国立研究開発法人
情報通信研究機構

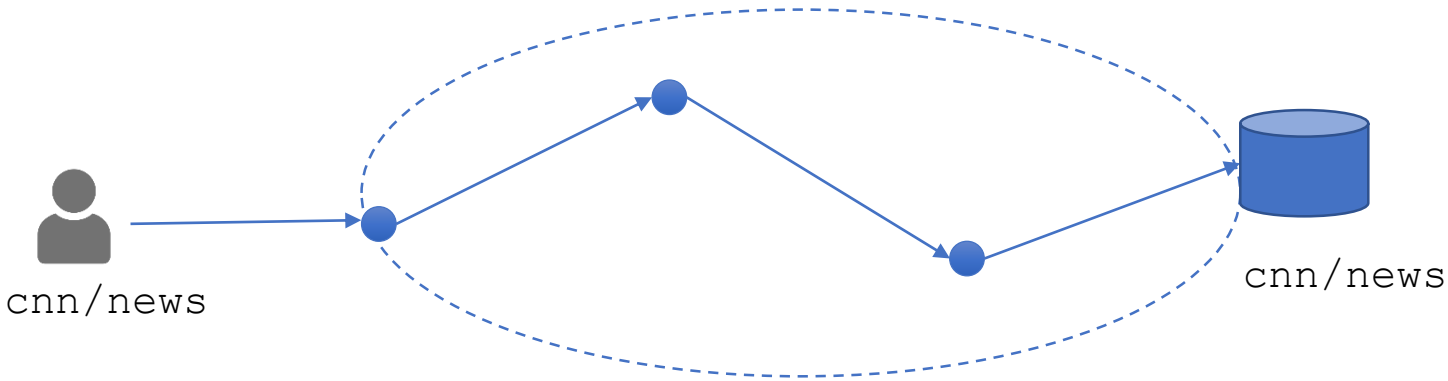
Small-size Low-replacement FIB Management

Dinh Nguyen, Hitoshi Asaeda
National Institute of Information and Communications Technology (NICT)
IEICE Technical Committee on ICN- Dec 14th, 2018

Content

- FIB management in NDN/CCN
  - Why needs FIB management?
  - FIB miss and FIB replacement problem
- Small-size low-replacement FIB management
- Evaluation
- Conclusion and future work

