# An Efficient Approach to Per-Flow State Tracking for High-Speed Networks

Brad Whitehead, **Chung-Horng Lung**
Dept. of Systems and Computer Eng.
Carleton University, Ottawa, Canada

Peter Rabinovitch
Alcatel-Lucent
Ottawa, Canada

# Outline

- Motivation
- Background
- Two main existing approaches:
  - BDFT – Binned Duration Flow Tracking
  - Fingerprint-Compressed Filter Approximate Concurrent State Machine (FCF ACSM)
- Proposed BDFT Hybrid
- Computational Analysis
- Experimental Analysis
- Conclusions

# Motivation

- Network monitoring is crucial.
- Obtaining per-flow information, e.g., flow state, has become increasingly important.
- High-speed routers have limited CPU and memory resources.
- Packet sampling, e.g., 1 in 20 sampling, normally has low accuracy.
- BDFT is CPU-efficient; FCF ACSM is memory-efficient.
- Need a **time and space efficient** method of tracking **per-flow state**.

# Background

- Not much work on tracking per-flow state.
- NetFlow is popular, but has scalability issue.
- Bloom filters or its variants are common in network monitoring due to the efficiency.
  - Space-code Bloom filters
  - Time-decaying Bloom filters
  - Shown to be able to scale to OC-192 speeds.
- Whitehead, et al.
  - Binned Duration Flow Tracking (BDFT)
    - CPU-efficient but requires larger memory space
- Bonomi, et al.
  - Fingerprint-Compressed Filter Approximate Concurrent State Machine (FCF ACSM)
    - Memory-efficient but has higher computational cost
- SCD (Symmetric Connection Detection) is adopted for this paper to filter out unsuccessful flows.

# Tracking State with Bins

- Challenges of flow tracking in practice:
  - Every packet
  - Arbitrary state transitions
- Observations:
  - Many flows share a common state
  - State transitions happen for many flows at the same time
- Idea of grouping flows into **"bins"**: a group of **flows sharing the same state** -> duration of flows
  - Much simpler state updates and smaller number of states

