08

Attack Traceback

Kai Bu
kaibu@zju.edu.cn
http://list.zju.edu.cn/kaibu/netsec2022

Attack Detected!

Attack From Where?

Attack From Where?

does source IP address say it all?

Attack From Where?

remember IP spoofing?

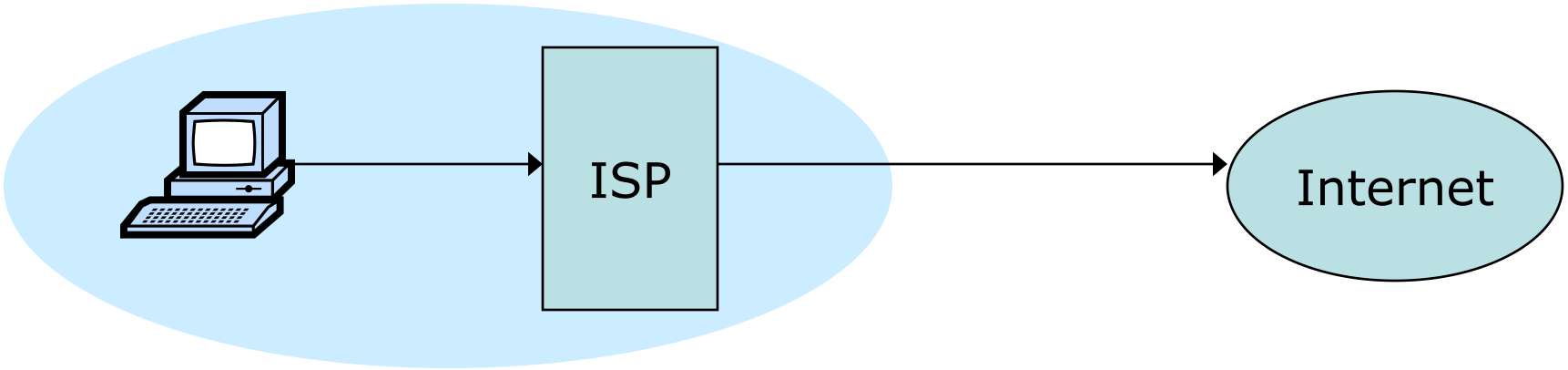
Attack From Where?

remember IP spoofing?

does ingress filtering filter them all?

Ingress Filtering

- How to find packet origin?



- Ingress filtering policy:

  ISP only forwards packets with legitimate source IP

Ingress Filtering

Implementation challenges:

- All ISPs need to do this — requires global coordination:

  If 10% of networks don't implement, there's no defense;

  No incentive for an ISP to implement — doesn't affect them;

Ingress Filtering

Implementation challenges:

- As of 2017 (from CAIDA):

  33% of autonomous systems allow spoofing;

  23% of announced IP address space allow spoofing;

Let Transit Routers Help!

“ To remember where you come from is part of where you’re going.

~ Anthony Burgess

IP Traceback

- Goal

  given set of attack packets

  determine path to source
- How

  change routers to record info in packets

IP Traceback

- Goal

  given set of attack packets

  determine path to source
- How

  change routers to record info in packets
- Assumptions

  trusted routers

  sufficient packets to track

  stable route from attacker to victim

IP Traceback

- Write path into packets

  router adds its own IP address to packet

  victim reads path from packet

- Deterministic Packet Marking

IP Traceback

- Write path into packets

  router adds its own IP address to packet

  victim reads path from packet
- Limitations

  requires space in packet

  path can be long

  no extra fields in current IP format (changes to packet format too much to expect)

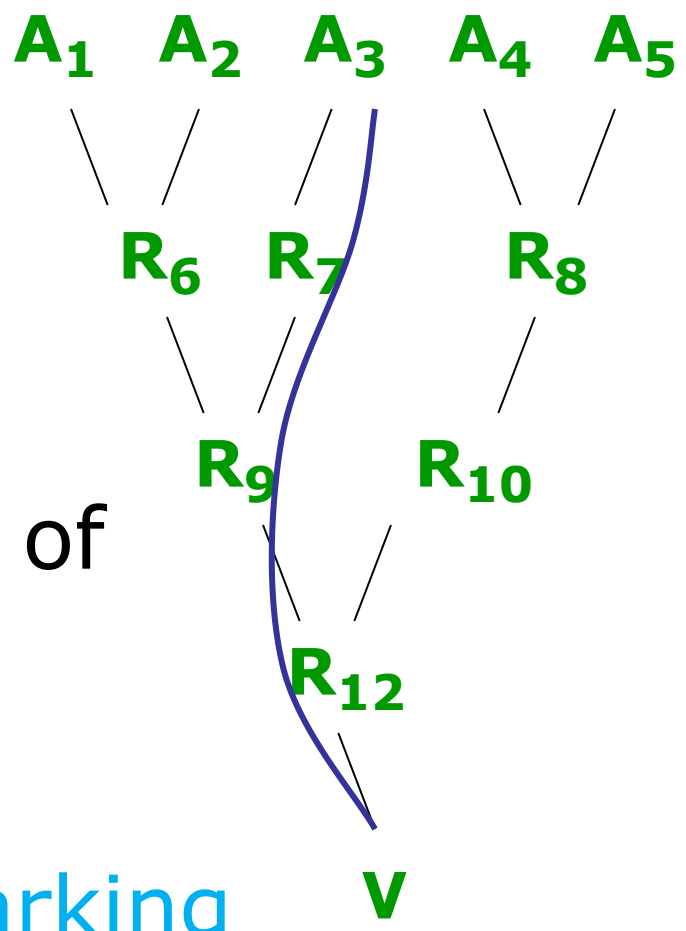
IP Traceback

- Sample and Merge

  store one link in each packet;

  router probabilistically stores own address;

  fixed space regardless of path length;



- Probabilistic Packet Marking

IP Traceback

- Edge Sampling: fields into packet

  edge: *start* and *end* IP addresses

  distance: no. of hops since edge stored

- Marking procedure of router R

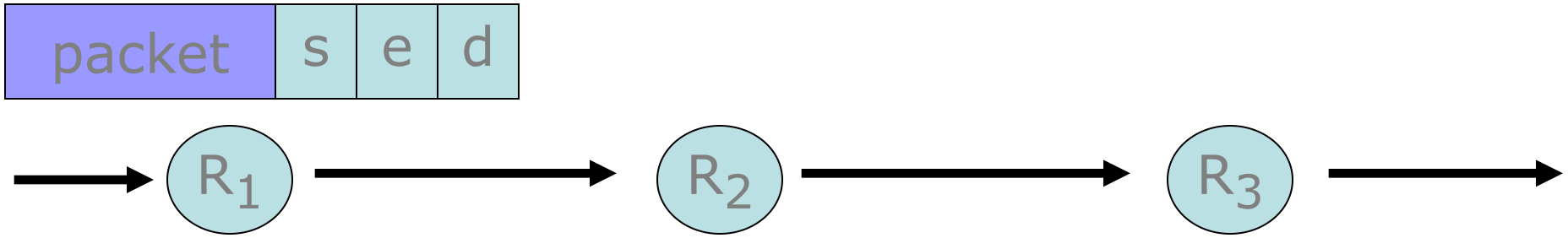
```
if coin turns up heads (with probability p) then
        write R into start address
        write 0 into distance field
else
        if distance == 0 write R into end field
        increment distance field
```

IP Traceback

- Packet received

  R_1 receives packet from source or another router;

