Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol

Li Fan, Pei Cao and Jussara Almeida

University of Wisconsin-Madison

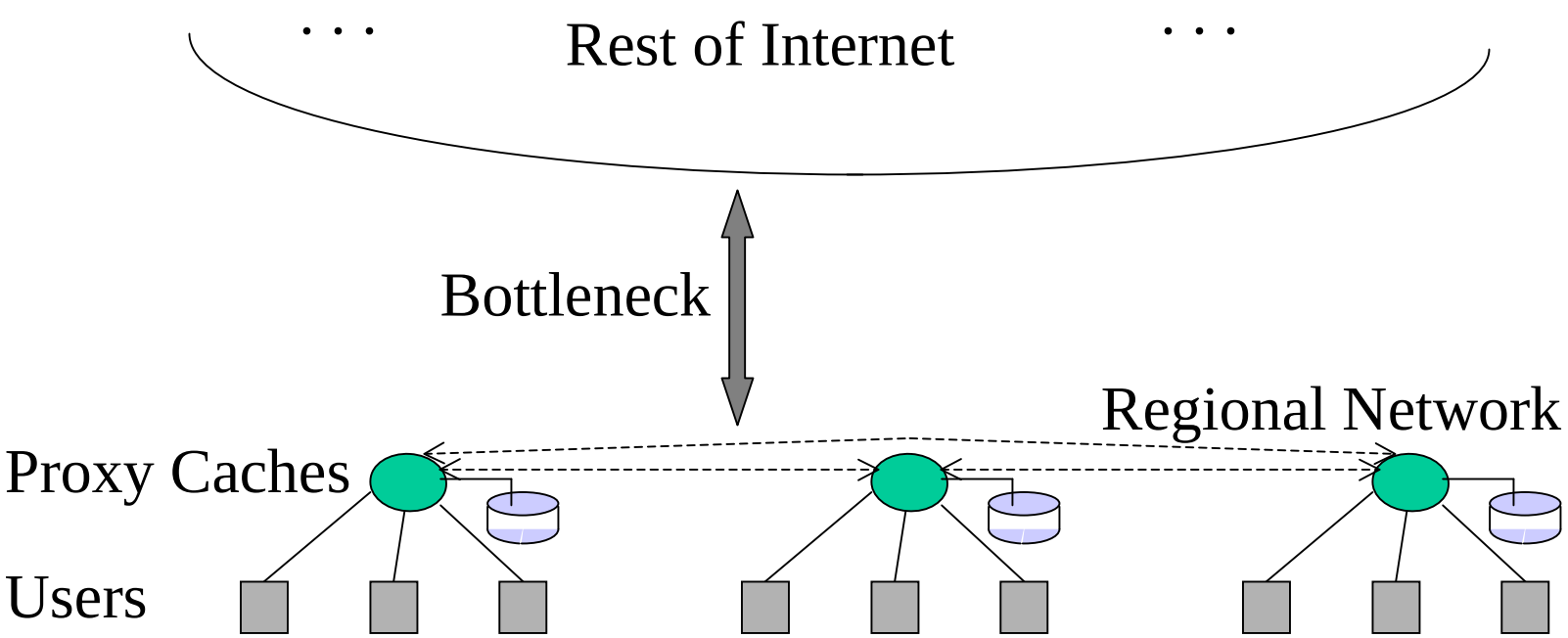
Andrei Broder

Compaq/DEC System Research Center

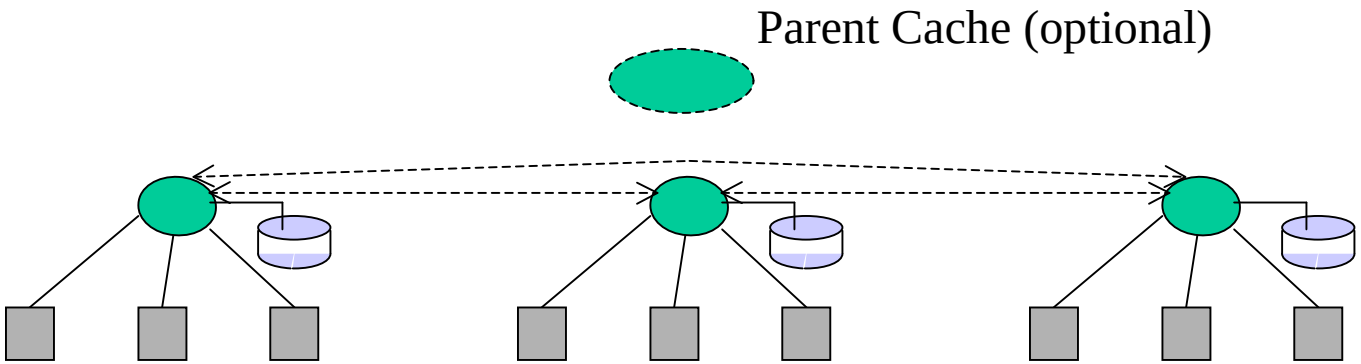
Why Web Caching

- One of the most important techniques to improve scalability of the Web
- Proxy caches are particularly effective

Why Cache Sharing?



Cache Sharing via ICP



- When one proxy has a cache miss, send queries to all siblings (and parents): “do you have the URL?”
- Whoever responds first with “Yes”, send a request to fetch the file
- If no “Yes” response within certain time limit, send request to Web server

Overhead of ICP

- #_of_queries = (#_of_proxies * average_miss_ratio) * #_of_proxies
- Experiments
  - **4** Squid proxies running on dual-processor 64MB SPARC20s linked with 100BaseT Ethernet links
  - Workloads: traces and synthetic benchmarks
- Compared with no cache sharing, ICP:
  - increases total network packets to each proxy by 8-29%
  - increases CPU overhead by 13-32%
