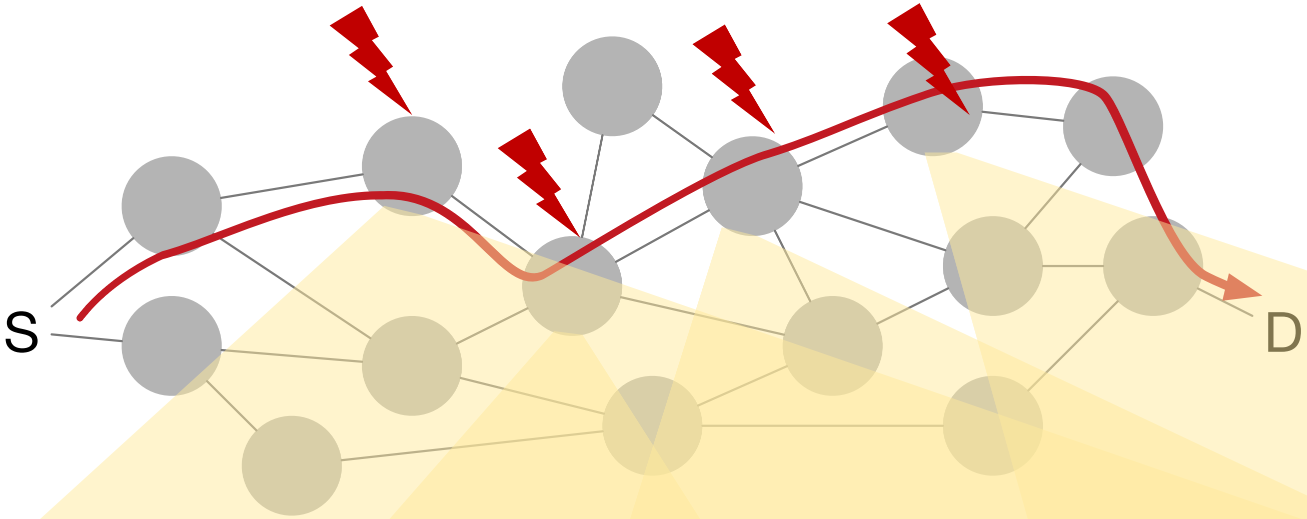
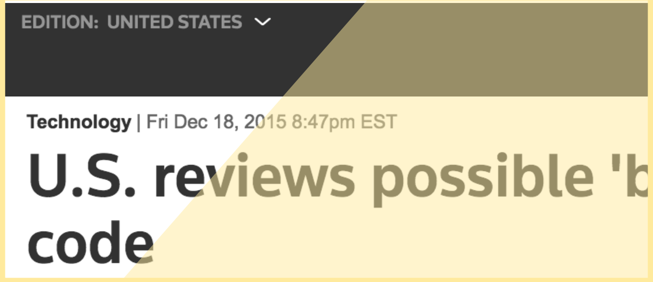
High-Speed Inter-Domain Fault Localization

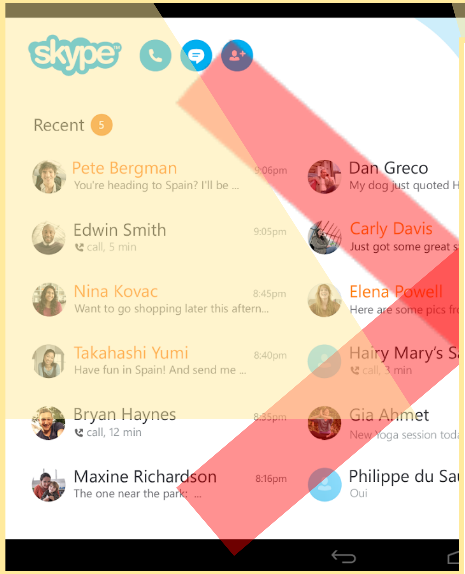
Cristina Basescu, Yue-Hsun Lin, Haoming Zhang, Adrian Perrig




EDITION: UNITED STATES

Technology | Fri Dec 18, 2015 8:47pm EST

U.S. reviews possible 'b
code



Recent 5

Pete Bergman 9:06pm
You're heading to Spain? I'll be ...

Edwin Smith 9:05pm
call, 5 min

Nina Kovac 8:45pm
Want to go shopping later this aftern...

Takahashi Yumi 8:40pm
Have fun in Spain! And send me ...

