


2004 IEEE Symposium on Security and Privacy



Large-Scale IP Traceback in High-Speed Internet

Jun (Jim) Xu

Networking & Telecommunications Group

College of Computing

Georgia Institute of Technology

(Joint work with Jun Li, Minho Sung, Li Li)

Introduction



- Internet DDoS attack is an ongoing threat
 - on websites: Yahoo, CNN, Amazon, eBay, etc (Feb. 2000)
 - on Internet infrastructure: 13 root DNS servers (Oct, 2002)
- It is hard to identify attackers due to IP spoofing
- IP Traceback: trace the attack sources despite spoofing
- Two main types of proposed traceback techniques
 - Probabilistic Packet Marking schemes: routers put stamps into packets, and victim reconstructs attack paths from these stamps [Savage et. Al. 00] [Goodrich 02]
 - Hash-based traceback: routers store bloom filter digests of packets, and victim query these digests recursively to find the attack path [Snoeren et. al. 01]

Scalability Problems of Two Approaches



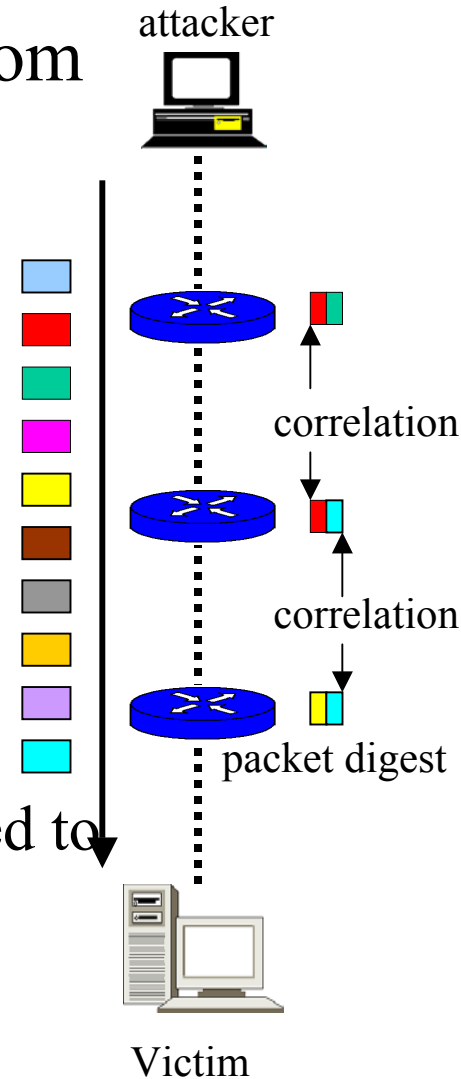
- Traceback needs to be scalable
 - When there are a large number of attackers, and
 - When the link speeds are high
- PPM is good for high-link speed, but cannot scale to large number of attackers [Goodrich 01]
- Hash-based scheme can scale to large number of attackers, but hard to scale to very high-link speed
- Our objective: design a traceback scheme that is scalable in both aspects above.

Assumptions “inherited” from the literature

- Attackers send lots of packets
- Traceback uses a small number of bits in IP header
- Attackers are aware of the traceback effort and can sabotage the effort: solutions has to work in this adversarial environment
- Count the number of routers in terms of evaluating false positive and negative; Partial paths reconstructed considered useful
- We can magically identify the DDoS packets

Design Overview

- Our idea: same as hash-based, but store bloom filter digests of sampled packets only
 - Use small sampling rate p (such as 3.3%)
 - Small storage and computational cost
 - Scale to 10 Gbps or 40 Gbps link speeds
 - Operate within the DRAM speed
- the challenge of the sampling
 - Need many more packets for traceback
 - Independent random sampling will not work: need to improve the “*correlation factor*”

