Bitmap Algorithms for Counting Active Flows on High Speed Links

Elisa Jasinska
jasinska@informatik.hu-berlin.de

Reference

- Cristian Estan, George Varghese & Mike Fisk, “Bitmap Algorithms for Counting Active Flows on High Speed Links”, Internet Measurement Conference 2003
- Paper: http://www.imconf.net/imc-2003/papers/p327-estan.ps
- Presentation: http://www.cs.ucsd.edu/users/cestan/papers/FlowCountingBitmaps.ppt

Agenda

- Flow?
- Counting Flows?
- Motivation
- Bitmap Algorithms
- Number of Flows?
- Tests and Results
- Summary

Flow?

- Constant values in certain fields of packet header
  - Eg. source and destination IP address, port numbers, etc.
  - “FlowID”
- Distinction between active and inactive flows

Counting Flows?

- Portscans
- Denial of Service attacks (DoS)
- General measurements
- Packet scheduling

Motivation

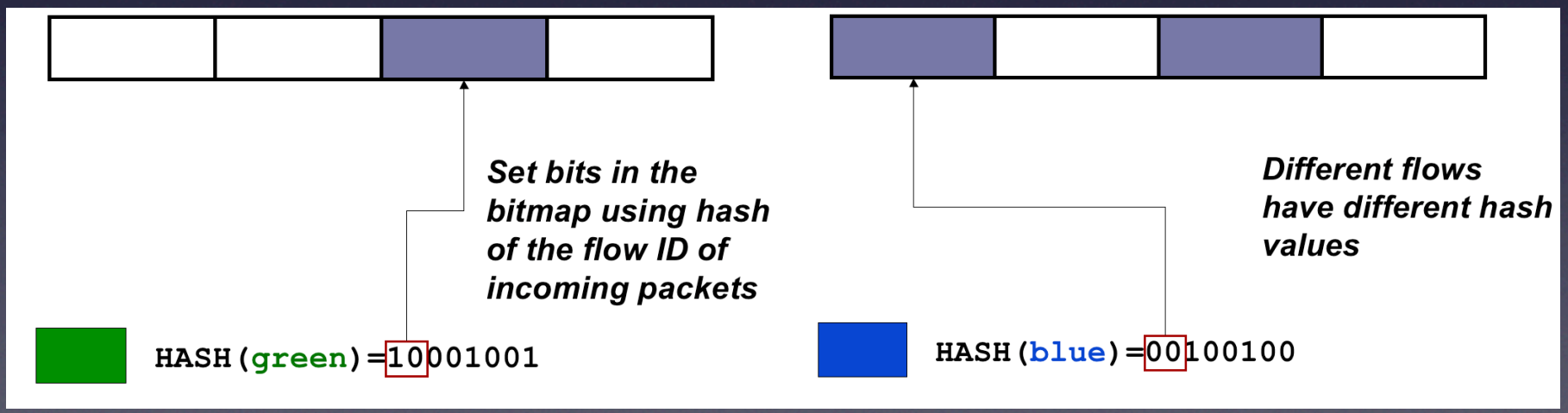
- Counting flows on high speed links needs:
  - Fast processing
  - Little (and fast) memory usage
- Bitmaps just set a bit
  no read-modify-write operation needed!

