University of Amsterdam
System and Network Engineering

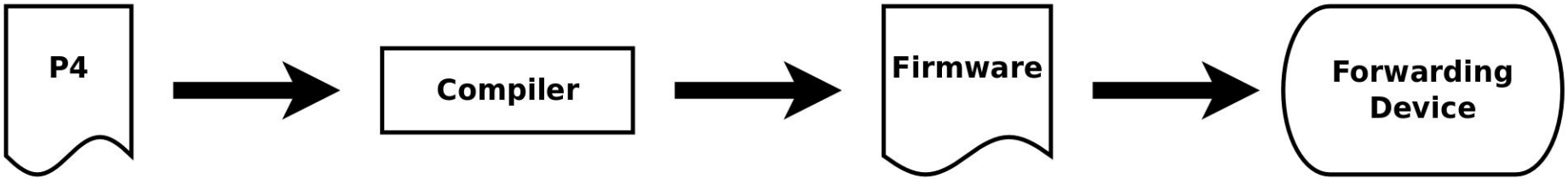
Tracking Network Flows with P4

Joseph Hill
Mitchel Aloserij
Paola Grosso

Funded by SURF NET

What is P4?

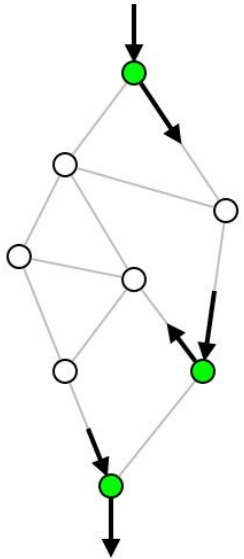
- P4: Programming Protocol-Independent Packet Processors
- Programming language that describes the behavior of the data plane
- Allows the rapid development and deployment of protocols
- An alternative to fixed function devices (e.g. Switches, Routers, Firewalls…)
- $P4_{14}$ with software switch provided by the P4 Language Consortium



Motivation

Can the features provided by P4 be used to enhance the ability to identify and track malicious traffic in networks?

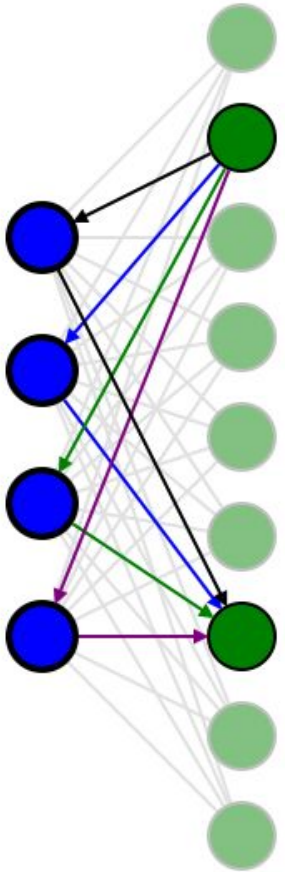
- Goals
  - Identify as much of the path of a flow as possible
  - Minimize impact on system resources
- Assumptions
  - Traffic may be classified as malicious at later time
  - NetFlow is unable to capture all flows at every node



Sampled NetFlow may result in incomplete knowledge of the path

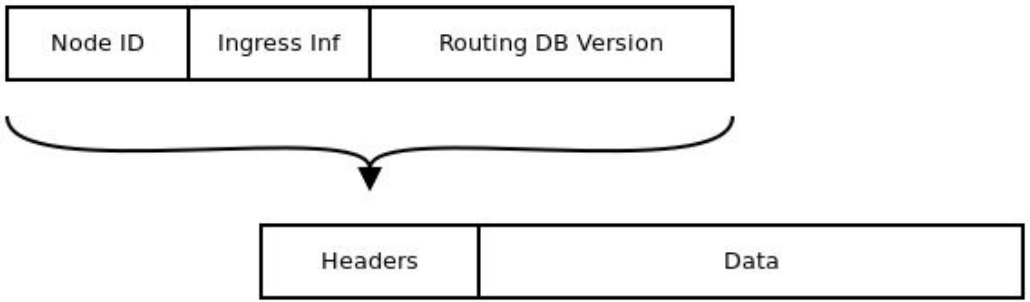
Path Tracking Methods: Hop Recording

- Each node adds its node ID to a header stack
- Flow and Path saved at egress node
- Not a novel approach
  - RFC 791 - Internet Protocol, 1981
  - In-band Network Telemetry (INT), 2015
- Variable amount of data added to packet
- Useful when load balancing over several short paths