  - increases user latency by 2-12%

Alternatives to ICP

- Force all users to go through the same cache or the same array of caches
  - Difficult in a wide-area environment
- Central directory server
  - Directory server can be a bottleneck

Ideally, one wants a protocol:

- keeps the total cache hit ratio high
- minimizes inter-proxy traffic
- scales to a large number of proxies

Summary Cache

- Basic idea:
  - Let each proxy keep a directory of what URLs are cached in every other proxy, and use the directory as a filter to reduce number of queries
- Problem 1: keeping the directory up to date
  - Solution: delay and batch the updates => directory can be slightly out of date
- Problem 2: DRAM requirement
  - Solution: compress the directory => imprecise, but inclusive directory

Errors Tolerated

- Suppose A and B share caches, A has a request for URL r that misses in A,
  - false misses: r is cached at B, but A didn't know

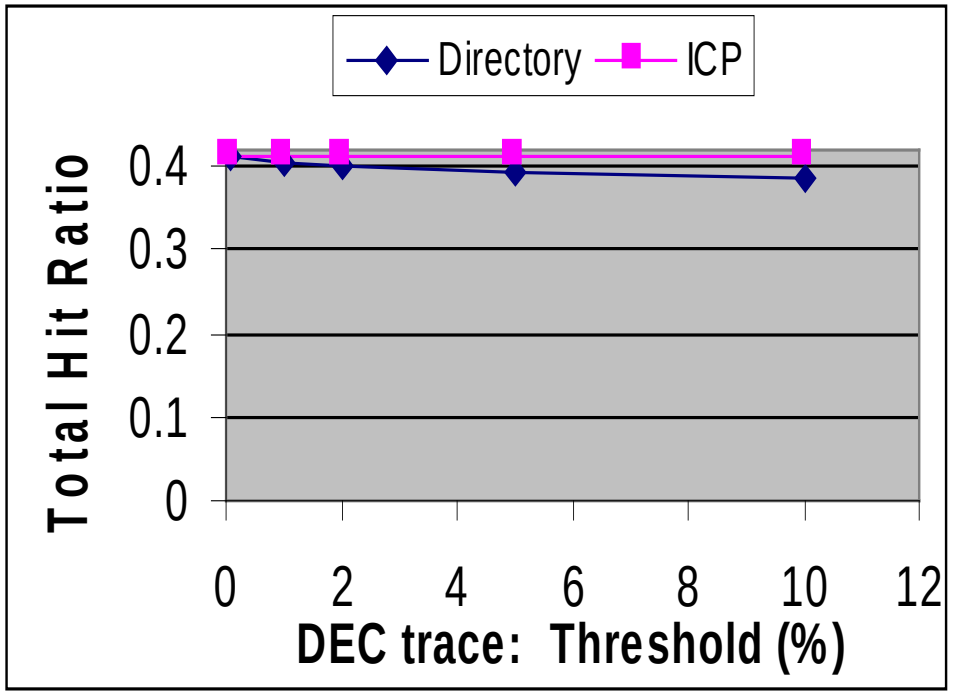
    Effect: lower total cache hit ratio
  - false hits: r is not cached at B, but A thought it is