Bryan Haynes 8:35pm
call, 12 min

Maxine Richardson 8:16pm
The one near the park: ...

Dan Greco
My dog just quoted H

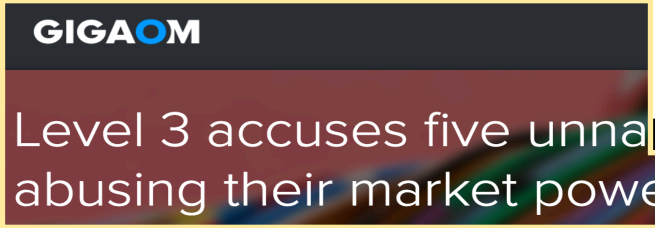
Carly Davis
Just got some great s

Elena Powell
Here are some pics fr

Hairy Mary's S
call, 3 min

Gia Ahmet
New yoga session tod

Philippe du Sa
Oui


GIGAOM

Level 3 accuses five unna
abusing their market power in peering

Service Provider Experienced Threats

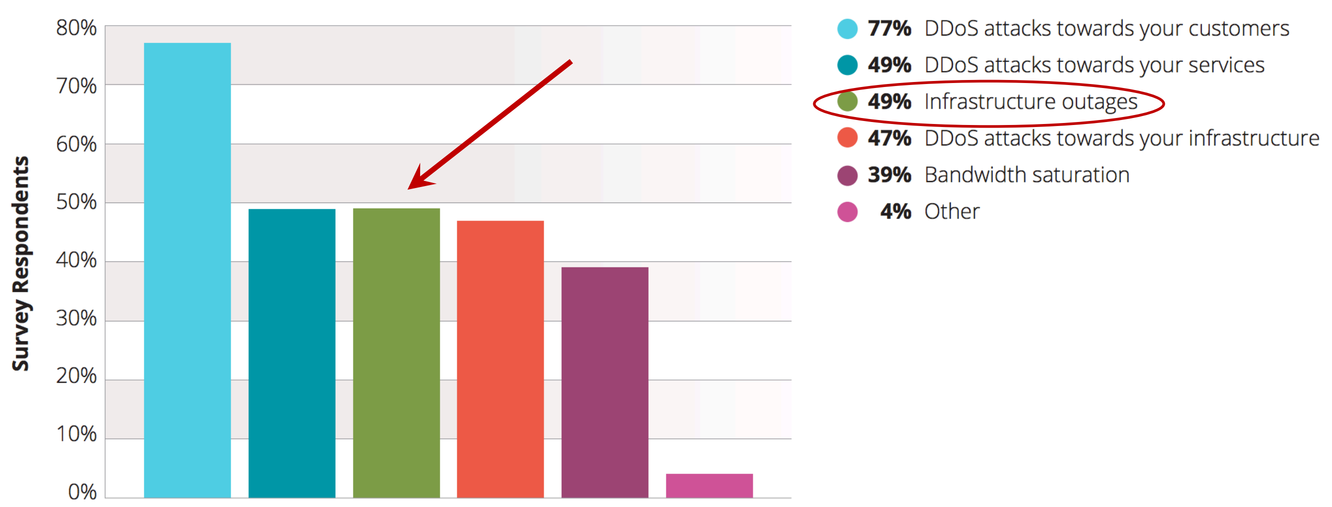
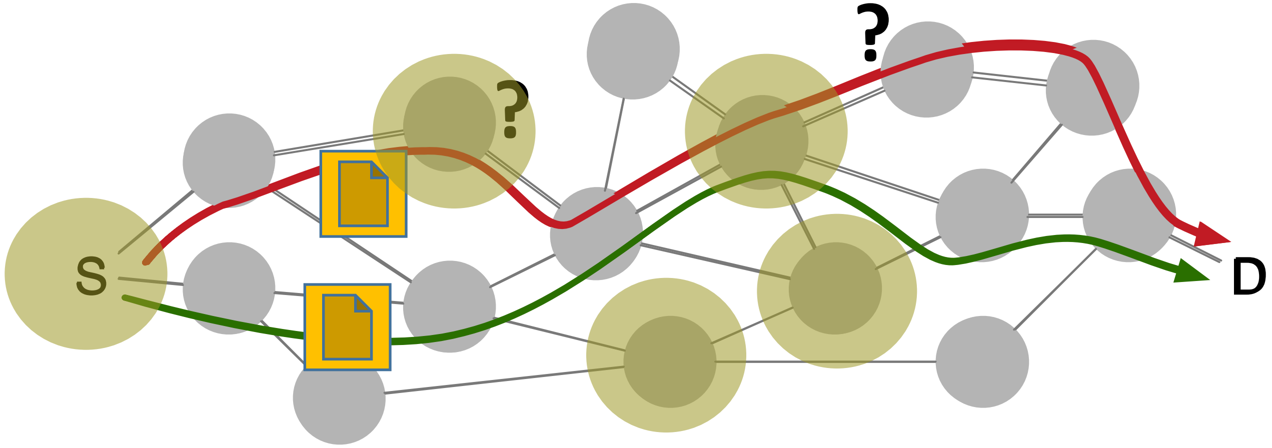