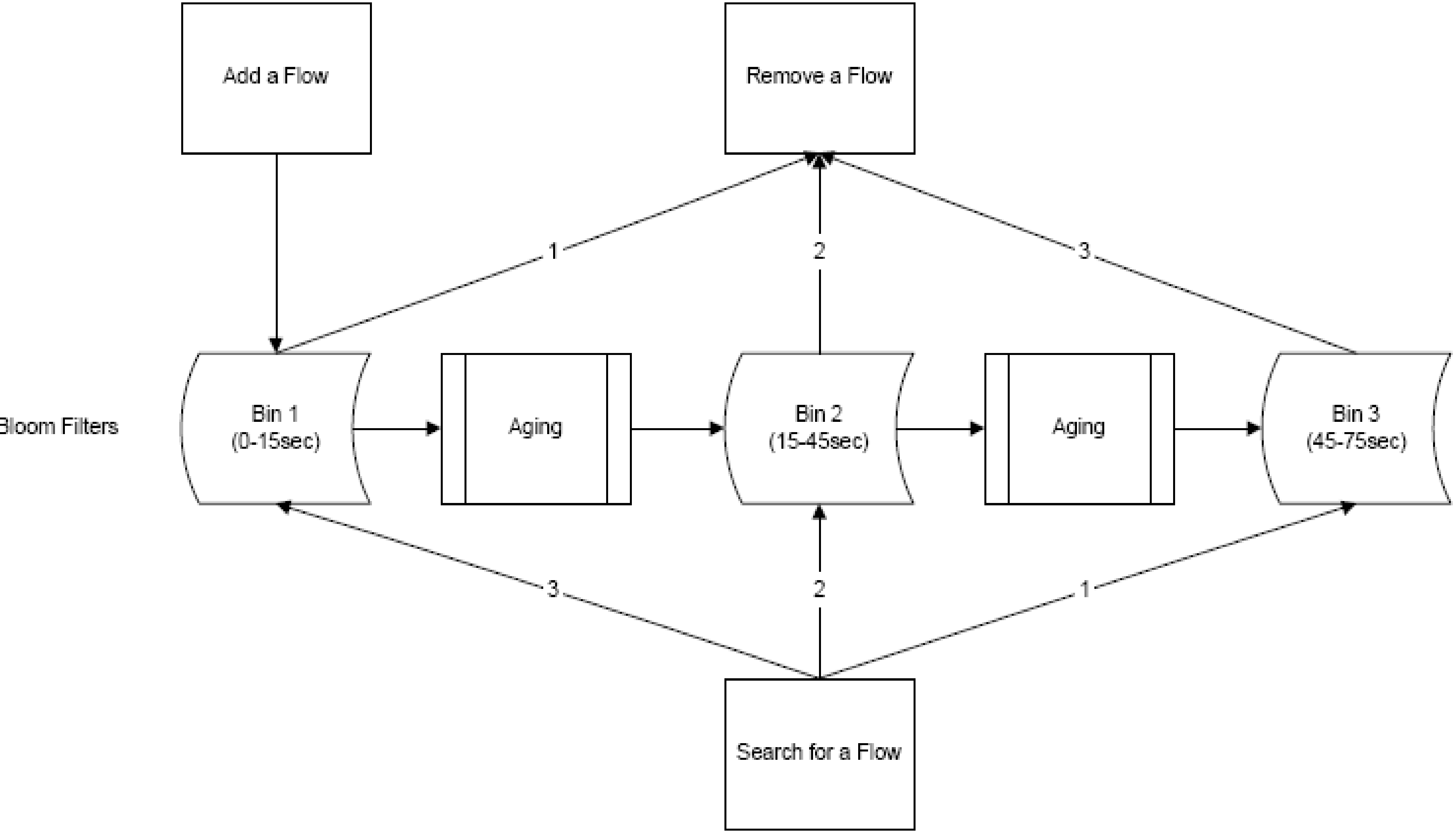
# BDFT – Binned Duration Flow Tracking

- BDFT is designed to track the approximate duration of all TCP flows seen on a high-speed router.
- Bins are the only data storage component of BDFT.
- Counting Bloom filters are adopted instead of just binary Bloom filters:
  - Replacing the flow ID information with hashes
  - Hashes are used to index counters in an array, incrementing them on insert (TCP SYN), and decrementing them on delete (TCP FIN or RST).

