref, ~5.62MPPS required for 20Gbps with 425-byte (avg) packets

Caveats and TODOs

- Experimenting with the data structs and algorithms, primarily
- No CS
- No congestion-control (of significance)
- Very few drops – most messages' names are in the FIB
- No fragments (no 4K/8K Content messages)
- Naming patterns and traffic patterns we haven't tried yet
- No Content signature verification
- We haven't measured everything we want to measure just yet
- So I may not be able to answer every question...