Figure 7 *Source: Arbor Networks, Inc.*

- **Fault localization problem statement**
  - Localize entities that **drop**, **delay**, or **modify traffic**
  - Practical for **inter-domain settings**



Who localizes faults?

Acceptable localization duration?

Acceptable communication overhead?

Storage overhead at nodes?

ODSBR – Awerbuch et al., ACM Trans. on Information and System Security (2008)

PAAI – Zhang et al., CoNEXT (2008)

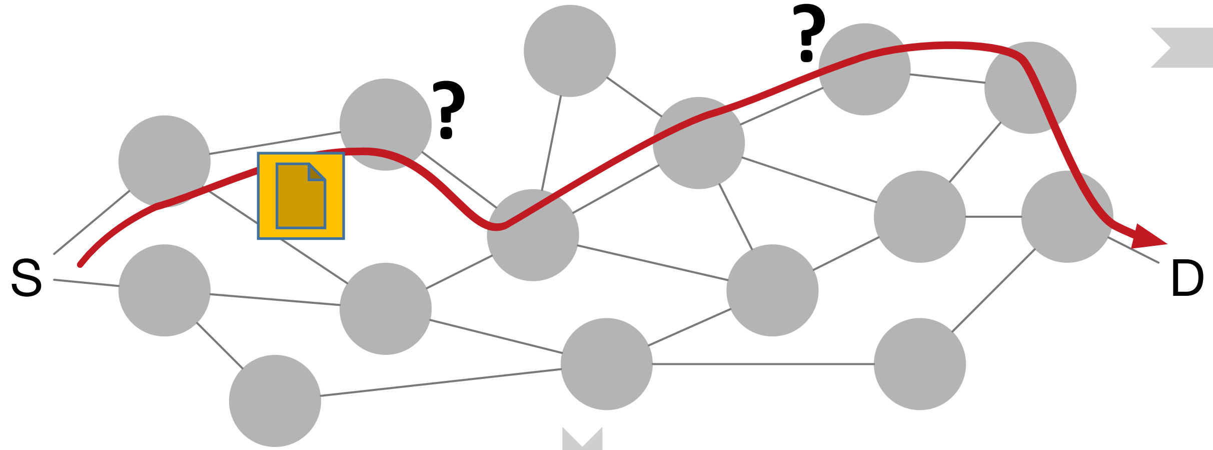
TrueNet – Zhang et al., ICNP (2011)

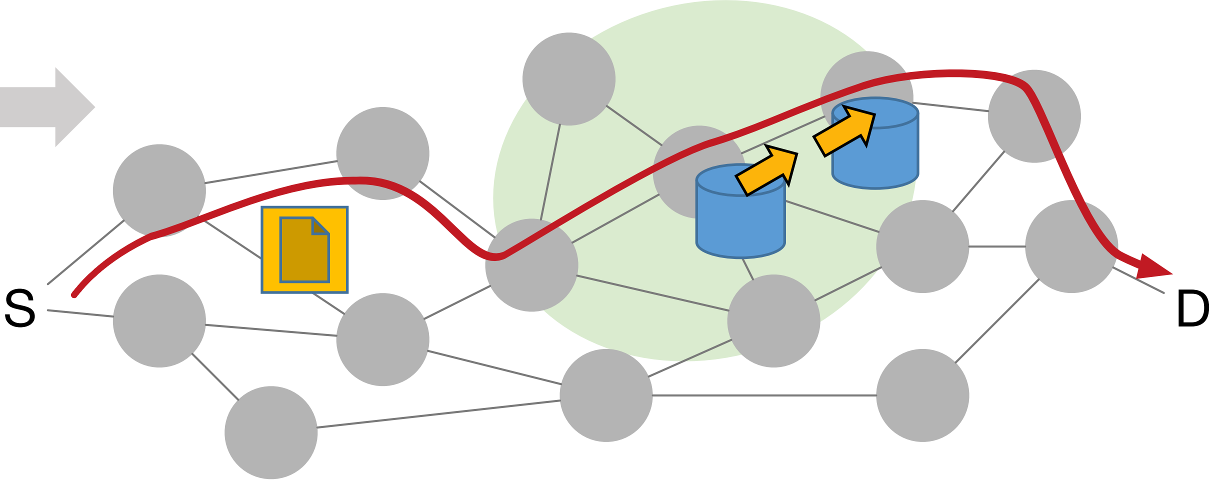
ShortMAC – Zhang et al., NDSS (2012)

