Osaka University

Design of a High-speed Content-centric-networking Router Using Content Addressable Memory

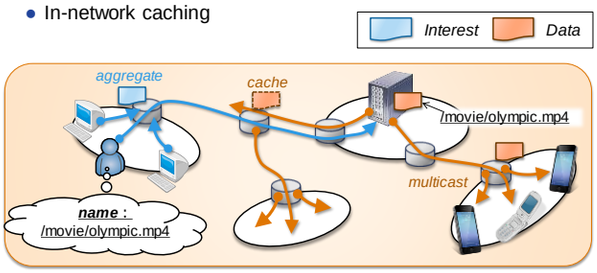
Atsushi Ooka
Osaka University
Japan

Outline

- Overview of Content-Centric Networking (CCN)
- Challenges to implement CCN router
- Propose an architecture of CCN router
  - Name Lookup Entity (NLE)
  - Interest Count Entity (ICE)
- Hardware Design
- Evaluation

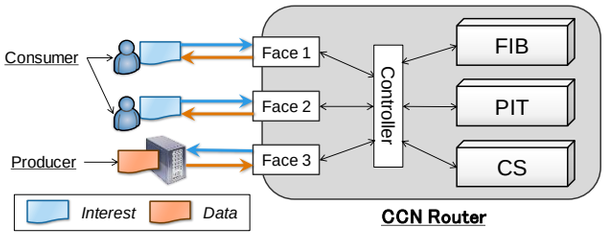
Content-Centric Networking (CCN)

- Content-based addressing with "*name*"
- Natively supported mechanisms to distribute content
  - Interest aggregation
  - Data multicasting
  - In-network caching



CCN Router Model

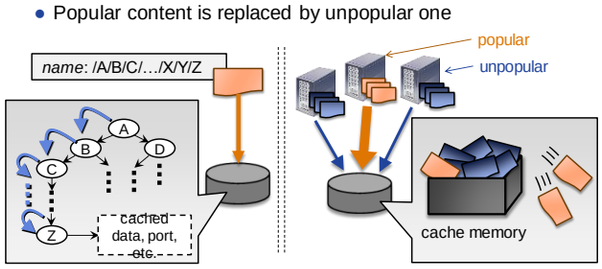
- Handles two types of packets (Interest and Data)
- Contains three types of tables to support new mechanisms
  - FIB*1: Forwards Interest
  - PIT*2: Aggregates Interest and multicasts Data
  - CS*3: Caches Data



*1 Forwarding Information Base, *2 Pending Interest Table, *3 Content Store

Challenge

- Variable-length, hierarchically structured names
  - The number of content names is very large.
  - The Latency of tree-based approaches is high.
- Limited cache capacity of CCN router
  - Popular content is replaced by unpopular one

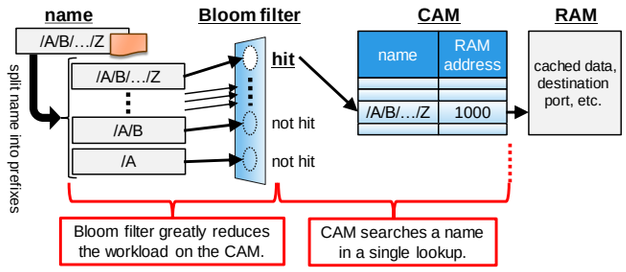


Approach

- Discuss an accurate communication model for CCN that properly handles all packets
- Design an architecture of CCN Router
  - Fast lookup hierarchical name using **CAM and Bloom filter**
  - Identify **content worth caching**
- Evaluate performance and cost of the hardware

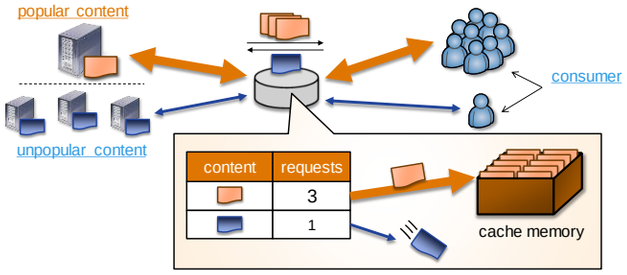
Lookup Using CAM and Bloom Filter

- CAM can search its entire memory in a single lookup.
- Bloom filter is low-cost and fast memory for testing membership.



