Building a better NetFlow

(to appear in SIGCOMM 2004)

Cristian Estan, Ken Keys, David Moore,
George Varghese

University of California, San Diego

IETF60 – Aug 4, 2004 – IPFIX WG

UCSD CSE

Disclaimers

- "NetFlow" used generically, no particular vendor or implementation implied
- Proposed changes are metering related, but can affect ipfix protocol design
- Not meant to be the definitive solution, but to help encourage discussion and improvements

Sampling pros and cons

- Reduces **processor load**
- Reduces **memory usage**
- Reduces **bandwidth** for reporting

- Results **less accurate**
- Cannot estimate non-TCP **flow counts**

- Finding the **sampling rate** that **balances** the pros and cons is **hard**
- The best choice **depends on traffic mix** ☹

Fixing NetFlow

NetFlow problem	How we solve it
Memory and bandwidth usage strongly depend on traffic mix	Adapting sampling rate (part 2)
Network operator must set sampling rate	
Mismatch of flow termination heuristics and analysis	**Measurement bins (part 1)**
Cannot estimate number of non-TCP flows	Sampling flows (part 3)

Operating with time bins

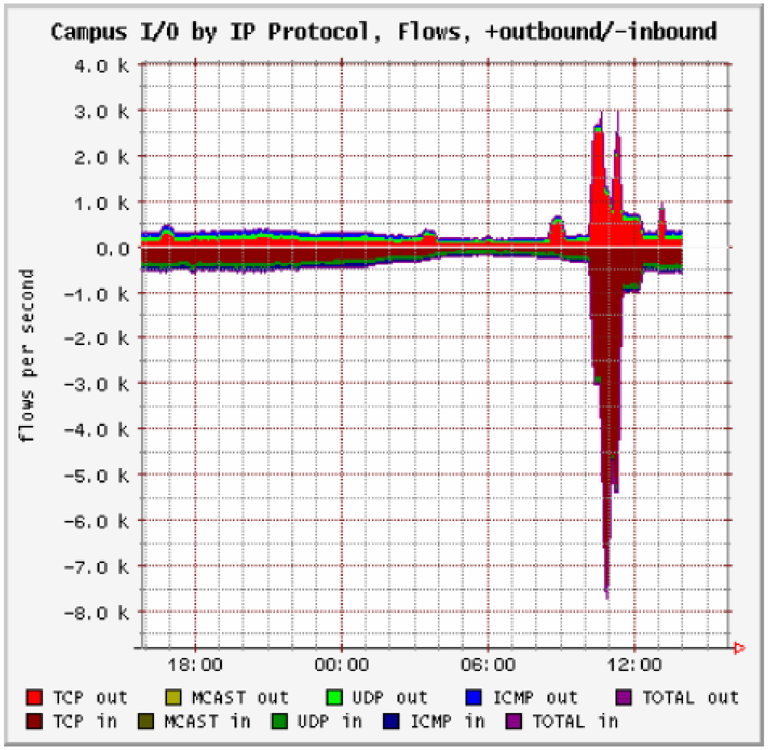
- Both operators and researchers usually prefer working with fixed time bins
- Use fixed size time bins (say 1 minute)
- Terminate all flow records at the end of the bin (but don't report immediately)
- Could use different sampling rates for each bin, including decreasing sampling within a bin as needed
- Simplifies analysis and reduces error
- Time bins allow reconstruction of flow timeouts

Analysis uses time bins anyway

IPMON

Application Breakdown			
Category	Packets (%)	Bytes (%)	Flows (%)
Web	54.35	61.48	47.33
File Sharing	3.35	2.43	3.74
FTP	0.52	0.54	0.07
Email	4.67	4.06	3.24
Streaming	7.26	13.07	1.60
DNS	6.13	1.16	27.26
Games	0.06	0.01	0.03
Other TCP	21.03	15.86	6.05
Other UDP	0.78	0.48	0.84
Not TCP/UDP	1.86	0.90	9.84

