Tracking Per-Flow State – Binned Duration Flow Tracking

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

Peter Rabinovitch
Alcatel-Lucent
Ottawa, Canada

Outline

- Motivation
- Background
- BDFT – Binned Duration Flow Tracking
  - Main components
- Experimental Analysis
  - Traces
  - Experimental setup
  - Results
- Computational Analysis
- Conclusions and Future Work

Motivation

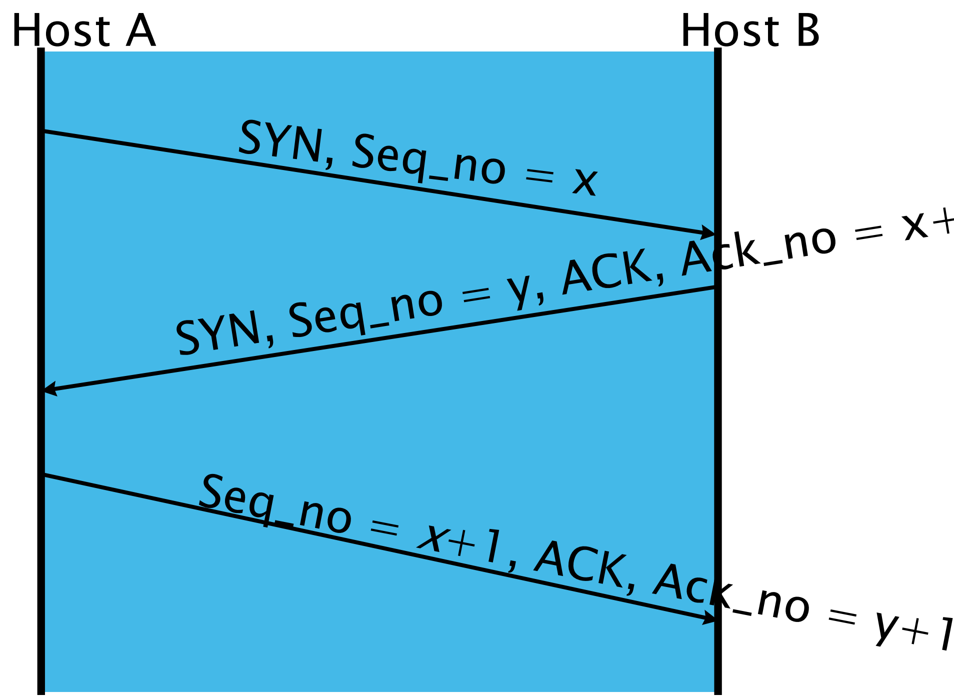
- Network monitoring is crucial.
- Obtaining per-flow information, e.g., flow state, has become increasingly important.
- High-speed routers have limited CPU and memory resources.
- Packet sampling, e.g., 1 in 20 sampling, normally has low accuracy or high memory usage for low bandwidth flow.
- Need 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.
- Bonomi, et al.
  - Finger-Compressed Filter Approximate Concurrent State Machine (FCF ACSM)
    - Very memory-efficient but has higher computational cost
- SCD (Symmetric Connection Detection) is adopted for this paper to filter out unsuccessful flows.

SCD (Symmetric Connection Detection)

- How does TCP establish a connection?
- 3-way handshake
- SCD: Once a TCP SYN has been detected from both sides of a connection, SCD will report that the connection was successful.
  - Keeps **state** and **direction**
- SCD is used for pre-filtering unsuccessful flows for BDFT
  - Can reduce the processing requirement by 95% for some traces.