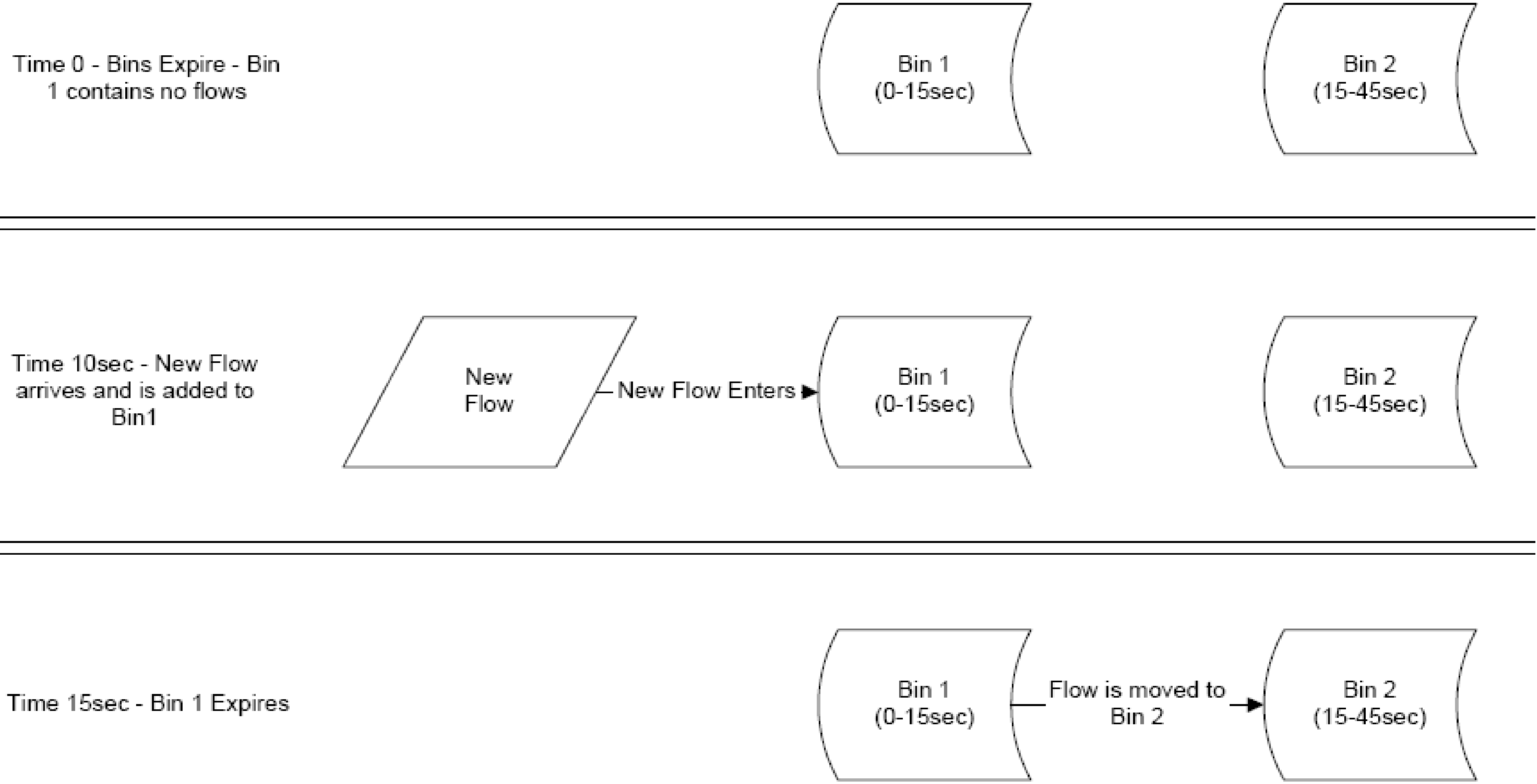
# BDFT – Main Components

- Add a flow
  - Add to Bin #1 ( at 2nd step of TCP 3-way handshake).
  - Unestablished flows are not added using SCD
  - k hashes are created from flow ID; increment counters
- Remove a flow
  - When the TCP FIN or RST flag is set, the flows are removed
  - Search the flow (from the shortest-duration bin)
  - Decrement counters
- Aging: a key step
  - Moving all flows in a shorter-duration (configurable time range) bin to the next longer-duration bin
  - No flow-specific info, e.g.. Flow start time, is stored
- Search for a flow
  - Based on requests
  - Starting with the oldest bin first and moving to younger bins sequentially to reduce chances of false positive

# BDFT Operations



# BDFT – Aging Process



Time 0 - Bins Expire - Bin 1 contains no flows

Bin 1 (0-15sec)

Bin 2 (15-45sec)

Time 10sec - New Flow arrives and is added to Bin1

New Flow

New Flow Enters

Bin 1 (0-15sec)

Bin 2 (15-45sec)

Time 15sec - Bin 1 Expires

Bin 1 (0-15sec)

