Network monitoring: Methods and challenges

Pere Barlet
(pbarlet@ac.upc.edu)

Introduction

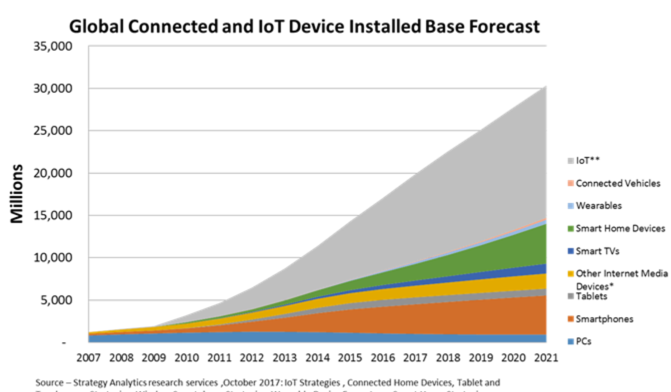
- Process of measuring network systems and traffic
  - Routers, switches, servers, etc.
  - Traffic volume, type, topology, etc.
- Network monitoring is important for:
  - Network planning (dimensioning the network, …)
  - Network management (troubleshooting, QoS, …)
  - Network security (IDS, NGFW, anomaly detection, …)
- Network monitoring is VERY challenging!

Introduction

- Internet is a huge system
  - The first node (ARPANET) was installed in 1969
  - >20 billion connected devices[1]
  - 50000 GB/s of data is moved by the Internet core[1]
  - 2300 Exabytes[2] of new data is created daily[3]
- Are we prepared for this growth?

[1] https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html
[2] 1 Exabyte (EB) = 10^{18} bytes = 1 billion GB
[3] https://www.ibmbigdatahub.com/infographic/four-vs-big-data

Number of connected devices



Introduction

- Internet is a complex system
  - It has grown organically
  - Without any centralized design or plan
  - By different independent organization, with different (often competing) goals
  - Grown exponentially, with most systems being connected recently (after 1990s)
  - Very dynamic, constantly changing in size, configuration, traffic and application mix

Introduction

- Internet behavior is difficult to model
  - Fully distributed system, without (almost) any central decision point
  - Decisions are made autonomously by different types of systems at different layers
  - Although protocols are well-defined, their mixed behavior is unpredictable
  - As a result, no analytical models exist that can explain the actual Internet behavior in the wild

Why measuring the Internet?

- As any **complex** system, measurement and monitoring are crucial to manage it
  - We cannot understand what we cannot measure!
- TMA is more important than ever
  - Internet has become a central piece of our lives
  - Current businesses depend on networks
  - Measuring the Internet is extremely challenging
  - TMA is the basis of most network security systems

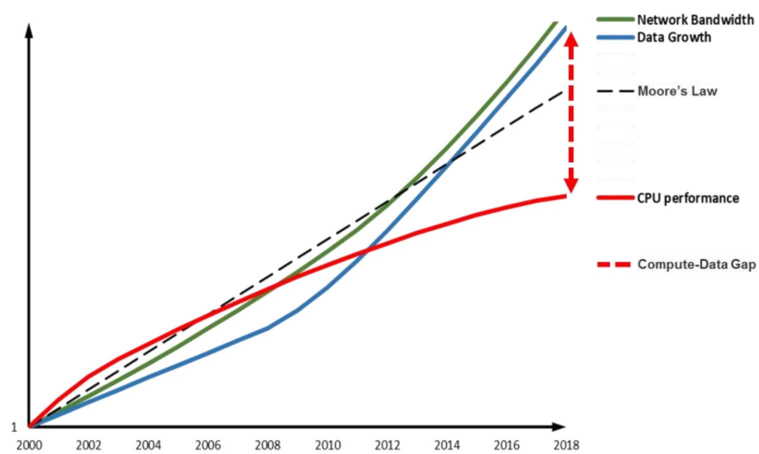
Why is TMA challenging?

- Modern Internet was designed without integrated monitoring mechanisms
- Monitoring capabilities of network devices (e.g., SNMP) are very limited
- Measurements have to be performed (mostly) by third-party tools
- Internet is fully decentralized, there is no central place where global statistics can be collected

Why is TMA challenging?

- Internet measurement is one of the biggest Big Data problems ever!
  - Huge data sets that are difficult to store, transfer, process and analyze
  - Optical speeds grow much higher than electronic speeds
  - Already in 1991, sampling was necessary to collect traffic statistics

Why is TMA challenging?