DynaFL – Zhang et al., S&P (2012)

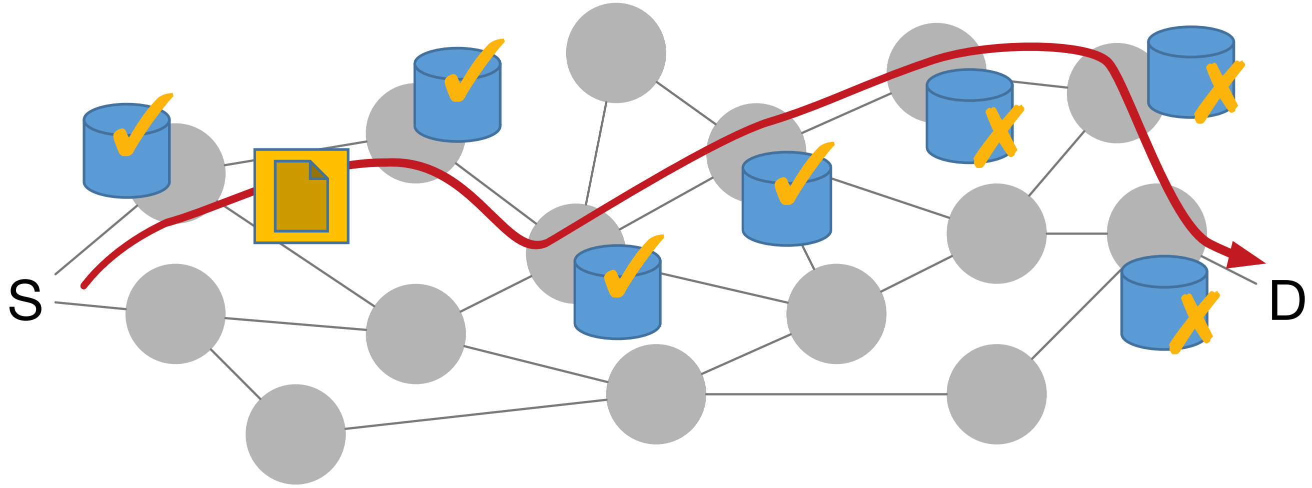
Secure sketch protocols – Goldberg et al., IEEE/ACM Trans. on Netw. (2014)