Tracking State with Bins

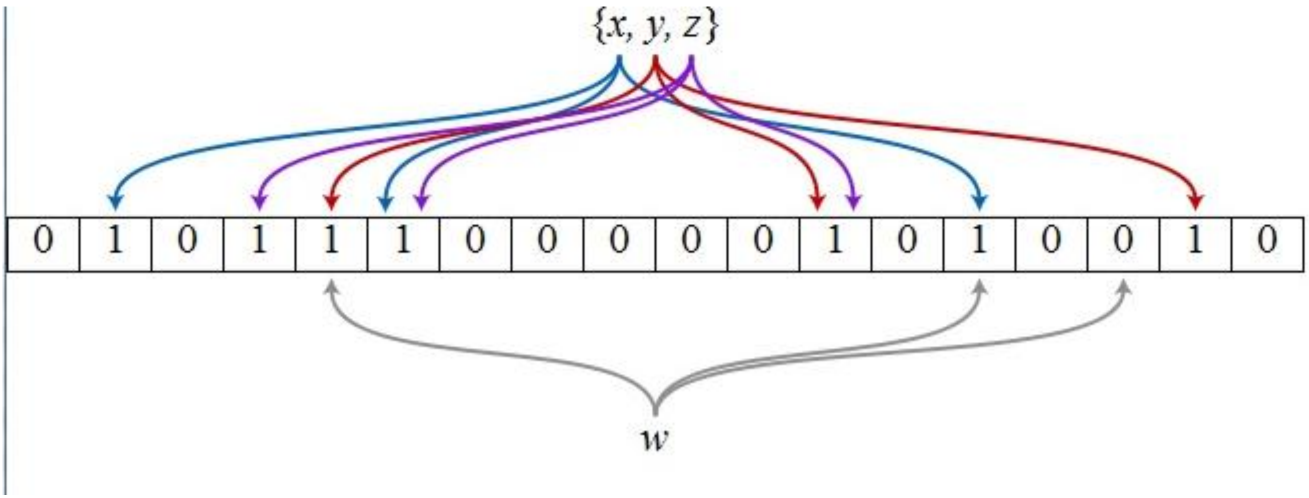
- Challenge of flow tracking:
  - A state tracking implementation which processes every packet and can allow arbitrary state transitions may not be practical on today's router
- Design tradeoff between **accuracy** and **efficiency**.
- Observations:
  - Many flows share a common state
  - State transitions happen for many flows at the same time
  - State transitions are singly-linked
- Main ideas: 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. The bin that a flow is in corresponds to its current duration.
- Counting Bloom filters are adopted:
  - 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).

Bloom Filters – an Example

K hashes
M slots



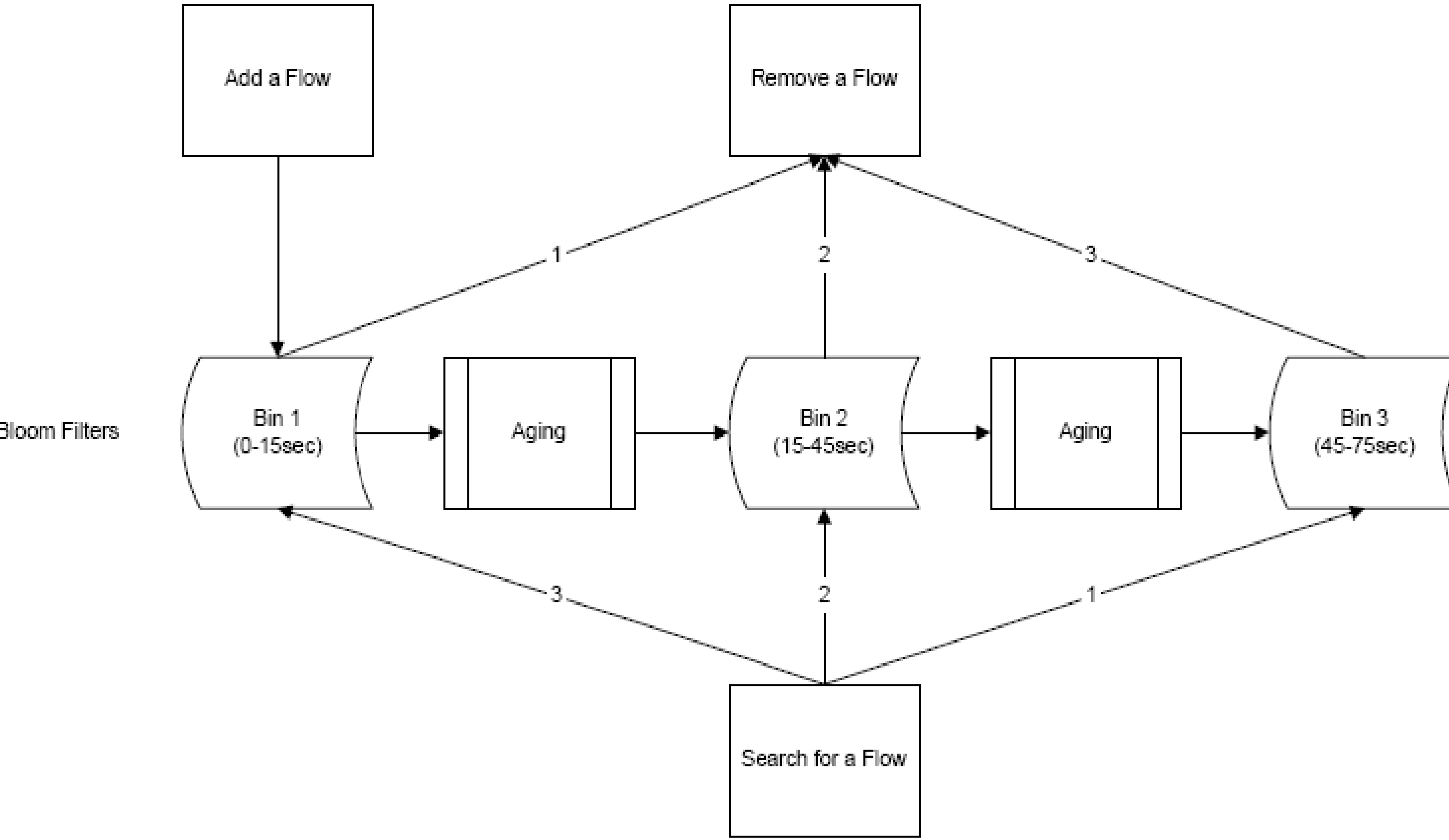
The element *w* is not in the set {x, y, z}, because it hashes to one bit-array position containing 0.
For this figure, m=18 and k=3