Bitmap Algorithms

- Direct Bitmap
- Virtual Bitmap
- Multiresolution Bitmap
- Derived Algorithms

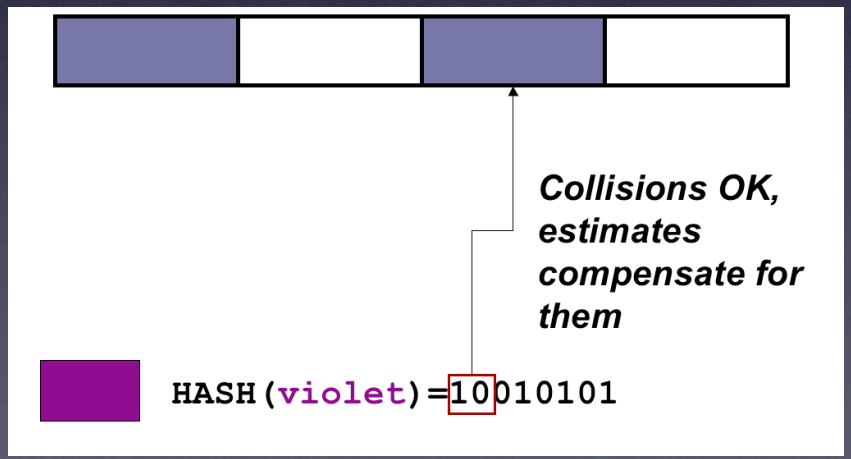
Direct Bitmap

- Hash function on flowID
- Set bit accordingly



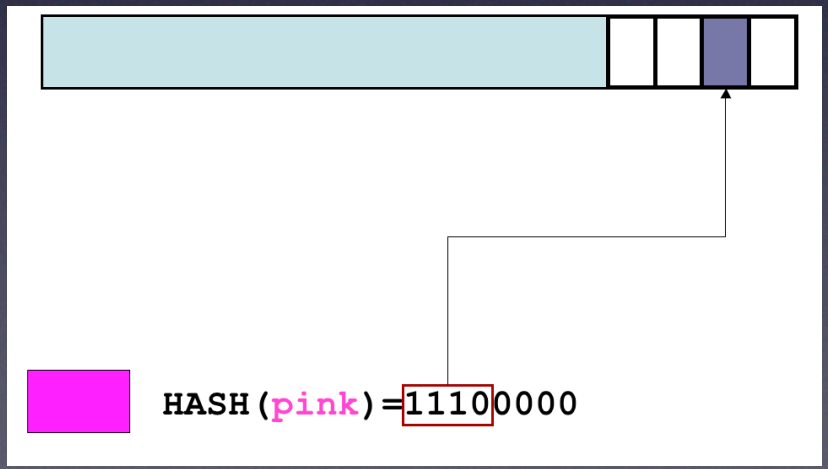
Direct Bitmap

- Collisions: Different flowID can result in same bit
- Estimation error large with fill grade > 50%
- Bitmap size must grow linearly with the amount of flows ⇒ more memory usage



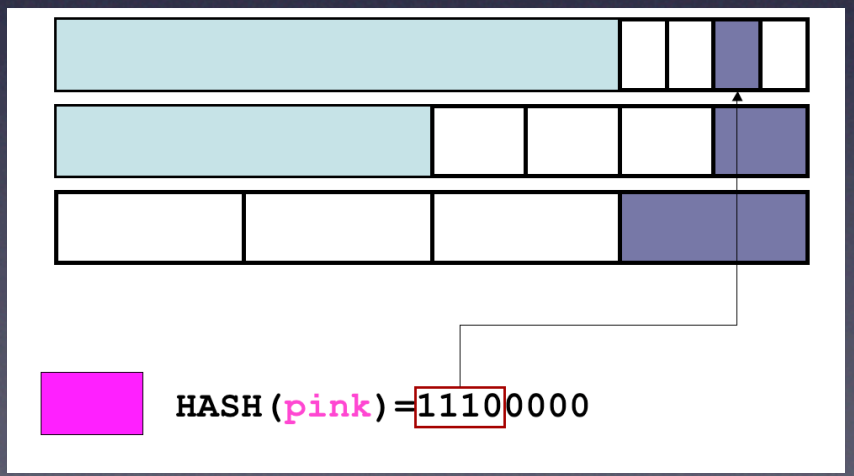
Virtual Bitmap

- Writes only part of the direct bitmap that would be needed for accurate result
- Reduced memory usage
- Information about flow amount needed upfront



