Identifying Frequent Items in Sliding Windows over On-Line Packet Streams

Alejandro López-Ortiz
School of Computer Science
University of Waterloo

Joint work with Lukasz Golab (Waterloo), David DeHaan (Waterloo), Erik Demaine (MIT), and J. Ian Munro (Waterloo)

Application

- Real-time analysis of network traffic
  - find frequently appearing packet types
    - Packet type: port #, protocol type, source IP.
  - But, interested in recent usage trends
    - E.g. for routing system analysis or anomaly detection
  - So, want to find frequently appearing packets in a sliding window of N most recent packets

If we could store the entire window:

University of Waterloo

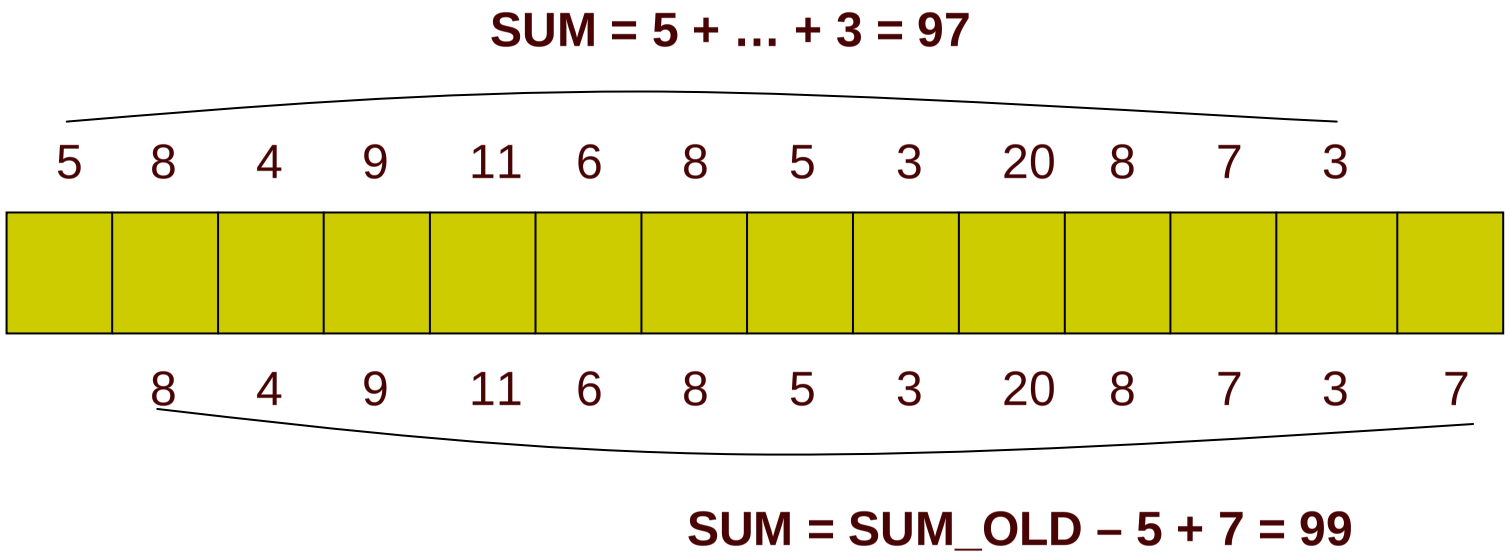
- Maintain frequency counts of each category in the window
- Update counters as new packets arrive and old packets are expired out of the window
- Periodically scan counters and return the packet types corresponding to the *k* largest counters (and possibly the actual counts too)

What if we can't store the entire window?

University of Waterloo

- Idea from [Zhu, Shasha, VLDB '02] :
  - Divide the sliding window into sub-windows, i.e. use a coarser time grain of T packets
  - Store summary for each sub-window
  - Every T packets:
    - Expire oldest sub-window
    - Add most recent sub-window
    - Update answer
  - Space req: $\frac{window}{T} \times summary$

Example: windowed SUM



Updating Top-k counters

- T_b = current count for packet of type *b*
- Update:

$$T_b = T_b - T_b(\text{old sub-window}) + T_b(\text{new sub-window})$$

- Problem is: T_b(*old sub-window*) might not be part of summary in *old sub-window*