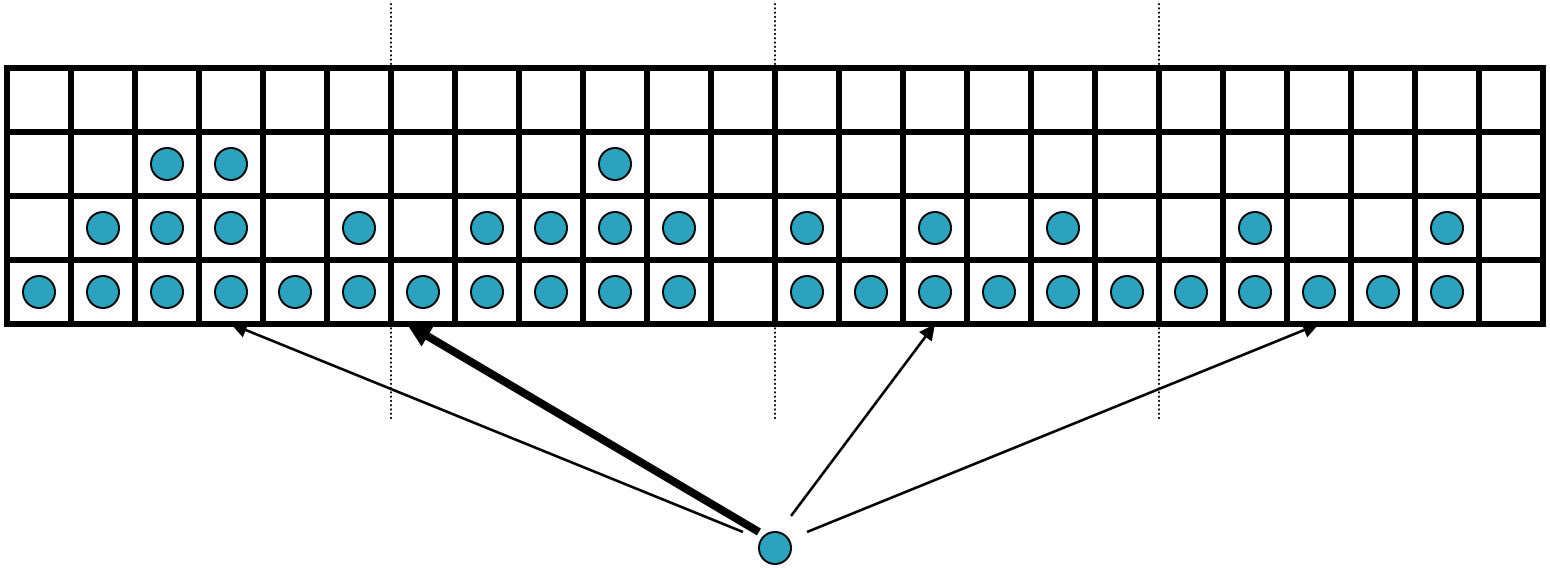
Flow is moved to Bin 2

Bin 2 (15-45sec)

# FCF ACSM

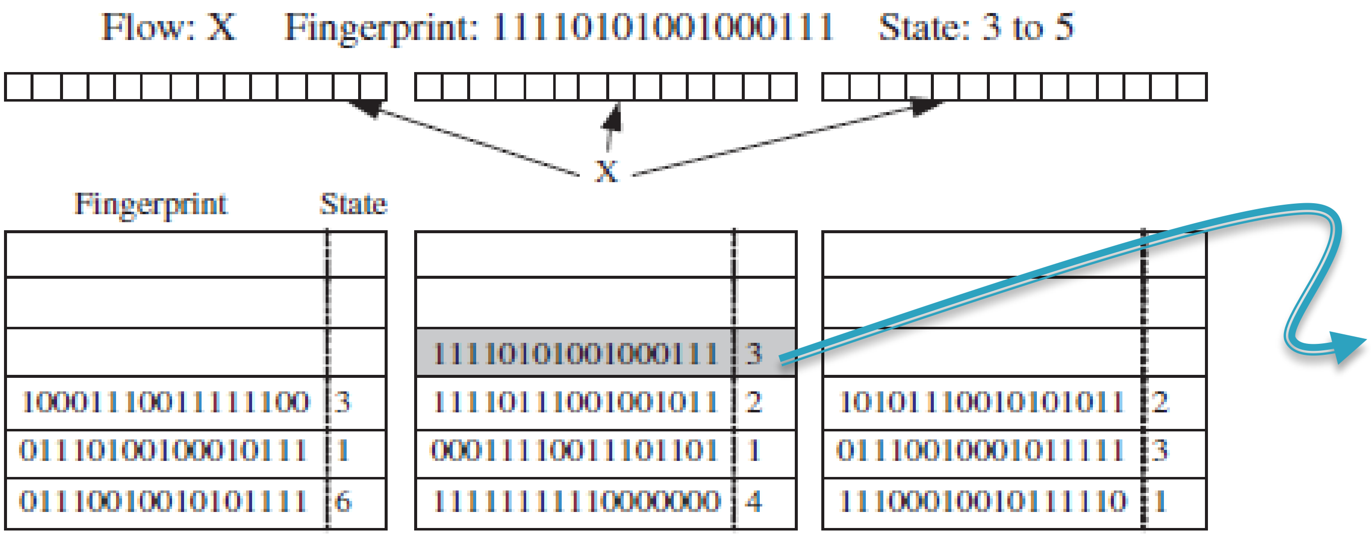
- Bonomi, et al. present 3 methods of tracking per-flow state
- FCF-ACSM is the most efficient
  - Employ **d-left hashing**
    - Accurate and good memory efficiency
    - Near perfect hash, even distribution of items in the buckets
    - Higher computational requirement