FlowScan



Site: San Jose (sj-20)
Date: February 5th, 2004

Relationship to IPFIX

- draft-ipfix-protocol-3, section 4:
  - 4.1: seems to require timeout based flows, allows for expiry based on resource constraints, but it is unclear on permissibility of using time bins
  - 4.2: allows for export of long-lasting flows on schedule determined by exporting process, but is unclear about what that entails
- draft-ipfix-protocol-3, section 8:
  - would it require putting the same start/end time (or bin #) in all of the Flow Records, or is there a way to specify the bin efficiently for an entire group of records

Fixing NetFlow

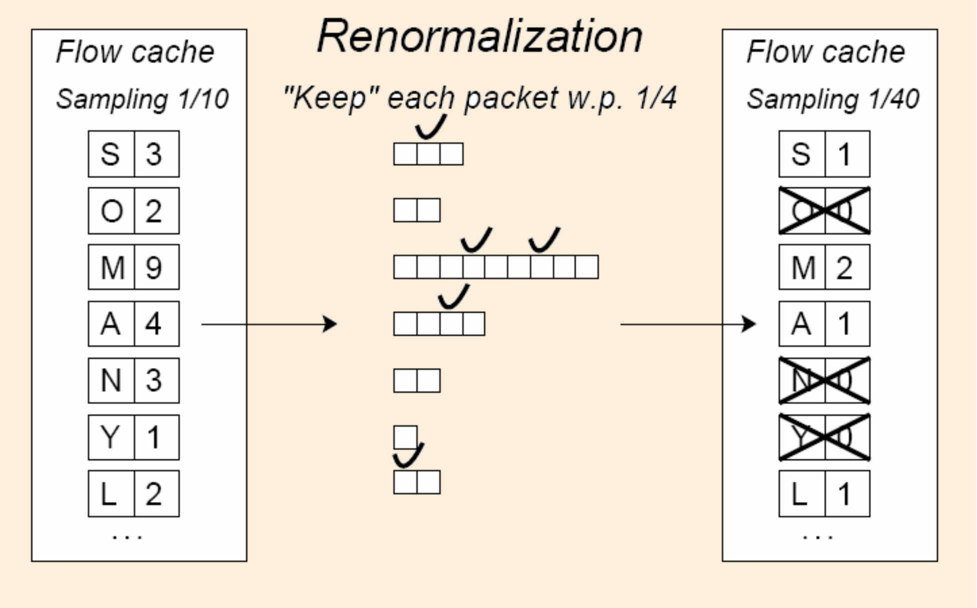
NetFlow problem	How we solve it
Memory and bandwidth usage strongly depend on traffic mix	**Adapting sampling rate (part 2)**
Network operator must set sampling rate	
Mismatch of flow termination heuristics and analysis	Measurement bins (part 1)
Cannot estimate number of non-TCP flows	Sampling flows (part 3)

Adaptive NetFlow

- Choose the sampling rate based on traffic
  - Use a high sampling rate when traffic allows
  - Keeping counters meaningful as sampling rate varies
  - Ensuring we never overload CPU
  - Ensuring we never run out of memory