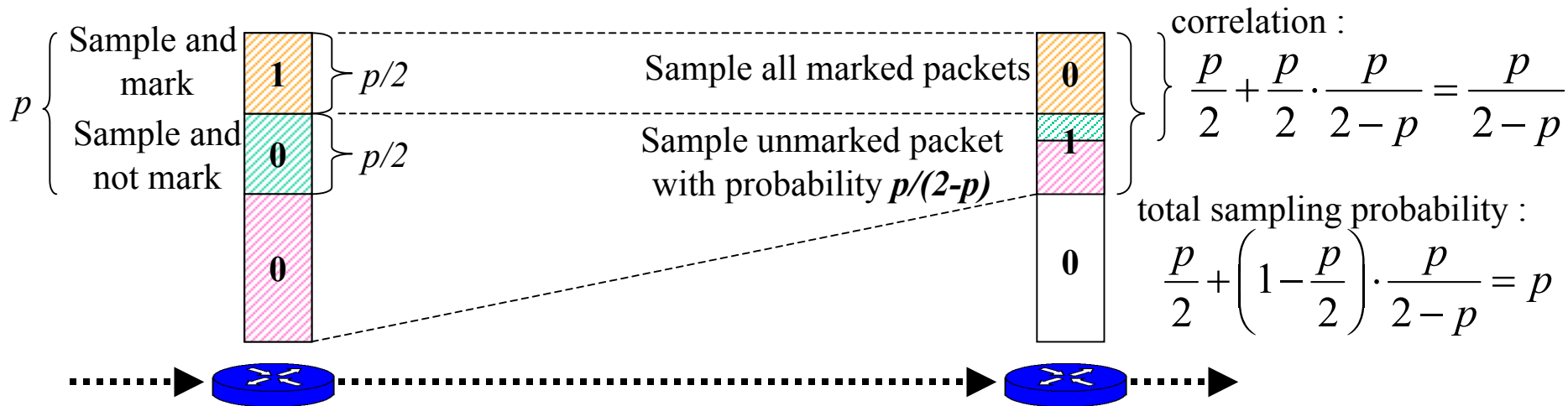


Overview of our hash-based traceback scheme

- Each router stores the bloom filter digests of sampled packets
- Neighboring routers compare with each other the digests of the packets they store for the traceback to proceed
 - Say P is an attack packet, then if you see P and I also see P , then P comes from me to you ...
- When correlation is small, the probability that both see something in common is small

One-bit Random Marking and Sampling(ORMS)

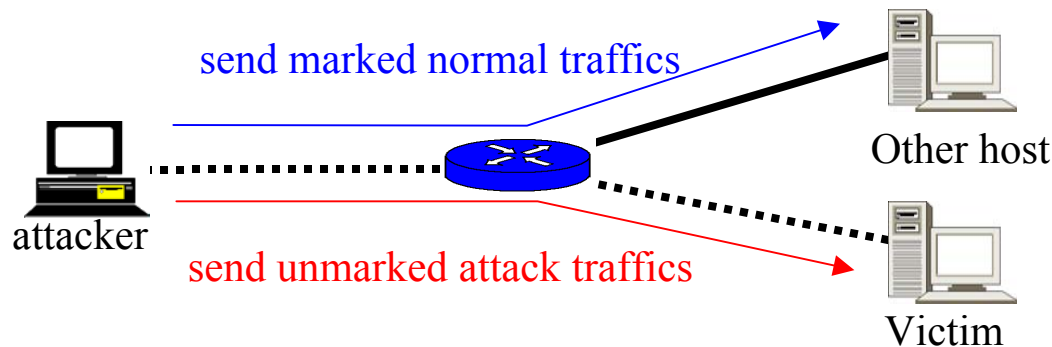
- ORMS make correlation factor be larger than 50%
- ORMS uses only one-bit for coordinating the sampling among the neighboring routers



correlation factor (sampled by both) : $\left(\frac{p}{2-p}\right) / p = \frac{1}{2-p}$
 (> 50% because $0 < p < 1$)