Previous approaches





Per-neighbor monitoring: incoming and outgoing flows



Per-packet monitoring: packet fingerprint

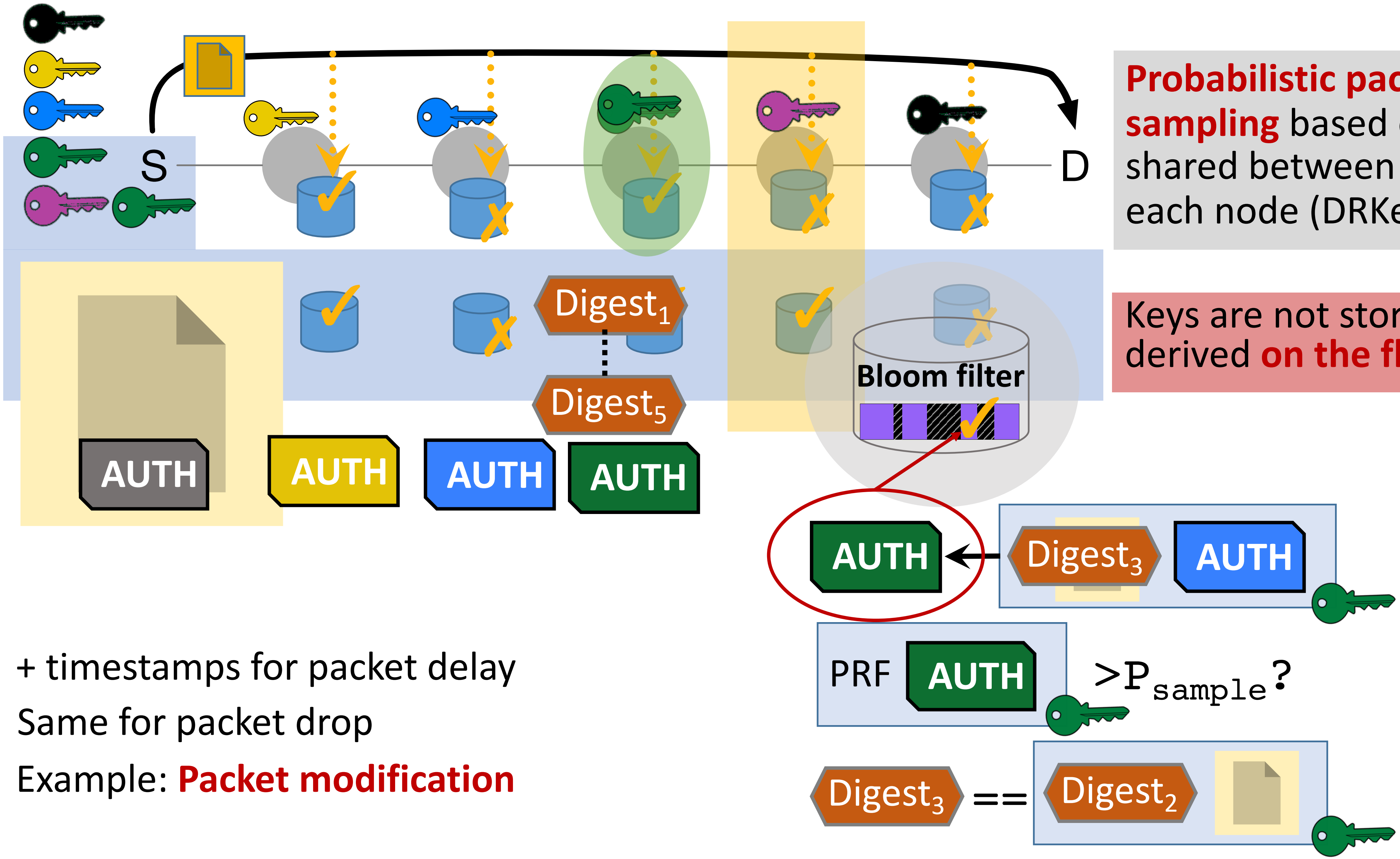
Per-flow or **per-source storage**

Traffic: 10 Gbps	**Fast path storage**
Secure sketch	~149 GB + per-source
ShortMAC	~4.6 GB + per-source
Faultprints	~46 MB

How to bound fast-path storage?

Probabilistic packet sampling based on keys shared between S and each node (DRKey*).

Keys are not stored, but derived **on the fly**.