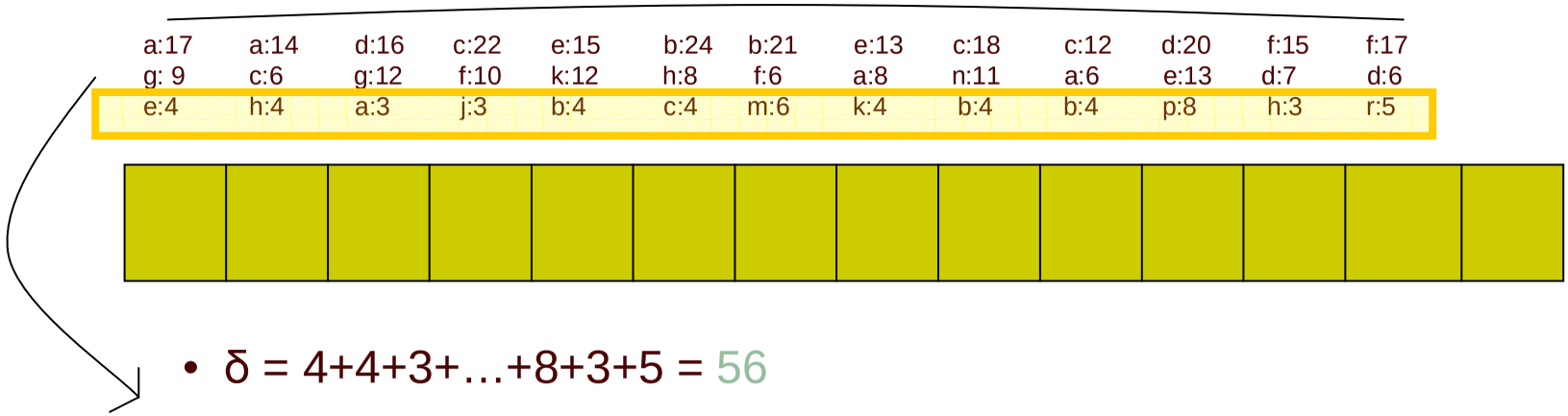
…but, let's use the technique anyway

University of Waterloo

- Sub-window summary: IDs and counts of the *k* most frequent categories
- δ = sum of the occurrence count of least frequent item in summary of each sub-window
- Compute overall occurrence count for each packet type from sub-window summaries
- Packets exceeding count δ are reported as top-*k*

The algorithm

- Let a, b, c, … be distinct packet types, let k = 3



- δ = 4+4+3+…+8+3+5 = 56

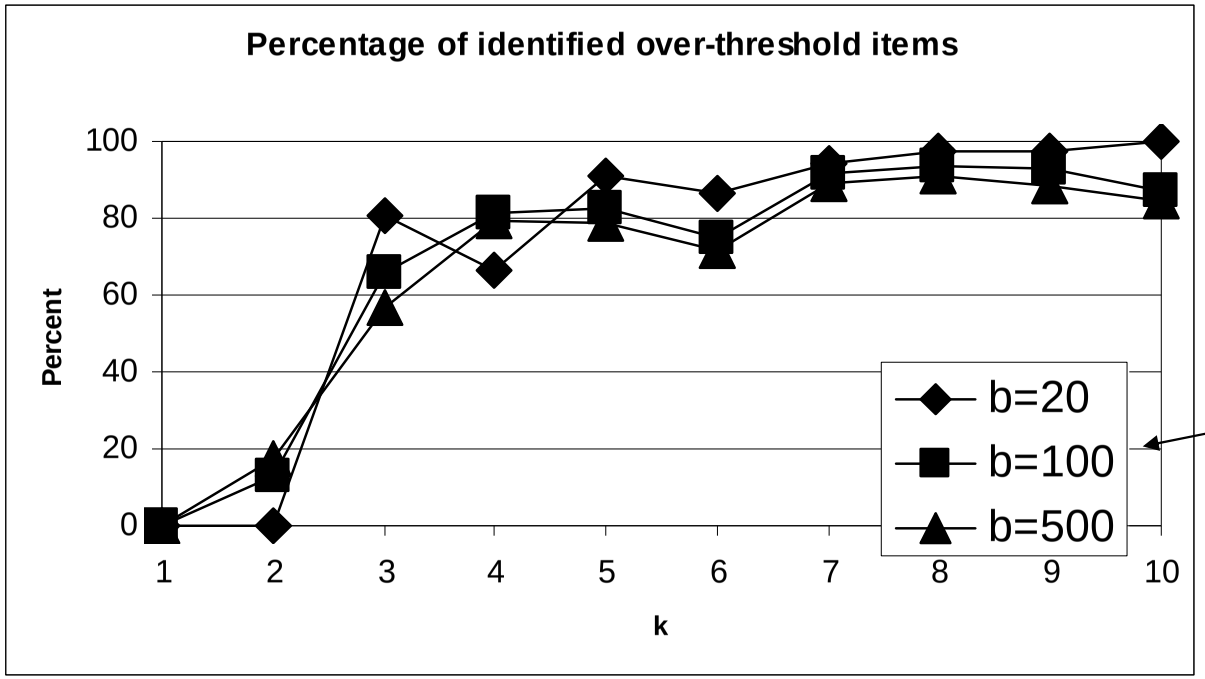
- Total frequency counts from the top-k lists: a=48, b=57,c=62,d=49,e=45, f=48,g=21,h=12,j=3,k=16,m=6,n=11,p=4,r=5