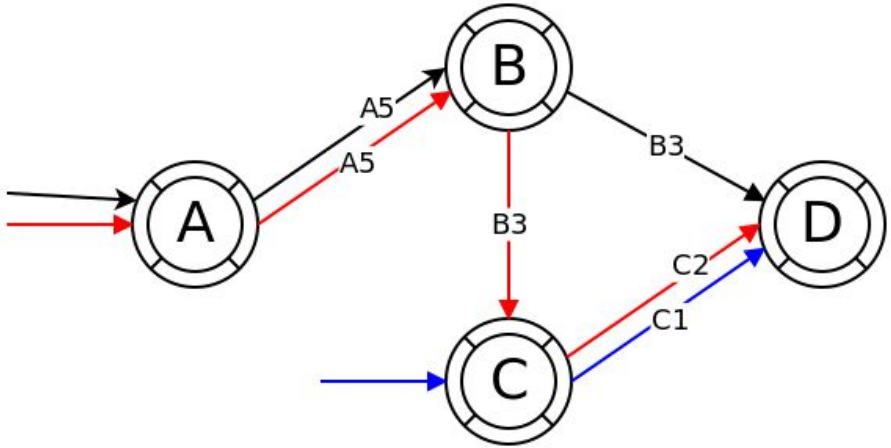
Path Tracking Methods: Forwarding State Logging

- Network forwarding state identifier added to packet
- Use a record of forwarding state to recreate path
- Requires global view of how routing will be performed
- Requires archives of each version of routing database



Path Tracking Methods: Dynamic Path Labeling

- MPLS style labels are generated for each path taken by packets
- Labels identify the history of a packet rather than future
- Final label stored with flow at egress node

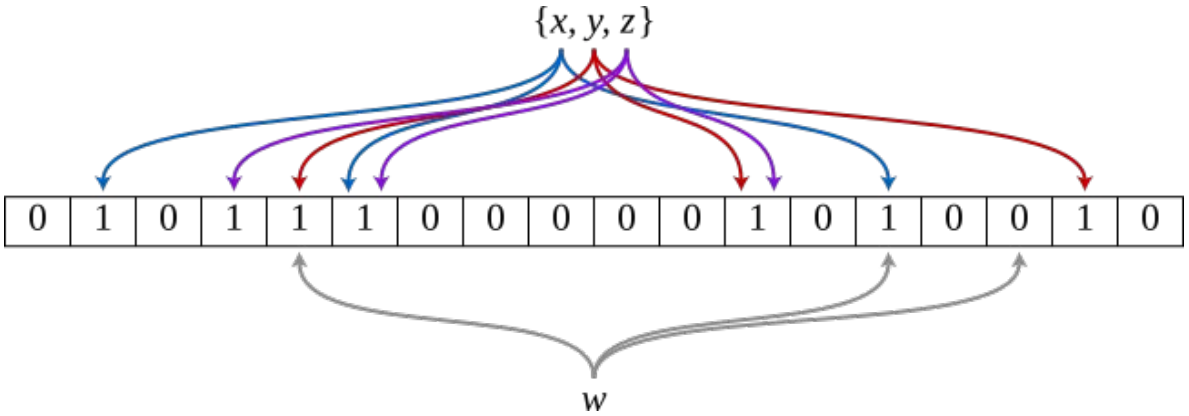


Bloom Filters

- Probabilistic Set
- Possible false positives, No false negatives
- Configurable Accuracy
- Fixed size data structure
- Independent of size of item being added

How Bloom Filters Work

- Bloom Filter is made up of fixed number of bits (m)
- Has predetermined number of hash functions (k)
- The hash functions determine which bits would be set if an item was in the set
- Items can only be added, not removed
- As items are added chance for a false positive increases



Bloom Filters in P4

- Can be implemented completely in data plane using P4
- Used to determine when a new flow has been seen
- Control Plane will only receive packets that are from a new flow
- Data Plane can send to centralized point without involving the Control Plane