Name-based Routing and Forwarding

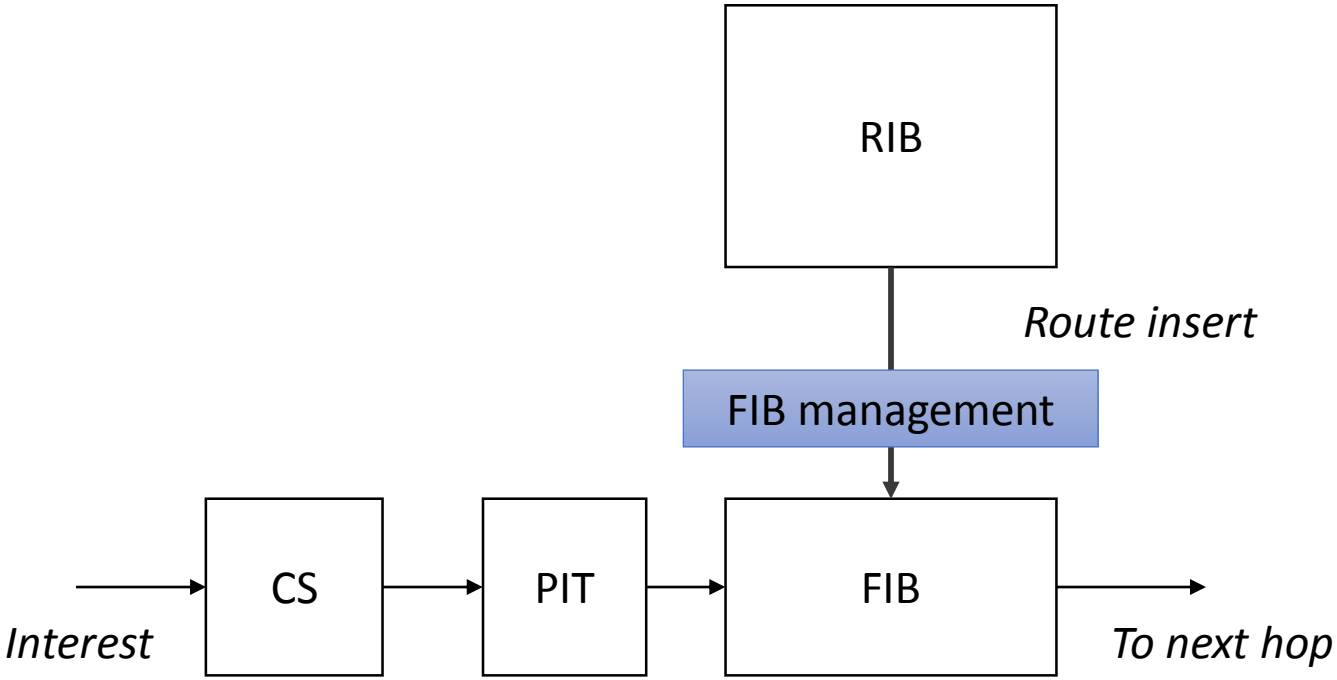


- Are centered components of NDN/CCN
  - Mapping between names and next hops
- Routing populates RIB (route information base)
  - Precomputed (proactive)
  - On-demand (reactive)
- RIB is then pushed to FIB (fwd. information base)
- FIB lookup on receiving every Interest
  - Find next hop to forward the Interest to
  - Must support line-speed forwarding
  - Deployed in fast memory like SRAM/TCAM

FIB is tiny compared to RIB

- RIB tables are large: (est.) 10^9 entries
  - Name prefixes: 10^8 (domain names) now, 10^9 soon
  - For comparison: BGP routing table size < 10^6
- FIB cannot hold large number of entries
  - Largest SRAM/TCAM size is about a few 100Mb [1] ~ 10^6 prefixes
  - Full RIB cannot fit into SRAM/TCAM for line-speed packet forwarding
- Routers have to work with partial FIB
  - Route discovery for prefixes not existed in FIB: flooding, looking up route service, or querying RIB

FIB management

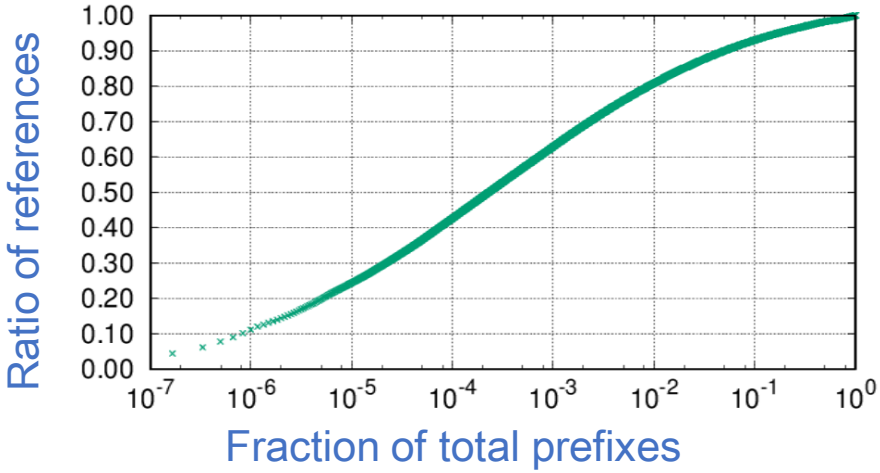


- FIB mgmt. controls which prefixes to insert/remove from FIB
- Required to keep the *best* prefixes in FIB
  - High probability that Interest finds its prefix in FIB
  - Low replacement rate to avoid FIB blocking

