Content-centric Networking (CCN) Research Activities at SNU

Taekyoung Kwon
tkkwon@snu.ac.kr
Seoul National University (SNU)

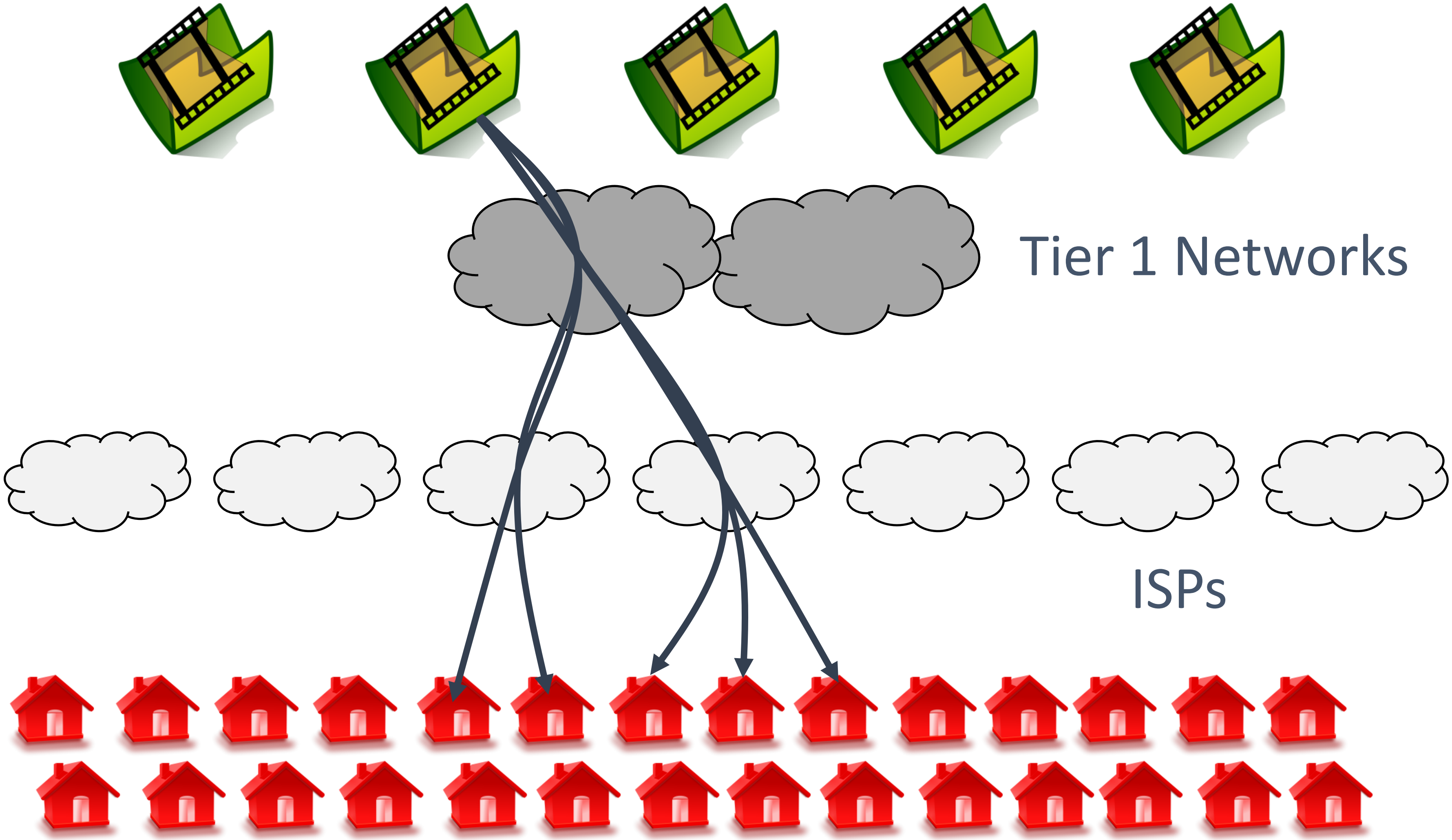
outline

1. Why ICN/CCN? What is it?

2. Routing/Caching

3. Forwarding

content distribution does not scale



source: Kutcher and Ohlman @ IRTF81

Problems of Today's Internet

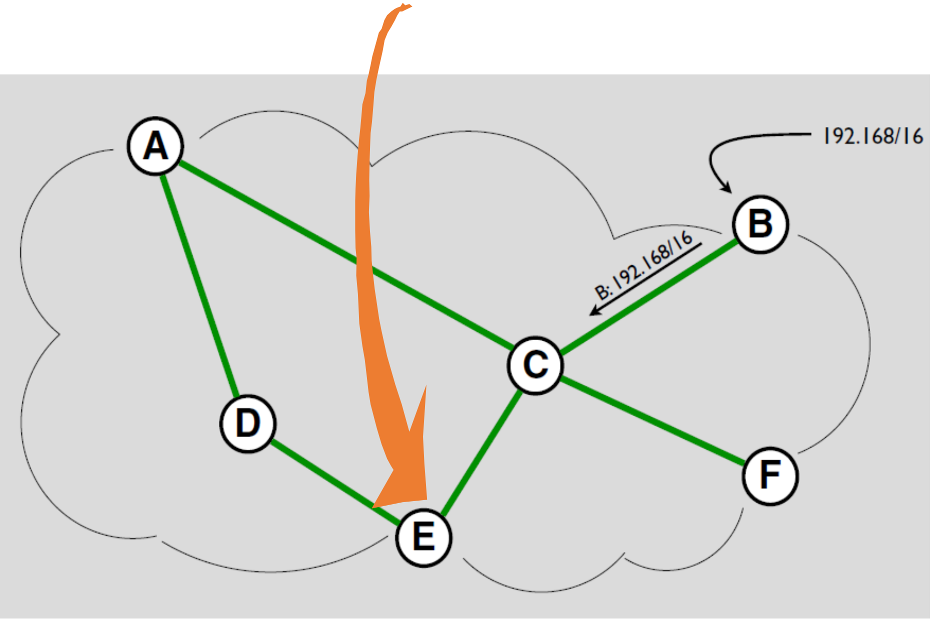
- URLs/IP addresses are overloaded with locator/ID functionality
  - Moving information = changing it's name => HTTP 404 file not found
- No consistent way to keep track of identical copies
- Information dissemination is inefficient
  - Can't benefit from existing copies (e.g. local copy on client)
  - No "anycast": e.g., get "nearest" copy
  - Problems like Flash-Crowd effect, Denial of Service, …
- Can't trust a copy received from an untrusted node
  - Security is host-centric
  - Mainly based on securing channels and trusting servers
- Application and content provider independence
  - CDNs focus on web content distributions for major players
  - What about other applications and other players?

source: Kutcher and Ohlman @ IRTF81

IP networking vs. ICN/CCN

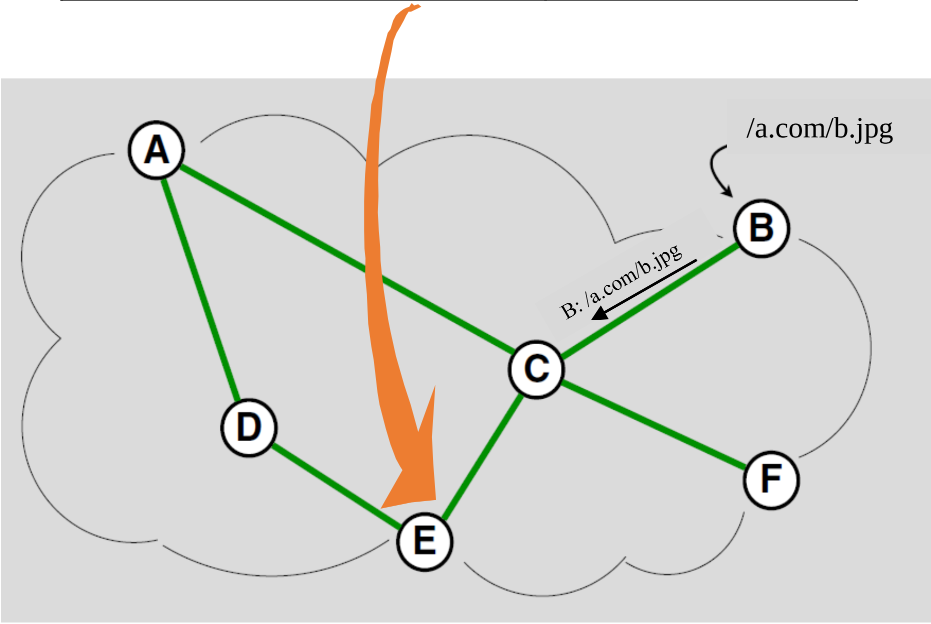
- Network prefix

Destination	Next Hop
192.168.0.0/16	Router C



- Content name

Content Name	Next Hop
/a.com/b.jpg	Router C



Where to put content name?

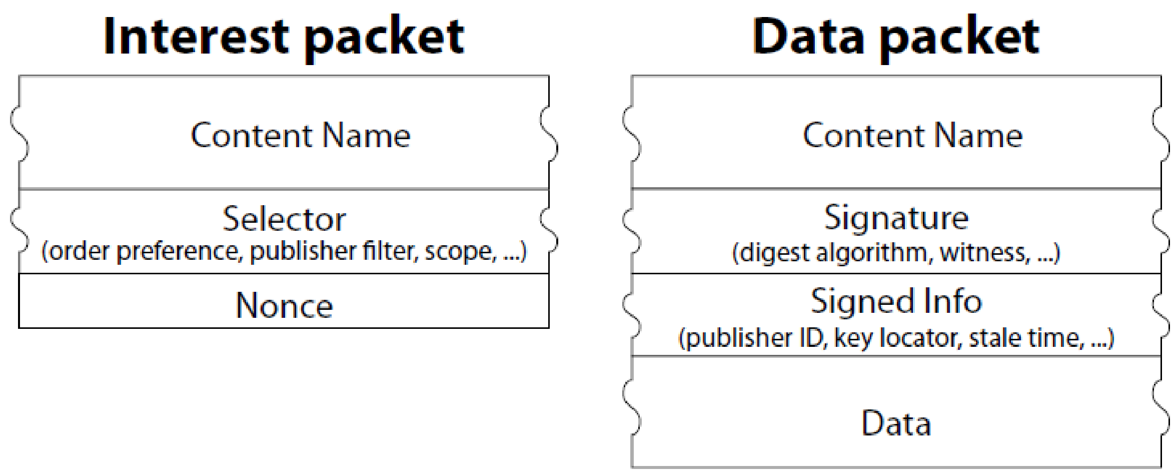
- In TCP/IP
  - Application layer header
    - E.g. HTTP, SIP
    - Deep packet inspection
  - IP option header
- New network layer header
  - A clean slate approach
    - content-centric networking (CCN)
    - named-data networking (NDN)

Content name or ID

- Content names (Cnames)
  - May replace the IP addresses
  - Content identifiers (CIDs)
- Cname/CID design choices
  - Hierarchical vs. flat
  - Semantics vs. semantic-free
    - Persistency
    - Location independence
  - Variable length vs. fixed length
- examples
  - /cnn.com/asia/sports/news.avi
  - /sonypictures.com/spiderman3.html
  - /yahoo.co.kr/image/logo.jpg
  - 0xF034BC....024A,
    - E.g. hash of content data, name, public key
  - Or hybrid