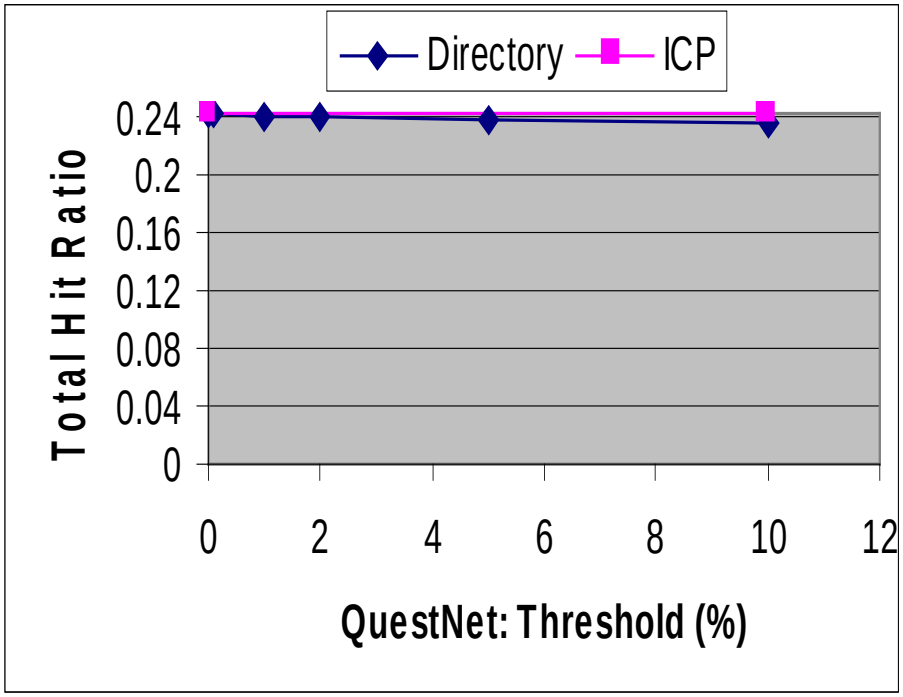
    Effect: wasted query messages
  - stale hits: r is cached at B, but B's copy is stale

    Effect: wasted query messages

Effect of Delay in Directory Updates

- Method: delay the updates until a certain percentage of the cached documents are “new”