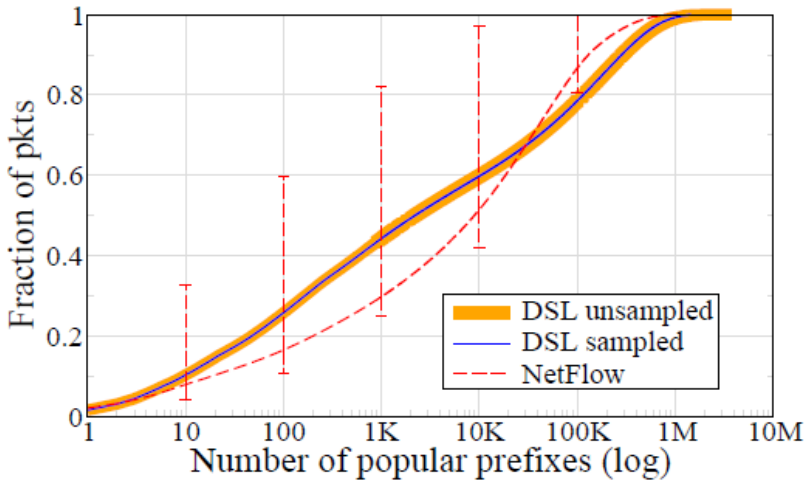
Pros & Cons of FIB management

- Pros
  - Much smaller FIB size: less than 10^{-3} full size
  - Deployable in fast memory SRAM/TCAM
  - Potentially faster lookup
- Cons
  - May have different lookup results compared with full table lookup. Avoidable by either
    - Specifying routable part of name & exact matching [3]
    - Use almost flat routable prefixes [2]
      - E.g.: use [www.bbc.com] instead of [com/bbc/www]

Name prefix skewness



6 million **URL names** from http://www.icn-names.net/



9.3 million /24 **IP prefixes** from traffic traces [5]