CCN basics

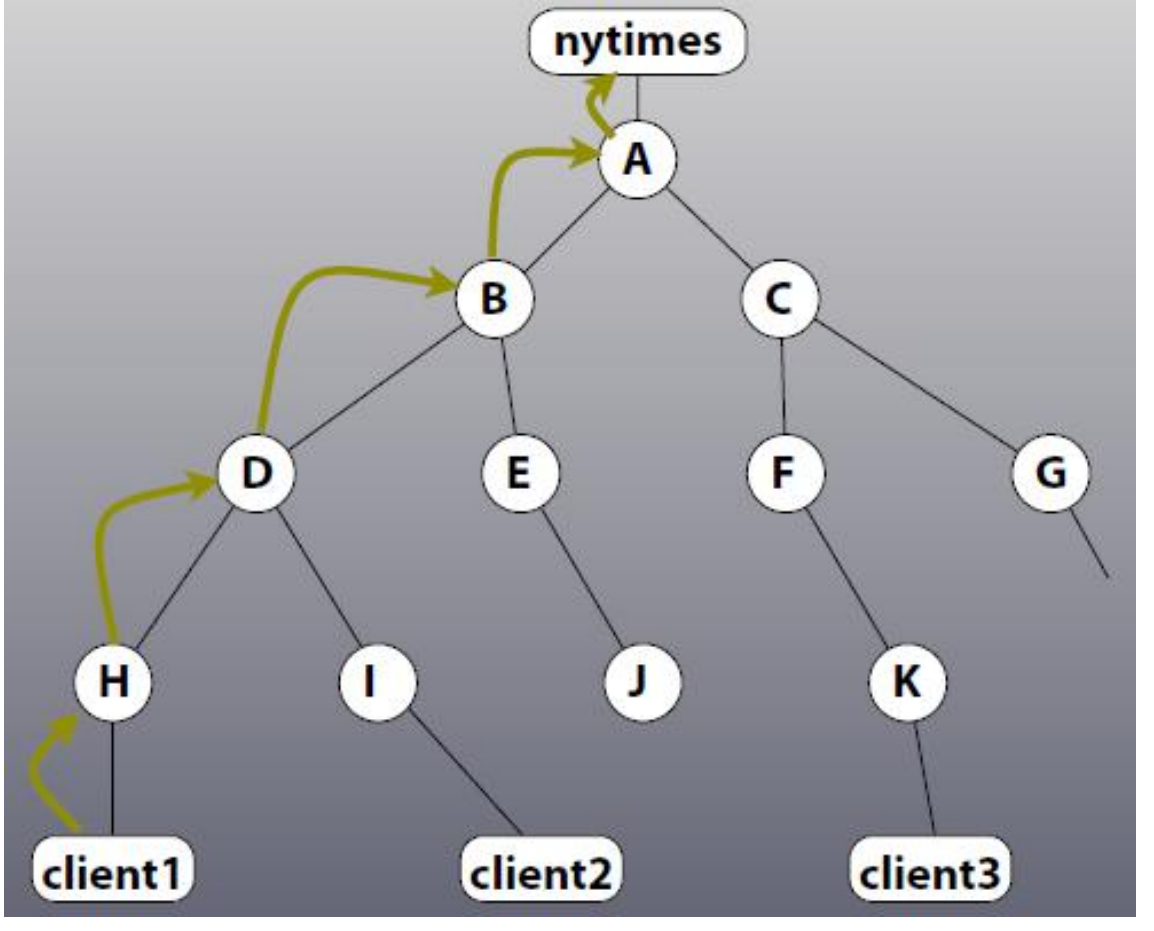
- Content name
  - Hierarchical, variable-length, semantics
- No IP address



- Consumers send Interest Packets
- Content holders send back Data Packets

Source: Van Jacobson@PARC

A user wants a particular object

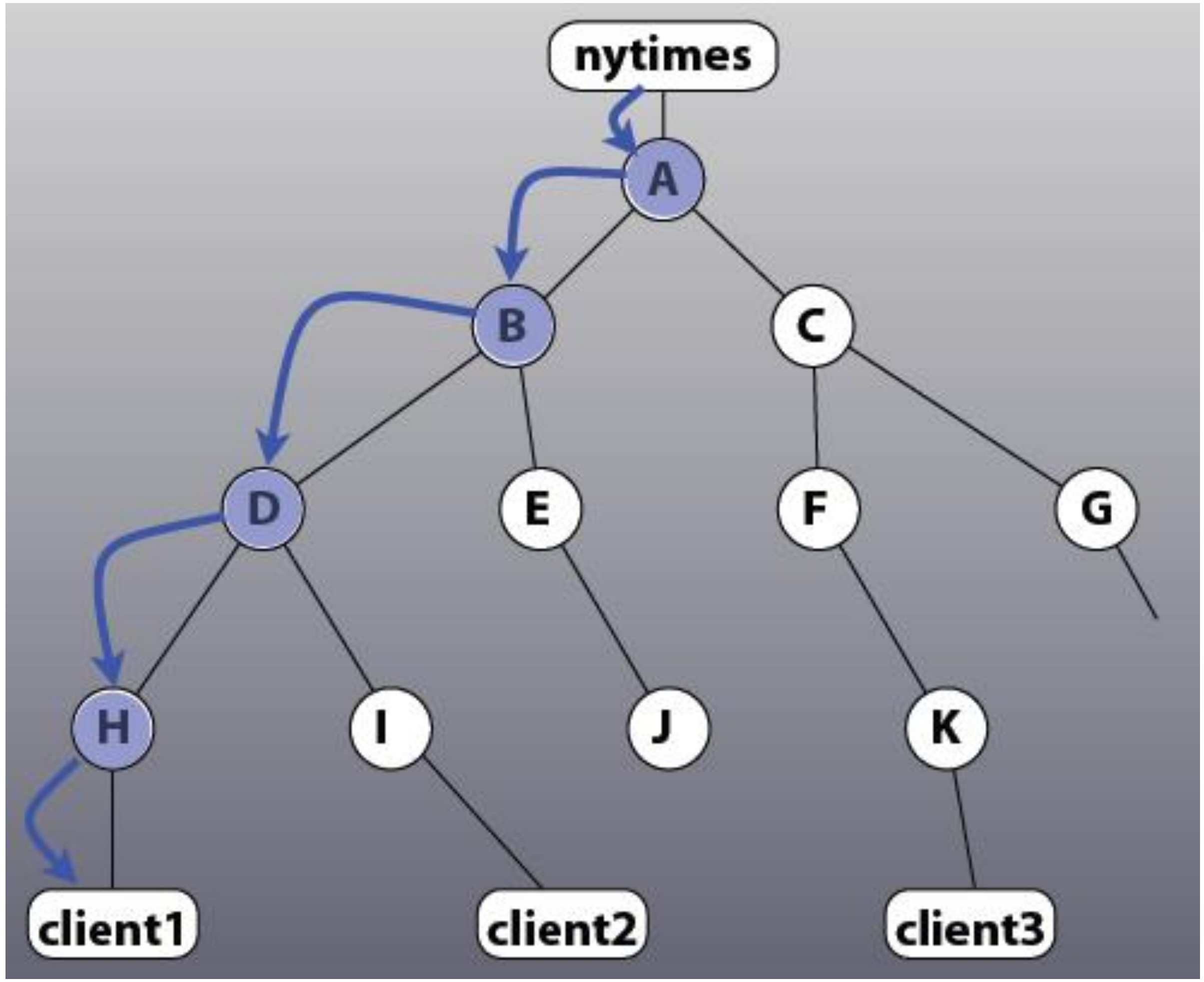


Source: Van Jacobson@PARC

The object is downloaded

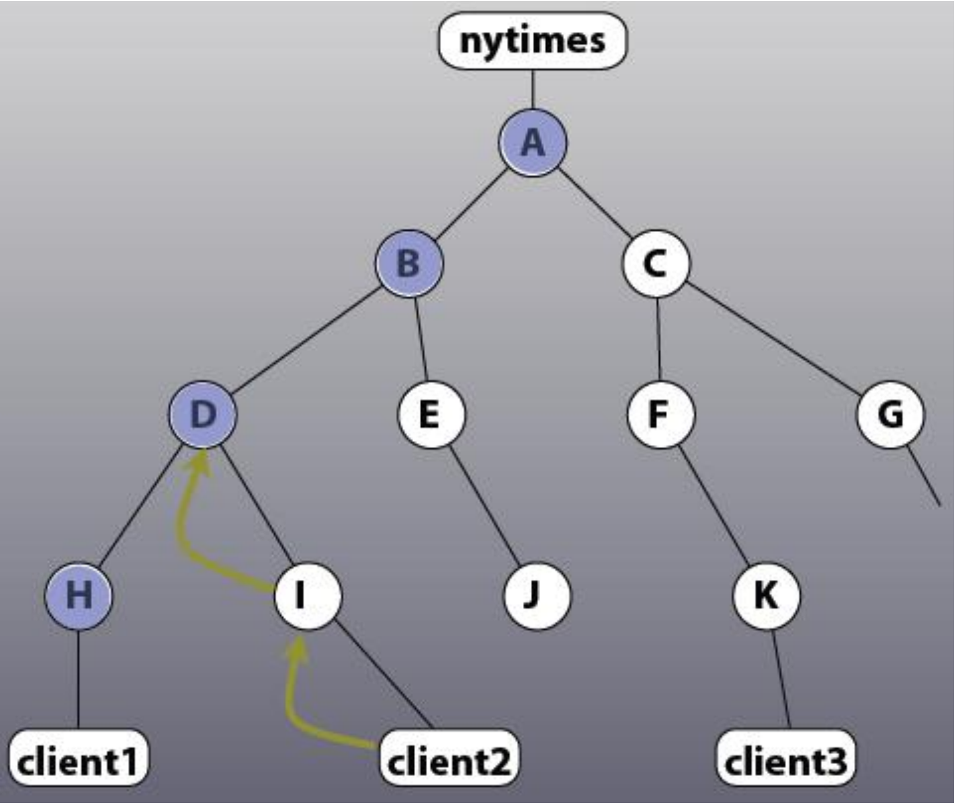
Content is cached!