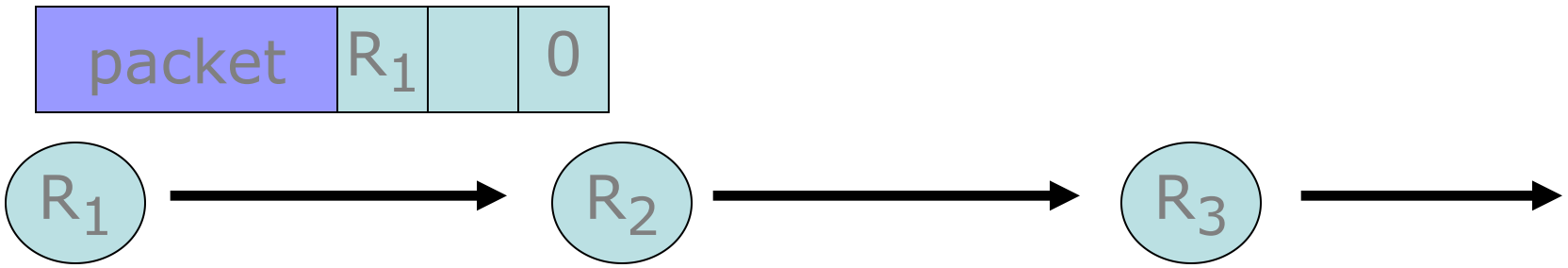
  packet contains space for start, end, distance;



IP Traceback

- Begin writing edge

  R_1 chooses to write start of edge; sets distance to 0;

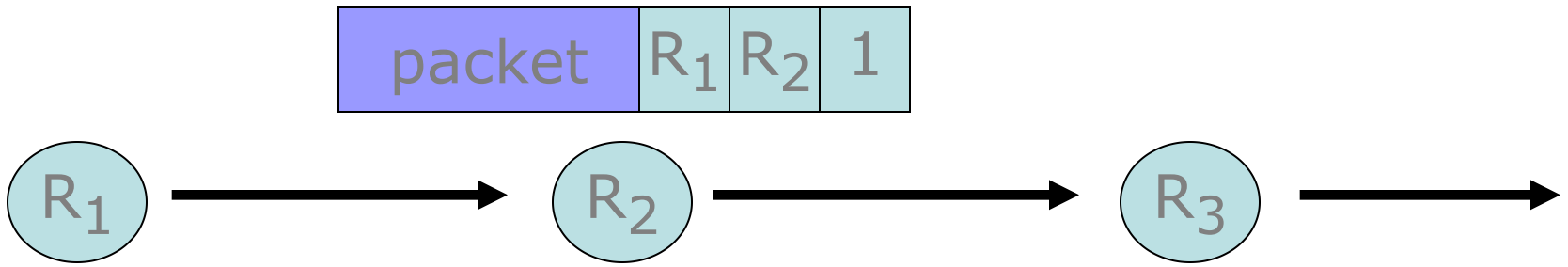


IP Traceback

- Finish writing edge

  R_2 chooses not to overwrite edge;

  distance is 0: write end of edge, increment distance to 1;

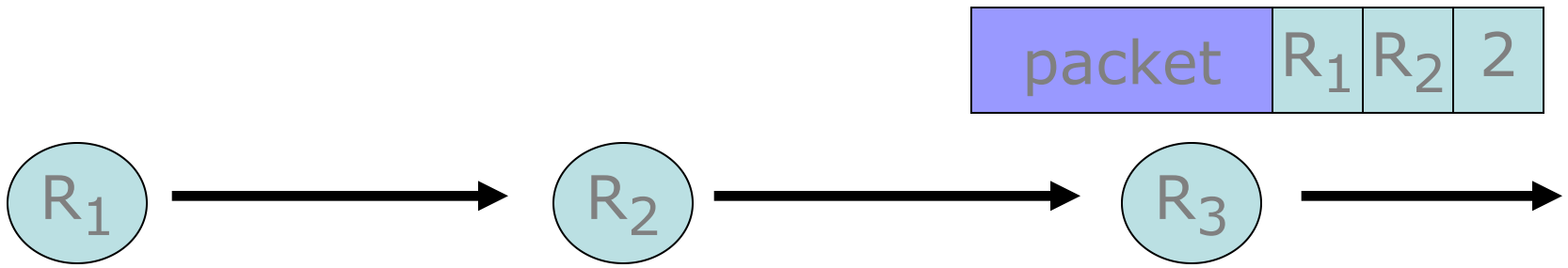


IP Traceback

- Increment distance

  R_3 chooses not to overwrite edge;

  distance>0: increment distance to 2;

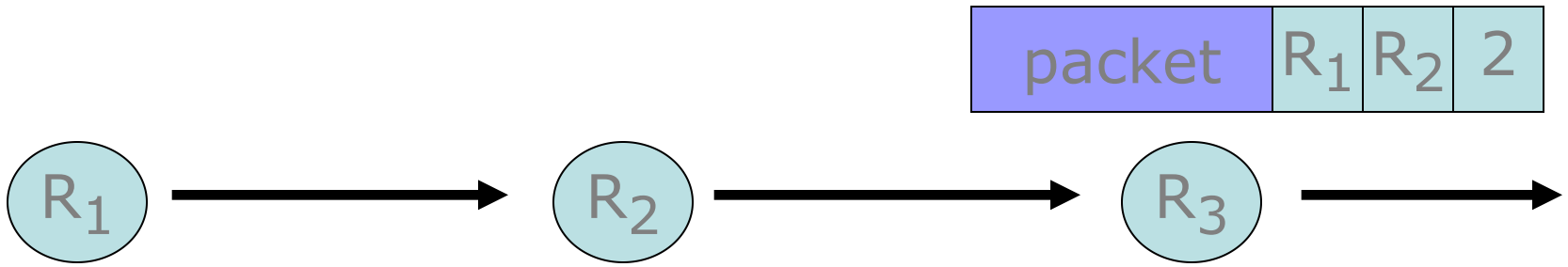


IP Traceback

- Increment distance

  R_3 chooses not to overwrite edge;

  distance>0: increment distance to 2;



- **What if traceback fields are tampered with…**

ICMP Traceback

- iTrace
- Each router samples one of packets it is forwarding and copies the contents and adjacent routers' info into an ICMP traceback message
- Router uses HMAC and X.509 digital certificate for authenticating traceback messages
- Router sends ICMP traceback messages to the destination

ICMP Traceback

- iTrace
- Each router samples one of packets it is forwarding and copies the contents and adjacent routers' info into an ICMP traceback message
- Router uses HMAC and X.509 digital certificate for authenticating traceback messages
- Router sends ICMP traceback messages to the destination

ICMP Traceback

- iTrace
- Require all the routers transmitting attack traffic be enabled with iTrace to construct an entire attack path
- yet ICMP packets are usually filtered… because of ICMP Ping Flood Attack…

ICMP Traceback

- iTrace
- Require all the routers transmitting attack traffic be enabled with iTrace to construct an entire attack path
- yet ICMP packets are usually filtered… because of ICMP Ping Flood Attack…
- yet not all packets are sampled on every hop