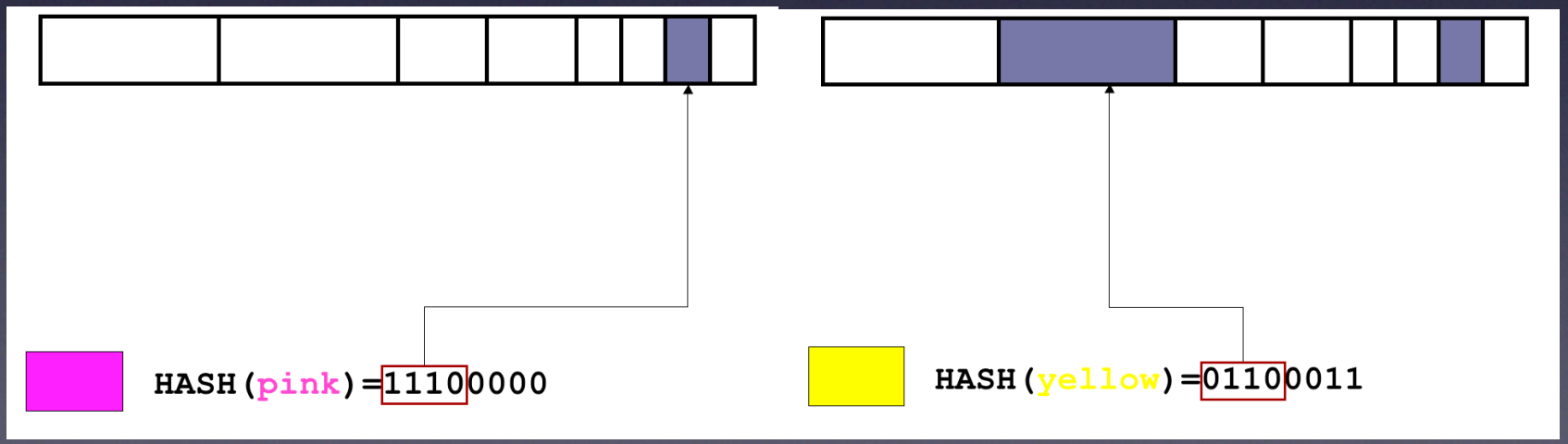
Multiple Bitmaps

- Multiple virtual bitmaps with different resolutions
- Allows us to choose the most accurate one
- More memory usage! More bit-set operations!



Multiresolution Bitmap

- One bitmap, different resolutions
- Less memory usage! Less bit-set operations!
- Multiple virtual bitmaps can be calculated back through OR operations



Derived Algorithms

- Adaptive Bitmap, combination of...
  - Large Virtual Bitmap and
  - Small Multiresolution Bitmap
- Triggered Bitmap, combination of...
  - Small Direct Bitmap and
  - Large Multiresolution Bitmap

