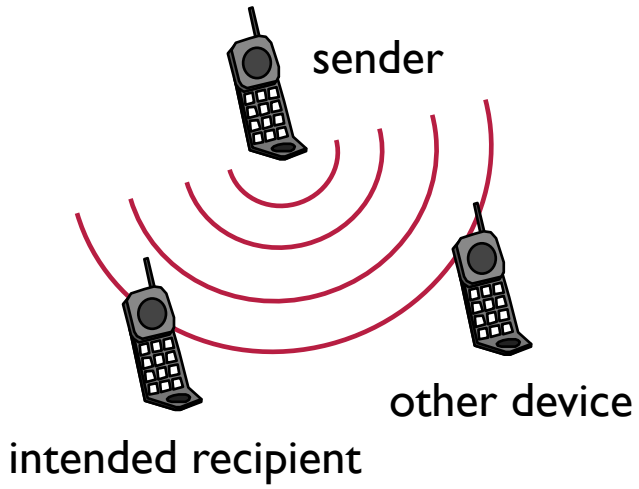
NDN-NIC: Name-based Filtering on Network Interface Card

Junxiao Shi, Teng Liang, Beichuan Zhang (University of Arizona)

Hao Wu, Bin Liu (Tsinghua University)

Communication over shared media

- Each device hears all signals transmitted within range, and needs to processes them.
  - Accept interested packets, discard those of no interest.
  - E.g., traditional Ethernet, and most wireless communication.
- In current systems, this is done at the network interface card (NIC) to save cycles and power of the main system.
  - Compare the L2 destination address with the host's own address.



NDN over shared media

- Can take advantage of the broadcast nature
  - Consumer doesn't need to specify the L2 destination address; any receiver with the content can reply.
  - Especially useful in wireless mobile networks.
- But implementing it in current systems means all packets are delivered to the main CPU for filtering.
  - Looking up PIT/CS/FIB before dropping unwanted packets.
- ***Can we design a NIC that can filter NDN packets based on their names?***

NDN NIC

- Goal: filtering packets based on names.
- Requirements:
  - No false negatives
    - Interests that match CS/PIT/FIB must be admitted.
    - Data that match PIT must be admitted.
  - Can have false positives but the fewer the better
  - Support regular end-hosts with small amount of memory.
    - 100s of thousands of names vs. 10s KB memory.
- Idea: put names into Bloom filters on NIC.
  - Incoming packets are admitted if they find a match in the BFs.
- Challenge: with limited memory, choose which names to go into the filters to satisfy the requirements.

Name Matching

- The naïve approach is to store all names from CS/PIT/FIB in BF, but it's actually more complicated than that.
  - Some names should not be stored in the BF. E.g., the default route in FIB, and PIT entries forwarded to local producers.
- NDN has two types of packets, and name matching is more than exact match.