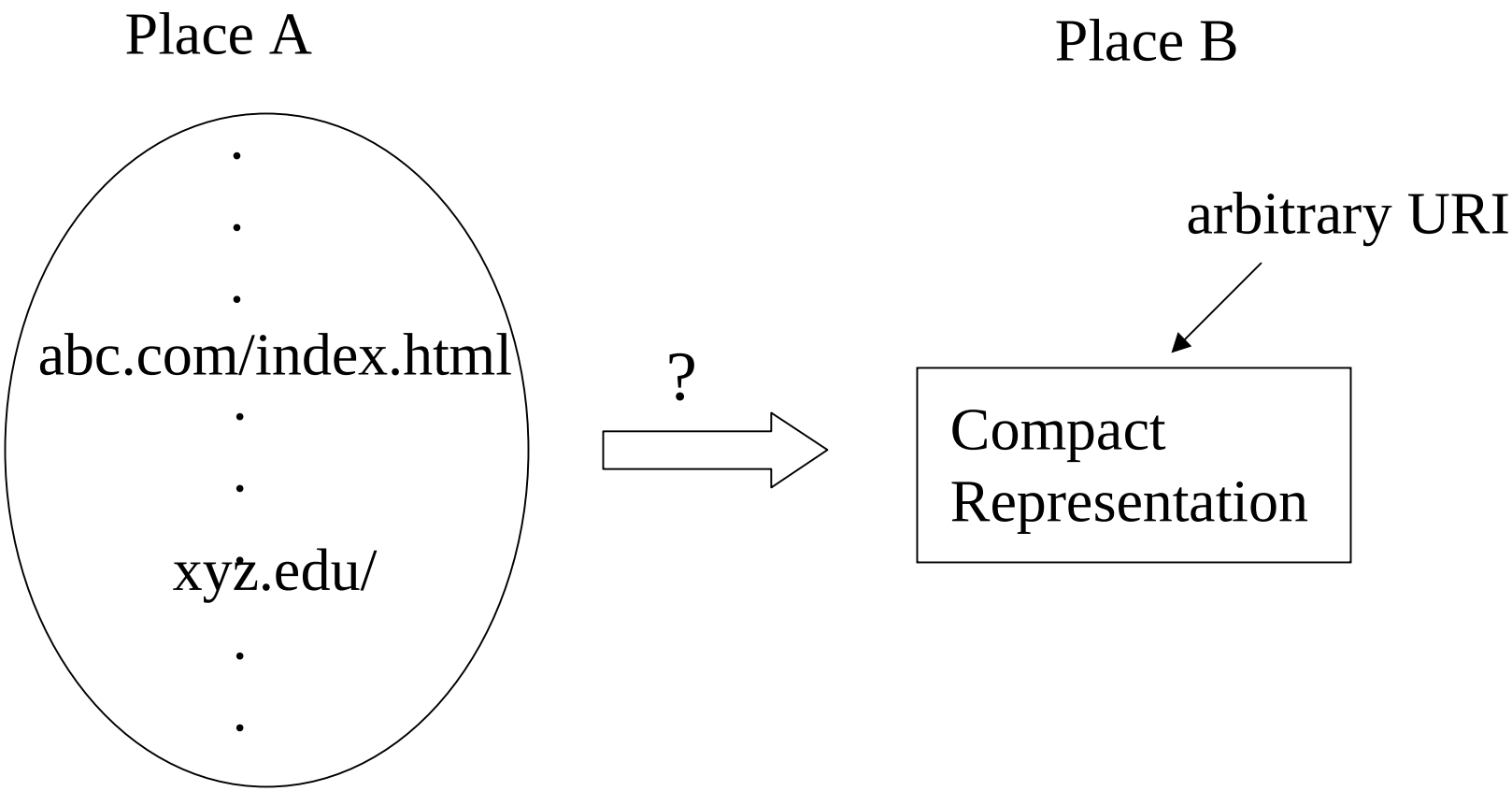


Compressing the Directories

- Requirements:
  - Inclusive
  - Low false positives
  - Concise

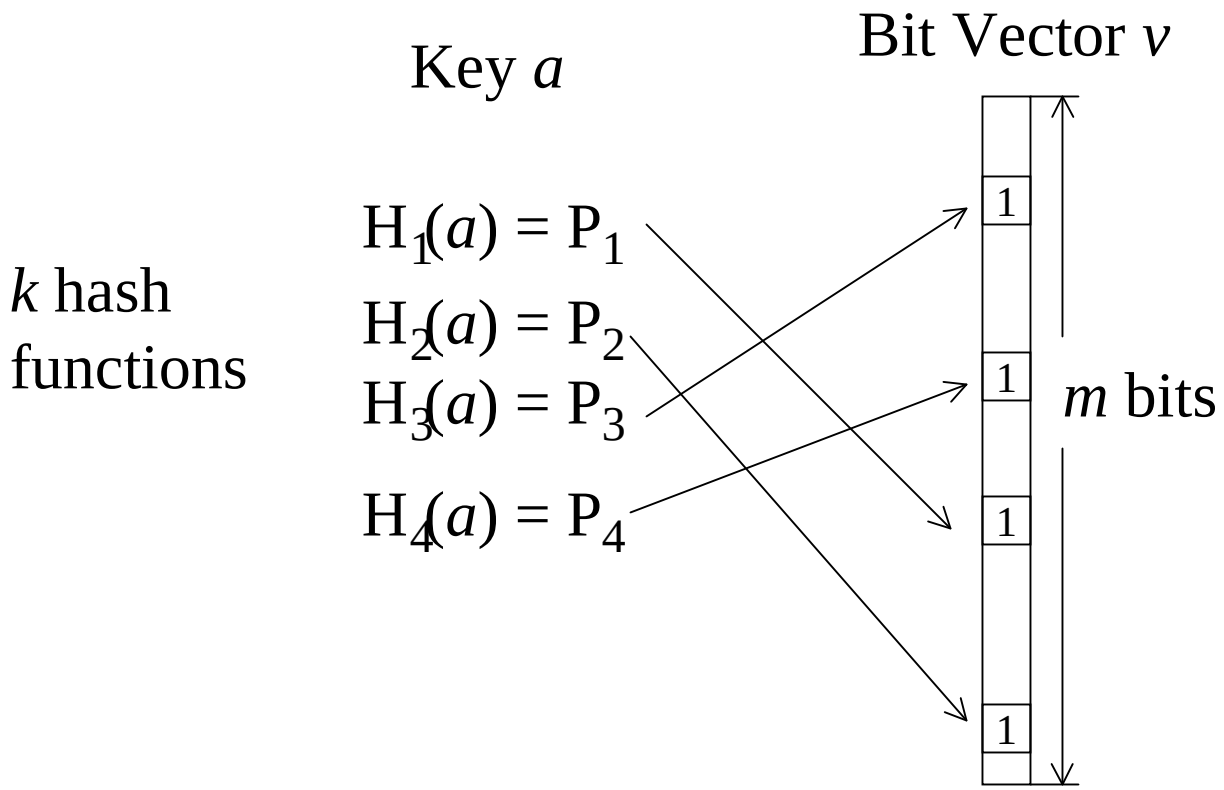
  we call the compressed directories “summaries”
- First try: use server URLs only
  - Problem: too many false hits, leading to too many messages between proxies

The Problem


Place A
Place B
abc.com/index.html