Source: https://blog.mellanox.com/2019/08/the-end-of-moores-law-and-the-return-of-cleverness/

Classification

- Classification of network monitoring methods
  - Hardware vs. software
  - Online vs. offline
  - LAN vs. WAN
  - Protocol level
  - **Active vs. passive**

Active monitoring

- Active tools are based on traffic injection
  - Probe traffic generated by a measurement device
  - Response to probe traffic is measured
- Pros: Flexibility
  - Devices can be deployed at the edge (e.g., end-hosts)
  - No instrumentation at the core is needed
  - Measurement does not directly rely on existing traffic

Active monitoring

- Cons: Intrusiveness
  – Probe traffic can degrade network performance
  – Probe traffic can impact the measurement itself
- Main usages
  – Performance evaluation (e.g., ping)
  – Bandwidth estimation (e.g., pathload)
  – Topology discovery (e.g., traceroute)

Passive monitoring

- Traffic collection from inside the network
  – Routers and switches (e.g., Cisco NetFlow)
  – Passive devices (e.g., libpcap, DAG cards, optical taps)
- Pros: Transparency
  – Network performance is not affected
  – No additional traffic is injected
  – Useful even with a single measurement point

Passive monitoring

- Cons: Complexity
  - Requires administrative access to network devices
  - Requires explicit presence of traffic under study
  - Online collection and analysis is hard (e.g., sampling)
  - Privacy concerns
- Multiple and diverse usages
  - Traffic analysis and classification, . . .
  - Anomaly and intrusion detection, . . .
- *Passive monitoring* is the basis of most Network Security Monitoring Tools (NSM)

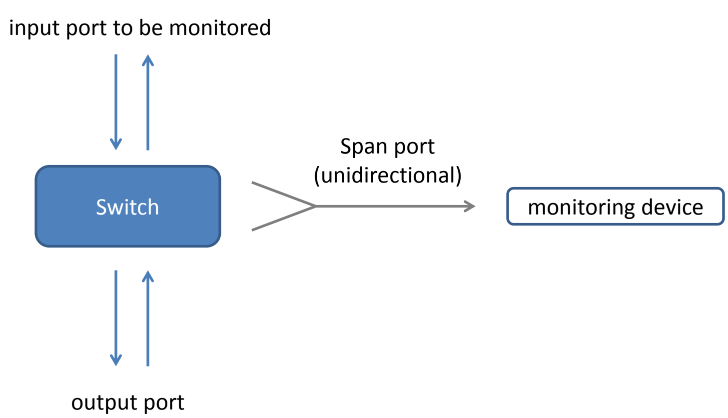
Traffic collection

- Different approaches for passive traffic collection
  - Port mirroring (SPAN port)
  - Test Access Port (TAP)
  - Flow monitoring (NetFlow/IPFIX)

Port mirroring

- Usually available in enterprise-grade routers and switches
- Traffic from one or more port (tx, rx or both) is copied (mirrored) to another port or interface
- Output port is called *SPAN* port (switched port analyzer)
- Both directions are transmitted in one direction

Port mirroring



Port mirroring

- Pros
  - Simplicity
  - Readily available, easy to deploy
- Cons
  - Full-duplex link mirrored to a single direction
  - Sum of throughput larger than mirror port tx.
  - Computational power (switching is prioritized)

Port mirroring

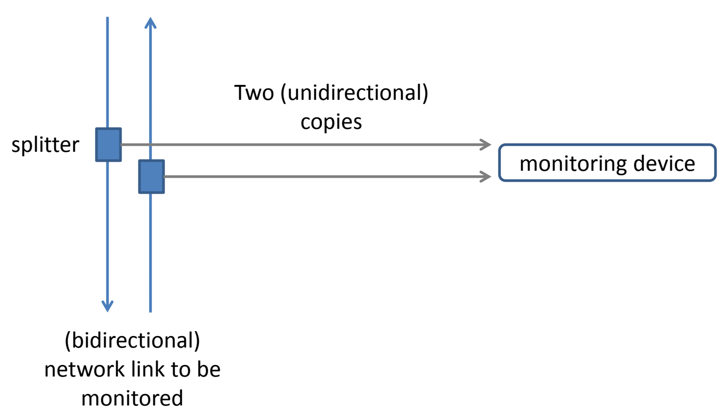
- Example configuration of a SPAN port on a Catalyst 2960 series switch

```
Switch(config)# no monitor session 1
Switch(config)# monitor session 1 source interface gigabitethernet0/1
Switch(config)# monitor session 1 destination interface gigabitethernet0/2
Switch(config)# end
```