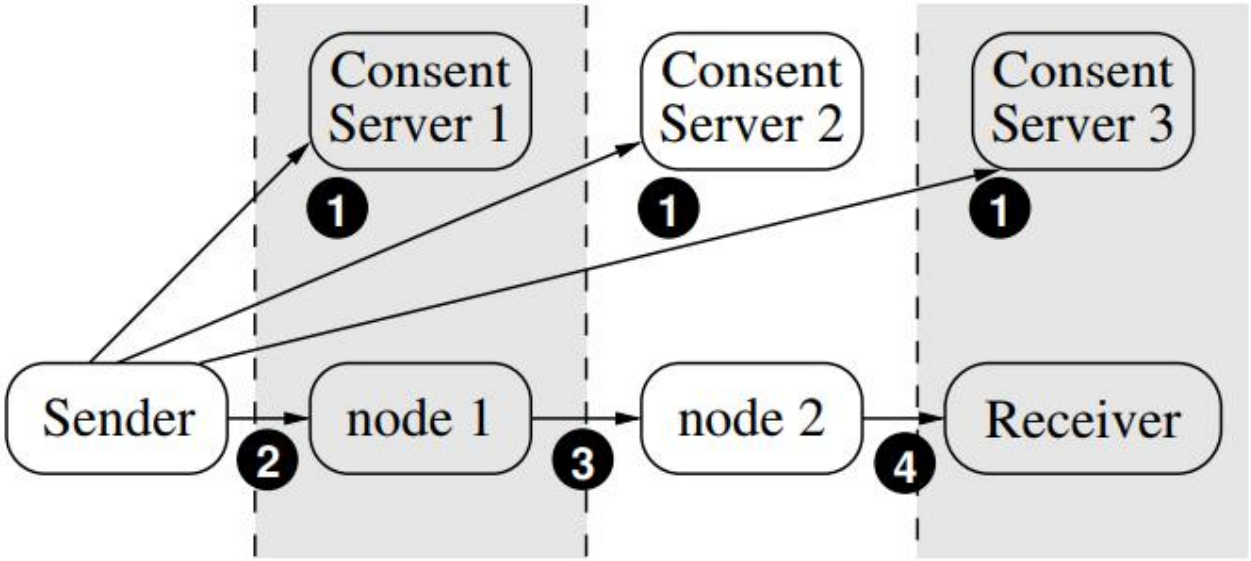
Path Validation

- PoC: Proof of Consent

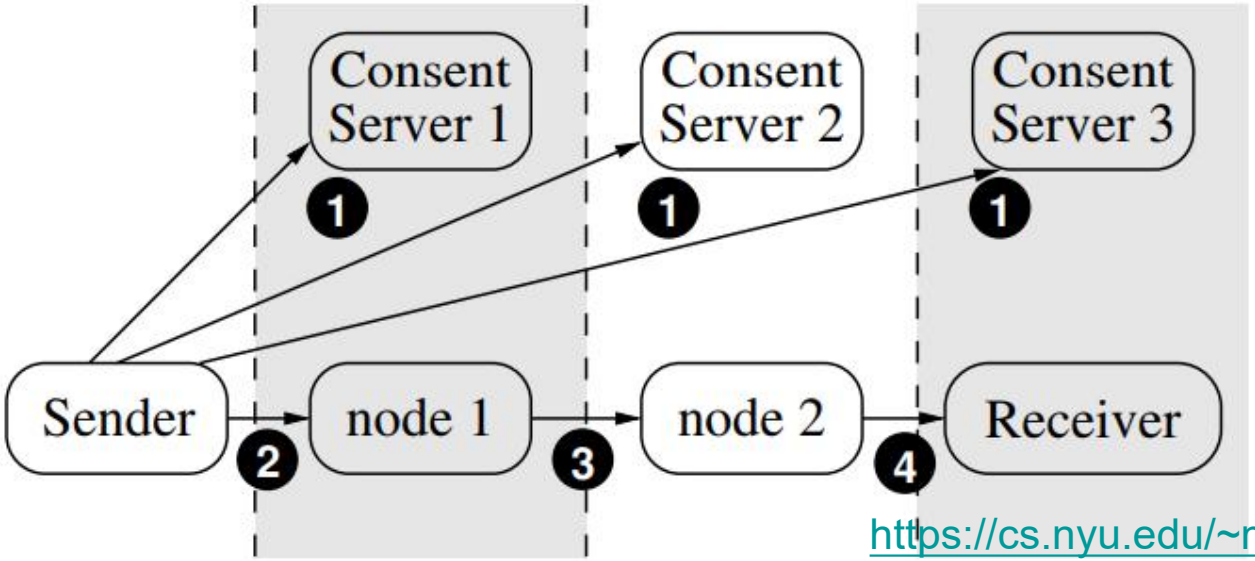
  certify the provider's consent to carry traffic along the path
- PoP: Proof of Provenance

  allow upstream nodes to prove to downstream nodes that they carried the packet

Path Validation



Path Validation



https://cs.nyu.edu/~mwalfish/papers/icing-conext11.pdf

P	N_0	N_1	N_2	N_3
V_1	$A_1 \oplus \text{PoP}_{0,1}$			
V_2	$A_2 \oplus \text{PoP}_{0,2}$			
V_3	$A_3 \oplus \text{PoP}_{0,3}$			
	Payload			

2

N_0	N_1	N_2	N_3
$A_1 \oplus \text{PoP}_{0,1}$			
$A_2 \oplus \text{PoP}_{0,2} \oplus \text{PoP}_{1,2}$			
$A_3 \oplus \text{PoP}_{0,3} \oplus \text{PoP}_{1,3} \oplus \text{PoP}_{2,3}$			
Payload			

4

how frequent is attack?

should every packet be always marked?

how frequent is attack?

should some packet be always sampled?

Let Transit Routers Help!

only when needed

Link Testing

- Traceback from the router closest to the victim
- Determine the upstream link that is used to carry out the attack traffic
- Recursively apply the previous technique until the attack source is reached

https://books.google.ca/books?id=cxdNEAAAQBAJ

Link Testing

- Traceback from the router closest to the victim
- Determine the upstream link that is used to carry out the attack traffic
- Recursively apply the previous technique until the attack source is reached
- Has to take effect while the attack is in progress

Link Testing

- Traceback from the router closest to the victim
- Determine the upstream link that is used to carry out the attack traffic
- Recursively apply the previous technique until the attack source is reached
- Input Debugging
- Controlled Flooding

Input Debugging

- Find attack signature, the common feature contained in all attack packets
- Communicate the attack signature to the upstream router, which then filters attack packets and determines the port of entry
- Recursively apply the previous technique on the upstream routers until reaching the attack source

Input Debugging

- Find attack signature, the common feature contained in all attack packets
- Communicate the attack signature to the upstream router, which then filters attack packets and determines the port of entry
- A considerable management overhead at the ISP level to communicate and coordinate the traceback

Controlled Flooding

- Need collaborative hosts
- Force the hosts to flood the links to upstream routers
- Since buffer on victim is shared by all incoming links, flooding the link carrying out attack leads to drops of attack packets
- Recursively apply the previous technique on the upstream routers until reaching the attack source

Controlled Flooding

- Need collaborative hosts
- Force the hosts to flood the links to upstream routers
- Since buffer on victim is shared by all incoming links, flooding the link carrying out attack leads to drops of attack packets
- Require an accurate topology map
  High overhead given multiple attacking sources (e.g., DDoS)

post-attack traceback?