[+] Zipf distribution

- good for FIB management

[−] Much larger name space

- Affordable FIB size is about 10^{-4}~10^{-3} of total number prefixes

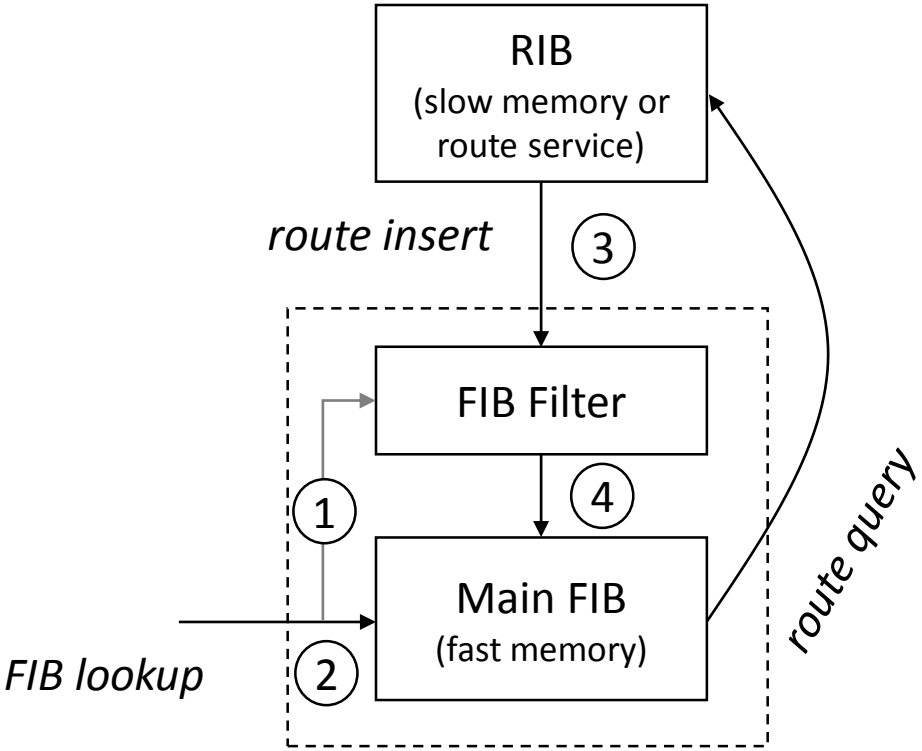
Existing FIB Management

- Use off-the-shelf policy
  - Recency-based (LRU) let every prefix come into FIB regardless of its popularity [3]
  - Frequency-based (LFU) not good in route caching, not adaptive to change [4]
  - Time-to-live (TTL) to deal with other issue (maintaining in-progress sessions) [2]
- High FIB miss rate
  - Happens if Interest carries a prefix not existed in FIB
  - Causing delay to retrieve missing routes: from centralized route information system, RIB, or flooding
- High FIB replacement rate
  - When a prefix is inserted/replaced another FIB entry
  - FIB insert/replacement blocks packet forwarding

Motivation

- FIB management for fast-memory FIB deployment
  - Extremely small-size: 10^{-4}~10^{-3} of est. # prefixes
  - Extremely low replacement rate
- Main idea: filter low-frequency prefixes from FIB
  - Increase hit
  - Minimize replacement

Proposed FIB management



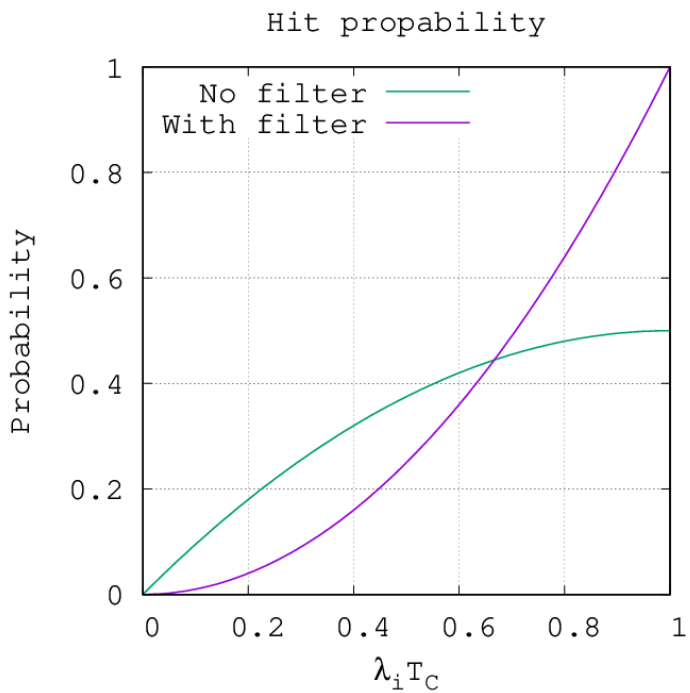
Proposed FIB Management

- <u>FIB Lookup</u>
  - ① Lookup FIB filter
    - Update frequency
  - ② Lookup main FIB
    - Return route if found
- <u>Route Insert</u>
  - ③ If prefix not in FIB filter
    - // low-freq. prefix
    - Insert to FIB filter
  - ④ Else
    - // high-freq. prefix
    - Insert into main FIB