Identify Content Worth Caching

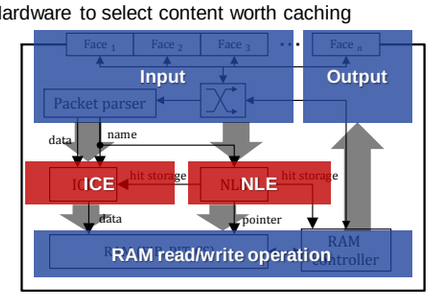
- Cache content depending on the number of requests
  - Small amount of popular content covers the majority of requests. (Ex. Only 10% of content requested more than three times covers 50% of traffic. [6])



[6]F. Guillemin, et al, "Experimental analysis of caching efficiency for YouTube traffic in an ISP network," in Proceedings of the 25th International Teletraffic Congress, September 2013, pp. 1–9.

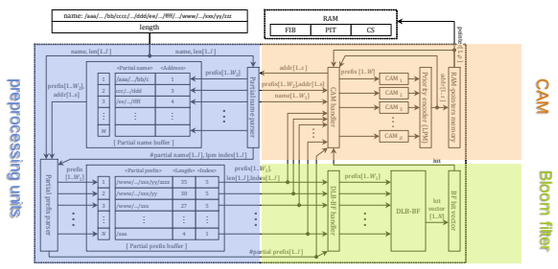
Hardware Design of Proposed CCN Router

- Name Lookup Entity (NLE)
  - Hardware to lookup name quickly
- Interest Count Entity (ICE)
  - Hardware to select content worth caching



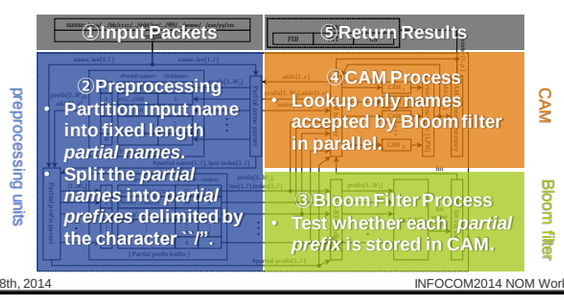
Hardware Design of Name Lookup Entity (NLE)

- NLE consists of Bloom filter, CAM, and two buffers.
- We split the monolithic CAM into multiple smaller CAM
  - Reduce the cost and power
  - Perform a lookup process in parallel and improve throughput



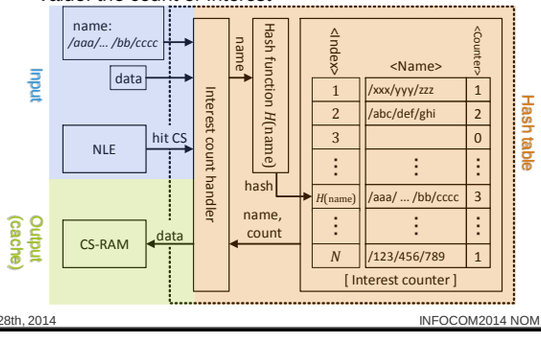
Hardware Design of Name Lookup Entity (NLE)

- NLE consists of Bloom filter, CAM, and two buffers.
- We split the monolithic CAM into multiple smaller CAM
  - Reduce the cost and power
  - Perform a lookup process in parallel and improve throughput



Hardware Design of Interest Count Entity (ICE)

- ICE is implemented as a simple hash table.
  - Key: a name address
  - Value: the count of Interest



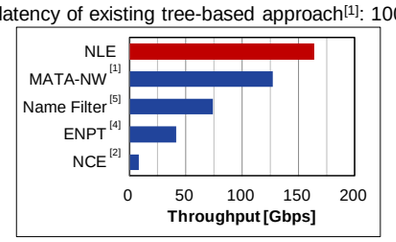
Evaluation

- Analyze the cost and performance of NLE.
  - Because the memory required for NLE implemented using CAM is the most serious problem.
  - We calculate the required memory size, cost of the memory, and throughput according to the hardware design.
- Parameter Setting
  - The number of entries: 10 million [1,2]
  - Average packet size: 256 B (Interest: 40 B, Data: 1500 B)
  - The length of name: 99% of names are no longer than 40 B and have no more than six components [3]