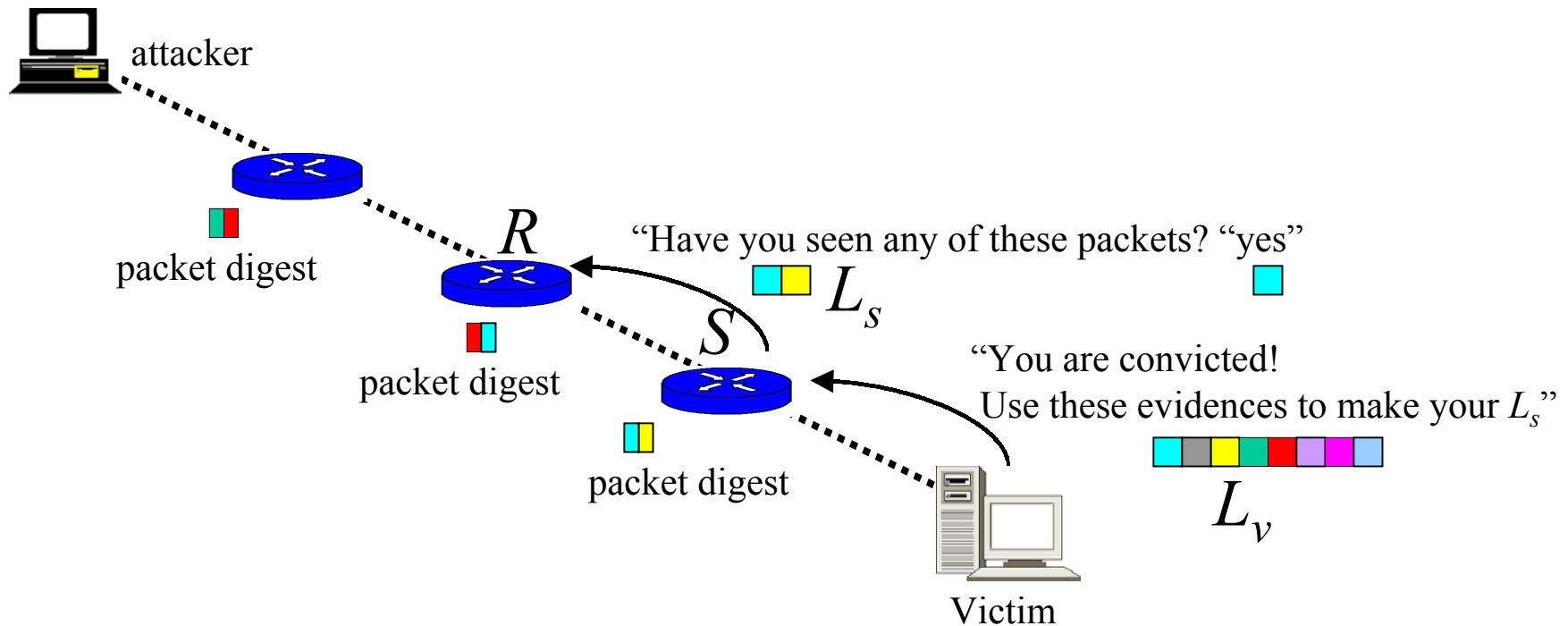
Why simpler schemes will NOT work?

- Why not trajectory sampling?
 - the attacker can use hash values that escape sampling
- Why save $p/2$ of marked packets, and $p/2$ of unmarked packets?
Why not simply save all packets that are marked with 1?