# Multiple Choices: *d*–left Hashing



- Split hash table into *d* equal subtables.
- To insert, choose a bucket uniformly for each subtable.
- Place item in a cell in the least loaded bucket, breaking ties to the left.

# FCF ACSM – d–left



Flow: X Fingerprint: 11110101001000111 State: 3 to 5

| Fingerprint | State |
|---|---|
| | |
| | |
| | |
| 10001110011111100 | 3 |
| 01110100100010111 | 1 |
| 01110010010101111 | 6 |

| Fingerprint | State |
|---|---|
| | |
| | |
| 11110101001000111 | 3 |
| 11110111001001011 | 2 |
| 00011110011101101 | 1 |
| 11111111110000000 | 4 |

| Fingerprint | State |
|---|---|
| | |
| | |
| | |
| 10101110010101011 | 2 |
| 01110010001011111 | 3 |
| 11100010010111110 | 1 |

*Add*: flow id (X) hashed (fingerprint), stored in one of subtales

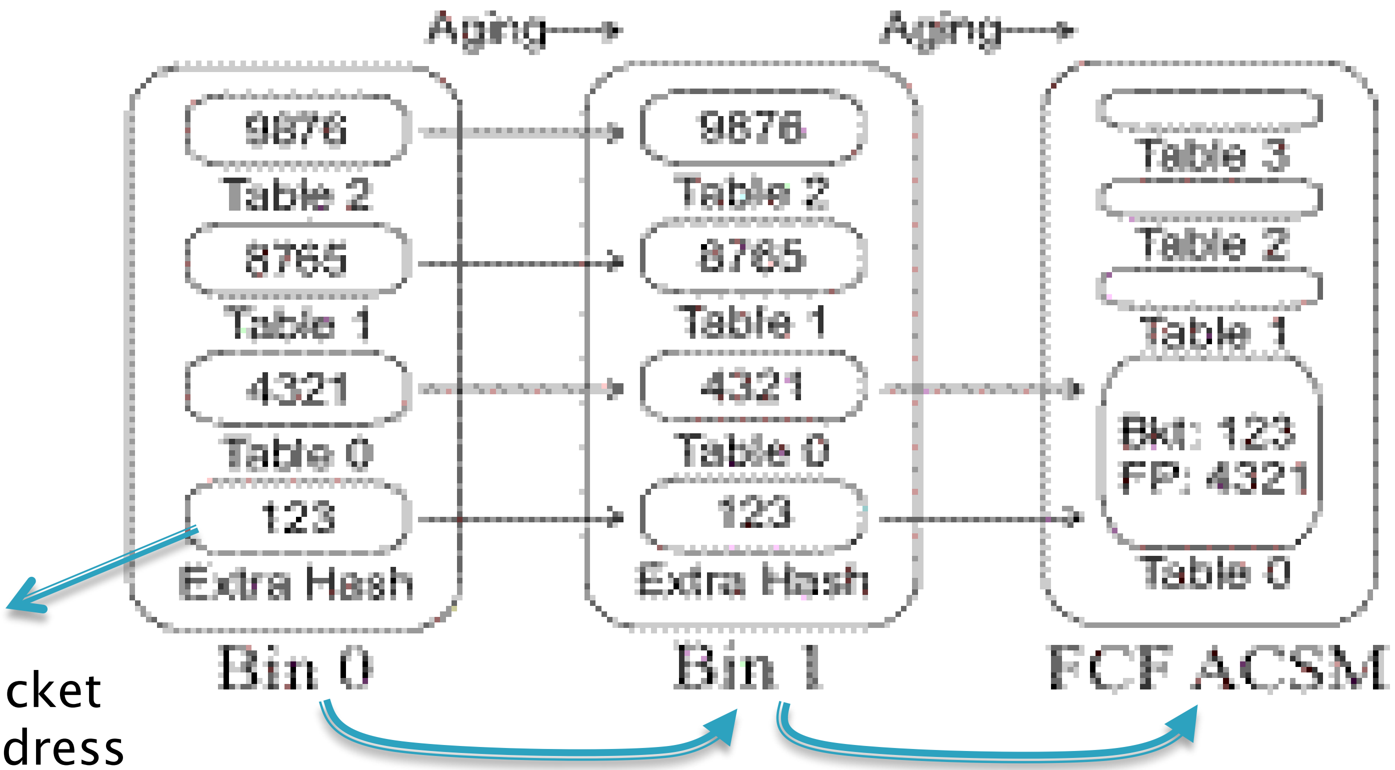
*Search*:
Flow found

- Number of subtables or hash functions *d;*
- Number of buckets **b** of each subblock of the hash table
- The height **h** of each bucket
- The size **f** of the fingerprint in bits. x additional bits for each flow (to represent the state)
- Total space is *dbh(f + x)* bits for the hash table

# BDFT Hybrid – Bloom and D-left

- Objective is to take advantages of best features of BDFT (speed) and FCF ACSM (space)
  - Idea:
    **replace older bins in BDFT with a single FCF ACSM**
  - BDFT: Short-lived flows in first few bins require frequent maintenance (add and remove operations)
  - FCF-ACSM: long-lived but seldom changing flows