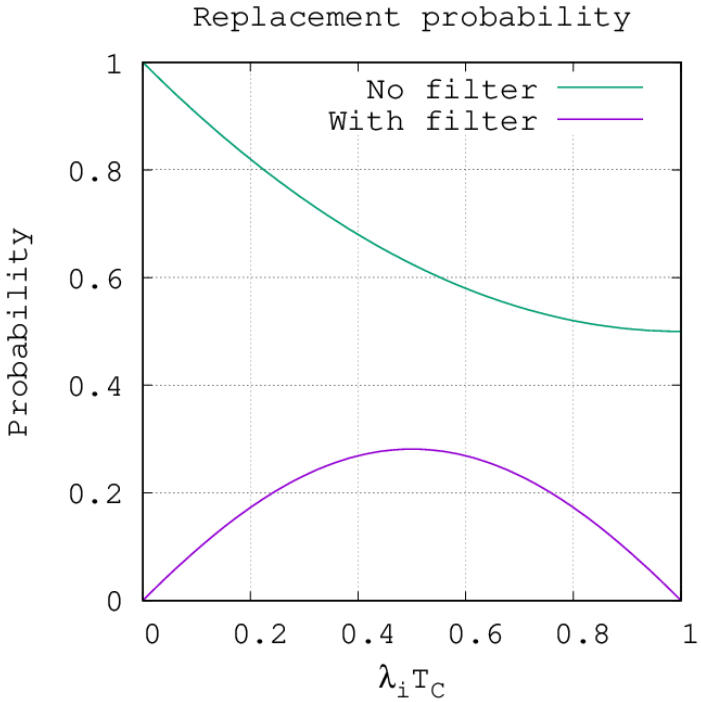
Replacement policy

- Main FIB
  - Recency-based policy (e.g. LRU) has been shown to work best in IP route caching
- FIB filter
  - Keep track of recent access frequency
  - Should be *cheap* in implementation: memory space, process time
- LRU is a good type of filter too
  - Simple
  - Recent highest-frequently accessed prefixes are more likely to exist

Analyzing LRU (filter)+ LRU (FIB)