[1] Y. Wang, et al, "Wire speed name lookup: a GPU-based approach," in Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation, April 2013, pp. 199–212.
[2] Y. Wang, et al., "Scalable name lookup in NDN using effective name component encoding," in Proceedings of the IEEE 32nd International Conference on Distributed Computing Systems 2012, June 2012, pp. 688–697.
[3] D. Perino and M. Varvello, "A reality check for Content Centric Networking," in Proceedings of the ACM SIGCOMM workshop on Information-centric networking, August 2011, pp. 44–49.

Throughput and Latency

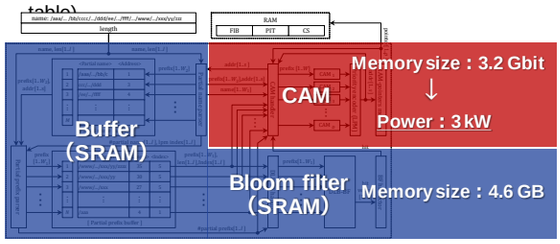
- Throughput: 163 Gpbs
  - We can greatly improve by concurrent lookups.
- Latency: 12.25 ns
  - The latency of existing tree-based approach[1]: 100 μs



[1] Y. Wang, et al, "Wire speed name lookup: a GPU-based approach," in Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation, April 2013.
[2] Y. Wang, et al., "Scalable name lookup in NDN using effective name component encoding," in Proceedings of the IEEE 32nd International Conference on Distributed Computing Systems 2012, June 2012
[4]H. Dai, et al, "On pending interest table in Named Data Networking," in Proceedings of the ACM/IEEE 8th Symposium on Architectures for Networking and Communications Systems 2012, October 2012
[5] Y. Wang, et al, "NameFilter: Achieving fast name lookup with low memory cost via applying two-stage bloom filters," in Proceedings of the IEEE INFOCOM 2013, April 2013

Required Memory Size and Cost

- SRAM (Buffer and Bloom filter): 4.6 GB
- CAM: 3.2 Gbit
  - 3.2 Gbit TCAM is impractical in terms of memory size and power.
  - We plan to use more scalable memory (ex. BCAM, hash table)