xyz.edu/
?
arbitrary URI
Compact
Representation

Bloom Filters

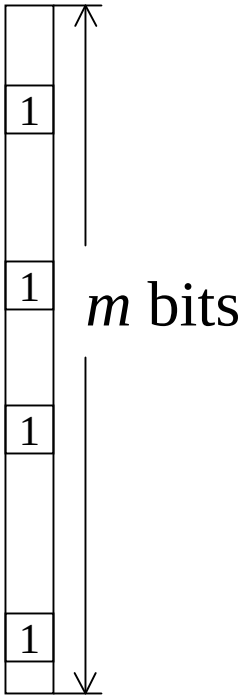
- Support membership test for a set of keys



Bloom Filters: the Math

- Given *n* keys, how to choose *m* and *k*?

Bit Vector *v*



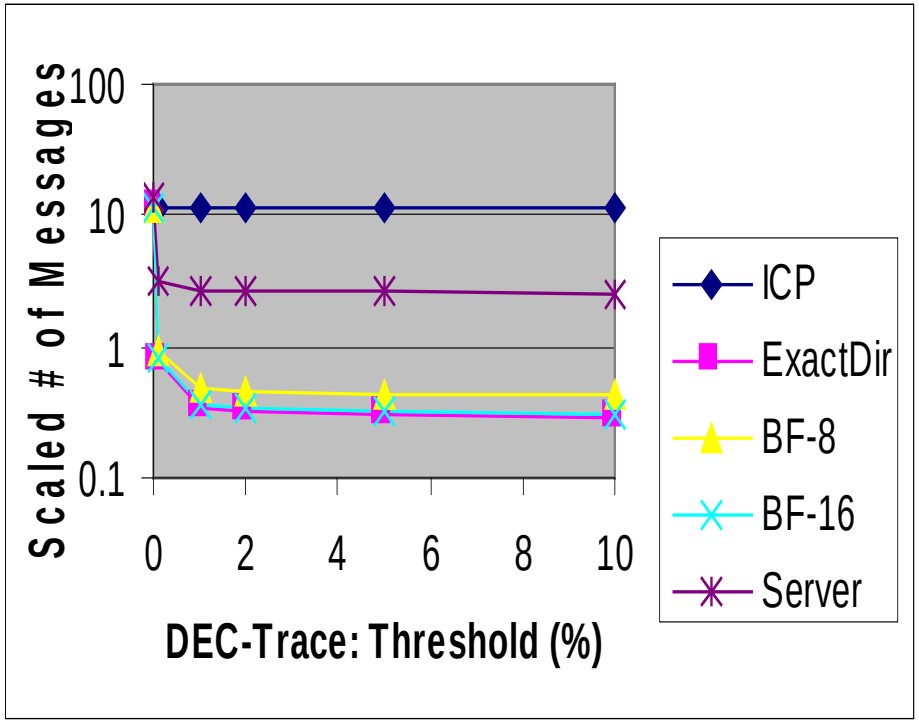
- Suppose m is fixed (>2n), choose k:
  k is optimal when exactly half of the bits are 0
  => optimal k = ln(2) * m/n