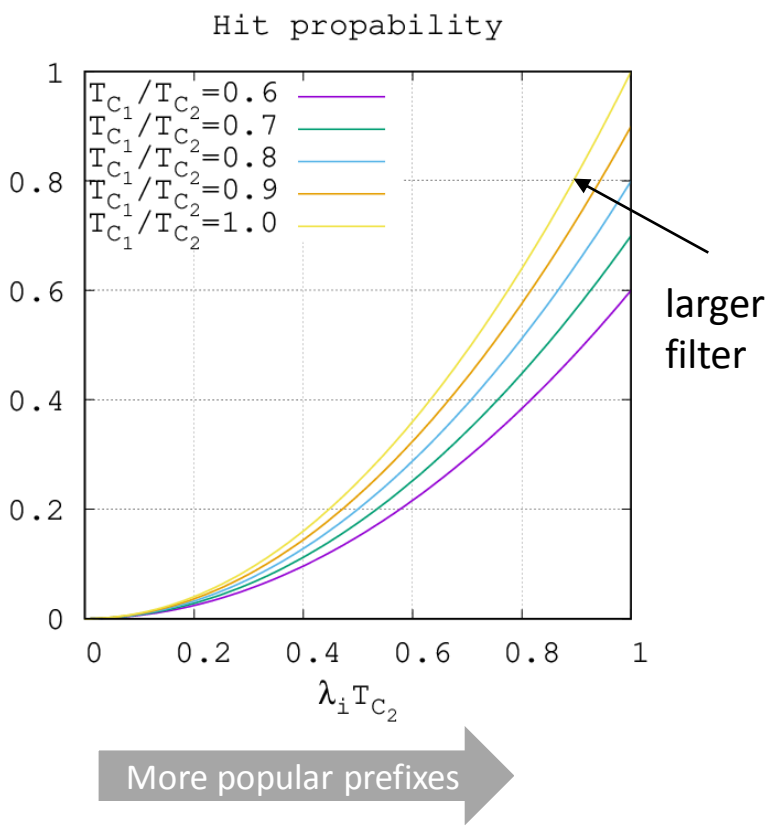
More popular prefixes

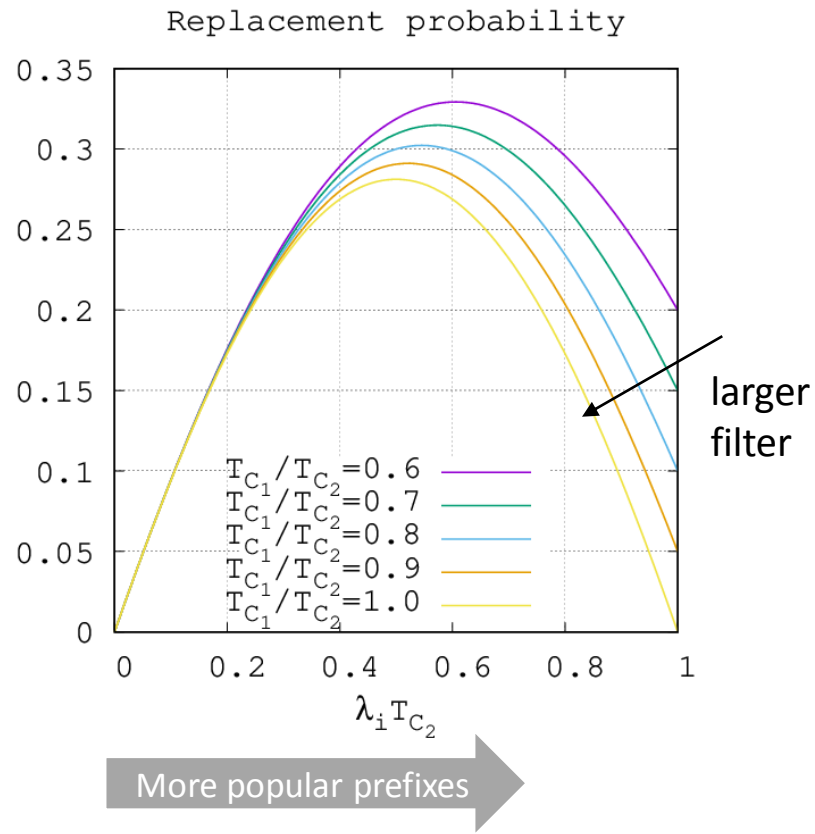


More popular prefixes

- FIB filter effectively lowers replacement probability
- FIB hit is higher with more popular content