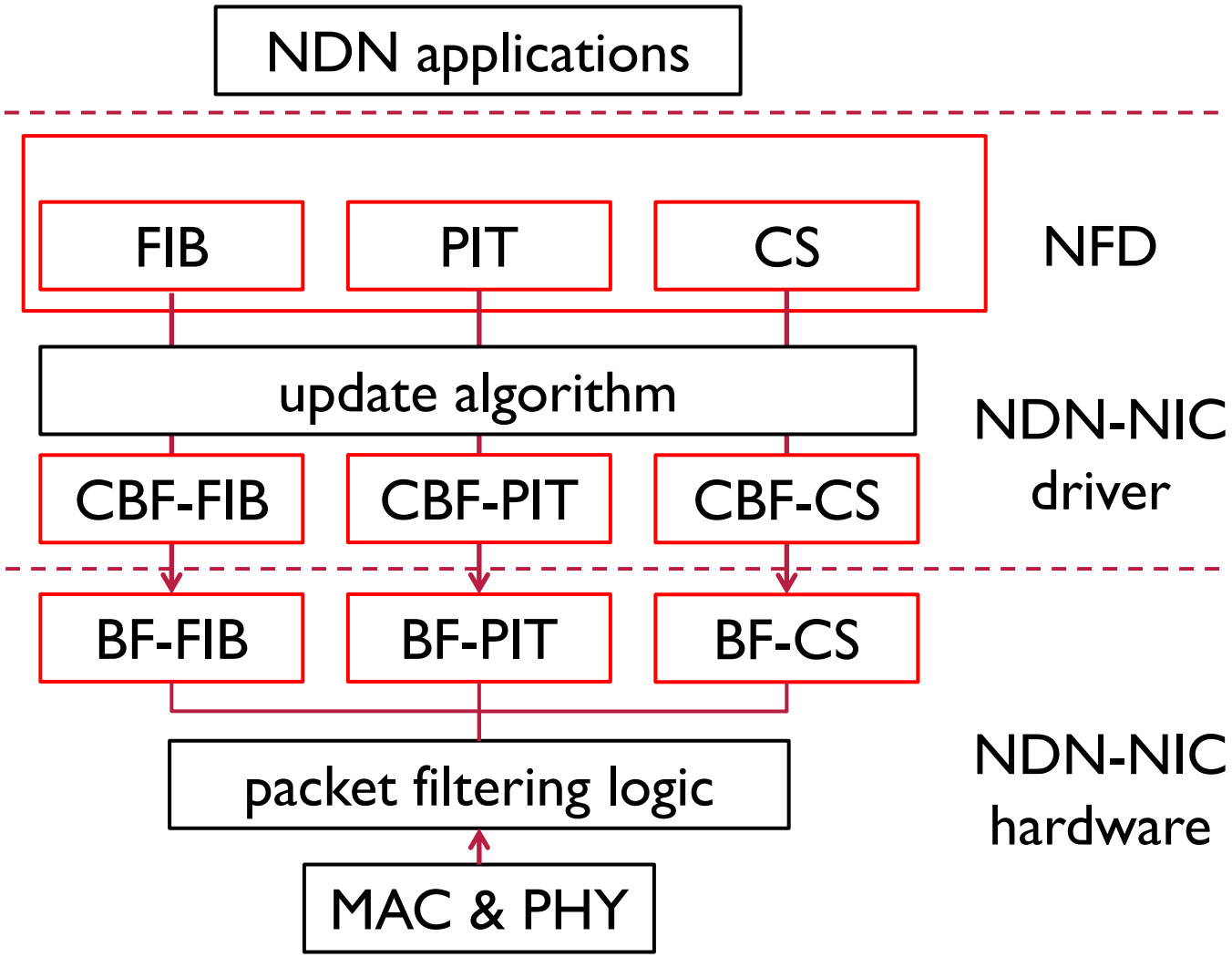
Name Matching and Bloom Filters

- FIB:
  - /A/B can match interest name /A/B, /A/B/C, or longer.
  - Store /A/B in BF
  - Filtering interests by looking up all prefixes of the interest name.
- CS:
  - /A/B/C/D can match interests /A, /A/B, /A/B/C, and /A/B/C/D.
  - Store /A/B/C/D and all its prefixes in BF.
  - Filtering interests by looking up the interest name.
- PIT:
  - /A/B/C matches Data with same or longer names, e.g., /A/B/C/D.
  - Store /A/B/C and all prefixes.
  - Filtering Data by looking up Data name and all its prefixes.
- Need three different Bloom Filters with different contents and different filtering procedures.

Name Removal From the Filters

- When CS/PIT/FIB are updated, the Bloom Filters may also need to be updated, otherwise will lead to false negatives or false positives.
- Name addition is easy for BF, but name removal needs counting Bloom filters, which increases memory usage.
- Maintain counting Bloom filters in the driver, only store BFs in NIC.