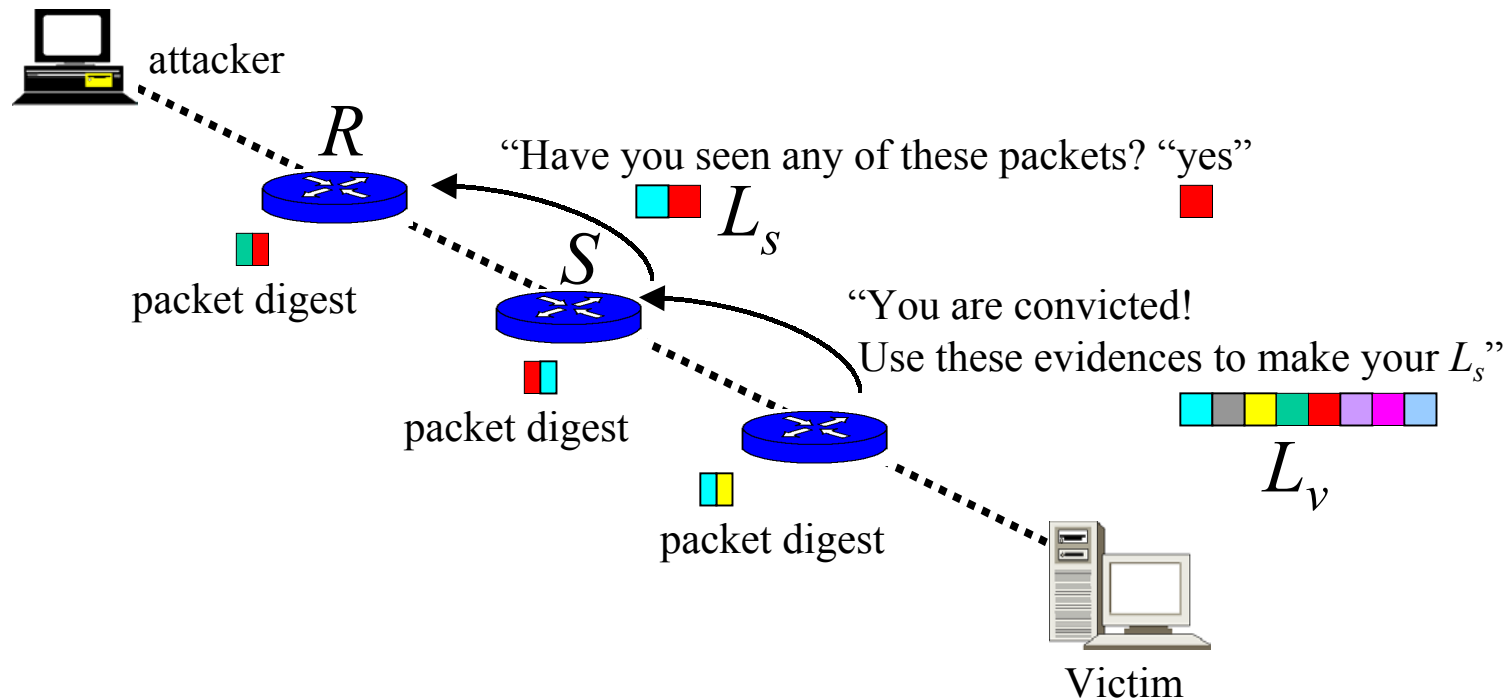
Traceback Processing

1. Collect a set of attack packets L_v
2. Check router S , a neighbor of the victim, with L_v
3. Check each router R (neighbor of S) with L_s



Traceback Processing


4. Pass L_v to R to be used to make new L_s
5. Repeat these processes



A fundamental optimization question

- Recall that in the original traceback scheme, the router records a bloom filter of 3 bits for each and every packets
- There are many different ways of spending this 3 bits per packet budget, representing different tradeoff points between size of digest and sampling frequency
 - e.g., use a 15-bit bloom filter but only record 20% of digests ($15 * 20\% = 3$)
 - e.g., use a 12-bit bloom filter but only record 25% of digests ($12 * 25\% = 3$)
 - Which one is better or where is the optimal tradeoff point?
- Answer lies in the information theory

Intuitions from the information theory



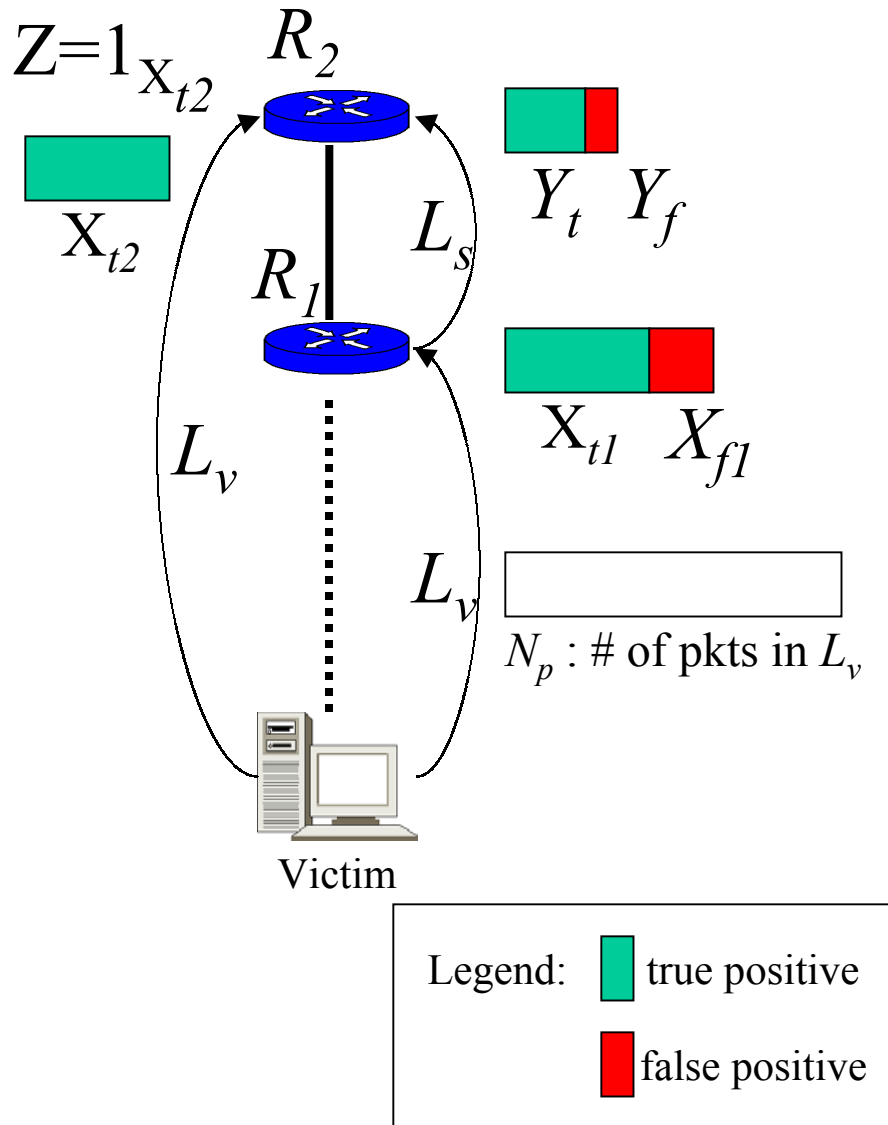
- View the traceback system as a communication channel
 - Increasing the size of digest reduces the false positive ratio of the bloom filter, and therefore improving the signal noise ratio (S/N)
 - Decreasing sampling rate reduces the bandwidth (W) of the channel
 - We want to maximize $C = W \log_2 (1+S/N)$
- C is the mutual information – maximize the mutual information between what is “observed” and what needs to be predicted – or minimize the conditional entropy
- Bonus from information theory: we derive a lower bound on the number of packets needed to achieve a certain level of traceback accuracy through **Fano’s inequality**