- Return **b** and **c** as frequent items in this window

Hypothesis

- If categories are +/- equally distributed, previous method may not work
- But, in a Power Law distribution, we expect a few heavy flows which should register on many top-*k* lists
- Experimented with a TCP trace
  - 1 month of traffic from Lawrence Berkeley Lab to the rest of the world; almost 800 000 packets in total
  - 1647 distinct source IP addresses, which we treated as distinct categories

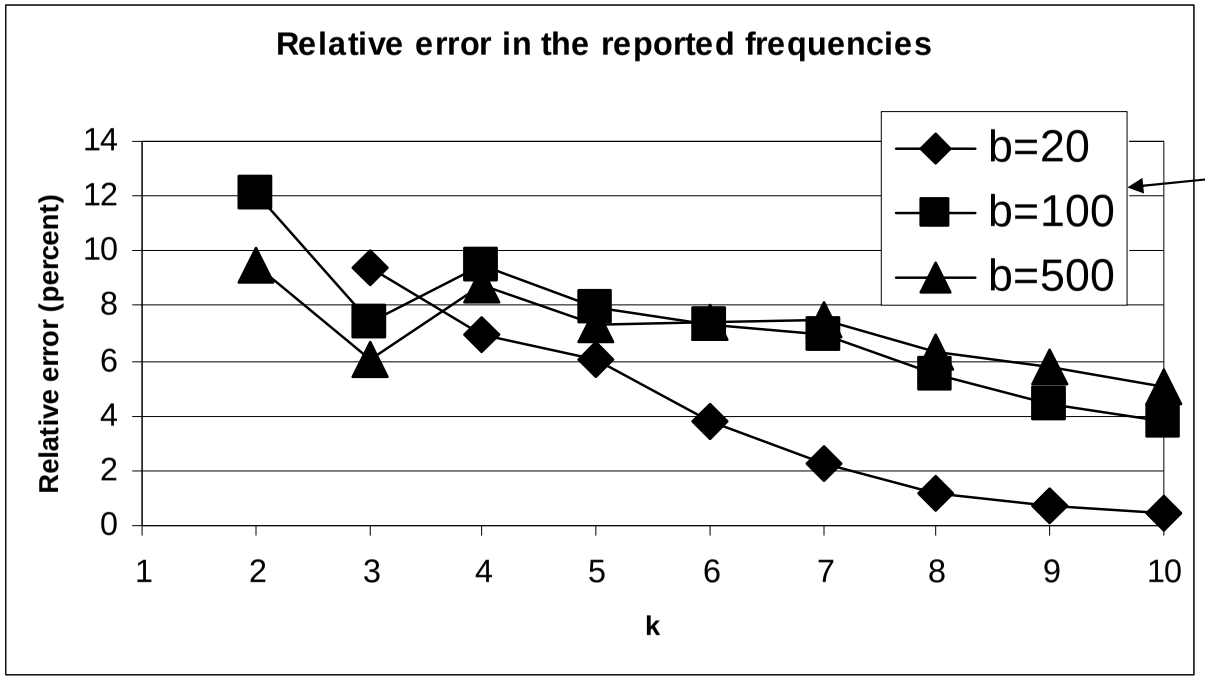
Results: accuracy



size of each sub-window

Window size = 100 000 packets

Results: precision of the reported frequencies



size of each sub-window

Window size = 100 000 packets

Conclusions

- Extended sub-window model to a holistic aggregate
- Good results due to the non-uniform distribution of Internet traffic
- Low space requirements