NDN-NIC Architecture

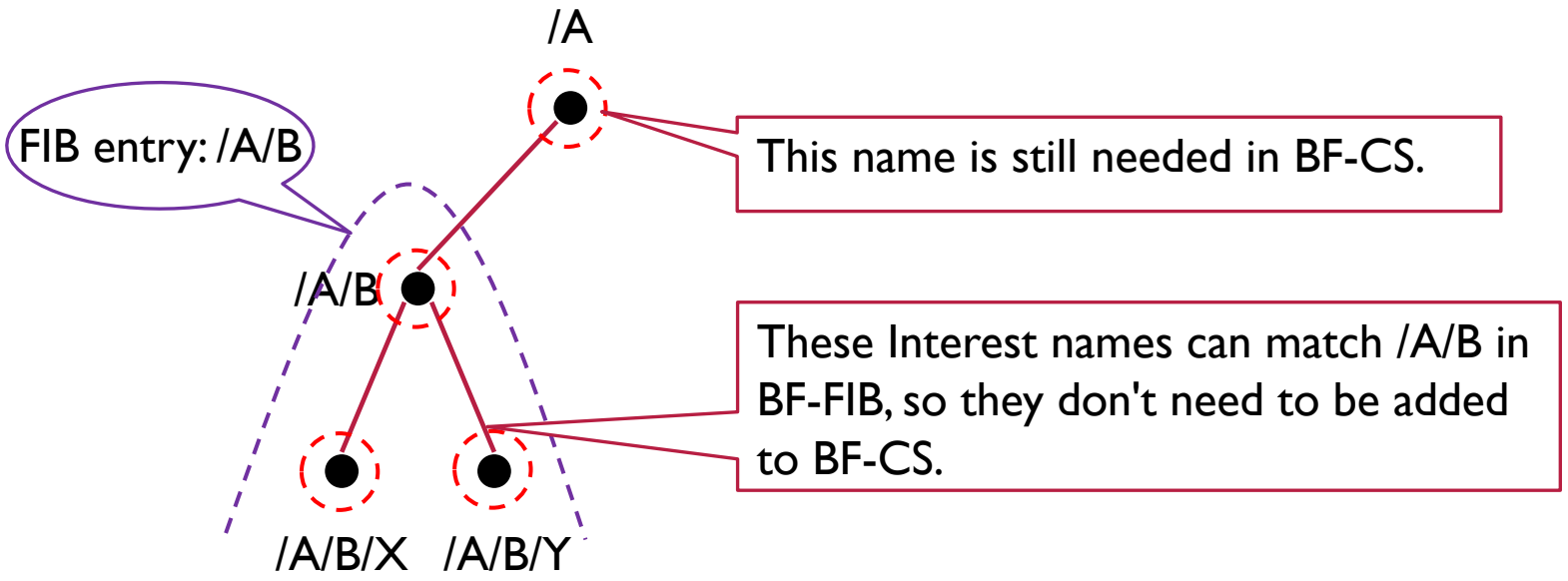


Reducing False Positives

- The more names added to a Bloom filter, the higher its false positive rate.
  - E.g., a 8KB BF with optimal hash functions, false positive 4.3% with 10K names, but 54% with 50K names.
- BF-CS is the primary optimization target
  - A regular host usually has a small FIB and a relatively small PIT.
  - But it can cache many Data objects in the CS, and all those names and their prefixes need to go into BF-CS.
- The idea is to introduce a shorter prefix in the BF to replace multiple CS names.

The Basic CS Algorithm

- Skip a CS name if it is already covered by a FIB entry.
- Doesn't introduce any new false positive.