Information Theory Background



- Concepts
 - Entropy $H(X)$: measures the uncertainty of X
 - Conditional entropy $H(X|Y)$: measures how much uncertainty remains for X given the observation of Y
- Fano's inequality: $H(p) \geq H(X|Y)$, where X is a binary random variable
 - p is the probability that the estimation of X does not match X , i.e., probability of error

Applications of Information Theory



- What we can observe :

$$X_{t1} + X_{f1}, Y_t + Y_f$$
- We want to estimate Z

$$Z = \begin{cases} 1 & \text{if } X_{t2} > 0 \\ 0 & \text{otherwise} \end{cases}$$
- Question : How to maximize our accuracy in estimating Z ?
- Answer :

$$\text{minimize } H(Z | X_{t1} + X_{f1}, Y_t + Y_f)$$

The optimization problem

$$k^* = \underset{k}{\operatorname{argmin}} H(Z | X_{tl} + X_{fl}, Y_t + Y_f)$$

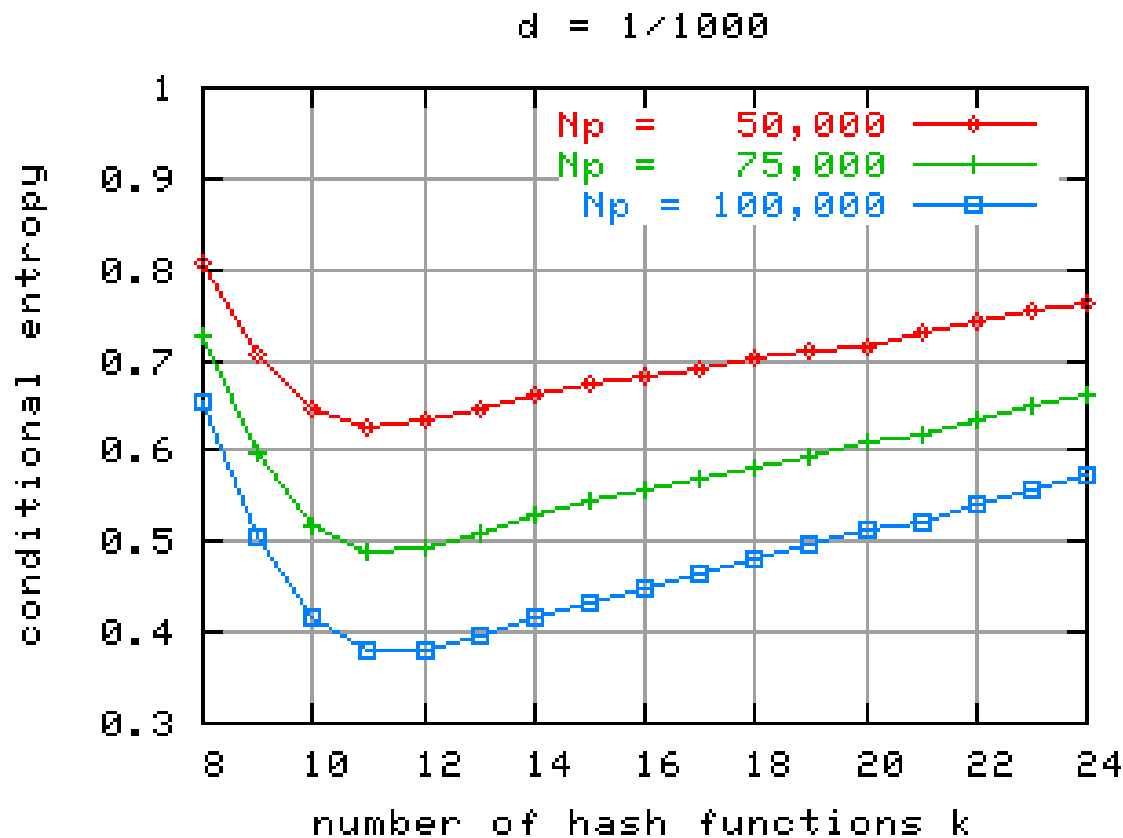
subject to the resource constraint ($s = k \times p$)