Counting Bloom filters: each location has a counter

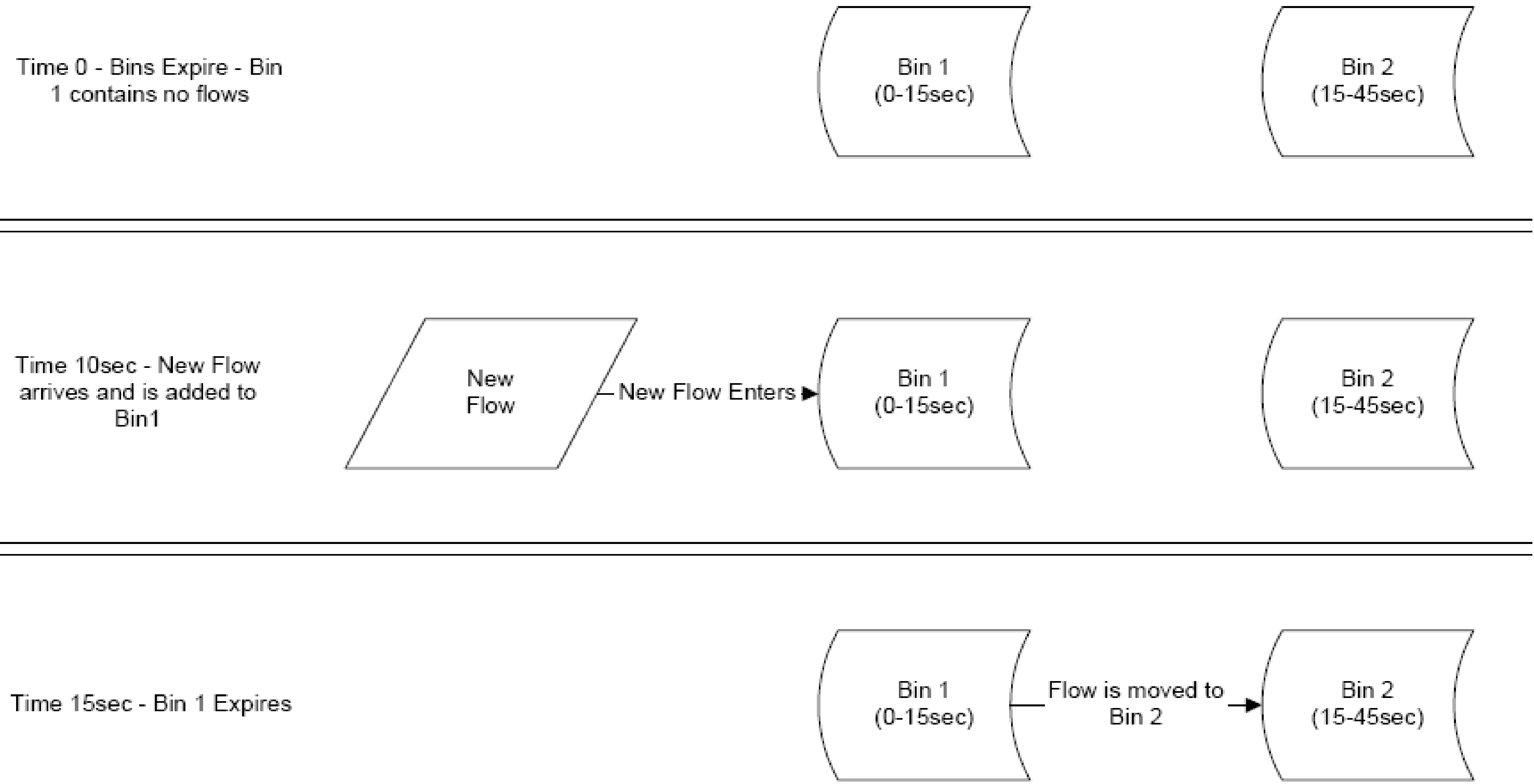
BDFT – Main Components

- Add a flow
  - Add to Bin #1 (at 2nd step of TCP 3-way handshake).
  - Unestablished flows are not added
  - 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



BDFT Steps – An Example

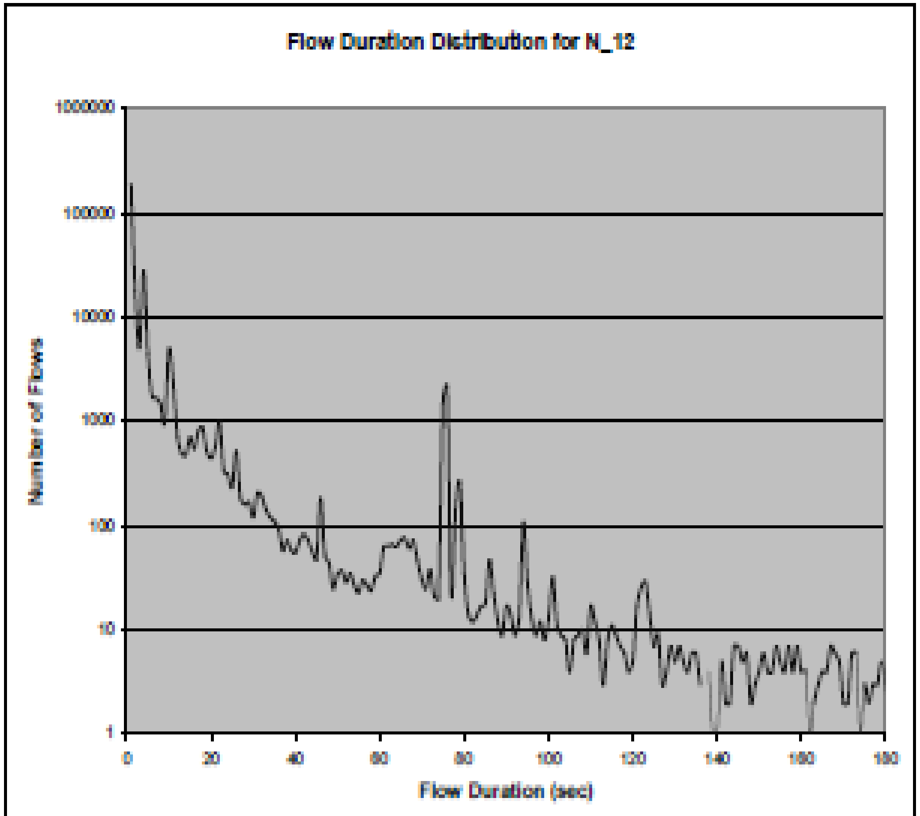
- The new flow arrives; its hashes are calculated based on IP Src/Dst, Port Src/Dst, and protocol type
- The flow is added to Bin 1 (0-15 sec) by incrementing the counters corresponding to the hashes
- After 15 seconds Bin 1 expires and its flows are moved to Bin 2 (15-30 sec)
- After an additional 30 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