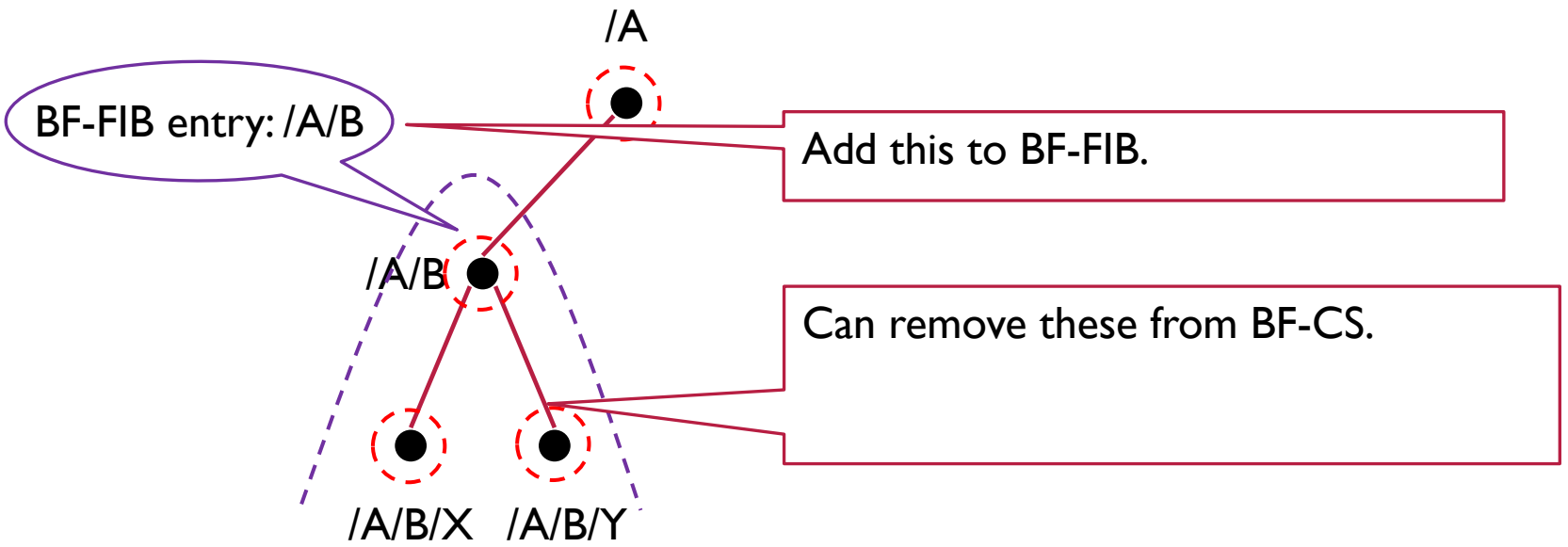


When Basic CS will be effective?

- Basic CS is effective when the CS caches many Data from local producer applications, because those Data names are covered by FIB entries registered by local applications.
  - i.e., content producers
- If most cached Data came from the network, their names aren't under FIB prefixes, and Basic CS will not work well.
  - i.e., content consumers.

The Active CS Algorithm

- Create a new prefix in BF-FIB so we can remove multiple names in BF-CS.
  - Reduce BF false positives since names are fewer.
  - But introduce ***prefix match false positive***, e.g., /A/B/Z will be admitted.