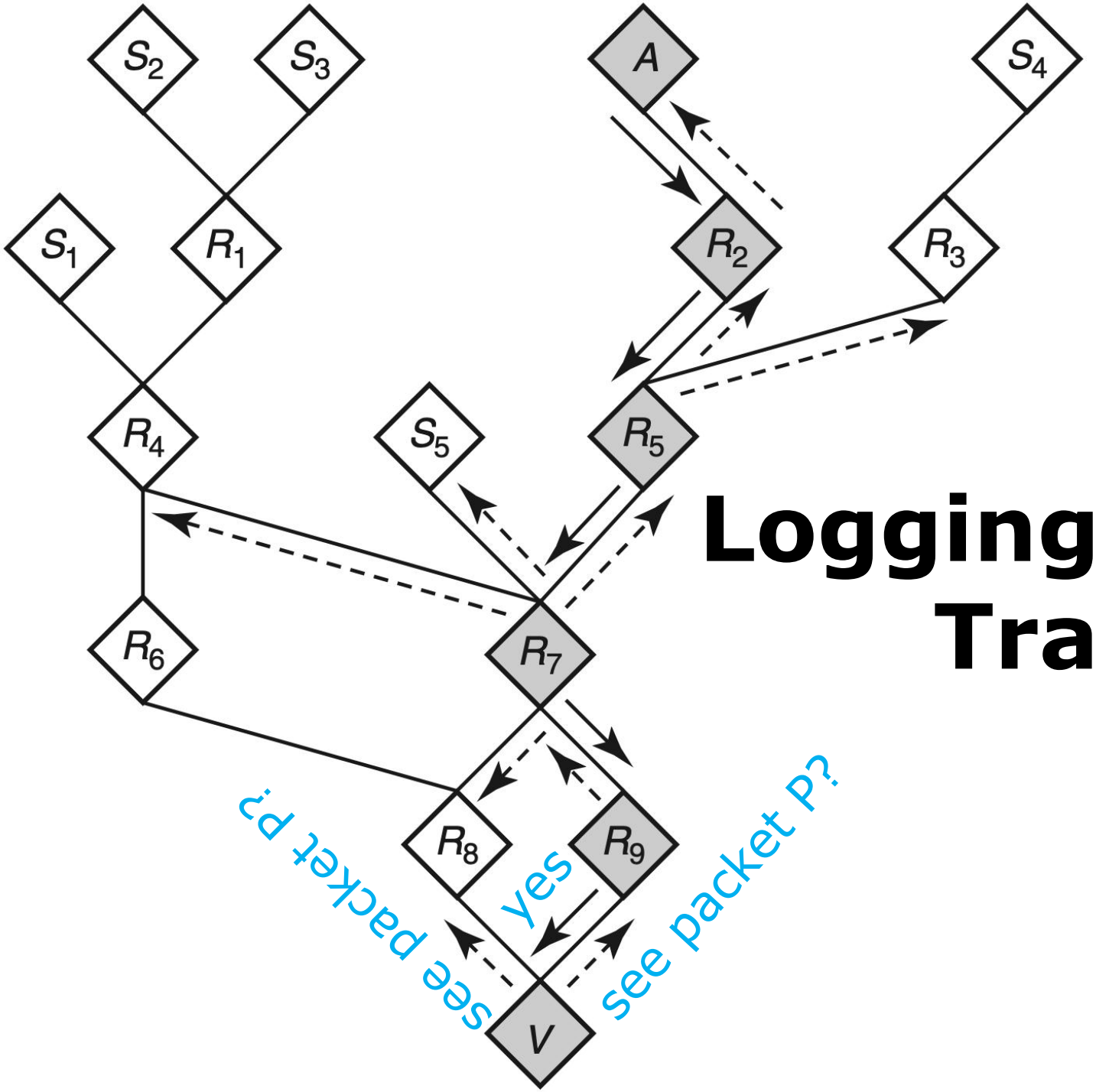
link testing requires ongoing attack…

Let Transit Routers Help!

log packets on routers to support query

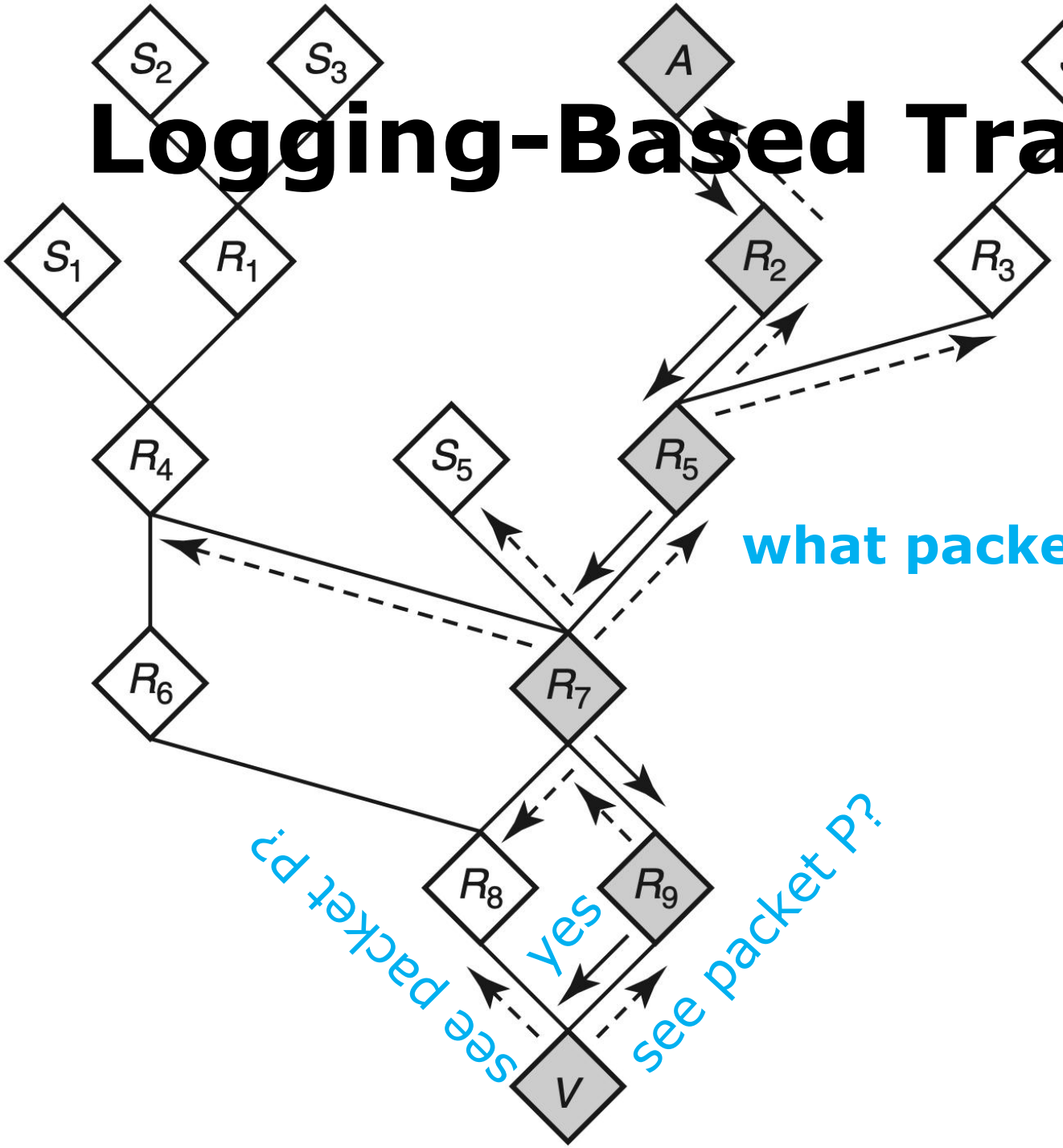
Logging-Based Traceback

- Routers store packet logs
- Victim queries the closest routers about packet appearance of attack packets
- The router containing attack packets recursively query upstream routers until reaching the attack source



Logging-Based Traceback

Logging-Based Traceback



what packet data to log?

Logging-Based Traceback

- Raw packets?
  high storage overhead on routers
- Hash of invariant content per packet?
  still high storage overhead given high traffic rate

Logging-Based Traceback

- Raw packets?
  high storage overhead on routers
- Hash of invariant content per packet?
  still high storage overhead given high traffic rate