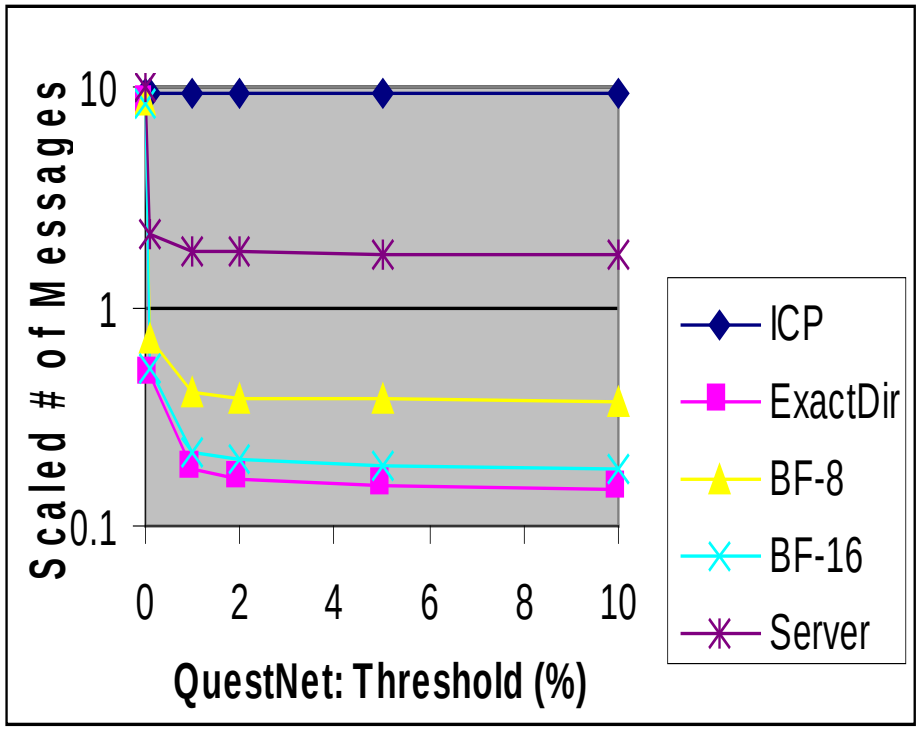
- False positive ratio under optimal k is $(1/2)^k$
  => false positive ratio = $(1/2)^{\ln 2 * m/n} = (0.62)^{m/n}$