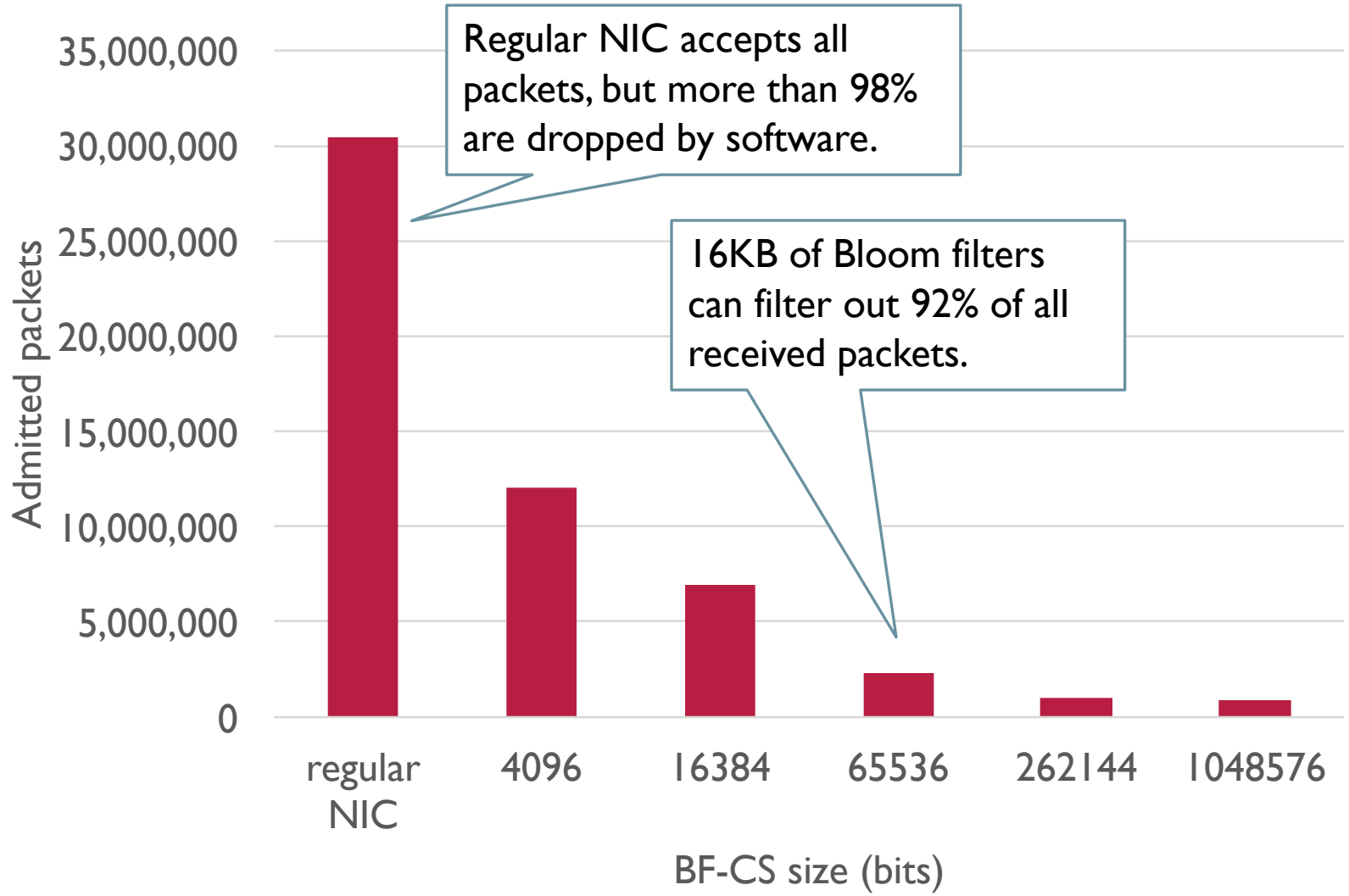
Active CS

- Goal of Active CS is to minimize overall false positives.
  - Bloom filter false positives can be estimated based.
  - But prefix match false positives depends on traffic pattern.
- We don't calculate prefix match false positives. Instead, Active CS tries to keep Bloom filter false positives between two thresholds.
  - Aggregate if BF false positive is high.
  - De-aggregate if BF false positive is low.