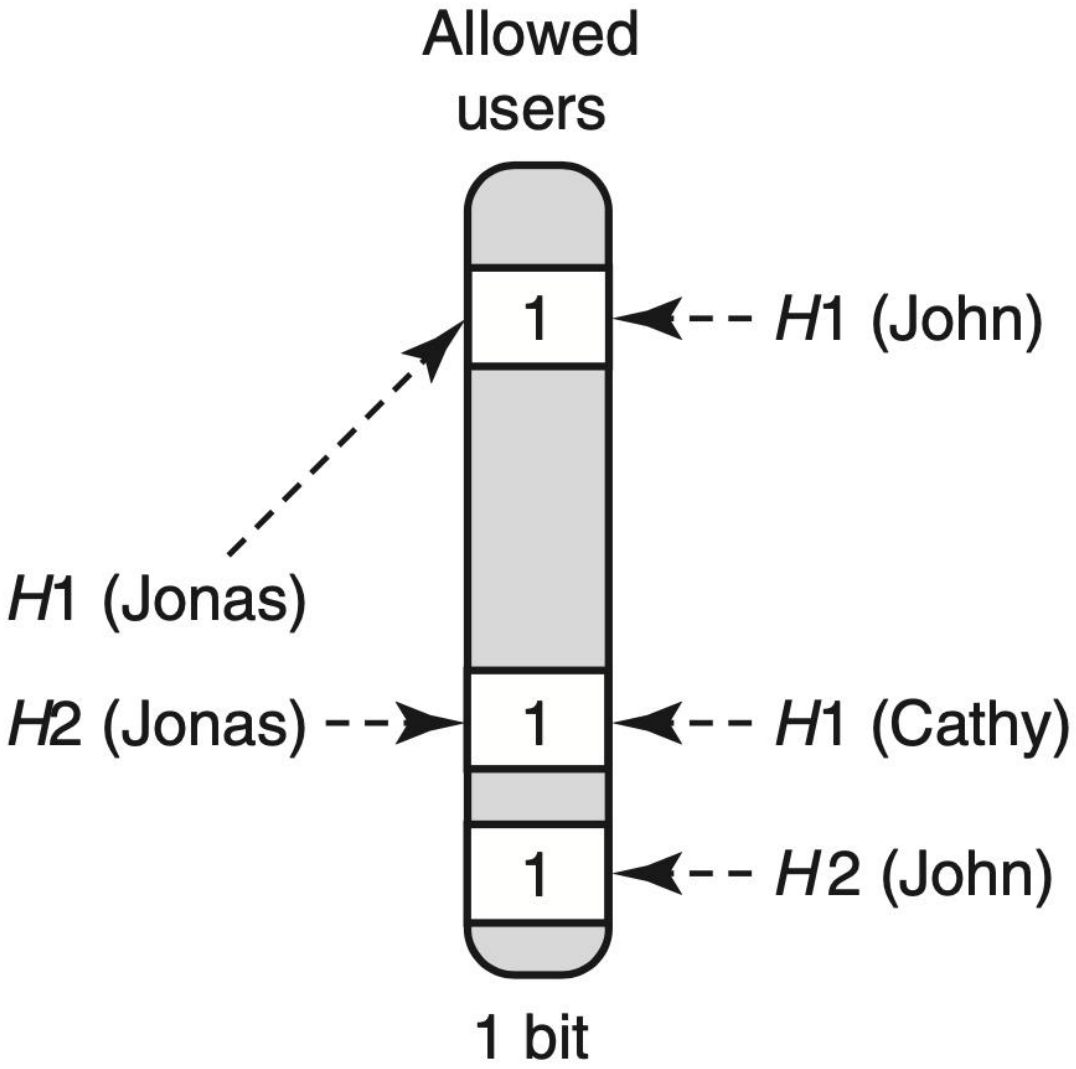
- How to efficient memership query?

Bloom Filter

- Effeficent set memership query using multiple hashes per set elements
- Use a bitmap, a bit of which is set of one element is hashed to this position

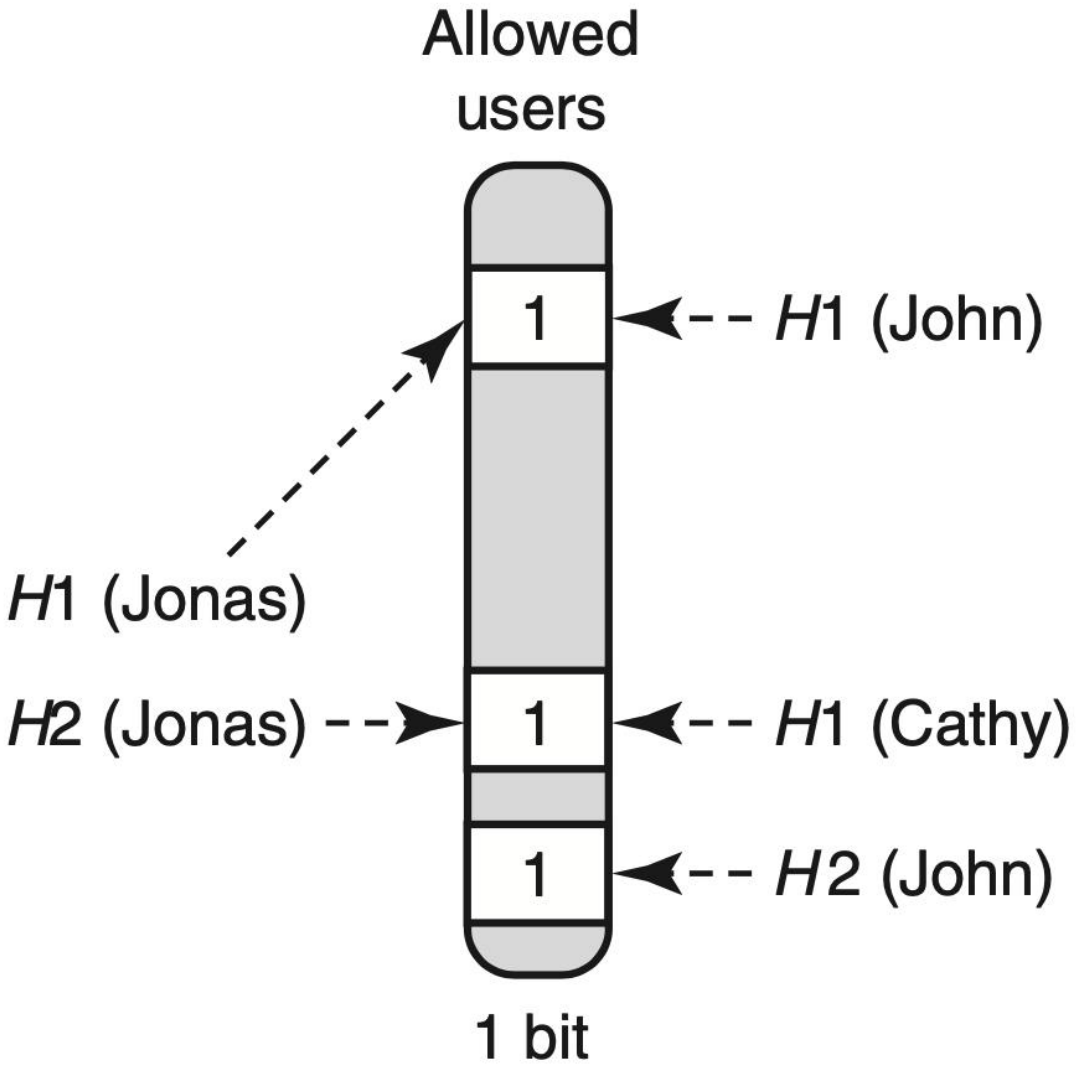
Bloom Filter

Is Cathy an allowed user?