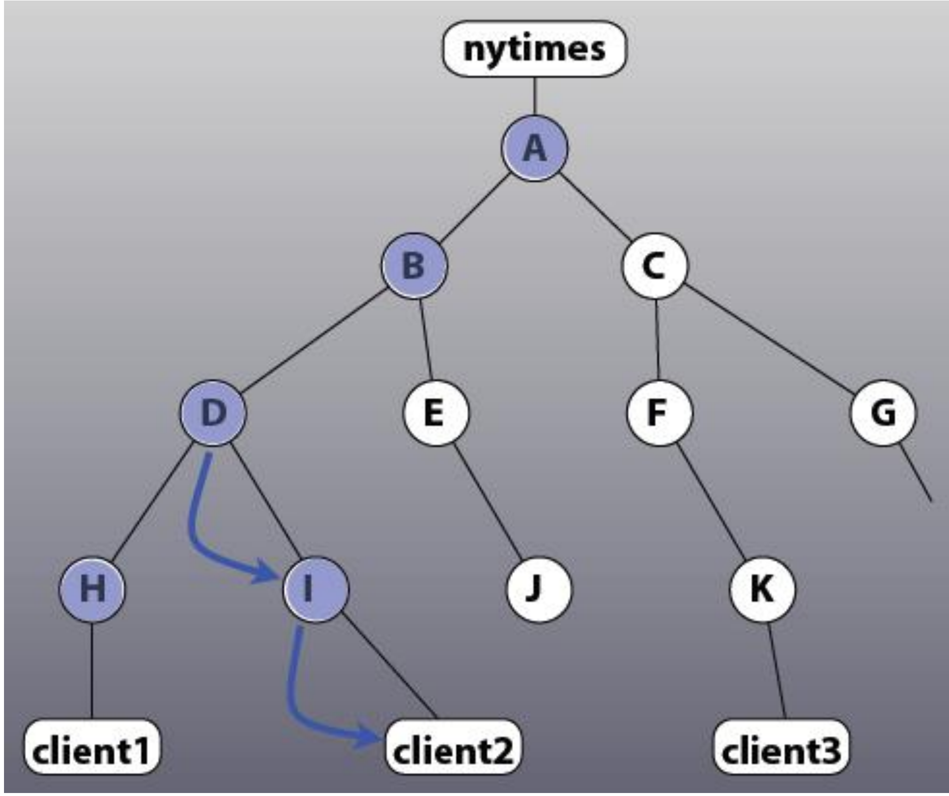
In-network caching



Source: Van Jacobson@PARC

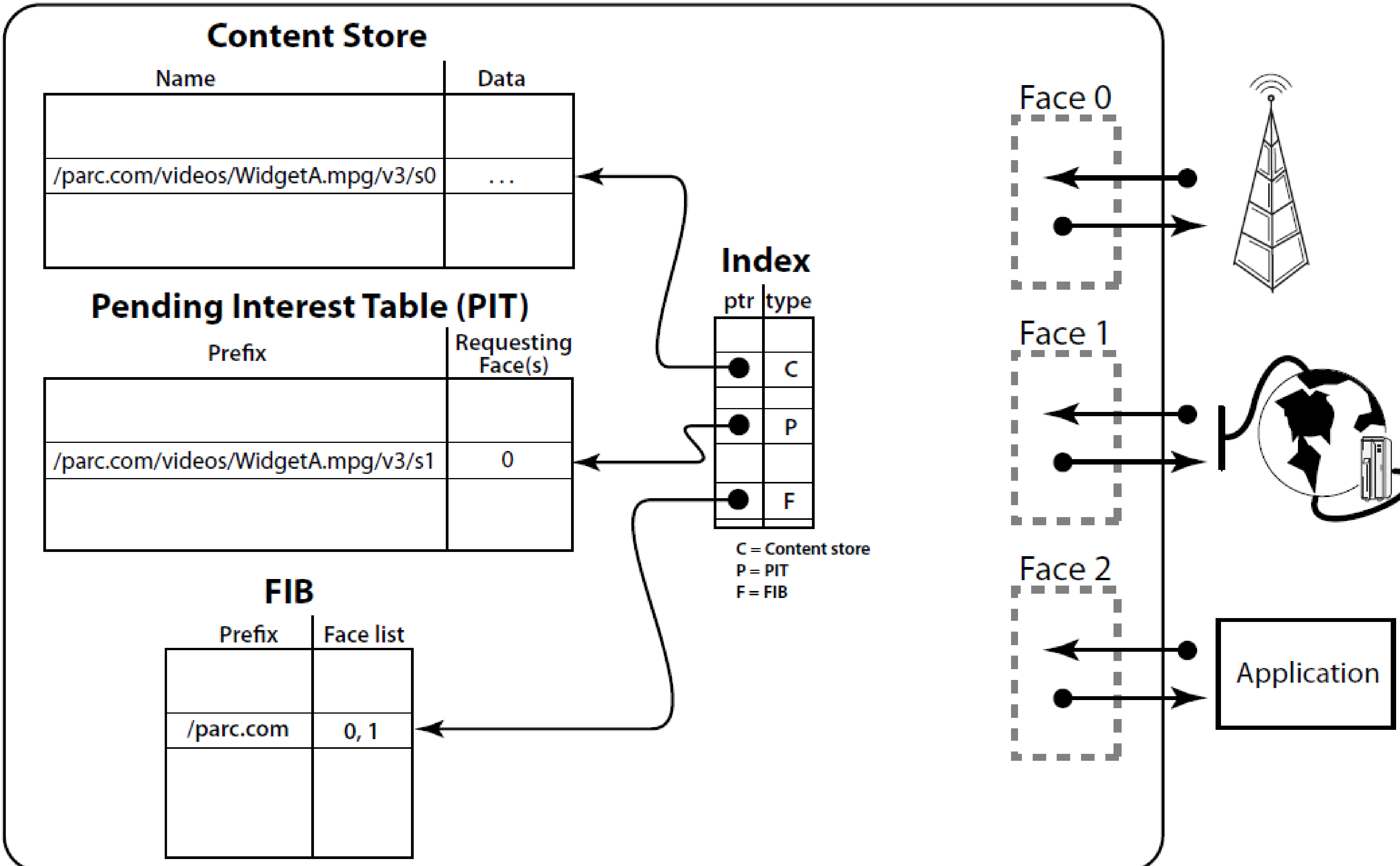
Another user requests the same object





Source: Van Jacobson@PARC

CCN forwarding



Source: Van Jacobson@PARC

ICN/CCN Recap

- Route-by-name
  - No indirection, better availability
  - Content name (or ID) is a routing entry
  - Huge scalability concern

- In-network caching

- Global-scale pure CCN may not be feasible
  - At least trillions of contents
  - Some aggregation may be possible
    - E.g. hierarchical names like URLs
    - One billion hosts now

- Other merits such as authentication

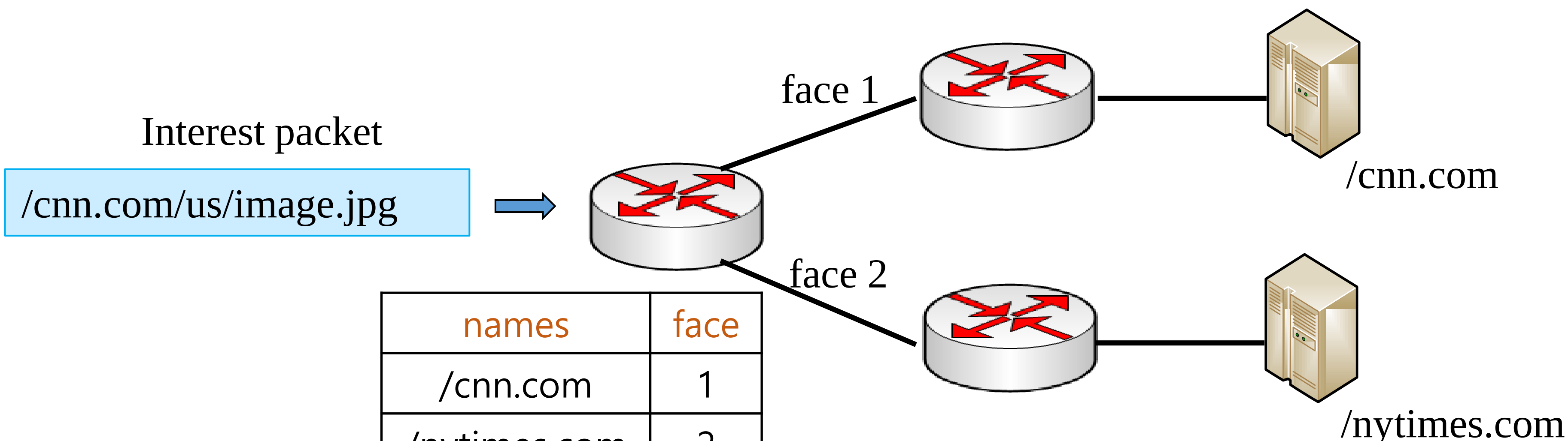
challenging issues

- routing
- forwarding
- caching
- applications/services
- naming
- mechanisms
  - e.g. interest control
- security and privacy
- migration
  - SDN programmability

CCN Routing/Caching

Routing in CCN/NDN

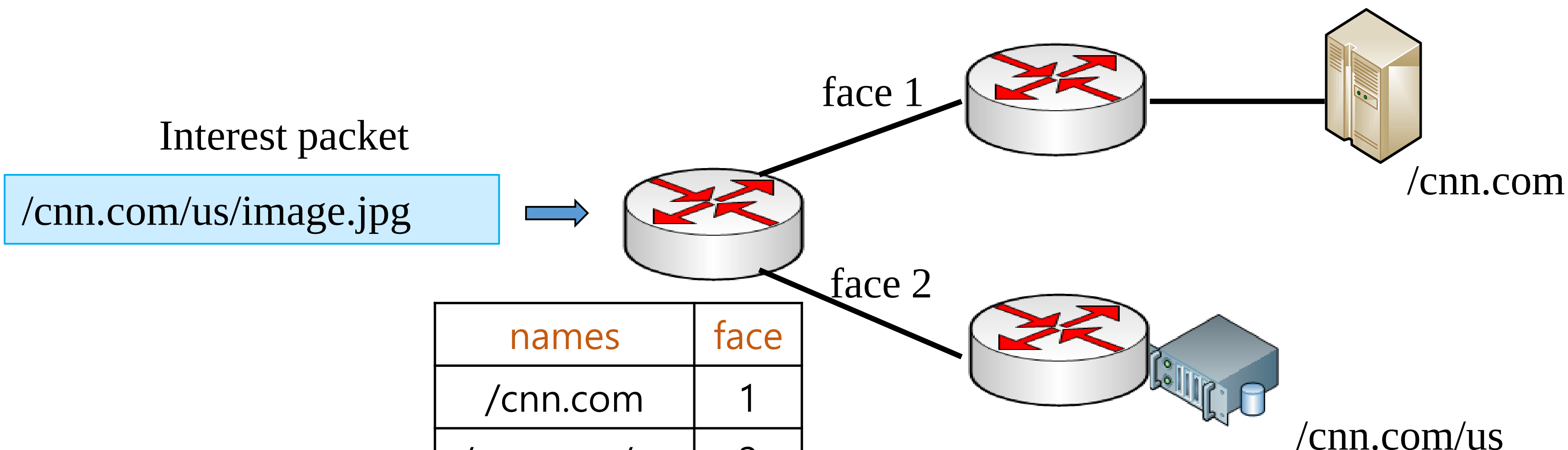
- No IP addresses
  - No DNS
- Use content names for forwarding
  - Content name (or content ID) is a routing entry



names	face
/cnn.com	1
/nytimes.com	2
…	

CCN routing: LPM

- An interest will be forwarded to a face with longest prefix matching (LPM)



names	face
/cnn.com	1
/cnn.com/us	2
…	

Issues in routing/caching for NDN

- Routing scalability
  - Too many content names
    - Number of contents in Google : O(10^{12})
  - Even with aggregation at host names
    - Number of domain names: O(10^9)
- Independent Caching at individual routers
  - Inefficient cache usage (i.e., redundancy)
- Uncoordinated routing and caching
  - If an item cached, that should be advertised
    - Worsen routing scalability
  - Otherwise, only on-path cache hit