Adapting sampling rate

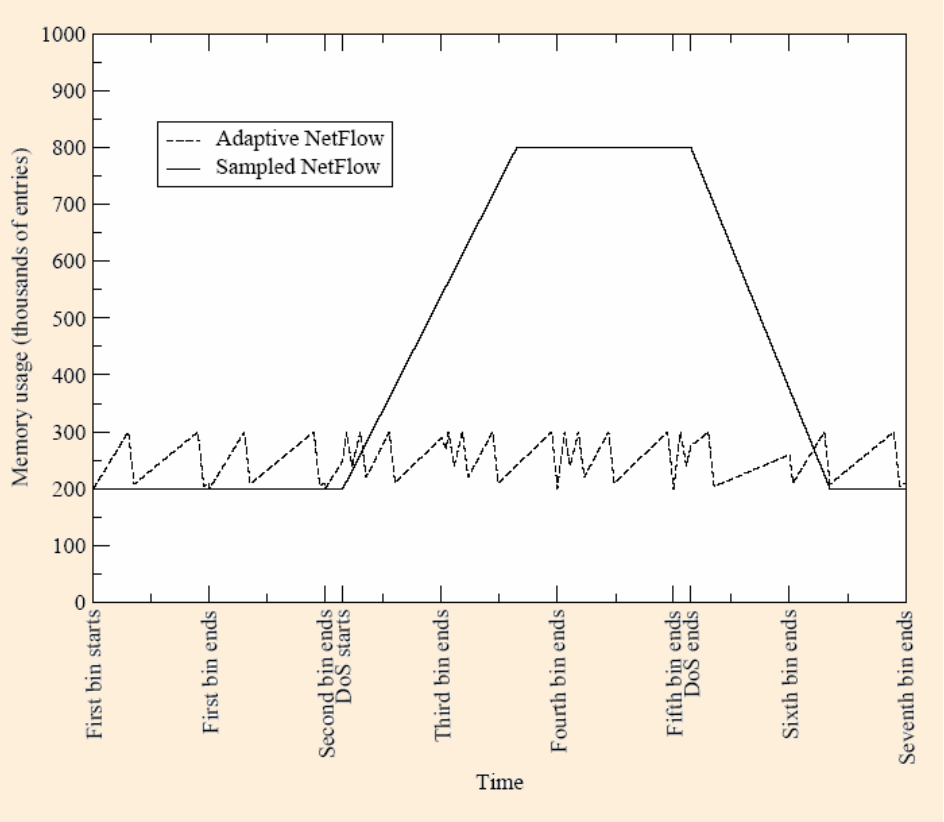
- If **multiple sampling rates** in effect while flow active, byte and packet counters meaningless
- Decreasing sampling rate – pretend to throw away sampled packets
- Increasing rate – not possible, since information discarded.
- Start each time bin with aggressive sampling

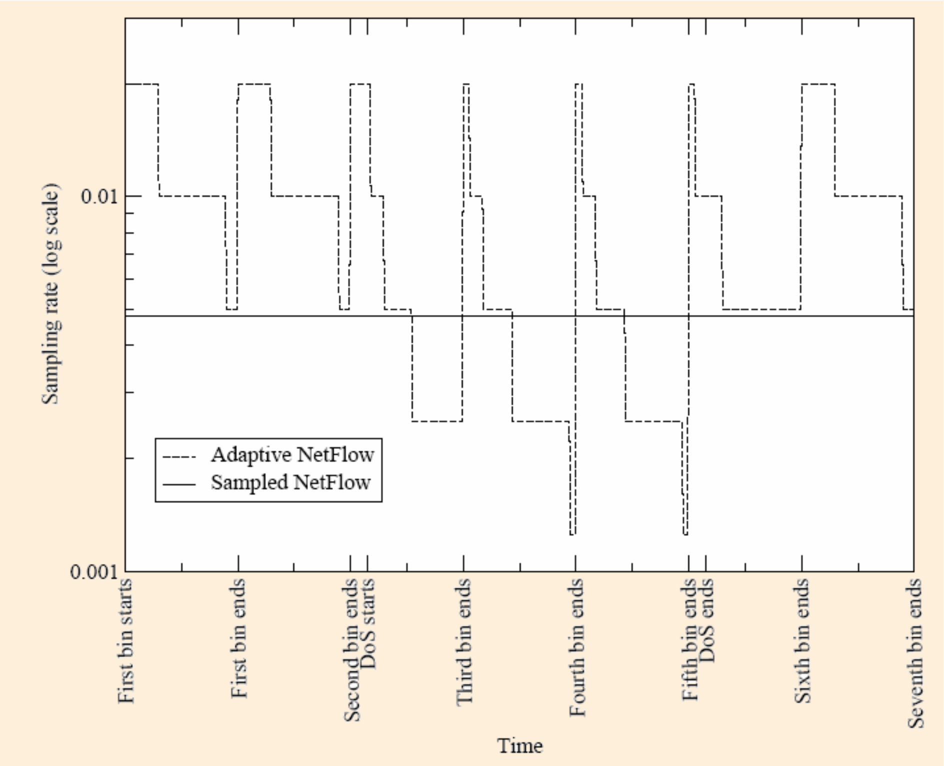


Limiting CPU usage

- Renormalization in parallel with operation
- Efficient renormalization – for most records only simple integer arithmetic, no random number generation
  - Updating 1 entry 3.4 μs
  - Renormalizing 1 entry 1.5 μs
- Vendor configures initial sampling rate high enough for CPU to keep up with minimum sized packets

Memory Usage:
What happens under DoS?