Evaluation

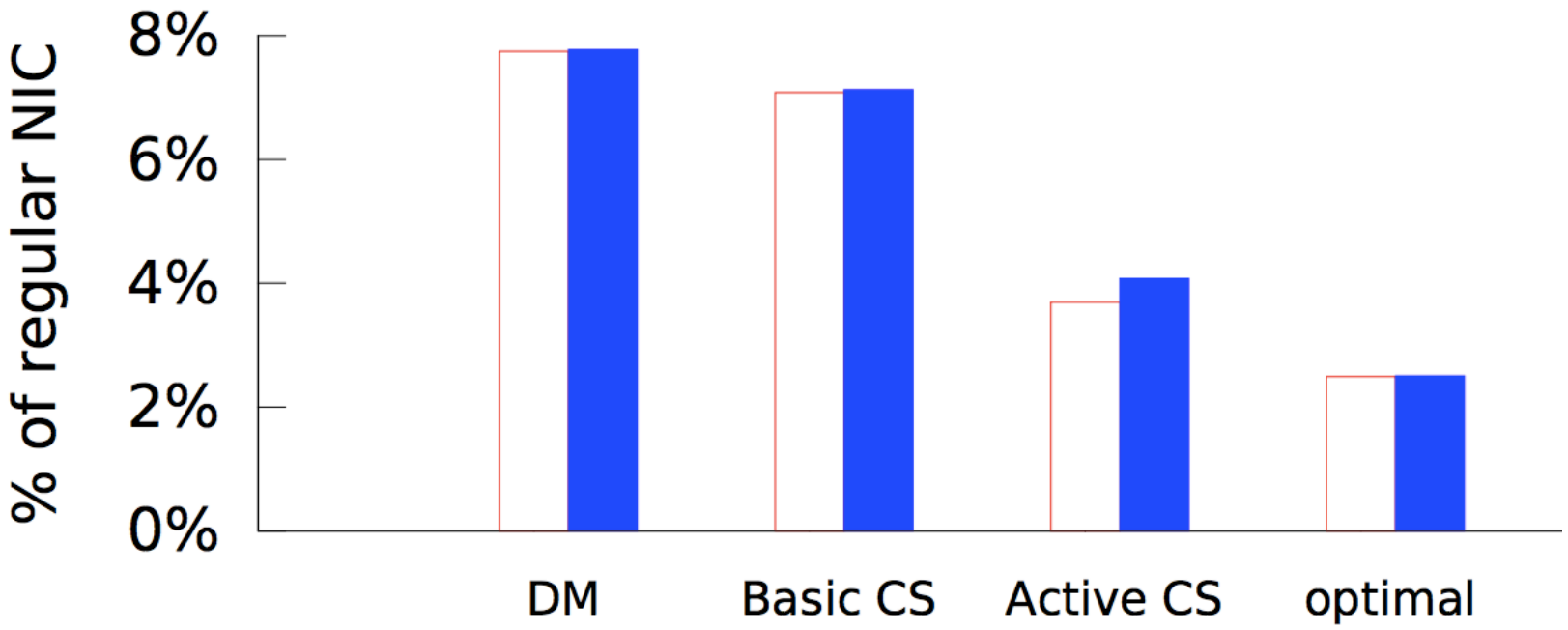
- Collect NFS traces in a department network.
- Use the NFS requests and replies to derive NDN interest names and packet timing, and use that to drive the emulation of NDN file access in mini-NDN.
- Record the interest and data packets, their timing, and table changes in NFD.
- Simulate and compare direct mapping, basic CS and active CS with different paramters:
  - How many packets are rejected by NDN-NIC?
  - How many packets are accepted but eventually dropped in NFD (false positives)?
  - The overhead?