CoRC addresses these problems

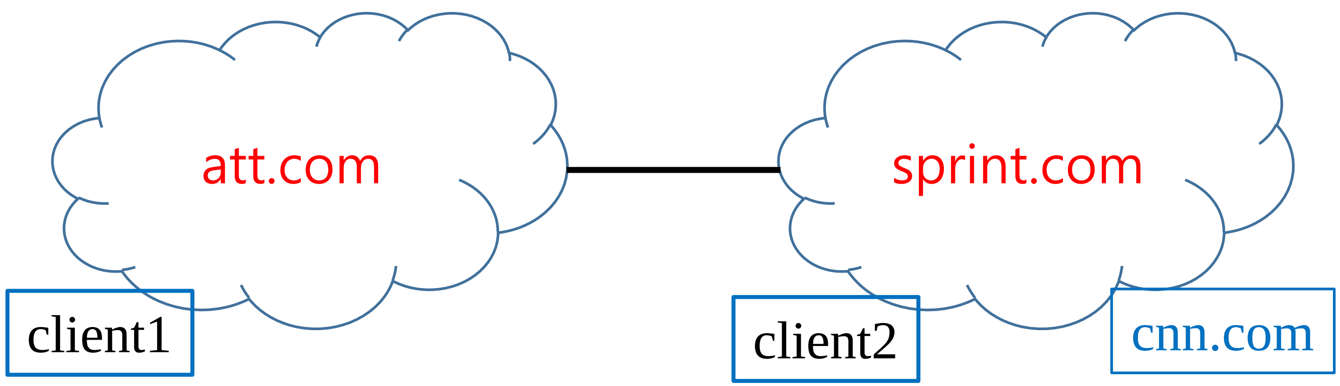
- Routing scalability
  - Partition FIB space
- Caching efficiency
  - Partition cache namespace
- Routing and caching are coordinated
  - Each router is in charge of the same namespace for routing and caching

* CoRC: Coordinated Routing and Caching

Hierarchy for scalability

❖ Some assumptions

- A host name (or publisher) is present in a content name
  - E.g. /cnn.com/asia/news.avi
- A host is connected to a particular ISP/AS
  - E.g. cnn.com is a subscriber to sprint.com
  - A mapping service between AS names and host names



- AS: autonomous system
- ISP: internet service provider

Inter-domain vs. intra-domain routing

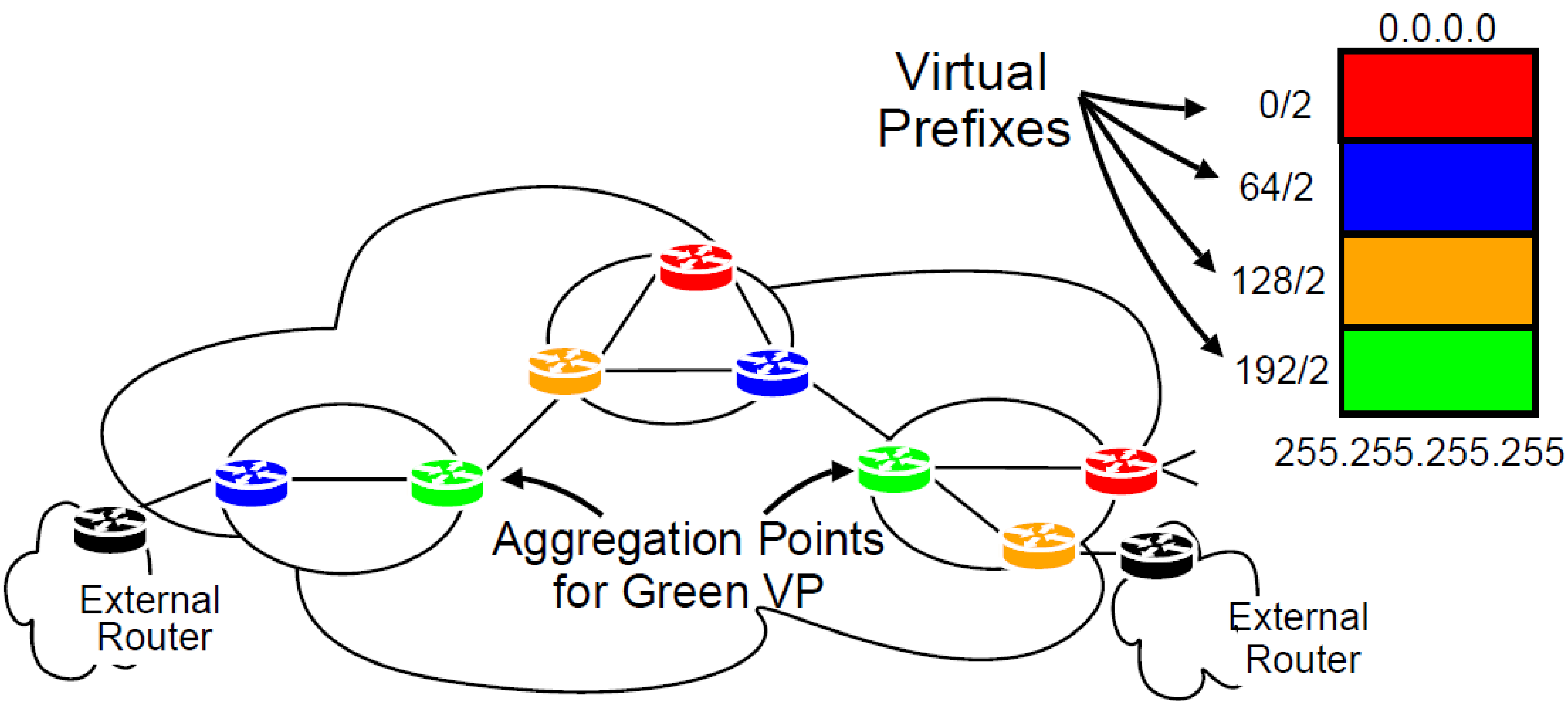
- A router in an AS needs to have FIB entries for host names in the AS
- For the hosts outside the AS,
  - Just have FIB entries for AS names of the hosts

Inter-domain routing

- Routing based on AS name
  - AS-FIB contains <AS name, next-hop IF>
  - AS name is advertised by an inter-domain routing protocol such as BGP
  - AS name can be used as-is or its hash

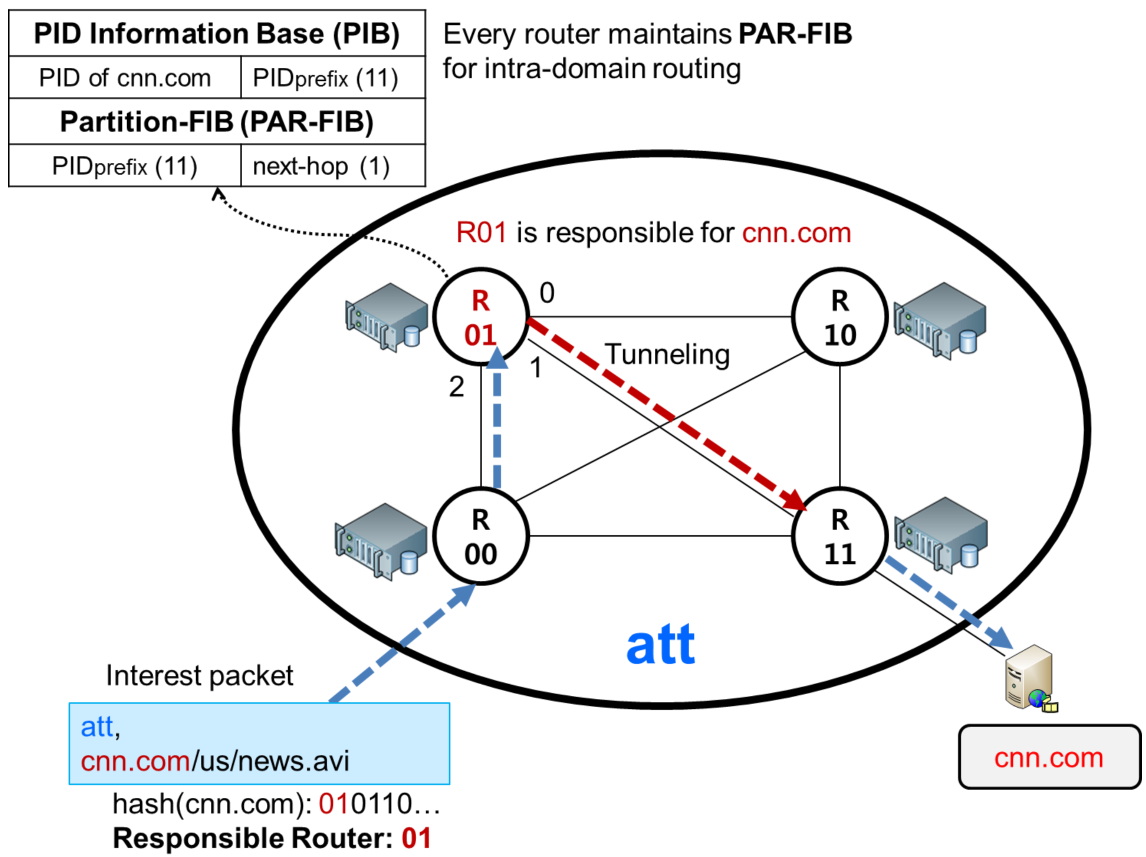
Split the hostname space for intra-domain

- Virtual Aggregation [NSDI '09]