Rate adaptation and memory usage

- Trigger renormalization whenever the number of entries reaches a fixed threshold
- Must choose new sampling rate so that enough records discarded by renormalization
  - Use partial histogram of packet counters
- Actual memory at router must exceed the desired number of records per bin *M* to allow renormalization and buffering of old records

Main tuning knob: # of records M

- Controlled resource usage
- User configures number of desired records to be exported
- More meaningful than sampling rate
  - Relative error in estimating an aggregate that is a certain fraction of the traffic depends on ***M***
- Can produce reports of various sizes and send them with different reliability levels
  - Dropping random records is worse than generating fewer records by using lower sampling rate

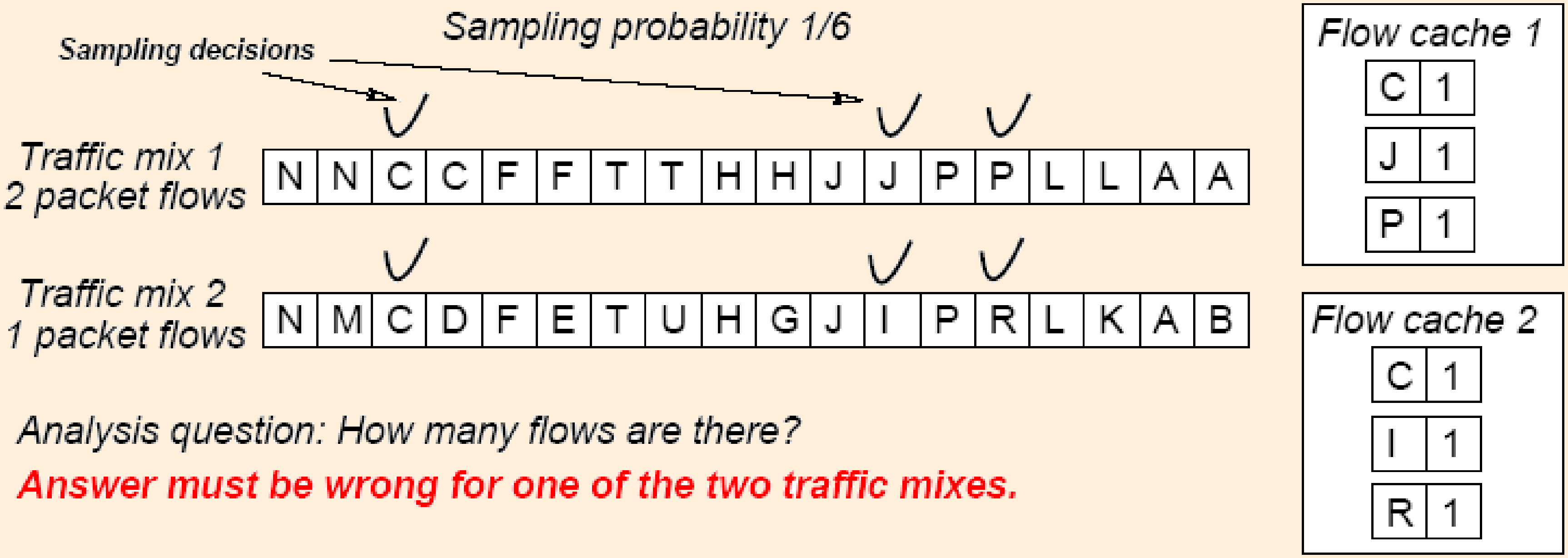
Relationship to IPFIX

- SCTP-PR: use different priority levels for different report sizes
- Reliable transport in general: may be able to share memory for flows from previous time bin with memory needed for retransmission
- draft-ipfix-protocol-3, section 8:
  - The sampling rate can vary frequently, should it be in the Flow Record or an Option Record?
  - If exporting multiple reports at different effective sampling rates, the same flow may be exported more than once, how should this be handled?