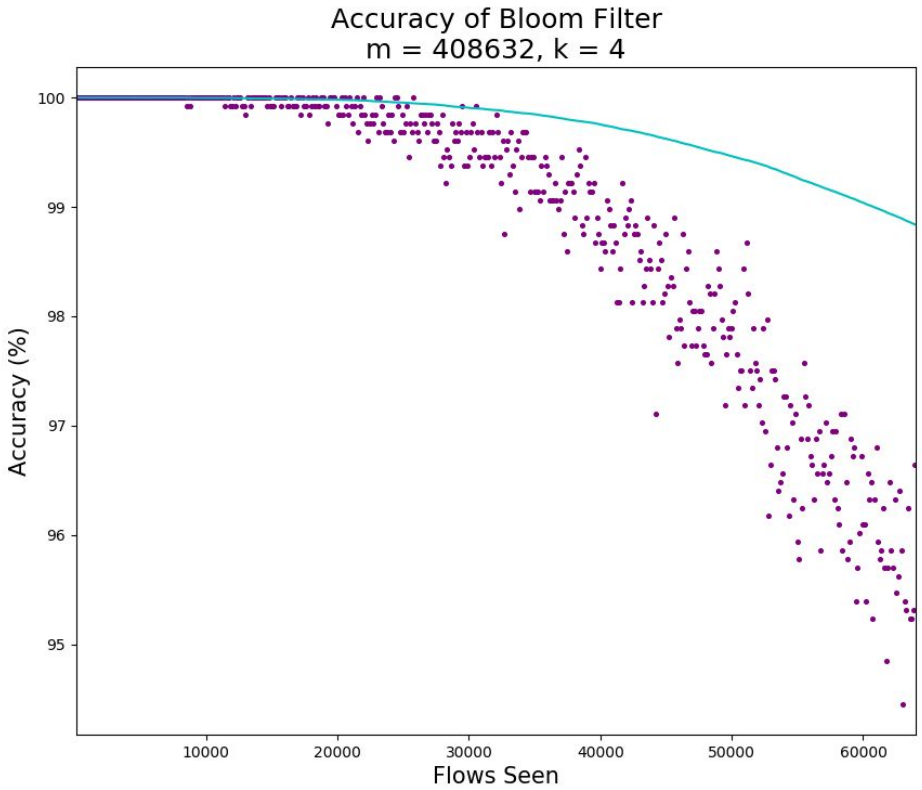
```
bit bloom_filter[BF_WIDTH]

action bf_check_bit(salt) {
      index = hash(hash_fields, salt) % BF_WIDTH
      bit = bloom_filter[index]
      hit = hit & bit
}

action bf_set_bit(salt) {
      index = hash(hash_fields, salt) % BF_WIDTH
      bloom_filter[index] = 1
}
```

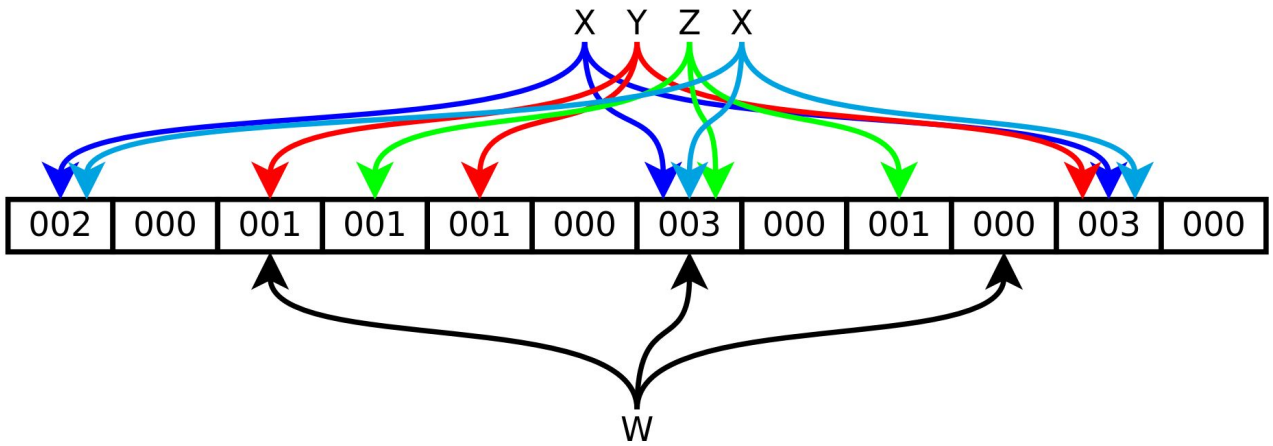
Practical Example

- Set a goal of 64000 flows (n) with a false positive rate of 5% (p)
- 64001st flow has a 5% chance of falsely being detected as previously seen
- Actual false positive rate for the first 64000 flows was 1.16%