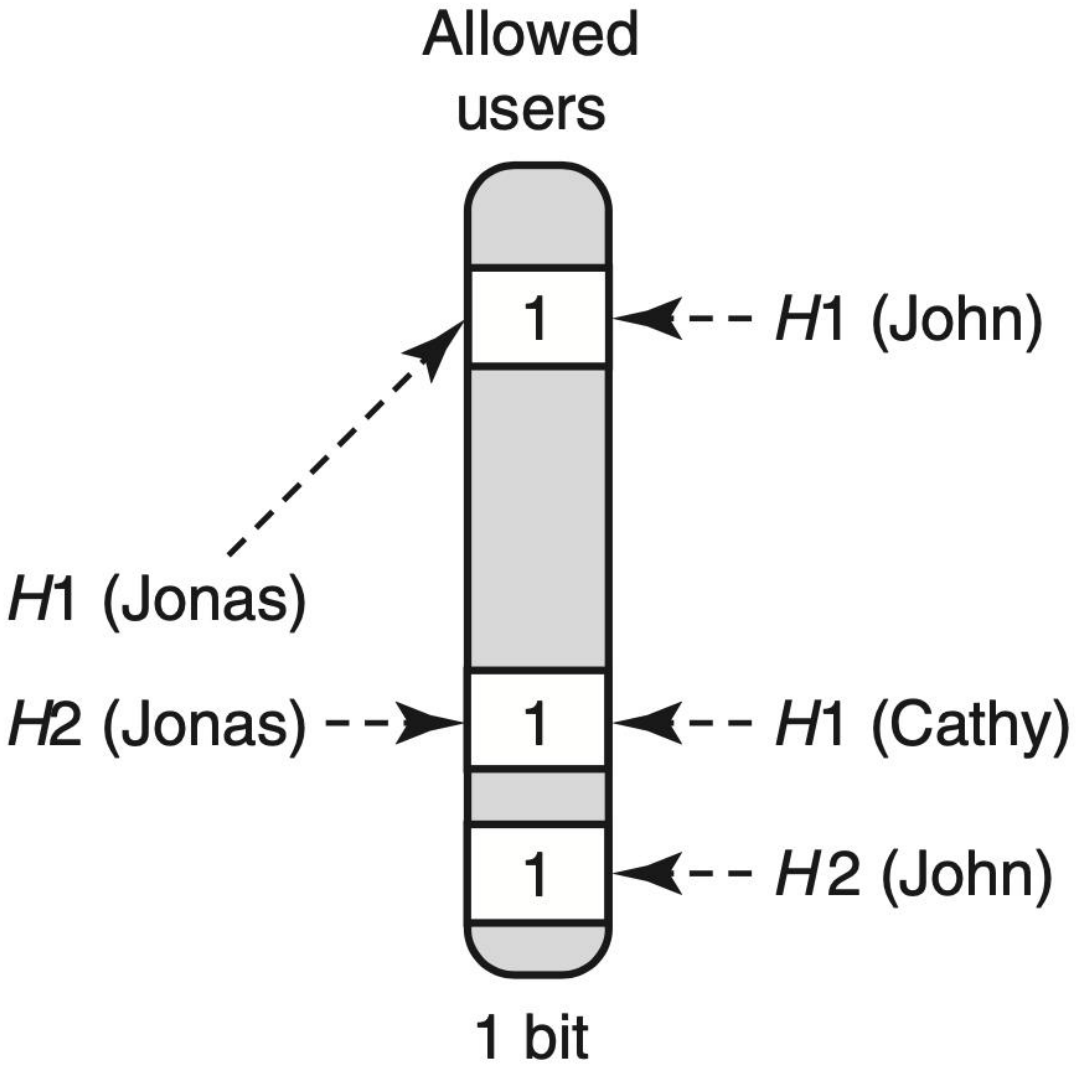
Bloom Filter

Is Jonas an allowed user?



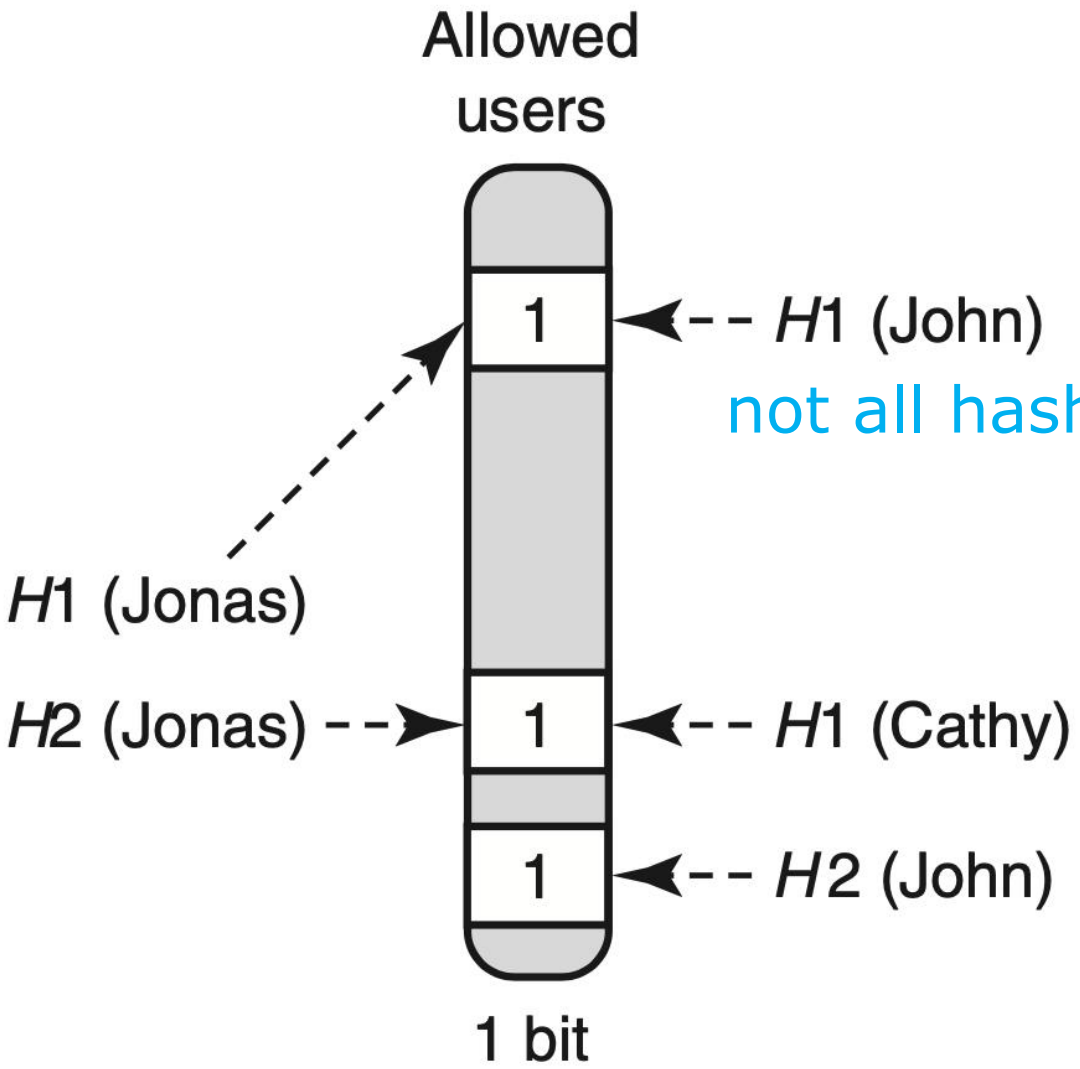
Bloom Filter

Is John an allowed user?



Bloom Filter

Is Cathy an allowed user?