TAP

- Test Access Port (TAP)
- Packet capture device in inline mode (e.g., optical splitter)
- Line is split, traffic duplicated passively
- Both directions are transmitted separately

TAP



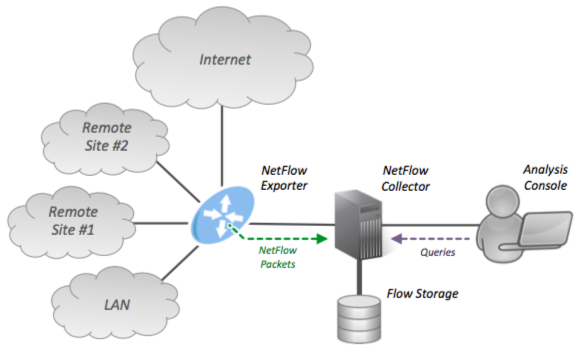
TAP

- Pros
  - Passive devices (do not require power)
  - No computational performance issues
- Cons
  - Passive connection attenuate the signal (original + copy)
  - Regeneration is needed if distance is large
  - Difficult to deploy, requires operational outage

Flow monitoring

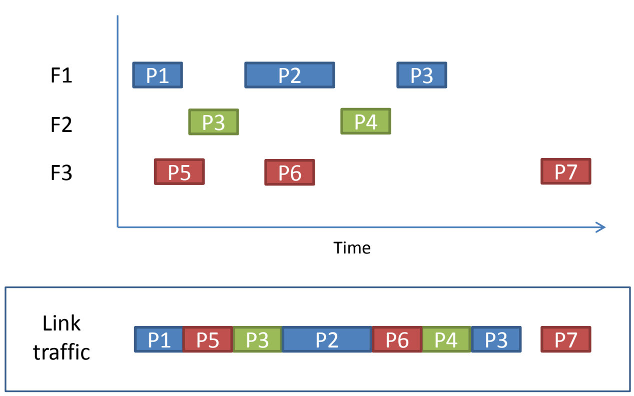
- Traffic is aggregated into flows by a router/switch and sent to an external collector (e.g., NetFlow, IPFIX)
- RFC 7011: *"A Flow is defined as a set of packets or frames passing an Observation Point in the network during a certain time interval. All packets belonging to a particular Flow have a set of common properties"*
- Usually identified by a 5-tuple: <src ip, dst ip, src port, dst port, L4 protocol>
- Simple metrics for each flow: <#pkts, #bytes, ts0, tsf, flags>

Flow monitoring



https://commons.wikimedia.org/w/index.php?curid=21685577

Flow aggregation example



Flow id	#pkts
F2	2
F3	3
F1	3

Flow table

F1 = h(<192.168.1.100, 147.83.2.135, 40764, 80, 6>)
F2 = h(<192.168.1.100, 147.83.2.135, 40765, 443, 6>)
F3 = h(<192,168.1.200, 8.8.8.8, 60346, 53, 17>)

Flow monitoring

- Pros
  - Easy to deploy, already integrated in devices
  - Traffic is aggregated, less storage requirements
  - Only packet headers are analyzed
  - Less privacy sensitive

- Cons
  - Computational requirements in routers
  - Commonly resort to sampling (e.g. 1/1000, 1/10000)
  - Only flow-level information is available

Network monitoring

Algorithms and challenges

Pere Barlet-Ros

Computer Architecture Department (UPC)
http://people.ac.upc.edu/pbarlet
pbarlet@ac.upc.edu

PART B

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Outline

Technological challenges

- Few *ns* per packet
  - Interarrivals 8*ns* (40Gb/s), 32*ns* (10Gb/s)
  - Memory access times $< 10ns$ (SRAM), tens of *ns* (DRAM)
- Obtaining simple metrics becomes extremely challenging
  - Approaches based on hash tables do not scale
  - Core of most monitoring algorithms
  - E.g., Active flows, flow size distribution, heavy hitter detection, delay, entropy, sophisticated sampling, . . .
- Probabilistic approach: trade accuracy for speed
  - Extremely efficient compared to compute exact answer
  - Fit in SRAM, 1 access/pkt
  - Probabilistic guarantees (bounded error)