Fixing NetFlow

NetFlow problem	How we solve it
Memory and bandwidth usage strongly depend on traffic mix	Adapting sampling rate (part 2)
Network operator must set sampling rate	
Mismatch of flow termination heuristics and analysis	Measurement bins (part 1)
Cannot estimate number of non-TCP flows	**Sampling flows (part 3)**

Counting flows



Sampling decisions

Sampling probability 1/6

Traffic mix 1
2 packet flows

		✓									✓		✓				
N	N	C	C	F	F	T	T	H	H	J	J	P	P	L	L	A	A

Traffic mix 2
1 packet flows

		✓									✓		✓				
N	M	C	D	F	E	T	U	H	G	J	I	P	R	L	K	A	B

Flow cache 1

C	1
J	1
P	1

Flow cache 2

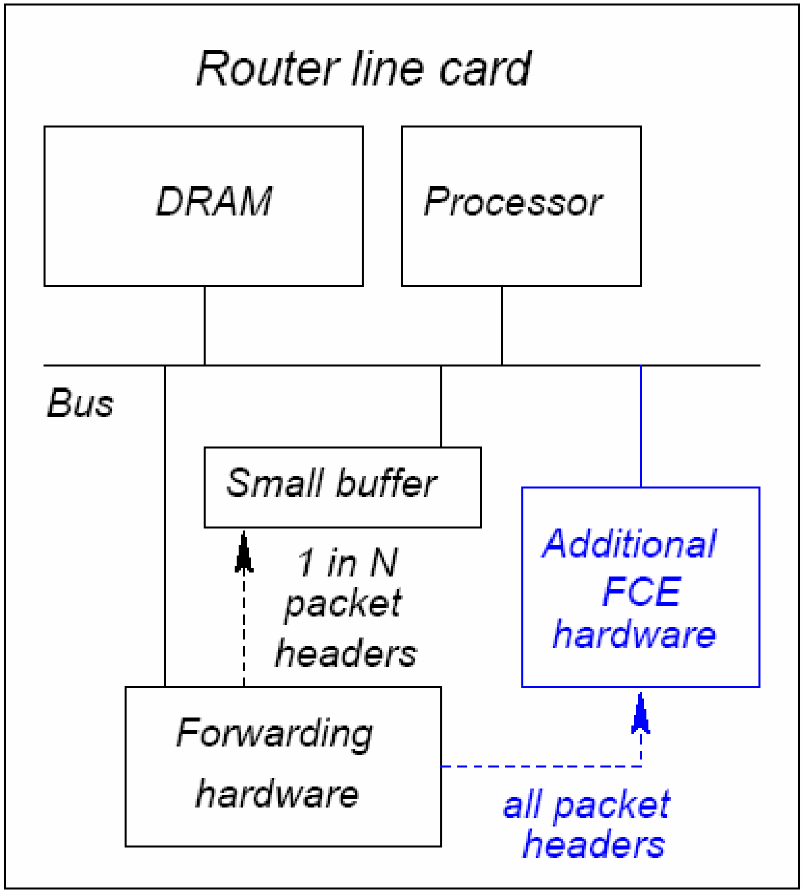
C	1
I	1
R	1

Analysis question: How many flows are there?

Answer must be wrong for one of the two traffic mixes.

- Goal: Unbiased, accurate flow counts for arbitrary post aggregation of the flows.

Flow Counting Extension



- Use “adaptive sampling” by Wegman and Flajolet
- Keep a table of all flow identifiers with hash(flowID)<1/2^{depth}
- At analysis scale flow counts by 2^{depth}
- Implement with CAM
- To fit memory, increase depth dynamically