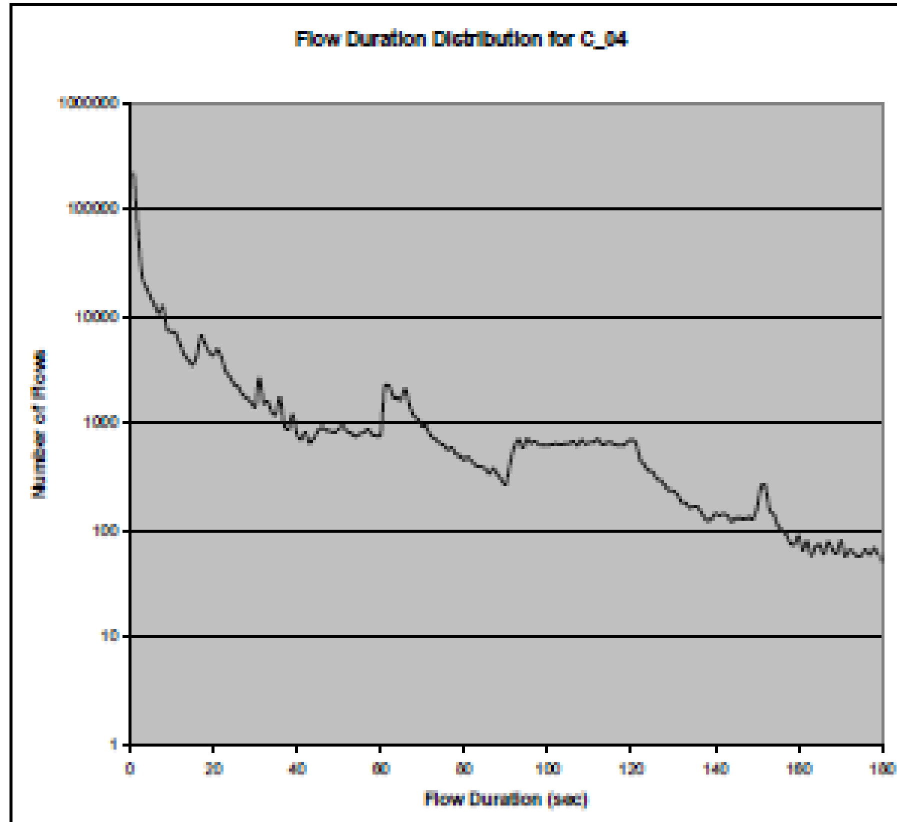
Experimental Analysis

- 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	196.9M		202.5M	
SYN	732,075	**0.37%**	15,608,680	**7.71%**
FIN	586,000	**0.30%**	6,084,826	**3.00%**
RST	52,628	**0.03%**	3,914,433	**1.93%**

Flow duration distribution for trace N_12 and C_04





Experimental Setup

- Distribution of flow durations critical to the design of 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 – Memory Usage vs. Accuracy

	with FRR removal		without FRR removal		
Trace	Memory Usage (bytes)	Accuracy	Memory Usage (bytes)	Number of Overflows	Accuracy
C_04	90112	95.46%	90112	902	79.24%
C_04	180224	99.19%	180224	134	96.15%
C_04	360448	99.87%	360448	16	99.59%
C_04	720896	99.97%	720896	1	99.98%
N_12	2816	96.85%			
N_12	5632	99.79%			
N_12	11264	99.98%			

0.257 bits/flow (C_04, 360448 bytes, 99.87%)
0.128 bits/flow (N_12, 5632 bytes, 99.79%)

Experimental Results – Computational Analysis

Operation	Mem. Reads	Mem. Writes	Branches	Total
Insert	3	3	3	9
Removal	6	3	6	15
Search (rare)	21	0	21	42
Aging (periodic)	2000	1000	1000	4000

- Insert and Removal (most frequent operations) are very efficient
- Search also efficient
- Aging depends on filter size

Assumptions:

- 3 hash functions
- Bloom filter size: 1000

Conclusions

- Proposed a per-flow state tracking – BDFT approach for high-speed networks
- Analysis of BDFT:
  - Computational performance
  - Memory usage
  - Accuracy
  - Simulations with real traffic traces
- The "binning" concept of BDFT appears to be efficient for traffic TCP flows