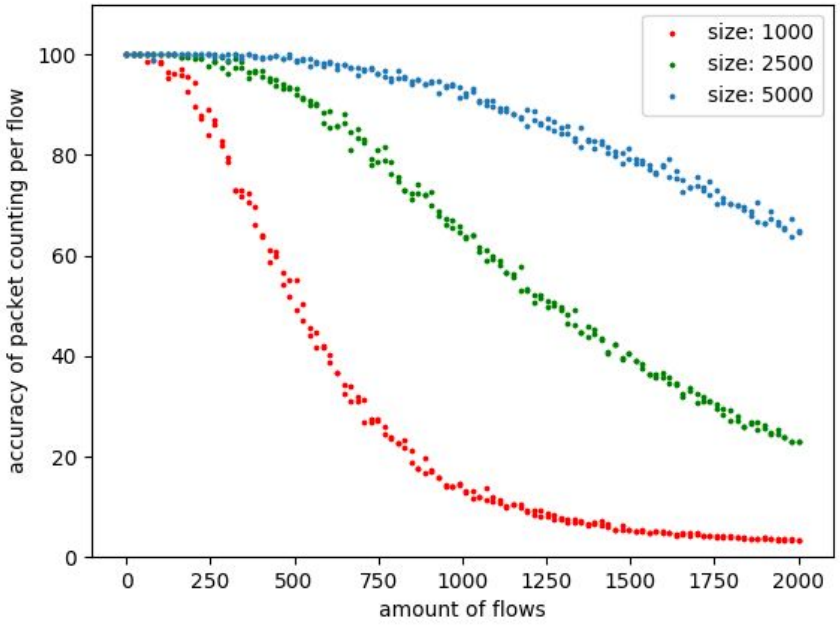
Counting Bloom Filters

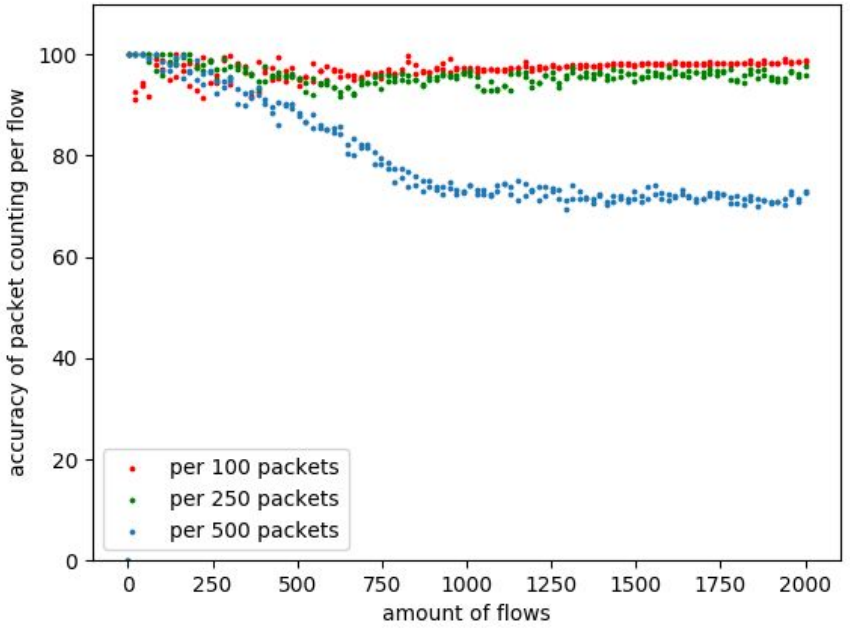
- Can now tell how many times a flow has been seen
- Uses substantially more memory
- Possible concurrency issues
- Inaccuracies now result in inflated counts