Packets accepted by NIC



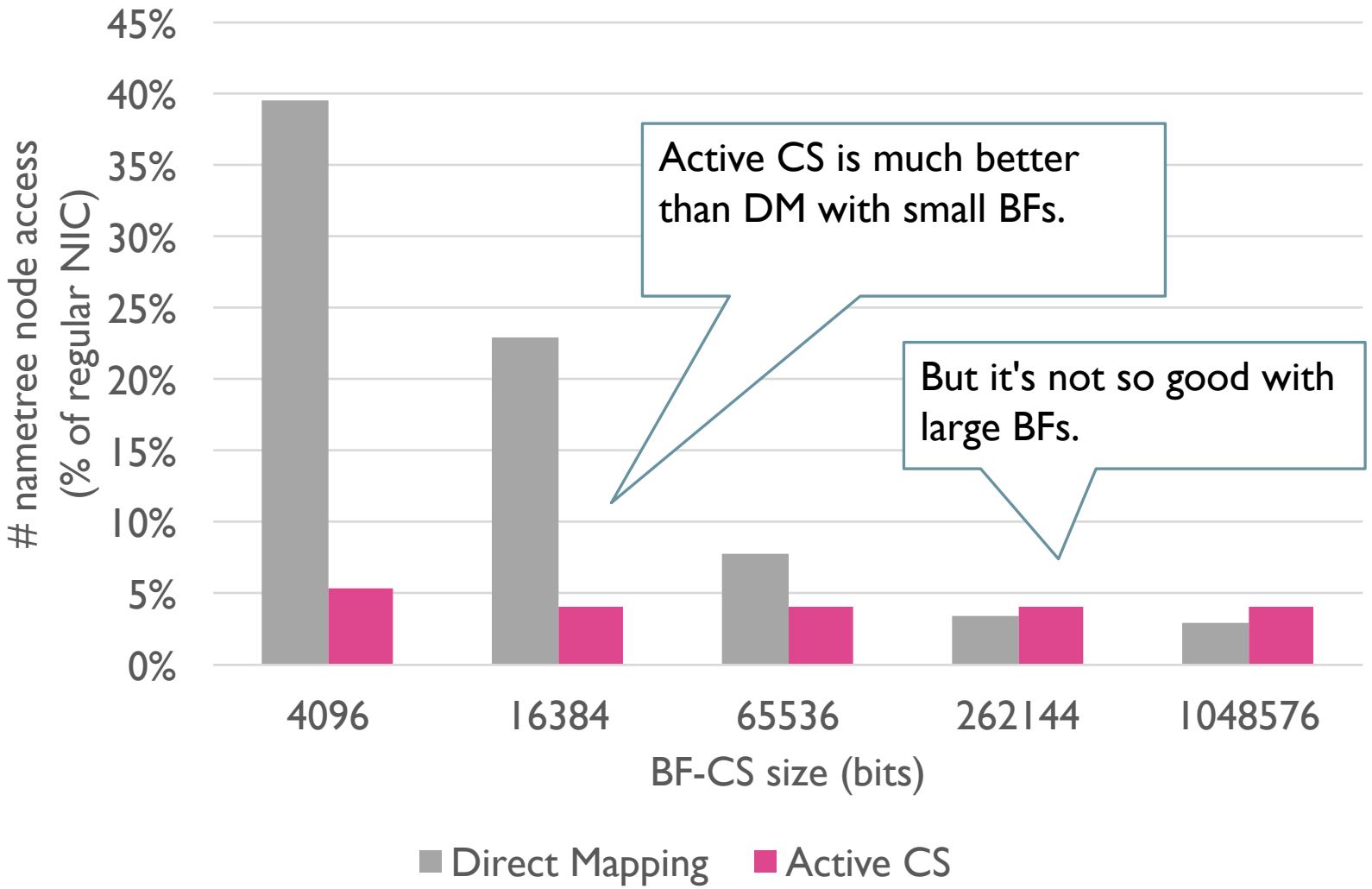
(65536-bit BF-FIB and 256-bit BF-PIT; Direct Mapping; two hash functions)

Compare different CS algorithms



- Improvement by basic CS is limited.
- Active CS helps more.

When to use Active CS?



(65536-bit BF-FIB and 256-bit BF-PIT)

Summary

- On shared media, filter packets in the NIC to save cycles and power of the main system.
  - Can be made effective using small amount of memory.
  - Active CS algorithm: tradeoff between two types of false positives.
- Future Work
  - Feasibility in hardware implementation: overhead and cost, update dynamics and the implications.

Q & A