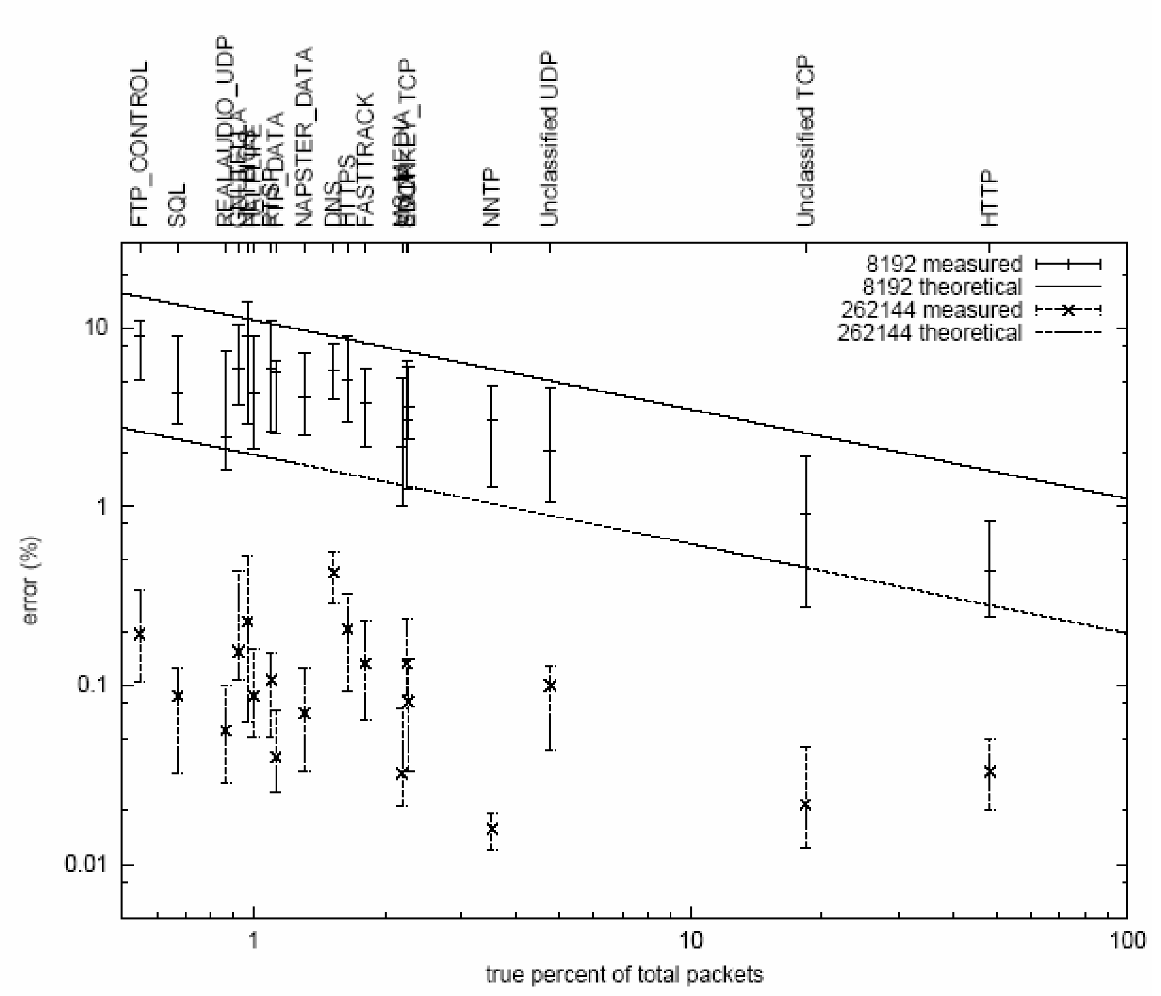
Relationship to IPFIX

- SCTP-PR: use different priority levels for different report sizes
- draft-ipfix-protocol-3, section 8:
  - The sampling rate can vary frequently, should it be in the Flow Record or an Option Record?
  - If exporting multiple reports at different effective sampling rates, the same flow may be exported more than once, how should this be handled?
- Would this require a separate template to export?
  - Basically the only thing to be exported here are the Flow Keys themselves.

Measurements

- Limited time, so for more details and results:
- http://www.caida.org/outreach/papers/2004/tr-2004-03/