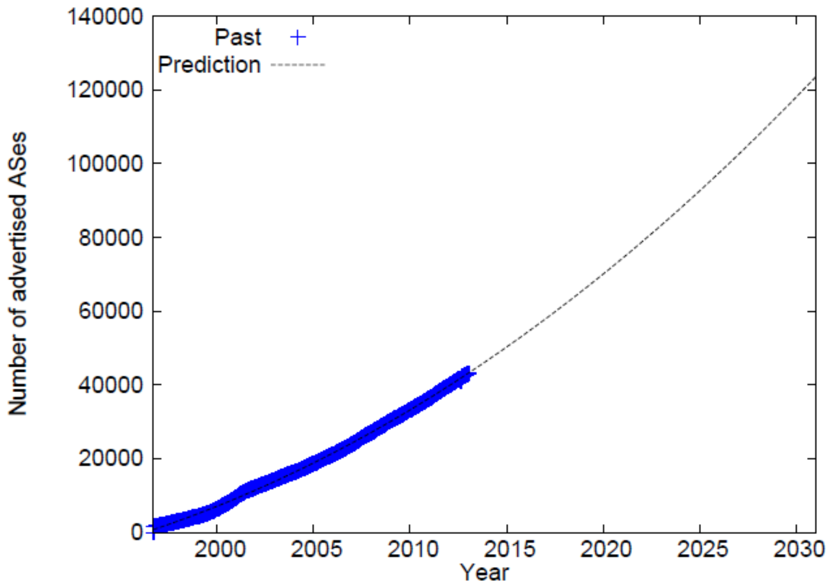
Intra-domain routing

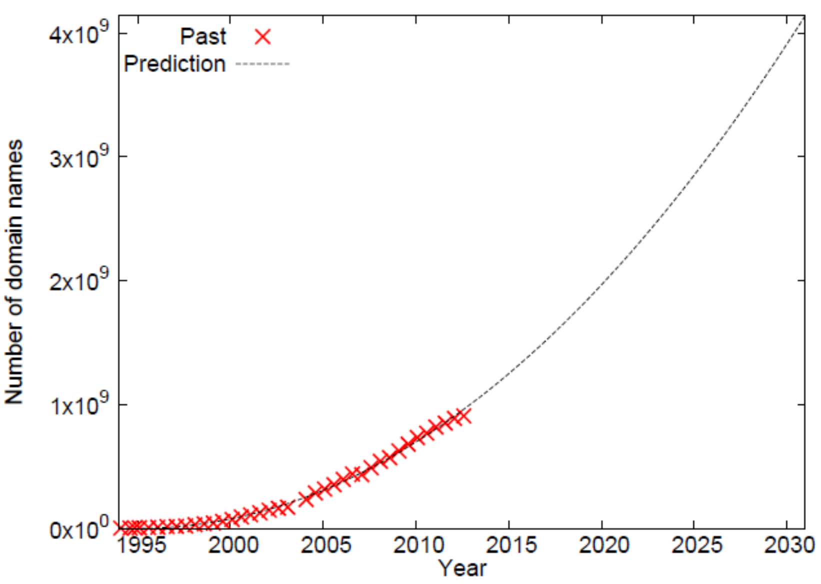
- Publisher identifier (PID)
  - hash of a publisher name, say 128 bit
- Partition routing and caching namespace
  - R01 should know the locations of all the publishers whose PIDs start with 0b01
  - Contents whose names starting with cnn.com are cached only at router R01



FIB size in 2030 (1)

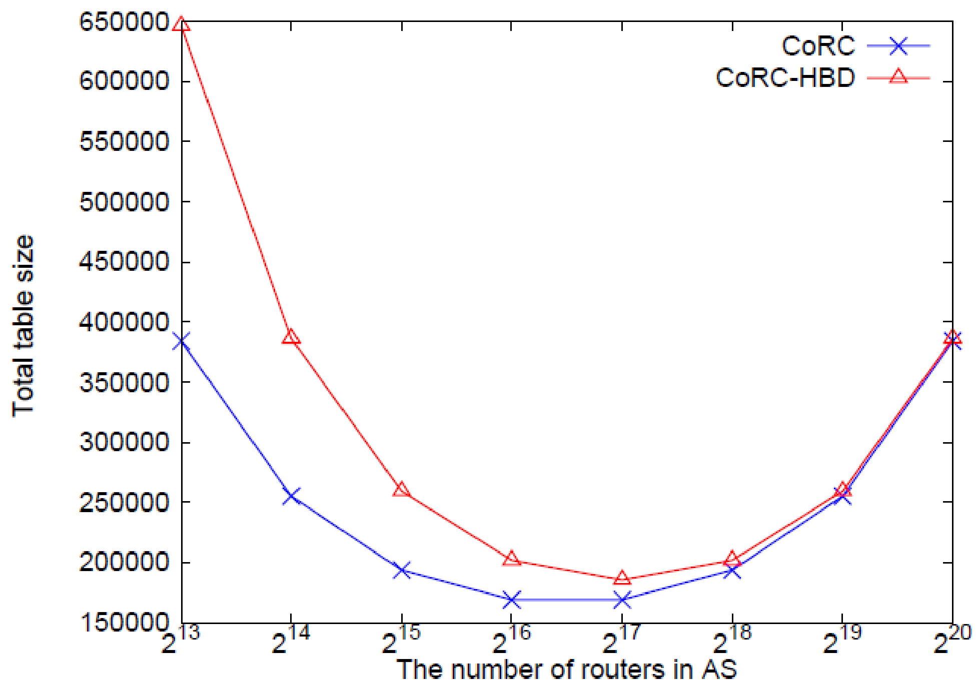
- AS-FIB
  - Number of ASes -> about 120,000
- PAR-FIB
  - the number of routers in an AS
- PIB
  - O(# of host names) / O(# of routers) in an AS





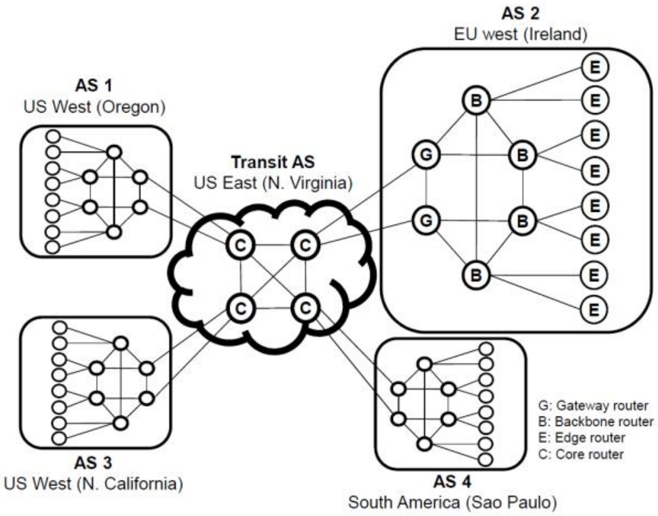
FIB size in 2030 (2)

- The Largest AS case
  - Contains 300 million publishers (assuming Zipf)
  - Has up to 2^{20} routers
  - ¼ of total routers operate as responsible routers
- Total FIB size of a CoRC router is comparable to that of a current DFZ router



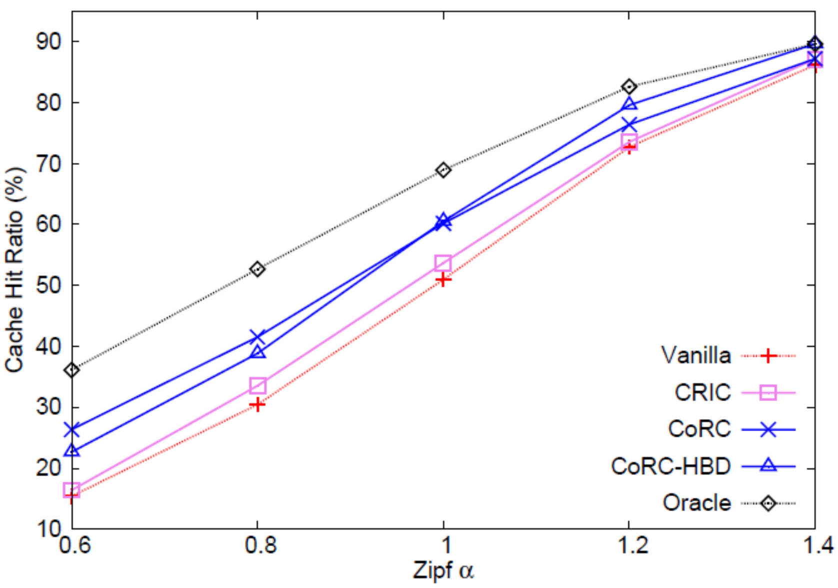
Comparison

- Flat routing, Independent Caching
  - Vanilla NDN
- Coordinated routing, Independent Caching (CRIC)
  - Interest packets are first sent to responsible routers
  - The routers cache content individually
- CoRC
- CoRC-Hybrid (CORC-HBD)
  - Some cache space for popular contents
- Oracle
  - The popularity distribution of items is known in advance
  - The border router stores all the top popular contents
- Network-wide cache space is equal