s : average number of bits “devoted” for each packet

p : sampling probability

k : size the bloom filter digest

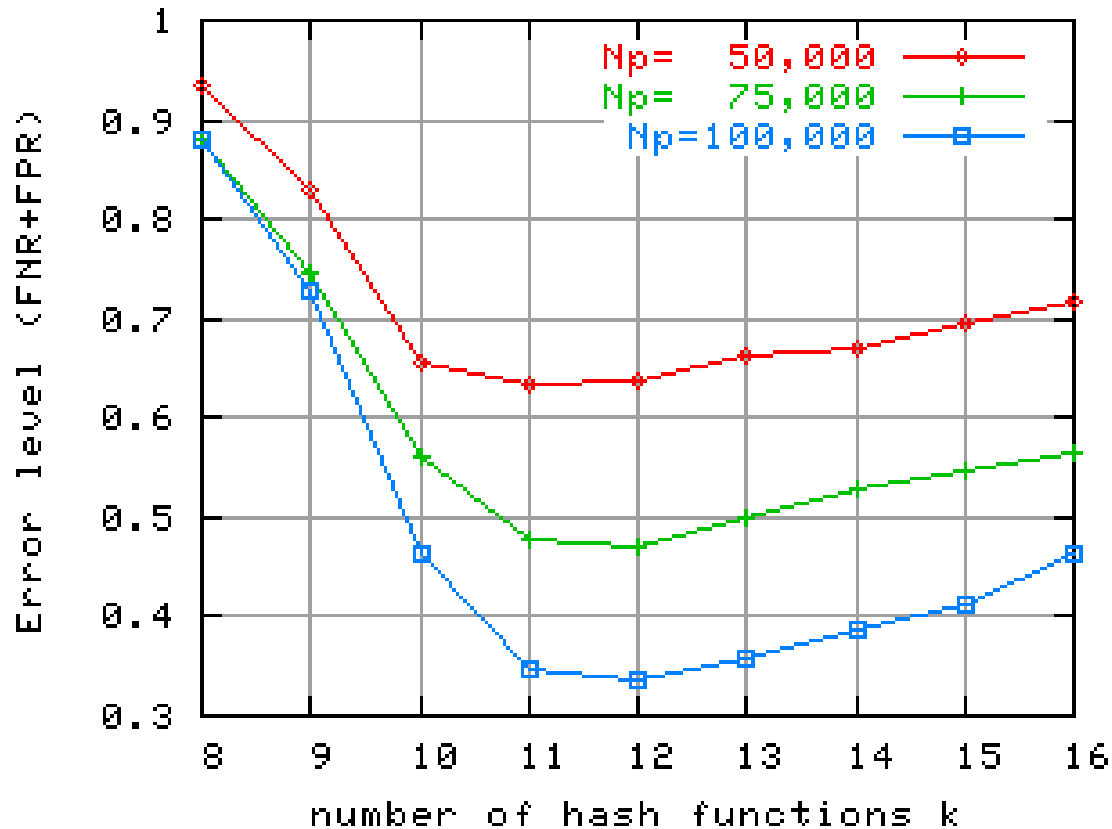
Applications of Information Theory



Resource constraint: $s = k \times p = 0.4$

Verification of Theoretical Analysis

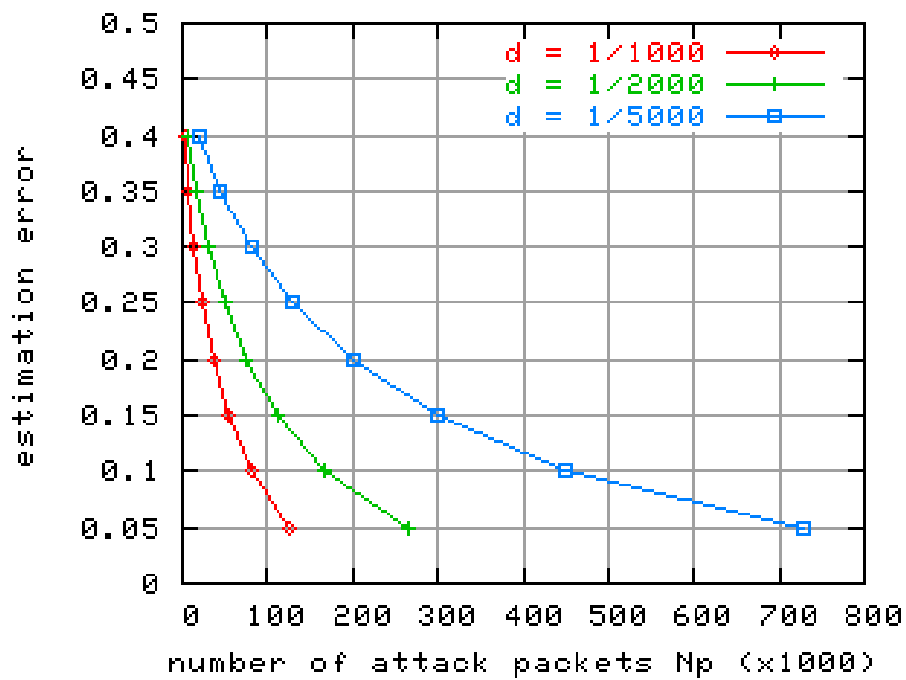