Technological challenges

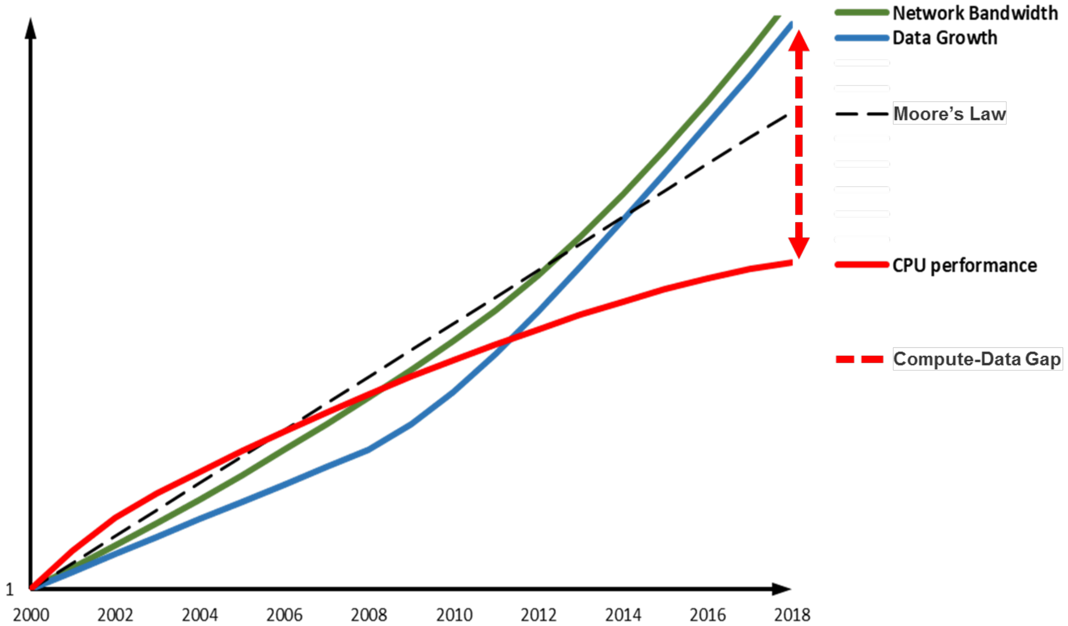


Figure: CPU-network data rates gap[1]

[1] https://blog.mellanox.com/2019/08/the-end-of-moores-law-and-the-return-of-cleverness/

Technological challenges

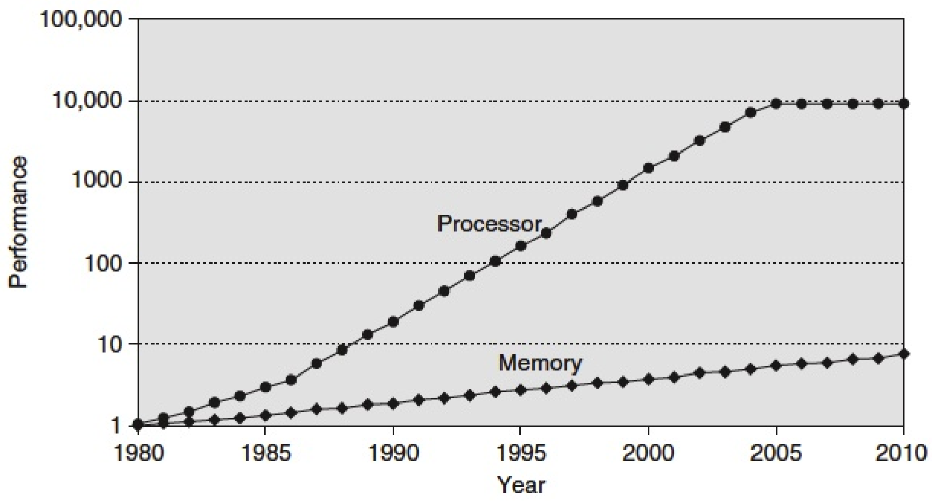


Figure: CPU-memory gap[2]

[2] Hennessy & Patterson. Computer Architecture: A Quantitative Approach, 2011

Technological challenges: : Example

- Simplest network monitoring program ever: Packet counter

```
LOAD   R1, address
ADD    R1, R1, 1
STORE  address, R1
```