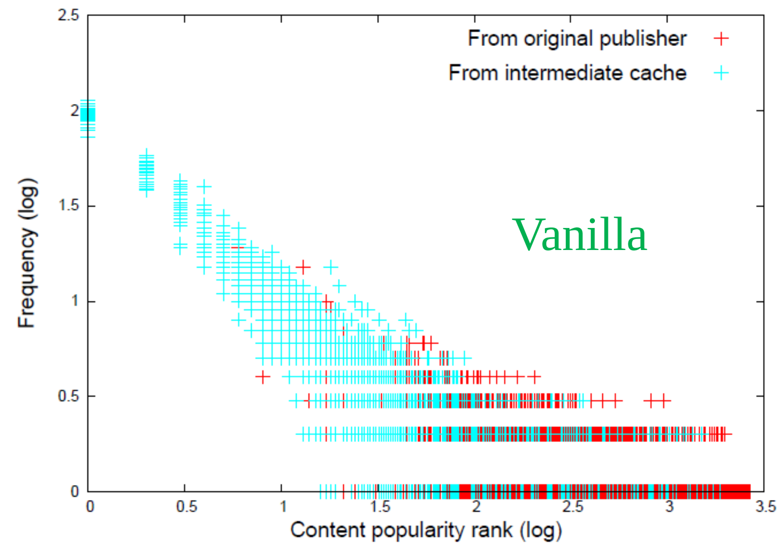


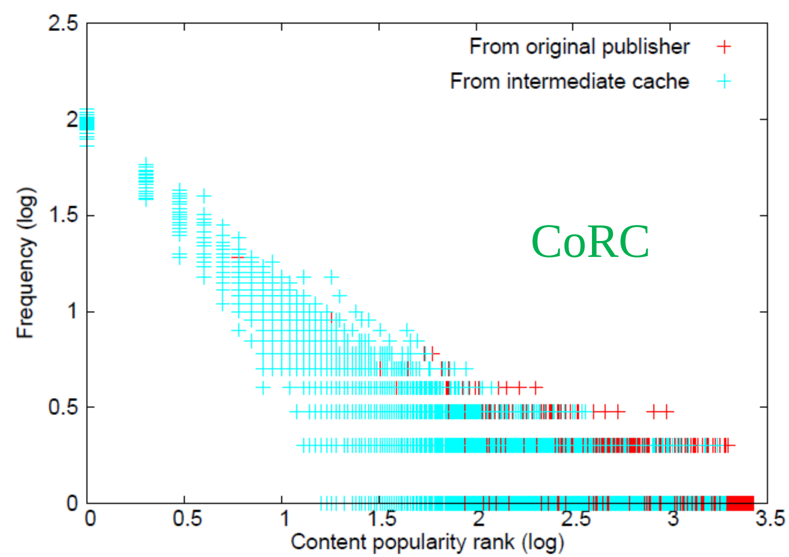
Cache utilization

Cache hit ratio



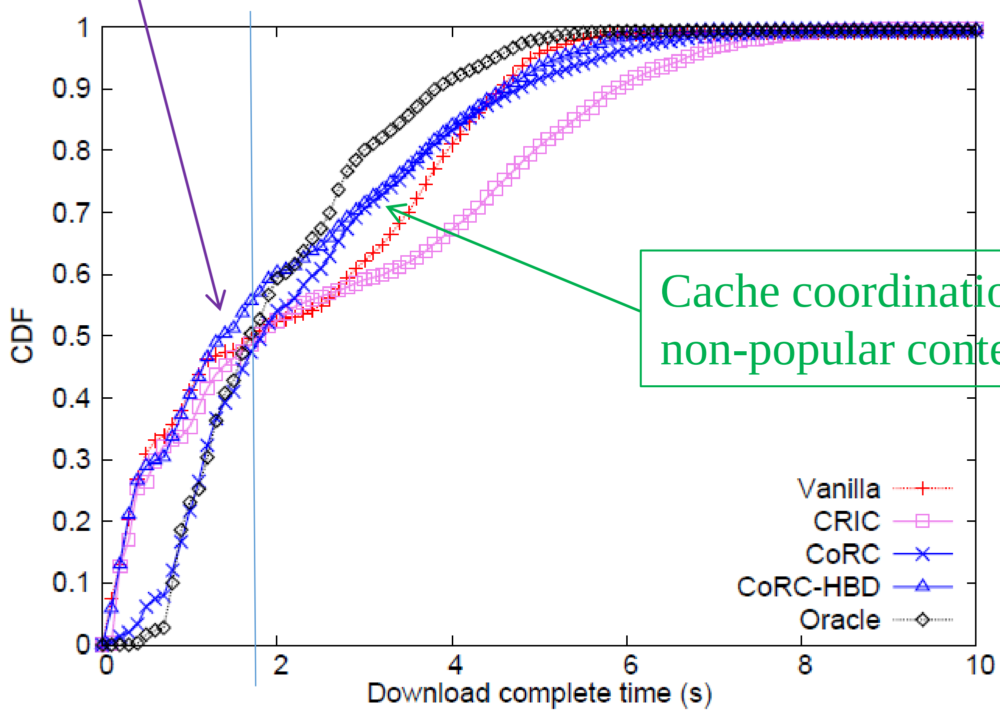
Cache hit/miss plot





Content retrieval time

Popular contents can be retrieved from nearby routers

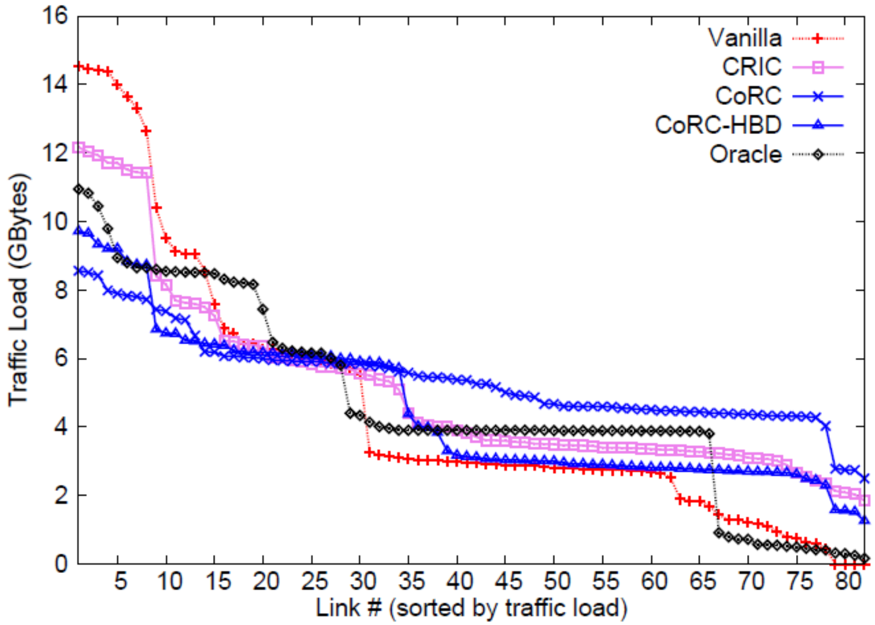


Cache coordination is beneficial for non-popular contents

(b) Content retrieval time (α = 1.0)

Link Load

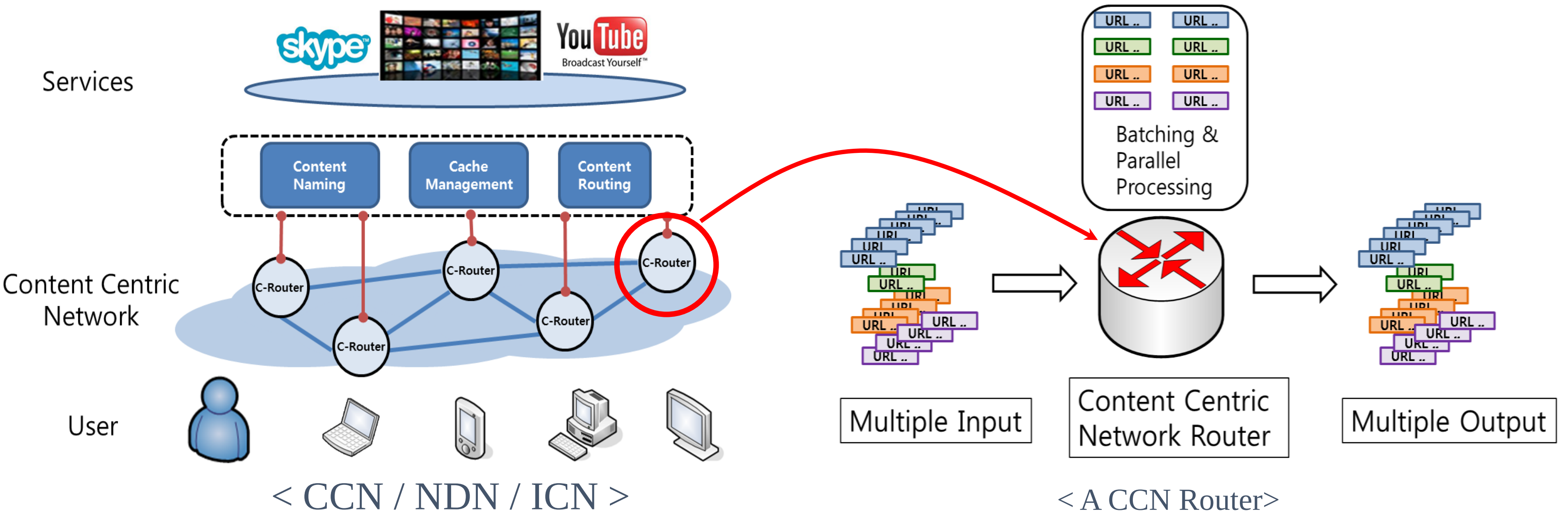
- Splitting the whole cache space to routers helps spread traffic over all links.



(c) Link load (α = 1.0)

CCN Forwarding

Motivation (1/2)



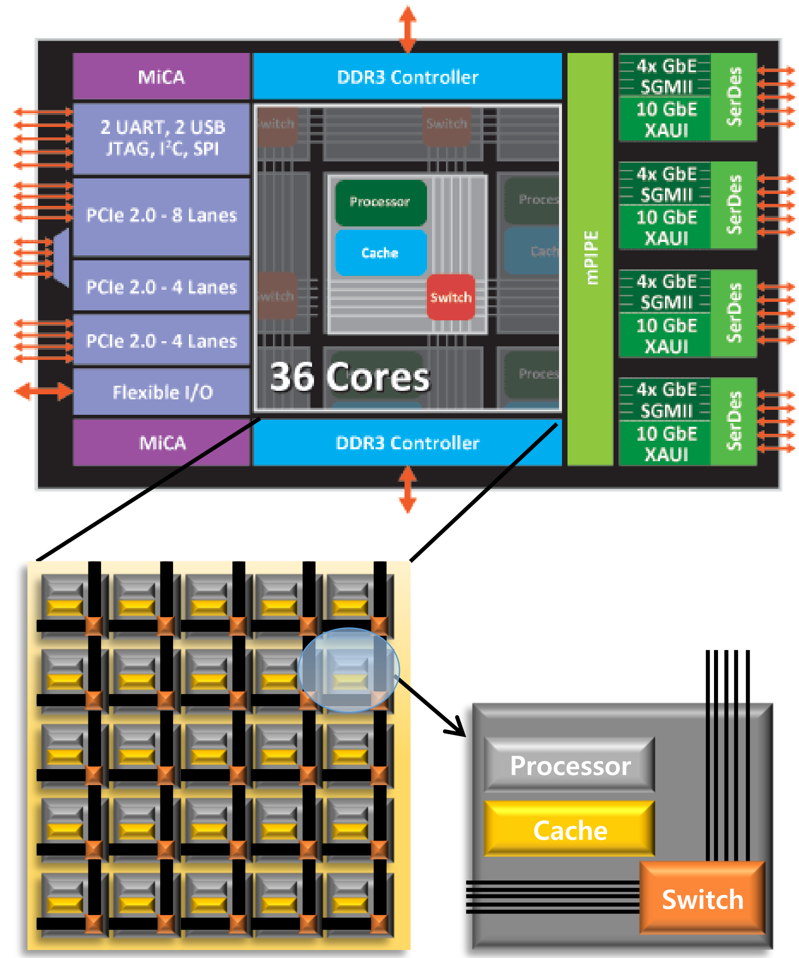
- High speed CCN router is crucial!
  - Many objects, many more chunks
  - large content like video

Motivation (2/2)

- CCN router speed up!
  - There have been a few solutions for a single lookup
    - High speed hardware acceleration (e.g., TCAM)
    - Good data structures and algorithms (e.g., hashing)
  - How can we deal with many lookups for many packets?
    - It must process a large number of chunks/packets
    - Memory access is a bottleneck of performance [SIGCOMM2010, SOSP2009]
  - Let's use parallelism!
    - on Multicore-based architecture (We use Tilera)

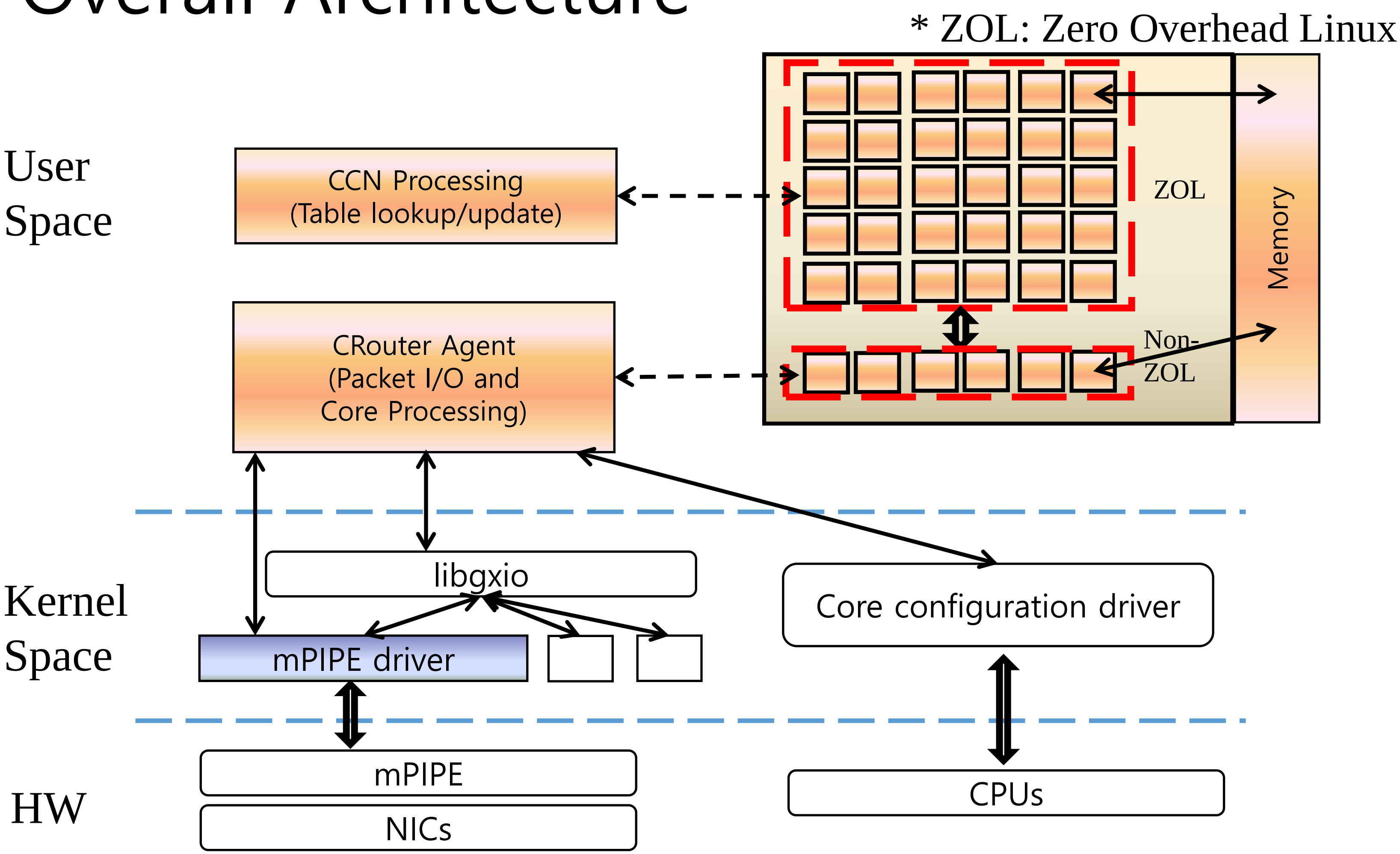
Tilera architecture (Gx8036)