  - Issue: aging of flows **from BDFT to FCF ACSM**

# BDFT-H Example



bucket address

BDFT

From BDFT to FCF

Assumptions. FCF has:

- 3 subtables
- 256 buckets each (8 bits)
- 16 bits for fingerprint

# Computational Analysis

| Name | Operation | Mem. Reads | Mem. Writes | Branches | Total |
|---|---|---|---|---|---|
| BDFT | Insert | 3 | 3 | 3 | 9 |
| FCF | Insert | 24 | 1 | 29 | 54 |
| BDFT | Removal | 6 | 3 | 6 | 15 |
| FCF | Removal | 12 | 1 | 12 | 25 |
| BDFT | Search (rare) | 21 | 0 | 21 | 42 |
| FCF | Search (rare) | 12 | 0 | 12 | 24 |
| BDFT | Aging (periodic) | 2000 | 1000 | 1000 | 4000 |
| FCF | Aging (periodic) | 2000 | 500 | 2000 | 4500 |
| BDFT-H | Aging (periodic) | 1 | 1+memset | 0 | 2+memset |
| BDFT-H | Aging (to d-left) | 3250 | 150 | 3000 | 6400 |

- Insert + Removal (frequent operations): FCF 3.5 times more
- Search: FCF is faster
- BDFT-H: fast insert-remove of short lived flows and quick search for long-duration flows

Assumptions:

- 3 hash functions
- 6 cells/bucket
- Bloom filter size: 1000

# Experimental Analysis

- Two traces
  - CAIDA (C_04): “dirty” traffic due to port scanning or DoS attacks
  - NLANR (N_12): clean traffic
- Characteristics for TCP control packets

| | N_12 | As a % of total | C_04 | As a % of total |
|---|---|---|---|---|
| Total established flows | 274,473 | 77.88% | 555,927 | 4.96% |
| Ave. active flows | 11,284 | | 901,245 | |
| Timed out flows | 430 | 0.16% | 4376 | 0.78% |
| Unique IPs | 97,036 | | 2,681,172 | |