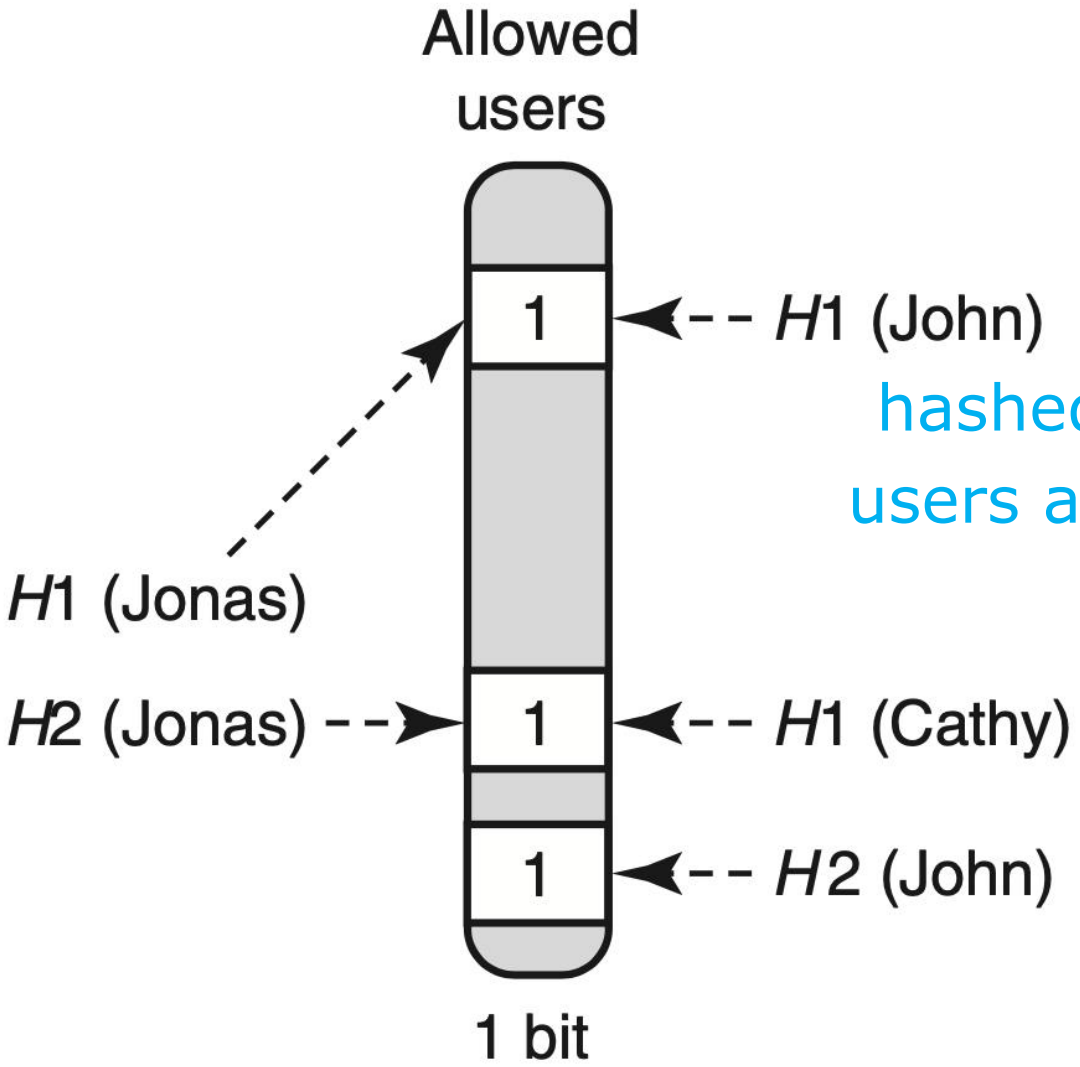


No False Negative:

For users not in, not all hashed positions are set.

Bloom Filter

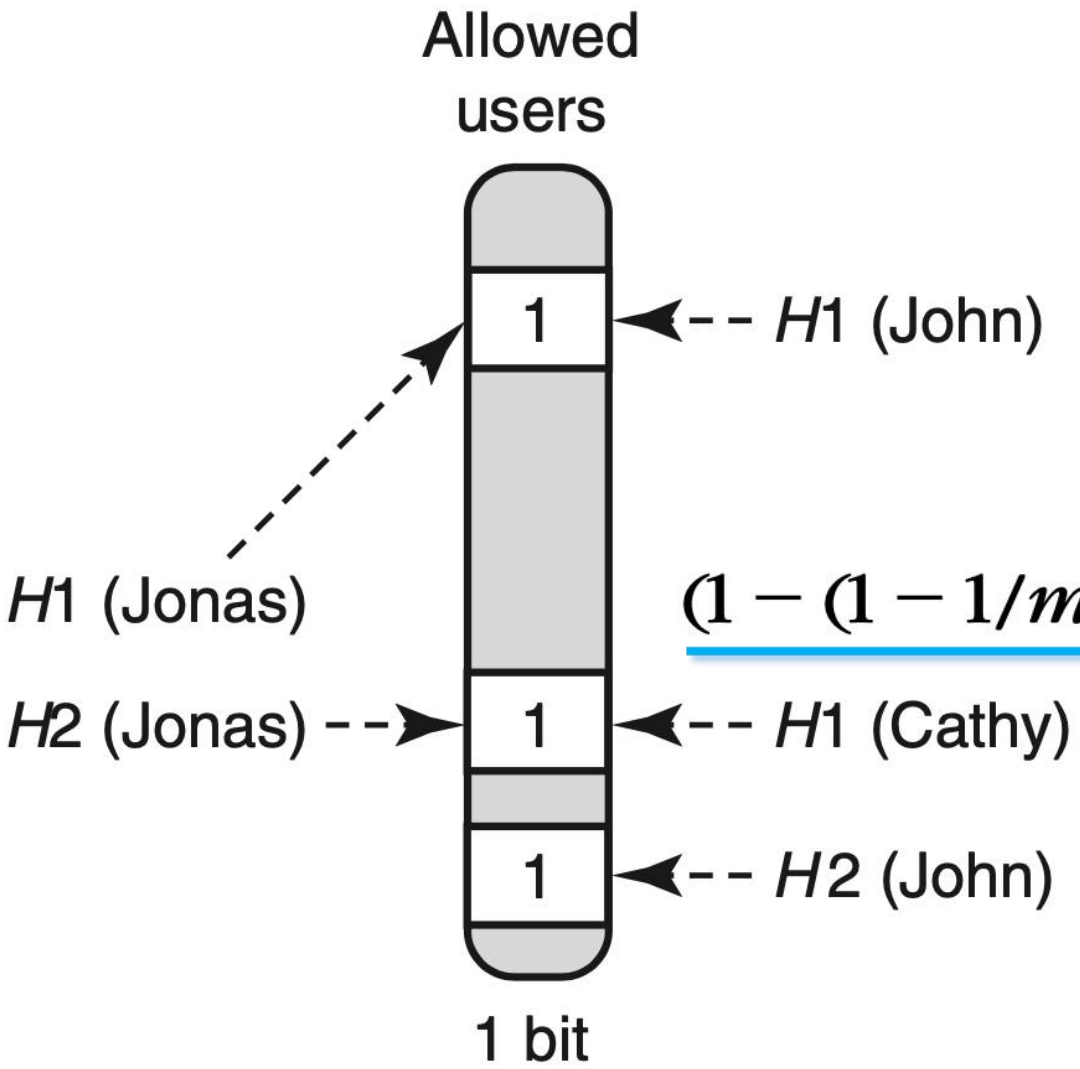
Is Jonas/John an allowed user?



Yet False Positive:
For users with all hashed positions being set, users are not necessarily in.

Bloom Filter

Is Jonas/John an allowed user?