- Good for I/O intensive work
  - PCIe bus is bottleneck
    - NICs are directly connected to cores
  - Memory is directly connected to cores
- 36 cores
  - Not expensive in terms of cost per core
- Utilizes additional hardware (mPIPE) for packet processing

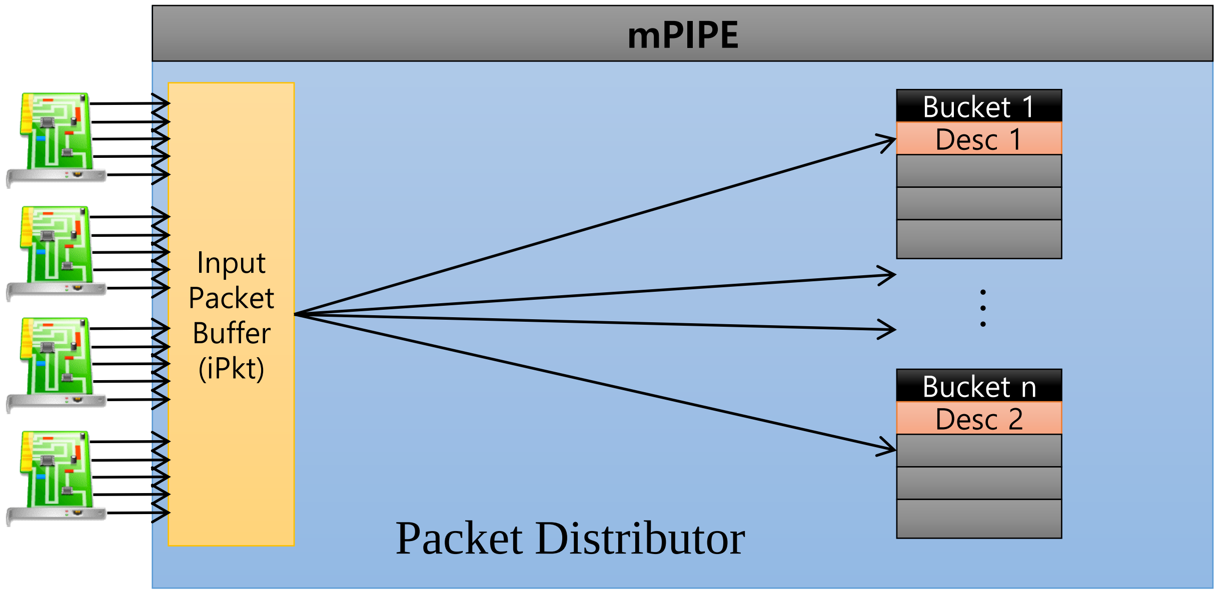


[TILERA Gx8036 (36 cores, 4x 10G NIC)]

Overall Architecture

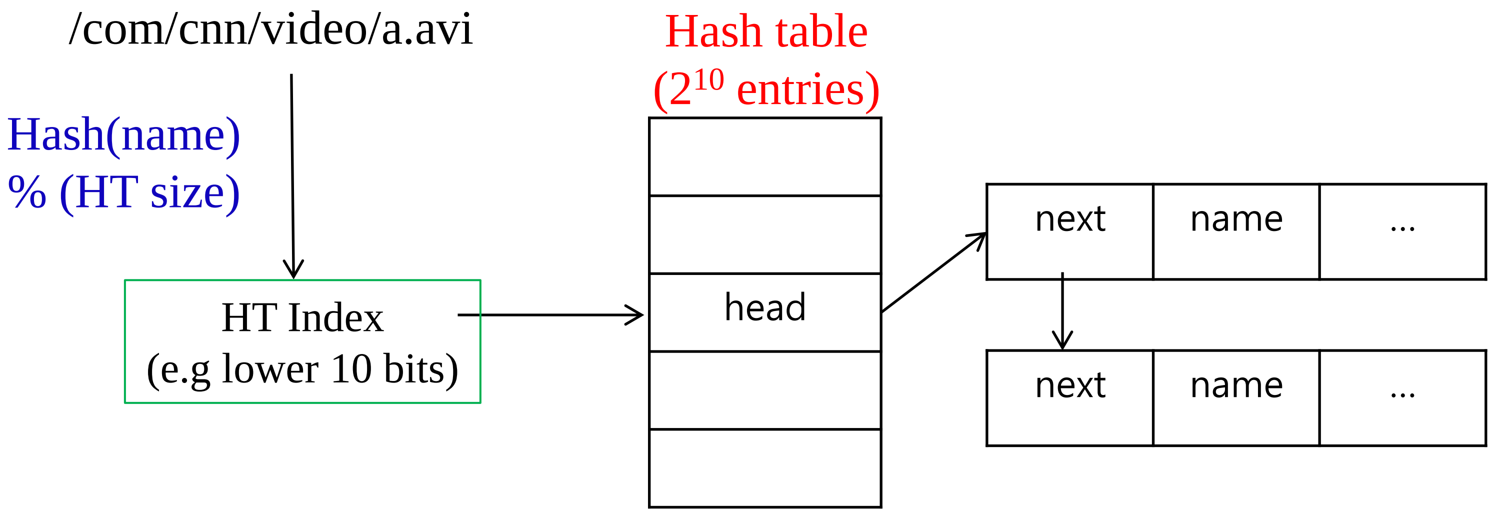


Packet Distributor



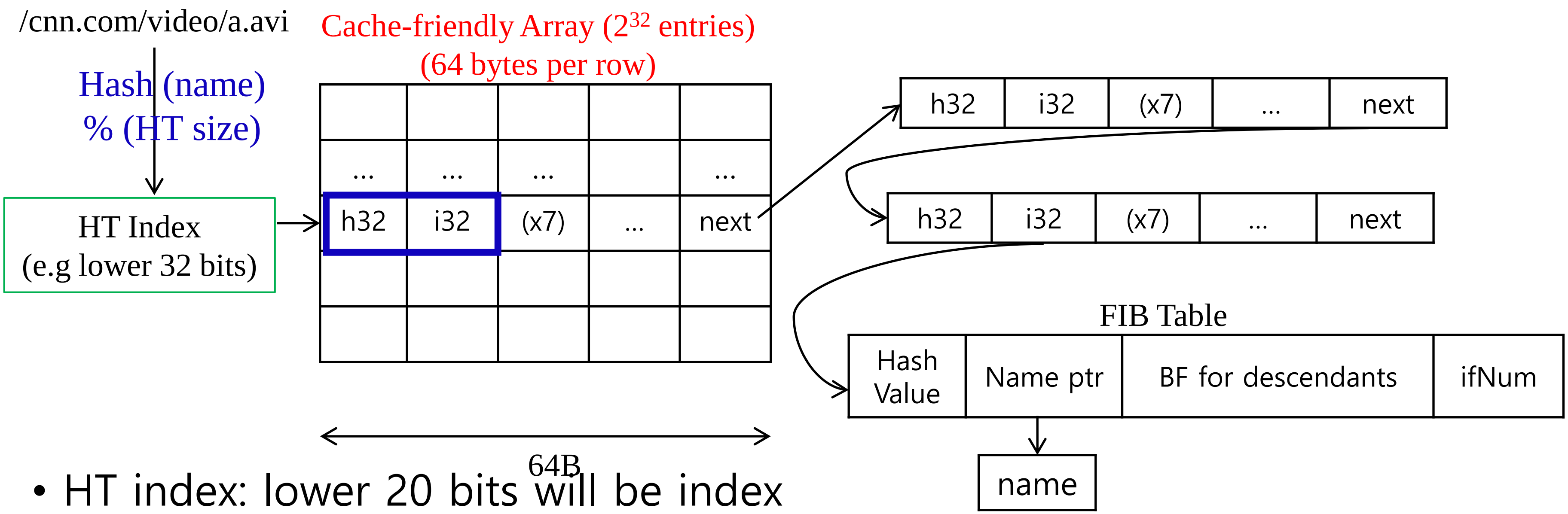
- Generates a packet descriptor for each packet
  - will go to memory partition for a core,
  - virtual address (pointer to a packet), duplication flag,…
- Assigns each packet descriptor to each core
  1. Execute a lightweight hash function for the content name
  2. Run modulo operation by the number of cores
     - each core has responsibility for a partition of namespace (i.e. hash)
- Packet payload is directly copied to memory by iDMA

Hash Table (HT) for CS/PIT



- Each core manages its own CS/PIT hash table
  - No lock operation required
- Hash table is utilized for CS/PIT lookup
- Consists of pre-allocated array & linked list
  - Nice for dynamic insertion & deletion