Bloom Filters: the Practice

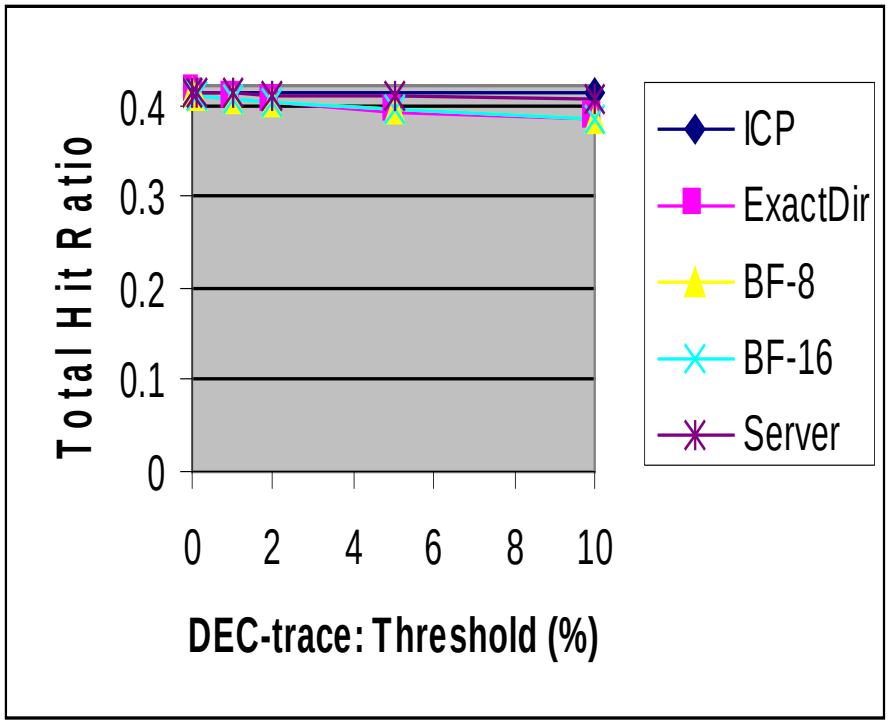
- Choosing hash functions
  - bits from MD5 signatures of URLs
- Maintaining the summary
  - the proxy maintains an array of counters
  - for each bit, the counter records how many times the bit is set to 1
- Updating the summary
  - either the whole bit array or the positions of changed bits (delta encoding)

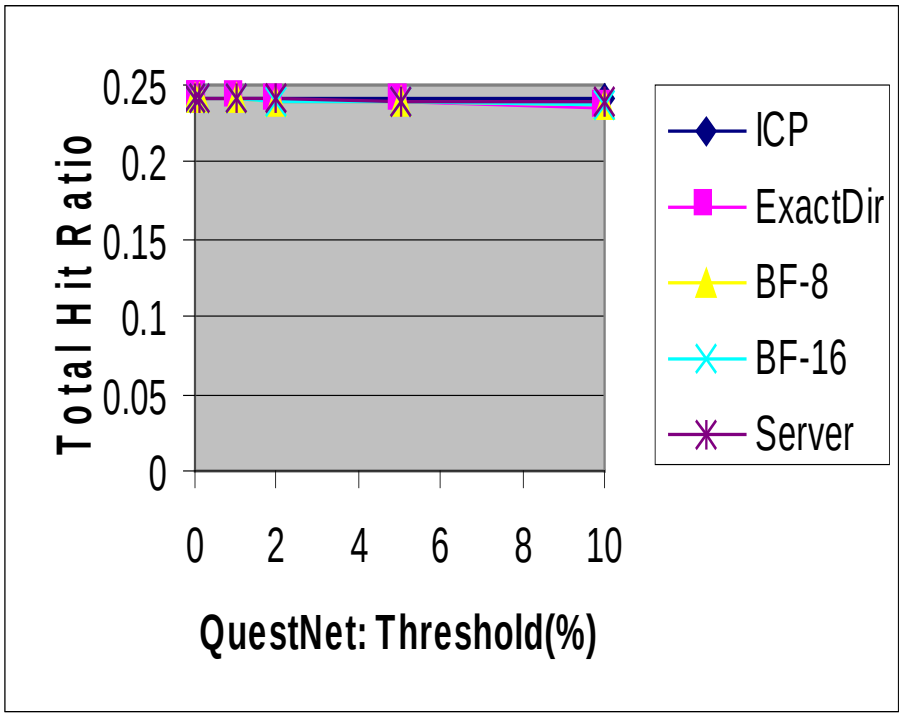
Result: Inter-Proxy Traffic





Result: Total Hit Ratio





Enhancing ICP with Summary Cache

- Prototype implemented in Squid 1.1.14
- Repeating the 4-proxy experiments, the new ICP:
  - Reduces UDP messages by a factor of 12 to 50 compared with the old ICP
  - Little increase in network packets over no cache sharing
  - increase CPU time by 2 - 7%
  - reduce user latency up to 4% with remote cache hits

Conclusions

- Summary cache enables caches to share contents with low overheads over wide area
- An alternative implementation called “Cache Digest” is in Squid 1.2.0
- Many other applications of bloom filters
- Technical report version available at:

  http://www.cs.wisc.edu/~cao/papers/summary-cache/