- Parameter tuning



Parameters: 1000 attackers, $s = k \times p = 0.4$

Lower bound through Fano's inequality

- $H(p_e) \geq H(Z | X_{tl} + X_{fl}, Y_t + Y_f)$



Parameters: $s=0.4$, $k=12$, $p=3.3\%$ ($12 \times 3.3\% = 0.4$)

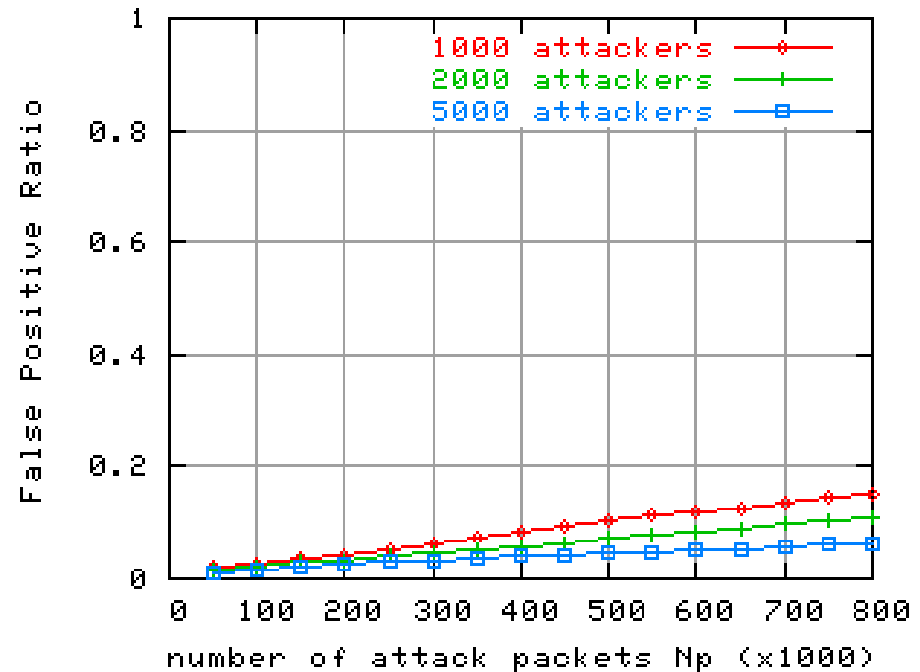
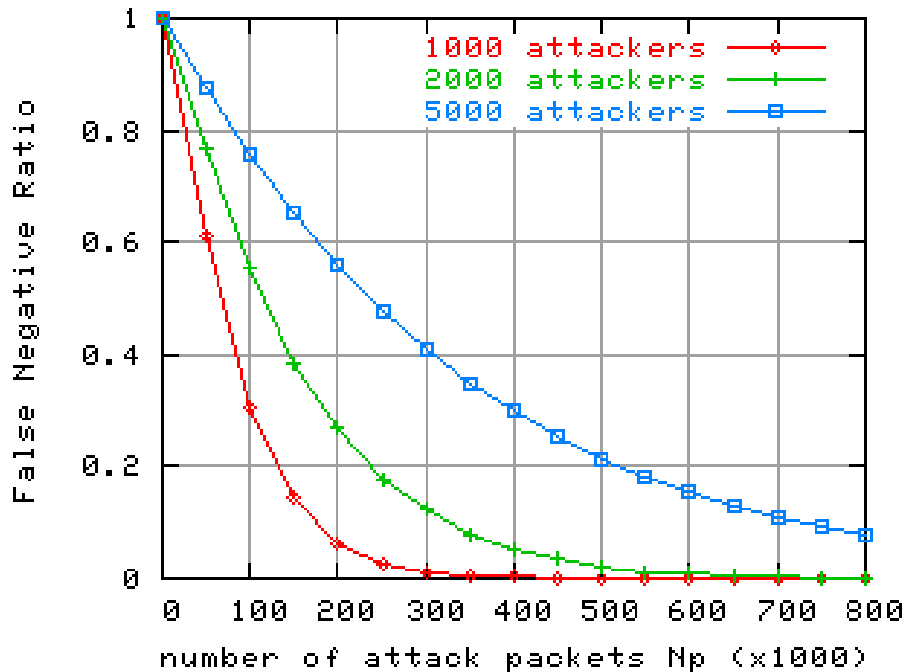
Simulation set-up