FIB filter size



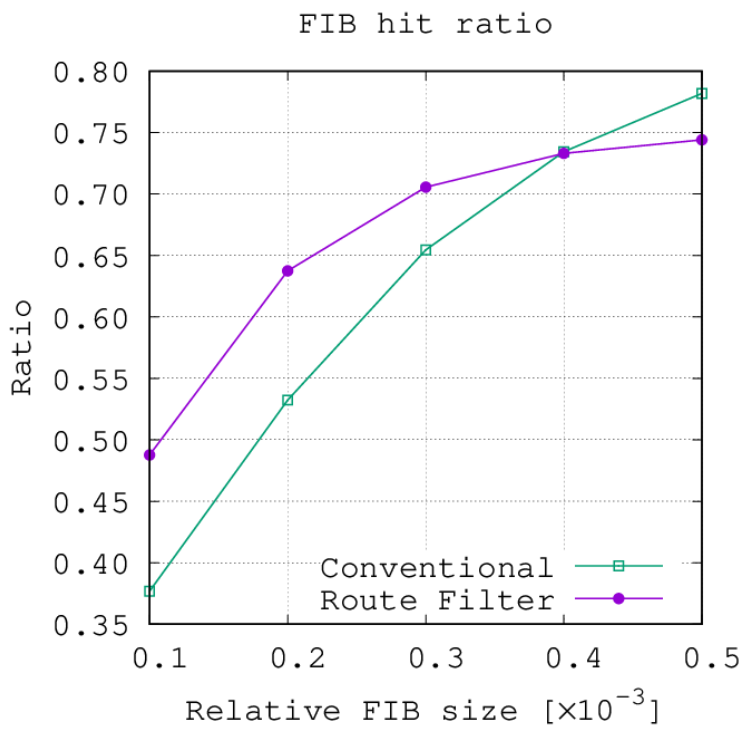


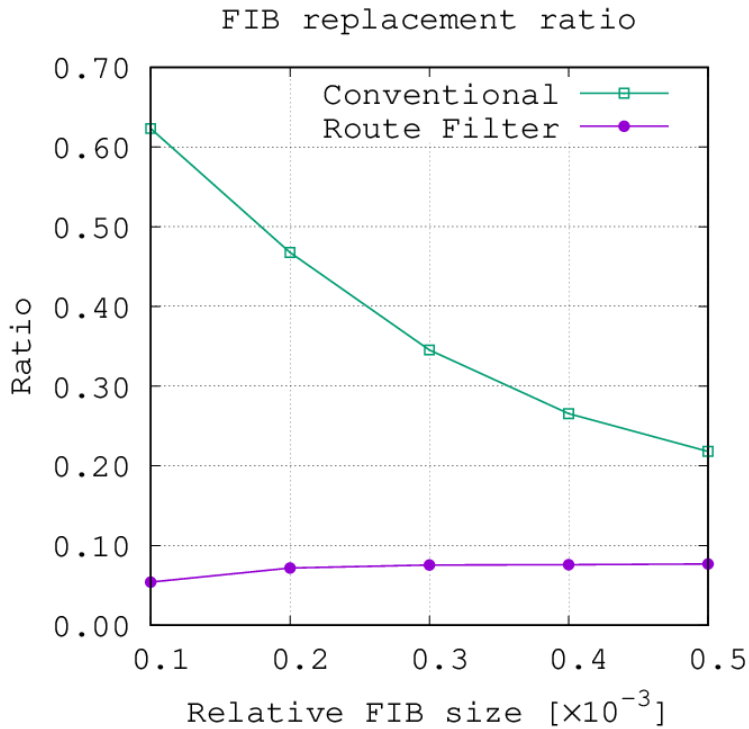
- Larger FIB filter has better performance
- But cannot be too large

Simulation evaluation

- Tier-1 topology: Rocketfuel AS3257 (Tiscali, EU)
  - 161 nodes, 328 links
  - select 80 lowest degree nodes as access nodes
- Traffic
  - 10^5 prefixes, Zipf 0.8, hosted at the access nodes
  - Content size distribution: Pareto(tail)/Lognormal(body)
  - Poisson arrival
  - Pipeline Interest control (W=1)
- Cache 2% catalog size
- Measure hit and replacement rates
  - Conventional FIB management (LRU)
  - Proposed FIB management (Route Filter)
    - Filter size ≈ FIB size: $\sim 10^{-4}$ #prefixes

Impact of FIB size





- Always has low replacement rate
- FIB filter works very well at small FIB size
  - Large FIB size: slight lower hit rate

Conclusion and future work

- Proposed FIB management
  - FIB filter to keep low-frequency prefixes out of FIB
  - Work with extreme small FIB size and reduce FIB replacement
- LRU FIB filter is good
  - At small FIB sizes
  - Very low replacement rate
  - Simple
- Future work
  - Evaluate with real traces
  - Incorporate with route protocols

Reference

1. Perino, D., & Varvello, M. A reality check for content centric networking, ICN'11
2. A. Detti et al., "Supporting the Web with an information centric network that routes by name," ComNetw2012
3. O. Ascigil et al., "On-Demand Routing for Scalable Name-Based Forwarding," *ACM ICN'18*
4. Schmidt, T. C et al. Let's collect names: How PANINI limits FIB tables in name based routing. *IFIP Networking'16*
5. Kim, C., Caesar, M., Gerber, A., & Rexford, J. (2009). Revisiting route caching: The world should be flat. *Lecture Notes in Computer Science*,

Thank you!

Dinh Nguyen, Hitoshi Asaeda
Small-size Low-replacement FIB Management
IEICE Technical Committee on ICN Dec.2018