Traffic sampling

- Most widely used technique to cope with high traffic rates
  - Standarized by the PSAMP working group at the IETF
  - Already implemented in most routers (e.g., Sampled NetFlow)
- Uniform packet sampling
  - Select a packet with a uniform probability p
  - Number of packets is estimated as $\widehat{n} = n/p$
  - Does not work for everyting (e.g., flow count, flow size distribution)
- Hash-based sampling
  - Compute $h(f)$, where h is a random hash function that maps $[1..k]$
  - If $h(f) < kp$ then the packet is selected
  - Total number of flows is estimated as $\widehat{f} = f/p$
  - All packets of a flow are either selected or discarded
  - Implements *trajectory sampling* if same h is used

Bloom filters[3]

- Space-efficient data structure to test set membership
  - Based on hashing (e.g., pseudo-random hash functions)
- Examples of usage in network monitoring
  - Replace hash tables to check if a flow has already been seen
  - Definition of flow is flexible
  - Traffic filtering
- Advantages
  - Small memory (SRAM) is needed compared to hash tables
- Limitations
  - False positives are possible
  - Removals are not possible (counting variants can support them)

[3] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. Commun. ACM, 13(7), 1970.

Bloom filters

- Parameters
  - k: #hash functions
  - b: size of the bitmap
  - p: false positive rate
  - n: #elements in the filter (max)

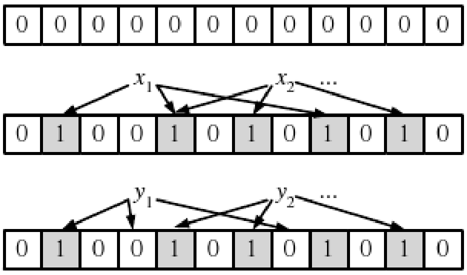


Figure: Example of a bloom filter[4]

[4] A. Broder and M. Mitzenmacher. Network Applications of Bloom Filters: A Survey. Internet Mathematics, 1(4), 2005.

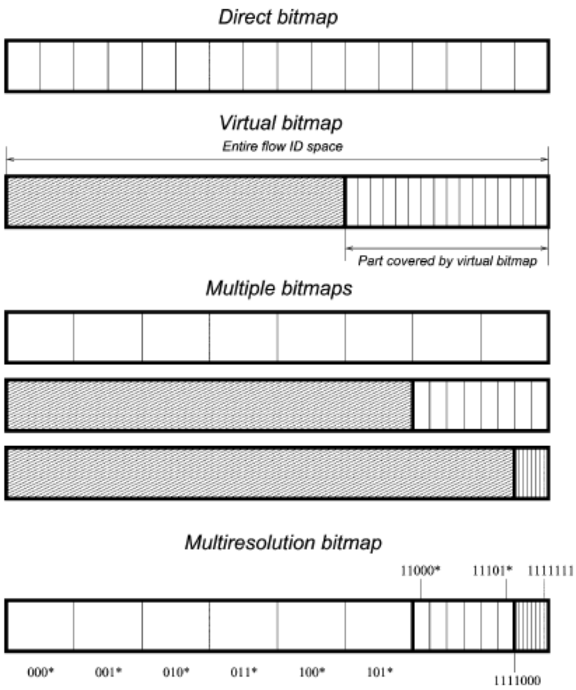
Direct bitmaps (linear counting)[5]

- Space-efficient algorithms to count the number of unique items
  - E.g., useful to count the number of flows over a fixed time interval
- Basic idea
  - Each flow hashes to one position (and all its packets)
  - Counting the number of 1's is inaccurate due to collisions
  - Count the number of unset positions instead
  - E.g., 20KB to count 1M flows with 1% error
- Estimate formulae
  - Flow hashes to a given bit: $p = 1/b$
  - No flow hashes to a given bit: $p_z = (1-p)^n \approx (1/e)^{n/b}$
  - Expected non-set bits: $E[z] = bp_z \approx b(1/e)^{n/b}$
  - Estimated number of flows: $\hat{n} = b \ln(b/z)$

[5] K.-Y. Whang *et al.* A linear-time probabilistic counting algorithm for database applications. ACM Trans. Database Syst., 15(2), 1990.

Bitmap variants[6]

- Direct bitmaps scale linearly with the number of flows
  - Variants: Virtual, multiresolution, adaptive, triggered bitmaps, . . .