ANF results

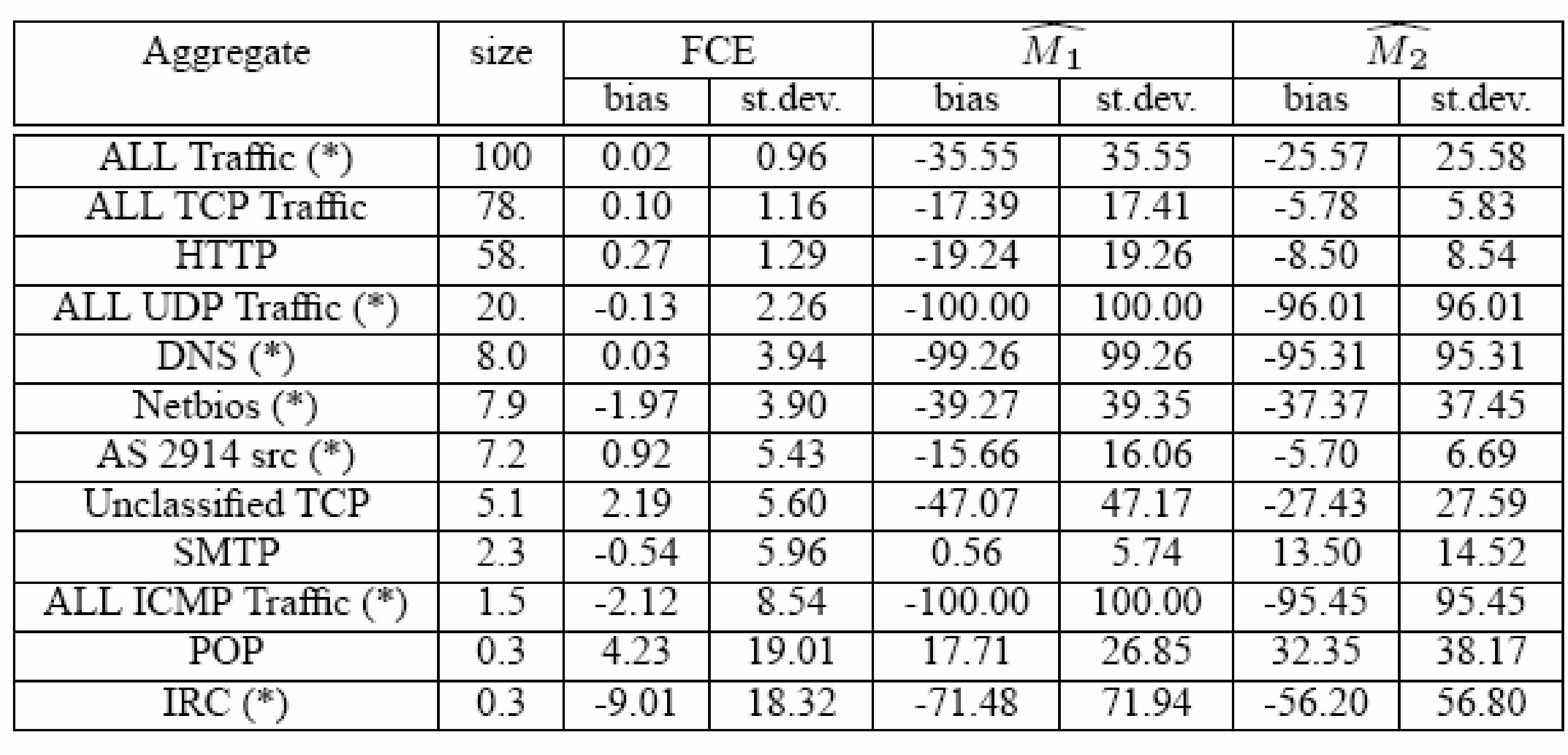


COOPERATIVE ASSOCIATION FOR INTERNET DATA ANALYSIS

University California, San Diego – Department of Computer Science

UCSD-CSE

FCE results



Aggregate	size	FCE		$\widehat{M_1}$		$\widehat{M_2}$	
		bias	st.dev.	bias	st.dev.	bias	st.dev.
ALL Traffic (*)	100	0.02	0.96	-35.55	35.55	-25.57	25.58
ALL TCP Traffic	78.	0.10	1.16	-17.39	17.41	-5.78	5.83
HTTP	58.	0.27	1.29	-19.24	19.26	-8.50	8.54
ALL UDP Traffic (*)	20.	-0.13	2.26	-100.00	100.00	-96.01	96.01
DNS (*)	8.0	0.03	3.94	-99.26	99.26	-95.31	95.31
Netbios (*)	7.9	-1.97	3.90	-39.27	39.35	-37.37	37.45
AS 2914 src (*)	7.2	0.92	5.43	-15.66	16.06	-5.70	6.69
Unclassified TCP	5.1	2.19	5.60	-47.07	47.17	-27.43	27.59
SMTP	2.3	-0.54	5.96	0.56	5.74	13.50	14.52
ALL ICMP Traffic (*)	1.5	-2.12	8.54	-100.00	100.00	-95.45	95.45
POP	0.3	4.23	19.01	17.71	26.85	32.35	38.17
IRC (*)	0.3	-9.01	18.32	-71.48	71.94	-56.20	56.80