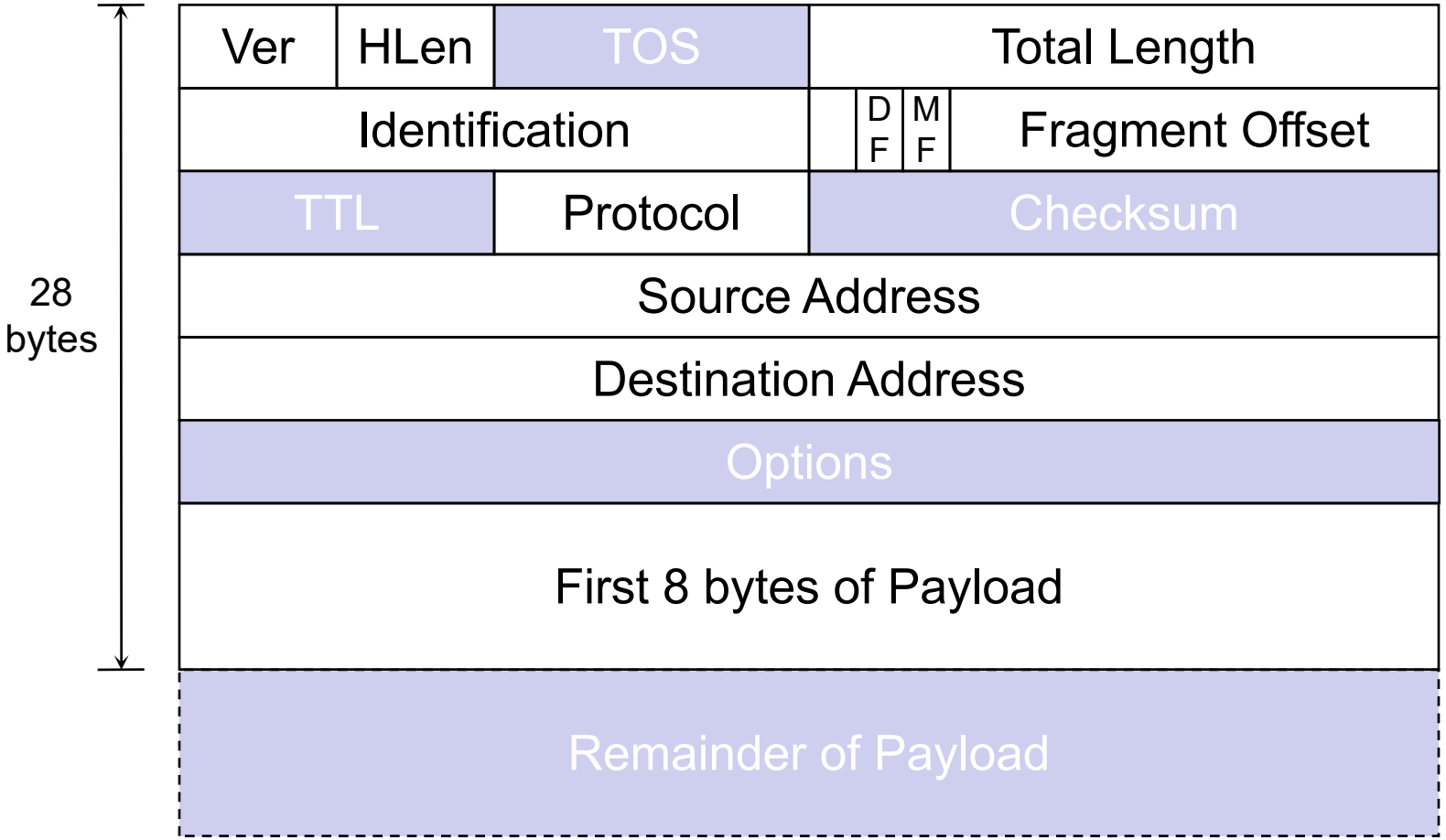


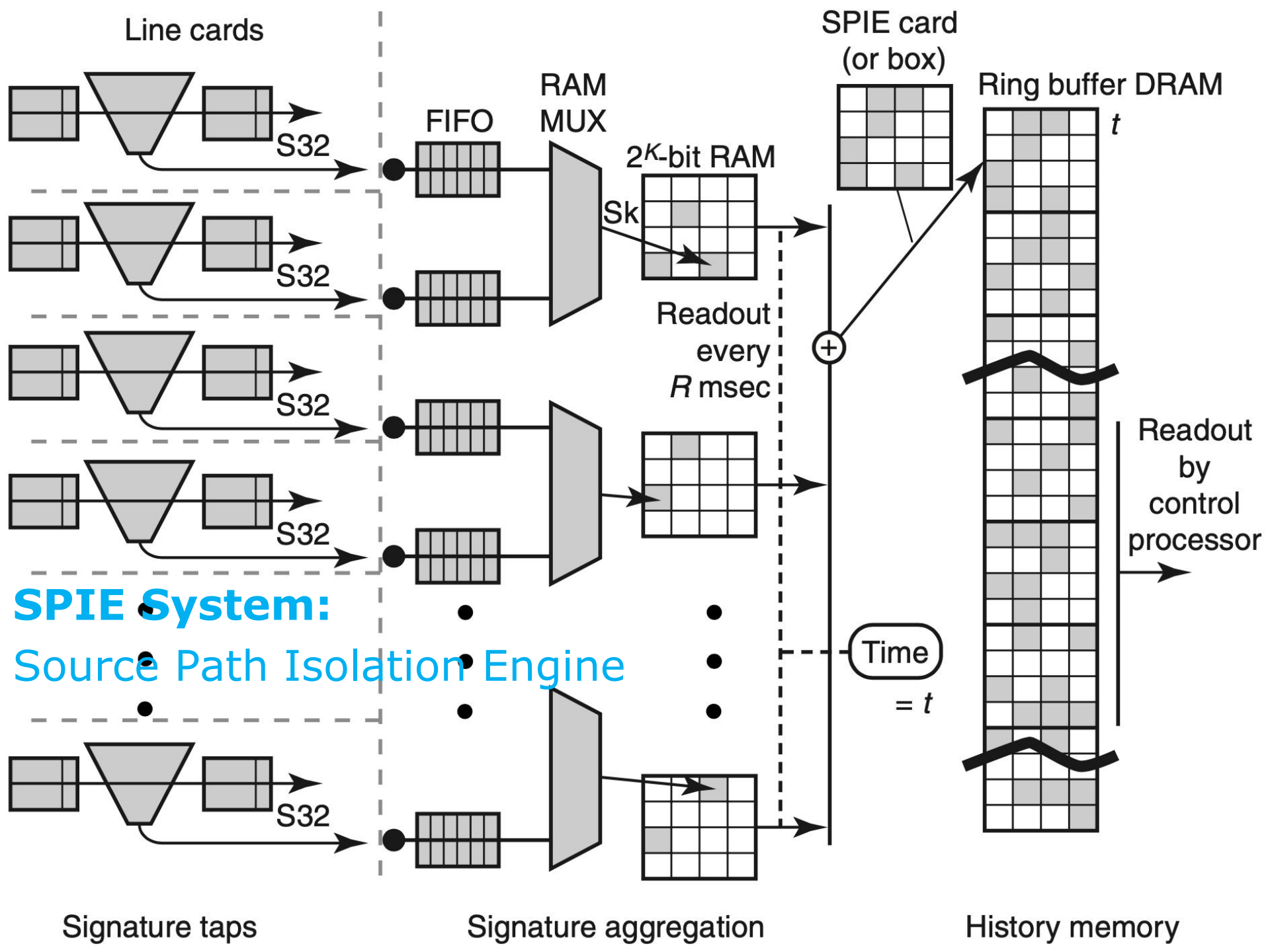
Yet False Positive:
m-size bitmap,
n members,
k hash functions:

$$(1-(1-1/m)^{kn})^k \approx (1-e^{kn/m})^k$$

Bloom Filter

- Hash invariant content of packets




Line cards
S32
S32
S32
S32
S32
FIFO
RAM
MUX
2K-bit RAM
Sk
Readout
every
R msec
SPIE card
(or box)
Ring buffer DRAM
t
Readout
by
control
processor
Time
= t
SPIE System:
Source Path Isolation Engine
Signature taps
Signature aggregation
History memory

Performance Comparison

	Management overhead	Network overhead	Router overhead	Distributed capability	Post-mortem capability	Preventative/ reactive
Ingress filtering	Moderate	Low	Moderate	N/A	N/A	Preventative
Link testing						
Input debugging	High	Low	High	Good	Poor	Reactive
Controlled flooding	Low	High	Low	Poor	Poor	Reactive
Logging	High	Low	High	Excellent	Excellent	Reactive
ICMP Traceback	Low	Low	Low	Good	Excellent	Reactive
Marking	Low	Low	Low	Good	Excellent	Reactive

https://dl.acm.org/doi/10.1145/347057.347560

were routing protocols modified…

Readings

- Practical Network Support for IP Traceback

  by Stefan Savage et al.
- Network Security Know It All

  by James Joshi

Thank You