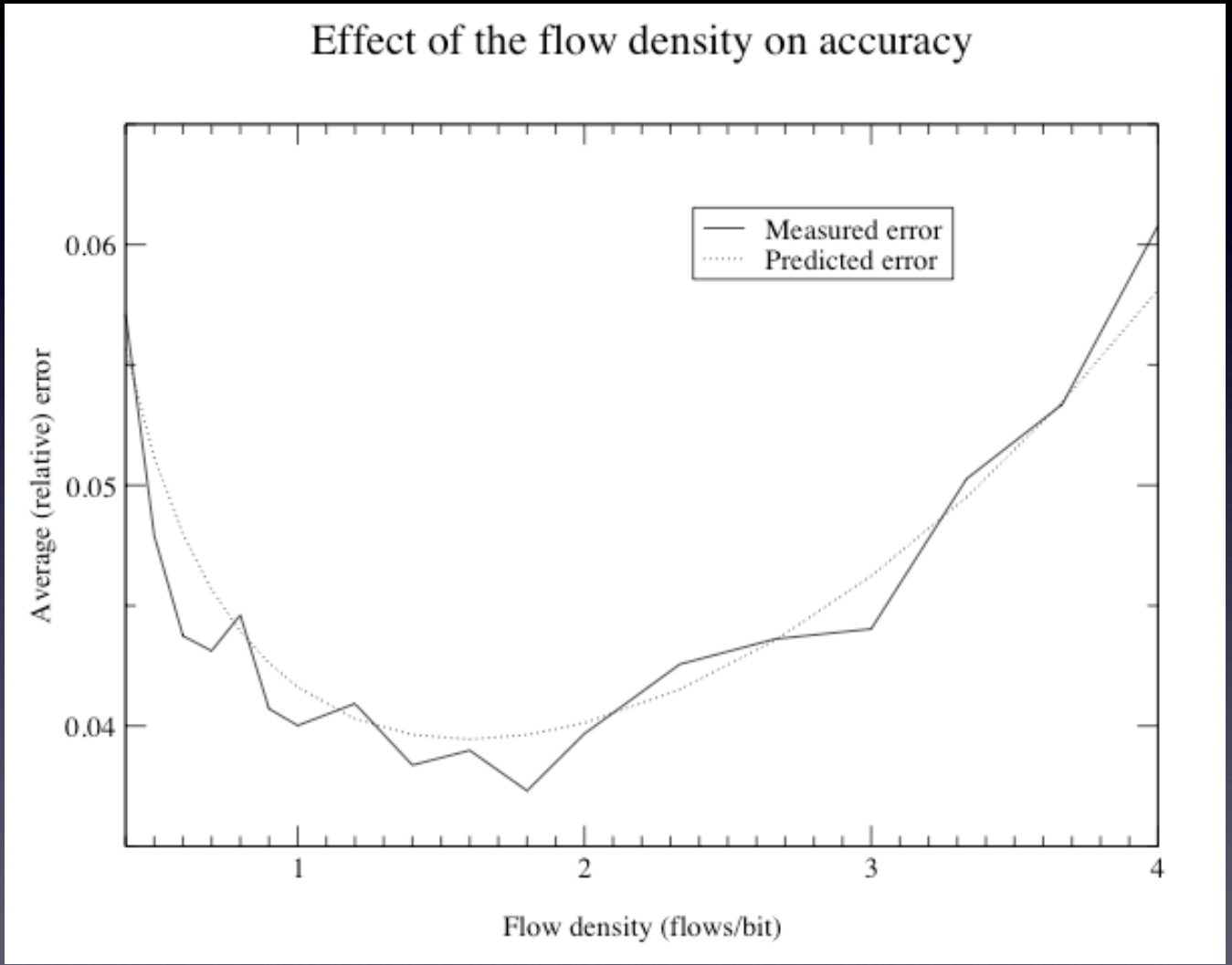
Number of Flows?

- Estimation due to collisions
- Expected error depends on flow density
- Important factor: configuration
  (size of bitmaps, etc.)

Tests and Results

- Error in # of flows with Virtual Bitmap with different flow densities
- Memory usage comparison of Snort, probabilistic counting and Triggered Bitmap

Virtual Bitmap Error



Triggered Bitmap Memory Usage

- Snort
  - Minimum memory usage in an implementation using naive algorithm
- Probabilistic counting
  - Based on multiresolution bitmap
- Triggered Bitmap

Triggered Bitmap Memory Usage

Interval	Snort	Calculation	Triggered Bitmap
12 sec.	1 968 K	2 474 K	381 K
600 sec.	50 791 K	22 876 K	5 725 K

Summary

- Use bitmaps to count flows
  - On high speeds
  - With reduced memory usage
- Only set a one bit
  - ➡ No read-modify-write operation needed
  - ➡ Less memory usage

Summary

- Hash function reduces number of results
  - ➡ Even less memory usage
  - ➡ Fast SRAM can be used
- Number of flows calculated statistically
  - ➡ Error estimations feasible

Thanks for listening!
Questions?