- Topologies: Skitter data I, Skitter data II, Bell-lab's data (routes from a host to 192,900, 158,181, 86,813 destinations)
- Host setting:
 - Victim: the original in the three topologies
 - Attackers: randomly selected among the destination hosts
- Performance Metrics
 - False Negative Ratio (FNR): the ratio of the number of missed routers to the number of infected routers
 - False Positive Ratio (FPR): the ratio of the number of incorrectly convicted routers to the number of convicted routers

Simulation results

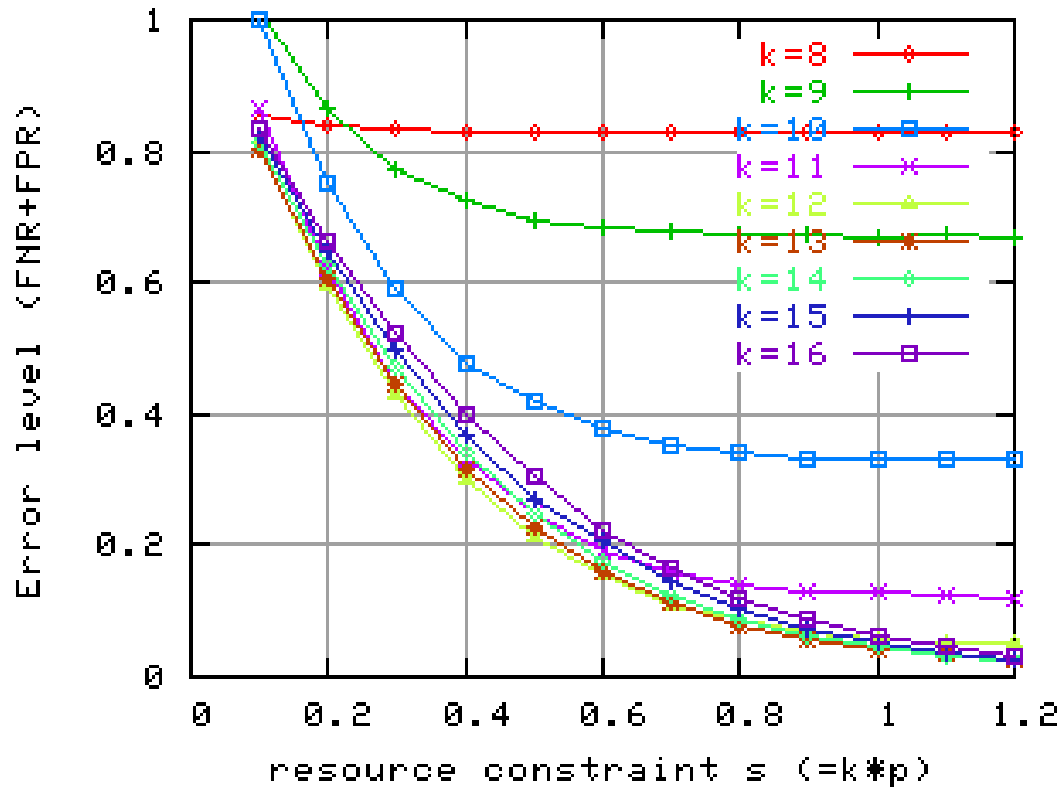
- False Negative & False Positive on Skitter I topology



Parameters: $s=0.4$, $k=12$, $p=3.3\%$ ($12 \times 3.3\% = 0.4$)

Verification of Theoretical Analysis

- Error levels by different k values



Parameters: 2000 attackers, $N_p=200,000$

Future work and open issues



1. Is correlation factor $1/(2-p)$ optimal for coordination using one bit?
2. What if we use more than one bit for coordinating sampling?
3. How to optimally combine PPM and hash-based scheme – a Network Information Theory question.
4. How to know with 100% certainty that some packets are attack packets? How about we only know with a certainty of p ?

Conclusions



- Design a sampled hash-based IP traceback scheme that can scale to a large number of attackers and high link speeds
- Addressed two challenges in this design:
 - Tamper-resistant coordinated sampling to increase the “correlation factor” to beyond 50% between two neighboring routers
 - An information theory approach to answer the fundamental parameter tuning question, and to answer some lower bound questions
- Lead to many new questions and challenges