[6] C. Estan, G. Varghese, M. Fisk. Bitmap algorithms for counting active flows on high speed links. IEEE/ACM Trans. Netw. 14(5), 2006.

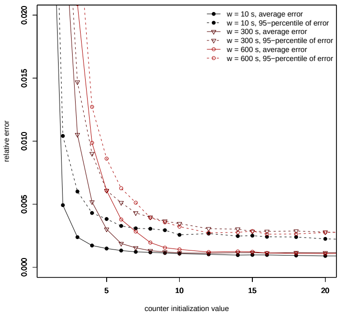
Bitmaps over sliding windows

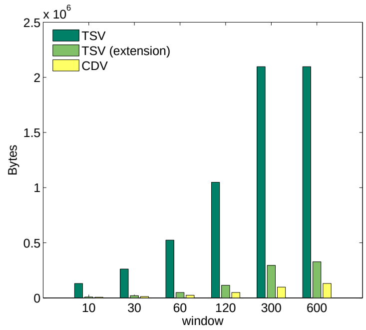
- Timestamp Vector (TSV)[7]
  - Vector of timestamps (instead of bits)
  - $O(n)$ query cost
- Countdown Vector (CDV)[8]
  - Vector of small timeout counters (instead of full timestamps)
  - Independent query and update processes
  - $O(1)$ query cost

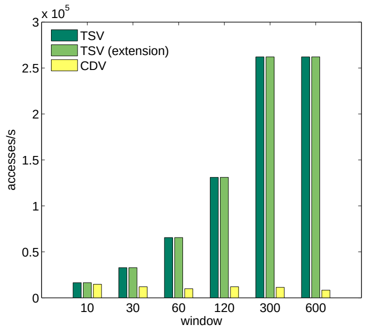
[7] H. Kim, D. O'Hallaron. Counting network flows in real time. In Proc. of IEEE Globecom, Dec. 2003.
[8] J. Sanjuàs-Cuxart *et al.* Counting flows over sliding windows in high speed networks. In Proc. of IFIP/TC6 Networking, May 2009.

CDV and TSV performance

- 30-min trace, 271 Mbps, 1 query/s, 50K/10s-1.8M/min flows[9]







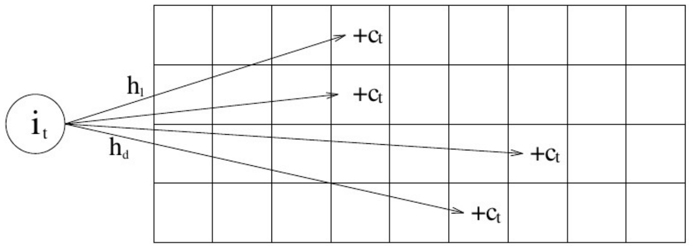
9 A hash table would require several MBs with these settings

Count-min sketch

- Limitations of Bloom filters and bitmaps
  - Estimate a single metric
  - Do not support per-flow measurements
  - Flow measurements require per-flow state
- Count-min sketch (CMS)
  - Compact summary of a large amount of data
  - Sub-linear probabilistic data structure (size $< N$)
  - BF represents sets, CMS multi-sets
- Examples of usage in network monitoring
  - Count number of packets per flow (flow size)
  - Heavy-hitter detection

CMS implementation[10]

- Data structure
  - Matrix of d rows and w columns ($w = \lceil \frac{e}{\varepsilon} \rceil, d = \lceil ln\frac{1}{\delta} \rceil$)
  - Query errors within a factor of ε with probability δ
  - Uses d pseudo-random hash functions that map $[1..w]$



- Algorithm
  - Update: $\forall_{1 \leq j \leq d} : count[j, h_j(i_t)] \leftarrow count[j, h_j(f)] + c_t$
  - Query: $\widehat{a_i} = min_j(count[j, h_j(i)])$

[10] G. Cormode, S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. J. Algorithms, 2005

One-way delay measurement

- Traditional approaches are expensive
  - Probing traffic (intrusive)
  - Trajectory sampling
  - Constant overhead per sample
- Alternative: LDA (Lossy Difference Aggregator)[11]
  - Send only sums of timestamps
  - Deal with packet loss (sampling + partition input stream)

[11] R. Kompella *et al.* Every microsecond counts: tracking fine-grain latencies with a lossy difference aggregator. SIGCOMM, 2010.

Lossy Difference Aggregator (LDA)