# Experimental Setup

- Distribution of flow durationsof BDFT
  - Estimation of the size of bins and total memory
  - In literature, 40% – 70% of flows last < **2 seconds**
  - N_12: 75% established flows < 2 seconds
  - C_04: 50% established flows < 2 seconds
- Unsuccessful connections filtered out with Symmetric connection detection (SCD)
- Flows after 2 minutes with no activity are removed
- **Tracking success**: estimated flow duration result within 50% of the actual flow duration if > 30 sec
- 3 hash functions are used
- Filter size: 1000 for1st and 2nd filters

# Experimental Results – BDFT Memory Usage vs. Accuracy

| Trace | Memory Usage (bytes) | Accuracy |
|---|---|---|
| C_04 | 90112 | 95.46% |
| C_04 | 180224 | 99.19% |
| C_04 | 360448 | 99.87% |
| C_04 | 720896 | 99.97% |
| N_12 | 2816 | 96.85% |
| N_12 | 5632 | 99.79% |
| N_12 | 11264 | 99.98% |

0.257 bits/flow
0.128 bits/flow

# Experimental Results – FCF ACSM Performance

| Trace | d-left | Memory | | |
|---|---|---|---|---|
| | (d/b/h/f) | Usage | Accuracy | |
| C_04 | 4/1024/6/16 | 67584 | 93.19% | |
| C_04 | 4/1024/9/16 | 101376 | 99.54% | |
| C_04 | 4/2048/6/16 | 135168 | 99.95% | → 0.096 bits/flow |
| C_04 | 4/4096/6/18 | 294912 | 99.98% | |
| N_12 | 4/64/4/12 | 2304 | 97.84% | |
| N_12 | 4/64/4/16 | 2816 | 99.90% | → 0.064 bits/flow |
| N_12 | 4/128/4/16 | 5632 | 99.98% | |

# Experimental Results – BDFT-H Performance

| Trace | BDFT Mem. | d-left (d/b/h/f) | Total Mem. | Accuracy | |
|---|---|---|---|---|---|
| C_04 | 65536 | 4/512/9/14 | 174336 | 99.75% | |
| C_04 | 131072 | 4/512/9/15 | 299520 | 99.94% | → 0.214 bits/flow |
| C_04 | 262144 | 4/512/9/16 | 547584 | 99.97% | |
| C_04 | 524288 | 4/512/9/16 | 645888 | 99.97% | |
| N_12 | 2048 | 4/16/4/15 | 7840 | 98.93% | |
| N_12 | 4096 | 4/32/4/15 | 12608 | 99.86% | → 0.286 bits/flow |
| N_12 | 8192 | 4/32/4/15 | 23104 | 99.98% | |

# Conclusions

- Proposed BDFT Hybrid approach for high-speed networks
- Analysis of BDFT Hybrid:
  - Speed: faster FCF ACSM for frequent operations
  - Space: lower BDFT generally
  - Accuracy: higher than BFDT and FCF ACSM
  - Simulations with 2 real traffic traces

# Thanks!

# BDFT Steps – An Example

- The new flow arrives; its hashes are calculated based on IP Src/Dst, Port Src/Dst, and protocol type
- The flow is added to Bin 1 (0-15 sec)  by incrementing the counters corresponding to the hashes
- After 15 seconds Bin 1 expires and its flows are moved to Bin 2 (15-30 sec)
- After an additional 30 seconds Bin 2 expires and its flows are moved to Bin 3 (45-75 sec)
- After 55 seconds from the flow start, a TCP FIN is received for the flow, and the removal process begins
- The flow’s hashes are calculated as above
- The Bins are searched for the flow's hashes starting with Bin 1
- The flow is found in Bin 3, so the counters corresponding to the hashes are decremented in Bin 3