Conclusions

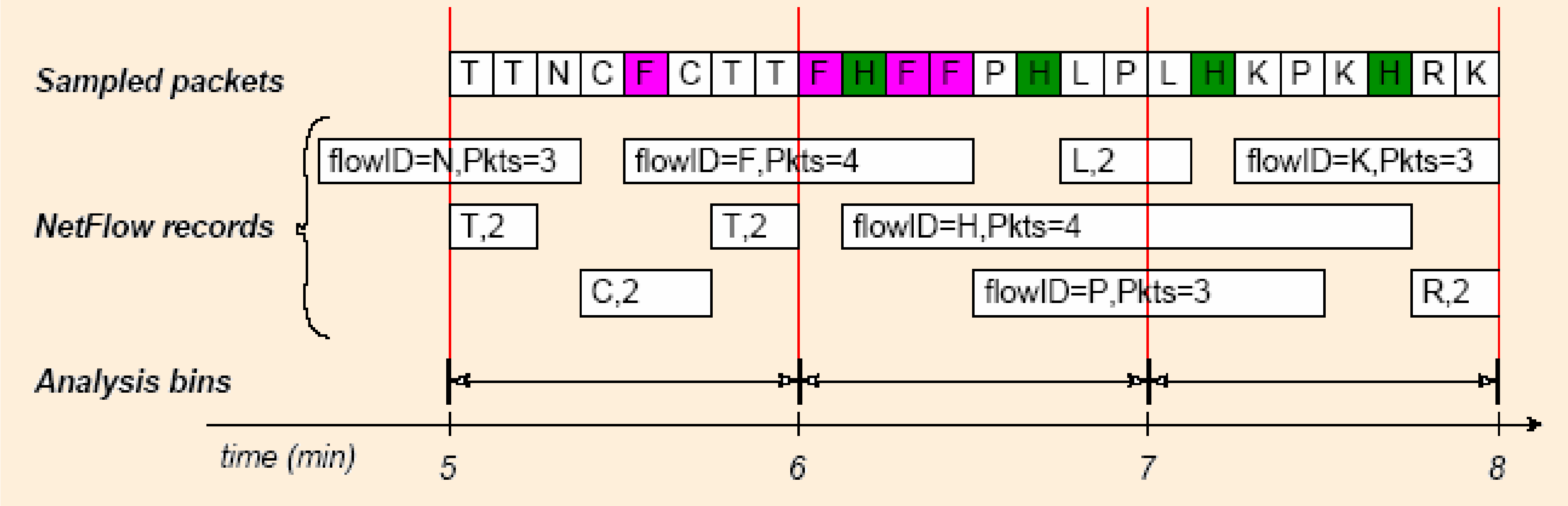
- Adaptive NetFlow improves NetFlow
  - Predictable resource usage even under adverse traffic
  - More meaningful tuning knob # or records *M*
  - Binned measurement matches analysis better
  - No hardware changes required
- Flow Counting Extension gives accurate flow counts for non-TCP flows too

Any more questions?

Theoretical results

- If ANF/NetFlow generates ***M*** entries, the relative standard deviation for aggregate that is fraction ***f*** of the traffic is at most *sqrt(1/**Mf**)* in packets and *sqrt(s_{max}/s_{avg}**Mf**)* in bytes
- If FCE generates ***M*** entries, the relative standard deviation for aggregate that is fraction ***f*** of the traffic is *sqrt(1/**Mf**)* in flows

Flow termination versus bins



- Flow termination heuristics require extra work to do the binning that can increase error in results
- Terminating flows at end of bin is backward compatible