Decrementing Bloom Filters

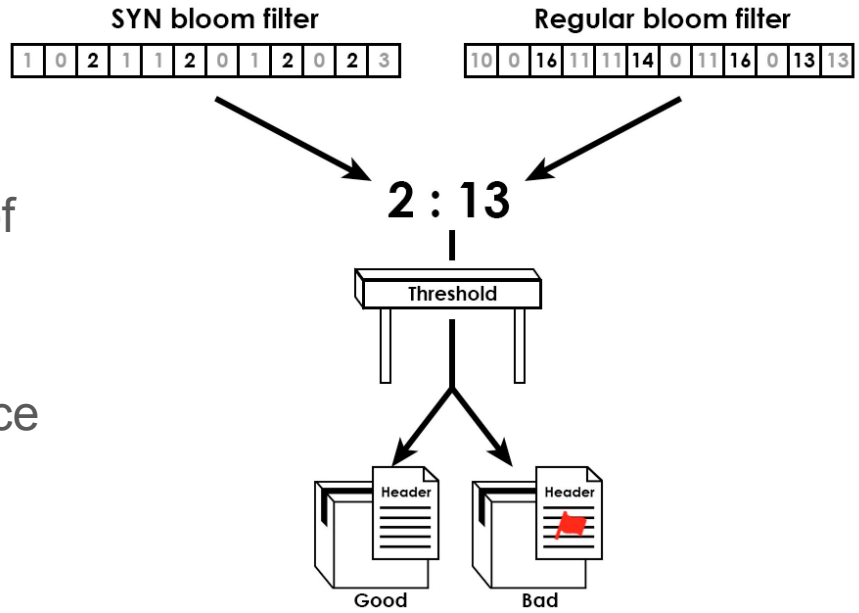
- Counteract a Bloom Filter ‘filling up’
- Can decrement based on time or packets





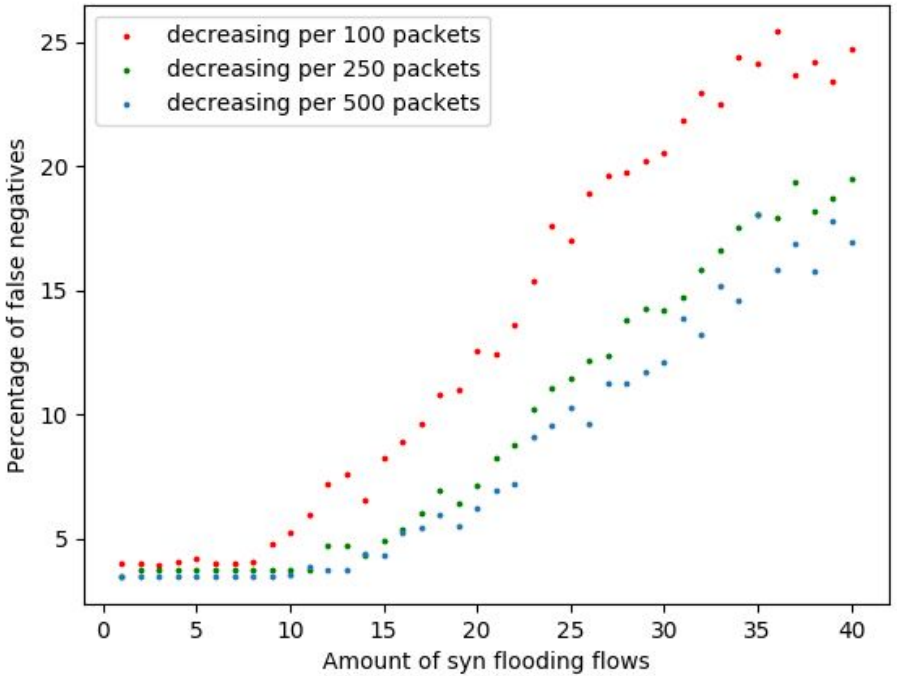
Application of Counting Bloom Filters

- Detecting SYN attacks using ratio of SYN to non-SYN packets
- Relatively low-bandwidth attack
- Assuming single or aggregate source



SYN Attack Detection Performance

- SYN attack is using 20 packets per second
- Lower decrement rate increases accuracy
- Lower performance the more distributed the attack



Conclusions

- Need to look at the additional constraints that hardware will impose
- The effectiveness of the flow tracking methods depends on the topology of the network and resource constraints
- Bloom Filters can provide high accuracy with minimal resource utilization
- What effect will moving to $P4_{16}$ have on implementation?

Questions?