Hash Table (HT) for FIB: Cache-friendly

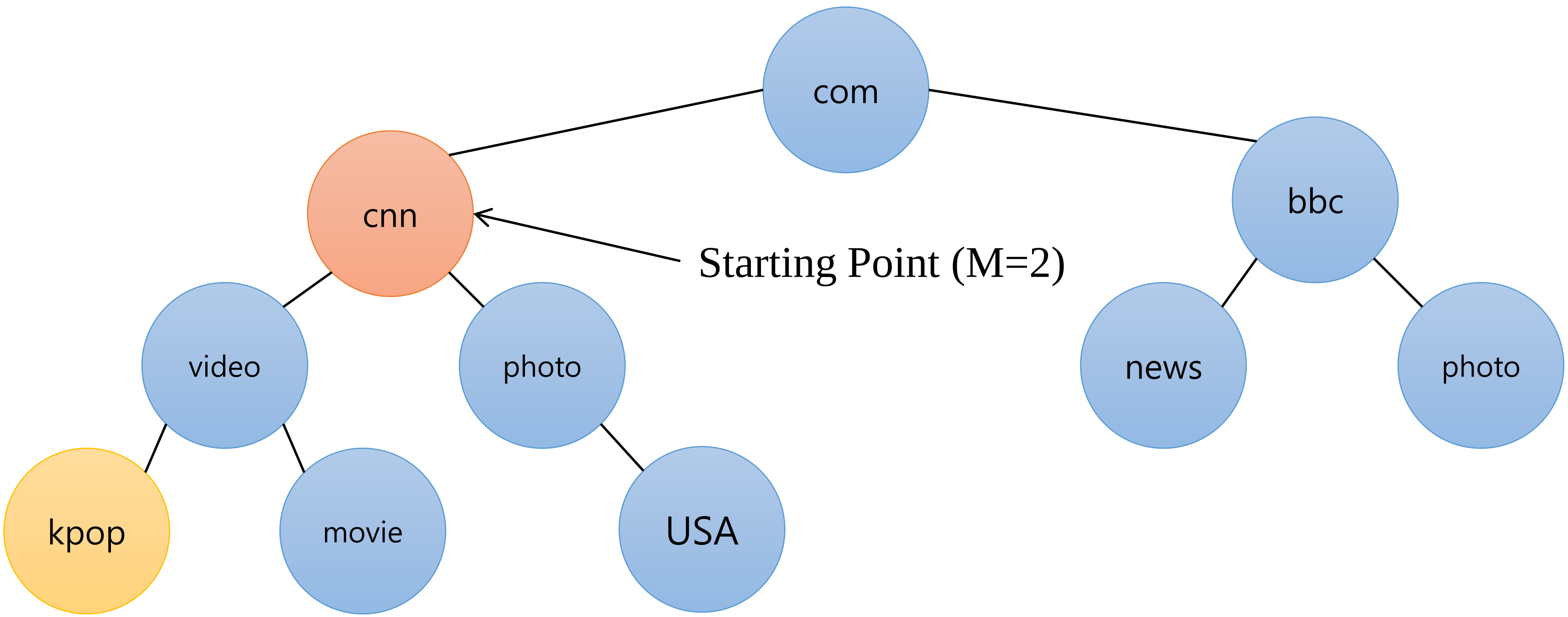


- HT index: lower 20 bits will be index
  - h32 : higher 32 bits of the 64 bit hash value
  - i32: FIB index (32 bits)
  - Hash value: 64 bits
- Cache-friendly array is utilized for efficient FIB lookup
  - 64 bytes per access
  - For one lookup request, 7 entries with the same h32 value are read to cache
    - If fails, lookup next 8 entries at one time
- Bloom filter for all children of the entry is stored in advance

LPM Operation & Name Update

- Starting at the specific point (M)
  - look at M components first
    - In current internet, many content providers do not usually publish contents with 1 component
- Bloom filter operation is utilized when a prefix is found
- For better FIB updates, counting bloom filter (CBF) is utilized
  - CBF is for FIB updates

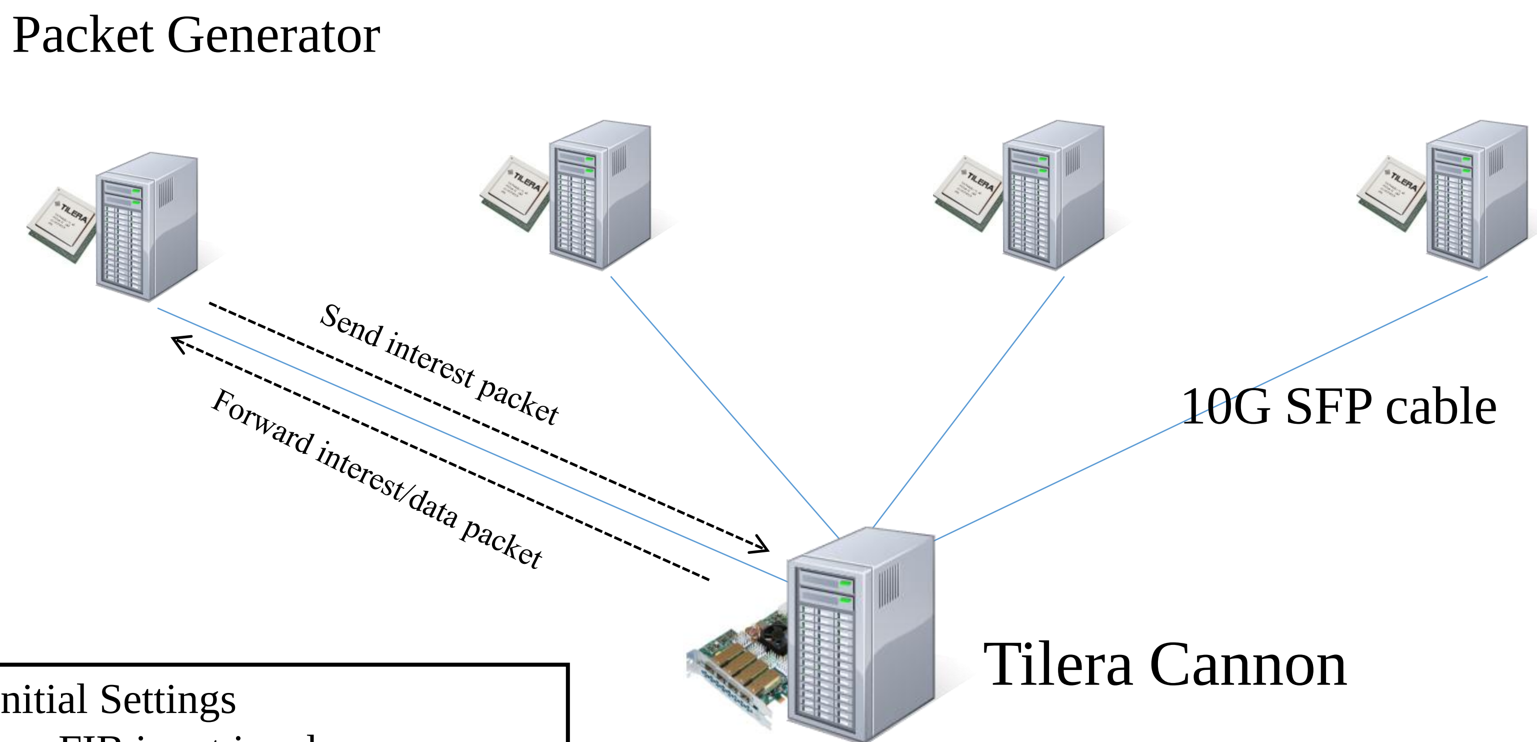
Illustration for LPM operations



For “/com/cnn/video/music/psy.mp3”,

1. Find /com/cnn & Check Bloom Filter (BF)
2. Find /com/cnn/video/music & Check BF
3. If bloom filter of “psy.mp3” exists,
    1. Find /com/cnn/video/music/psy.mp3

Testbed



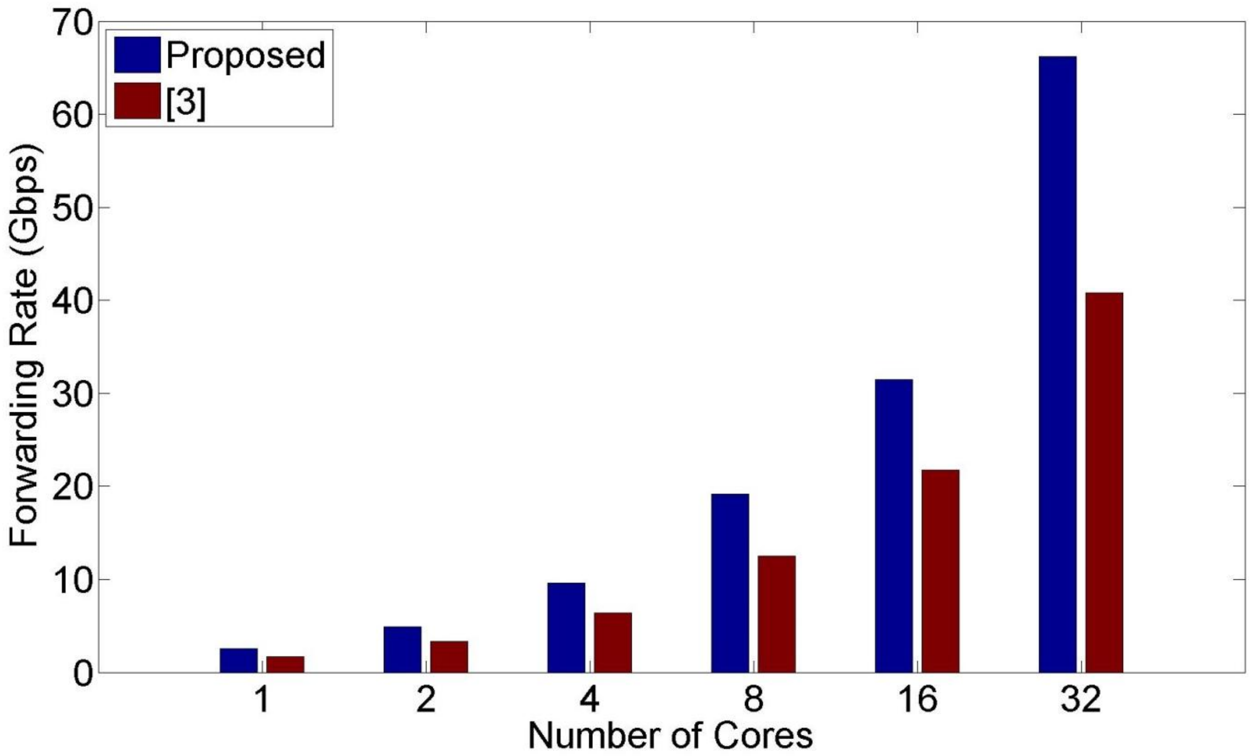
Initial Settings
1. FIB is set in advance
2. 4 transmitters & 4 receivers

Experiment setting

- Each machine runs 4 transmitters & 4 receivers for full rate test
  - Avg. 9 Gbps for each link (around 36 Gbps total)
- Content names of interest
  - Generated from the request URLs in IRCache dataset [1]
  - 1 week dataset from the IRCaches in 7 states in the USA
- FIB entries
  - Extracted from content names of interest
  - Distribution for the number of components follows URL BlackList dataset [2]
- Total forwarding time for 40 million interests are measured

[1] http://www.ircache.net/
[2] http://urlblacklist.com/

Max Forwarding Capacity (FIB access only)



- Forwarding rate increase linearly when the number of cores increase
- Our content router shows better forwarding performance (60%) than [3] (recently published research for multicore-based CCN router)

[3] Won so et al., “Named Data Networking on a Router: Fast and DoS-resistant Forwarding with Hash Tables”, in Proc. ANCS 2013

final remarks

- we introduced key technical challenges/our approaches in CCN
- routing and forwarding by variable-length name is feasible
- killer applications
- migration
    - SDN/NFV programmability