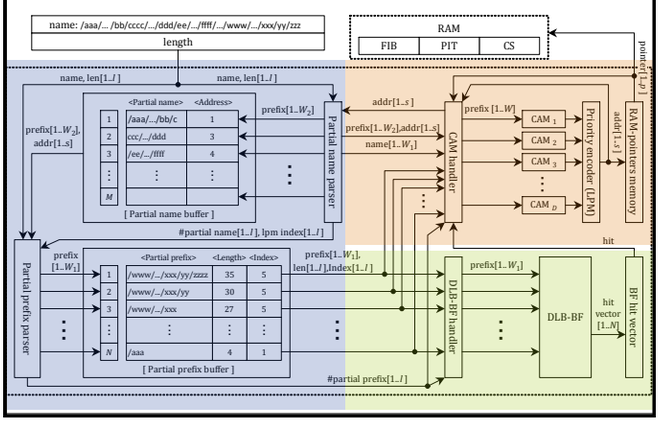


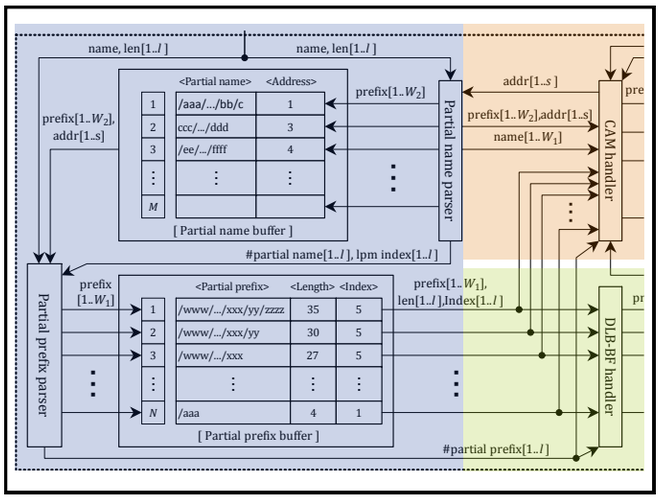
NLE

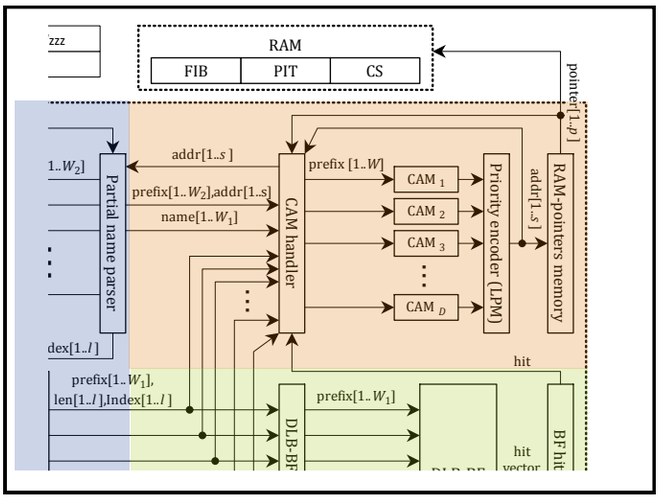
Summary and Future Work

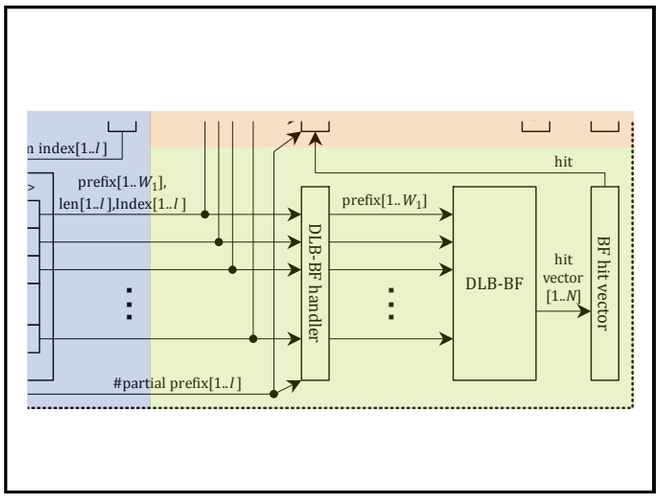
- Summary
  - We designed a CCN router hardware architecture.
    - NLE: Fast lookup name using CAM and Bloom filter
    - ICE: Select content worth caching
  - We evaluated the throughput and cost of a CCN router.
- Future work
  - Scale the memory capacity and the number of entries
  - Evaluate a CCN router performance based on a hardware implementation
  - Evaluate network-level performance using the implemented CCN router

Hardware Design of NLE




name, len[1..l]
name, len[1..l]
<Partial name>
<Address>
prefix[1..W2],
addr[1..s]
prefix[1..W2]
addr[1..s]
prefix[1..W2],addr[1..s]
name[1..W1]
Partial name parser
CAM handler
[Partial name buffer]
#partial name[1..l], lpm index[1..l]
Partial prefix parser
prefix
[1..W1]
<Partial prefix>
<Length>
<Index>
prefix[1..W1],
len[1..l],Index[1..l]
DLB-BF handler
[Partial prefix buffer]
#partial prefix[1..l]


RAM
FIB
PIT
CS
pointer[1..p]
addr[1..s]
prefix [1..W]
prefix[1..W2],addr[1..s]
name[1..W1]
Partial name parser
CAM handler
CAM 1
CAM 2
CAM 3
CAM D
Priority encoder (LPM)
addr[1..s]
RAM-pointers memory
hit
prefix[1..W1],
len[1..l],Index[1..l]
prefix[1..W1]
hit
vector


prefix[1..W1],
len[1..l],Index[1..l]
hit
prefix[1..W1]
DLB-BF handler
DLB-BF
hit
vector
[1..N]
BF hit vector
#partial prefix[1..l]