+ timestamps for packet delay

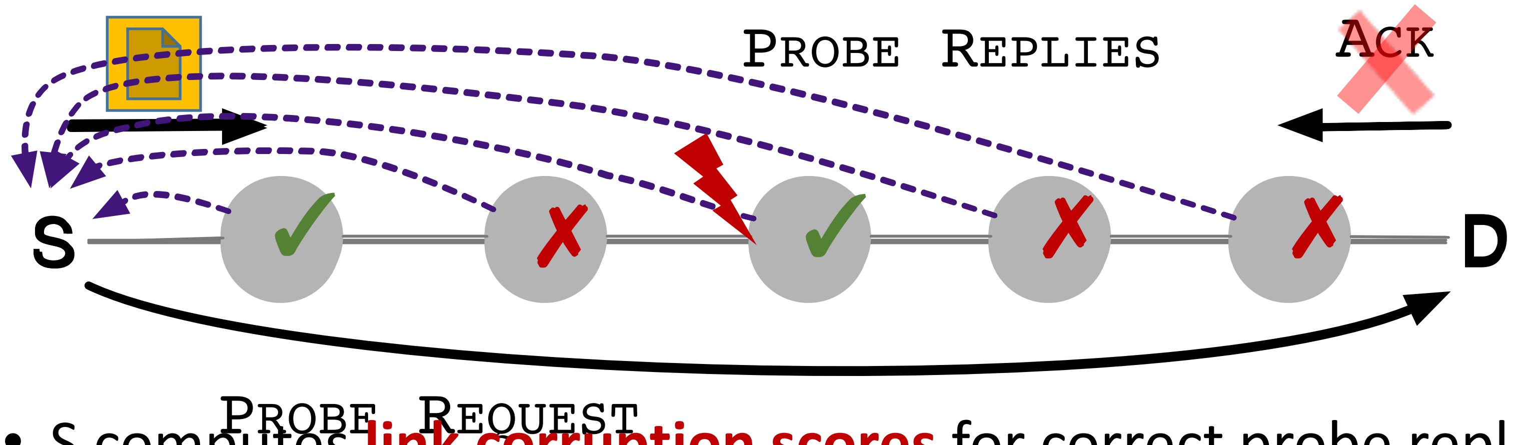
Same for packet drop

Example: **Packet modification**

*Lightweight source authentication and path validation – Kim et al., SIGCOMM 2014

Fault localization

- Localization performed when **fault is detected**



✓ packet observed

✗ packet not observed

? Incorrect reply

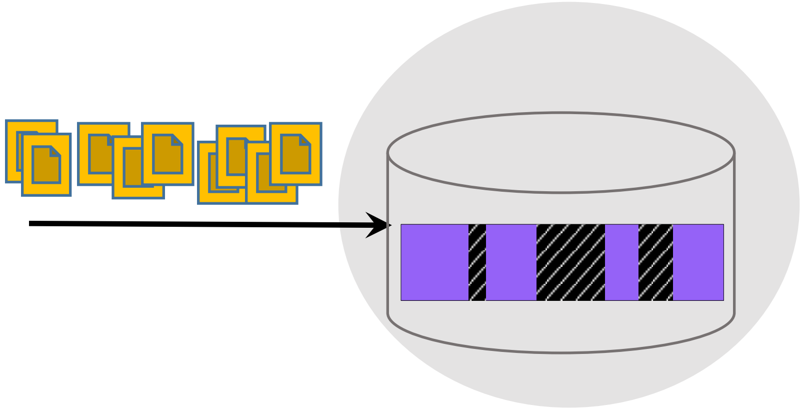
- S computes **link corruption scores** for correct probe replies

✓	✗	?	✗	✗	**Received values**
✓	✗	✓	✓	✗	**Expected values**

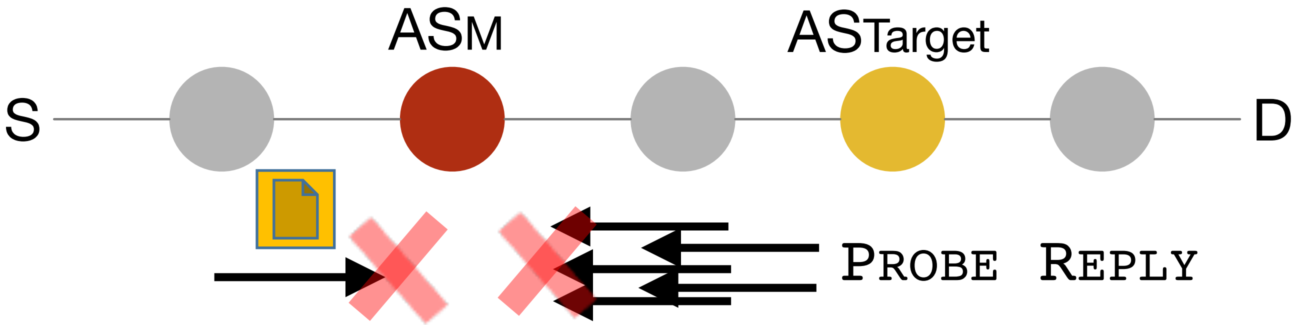
- S computes **node misbehavior probabilities** for incorrect probe replies (see paper)

Is Faultprints secure?

- **Storage exhaustion defense**
  - Epochs
  - **Worst case scenario**: ~46 MB per 10 Gbps traffic



- **Framing attacks**
  - Cannot guess packets sampled by target
  - Probe reply indistinguishability
  - Best strategy is to attack at random ⇒ reduce the attack surface



No Free Lunch

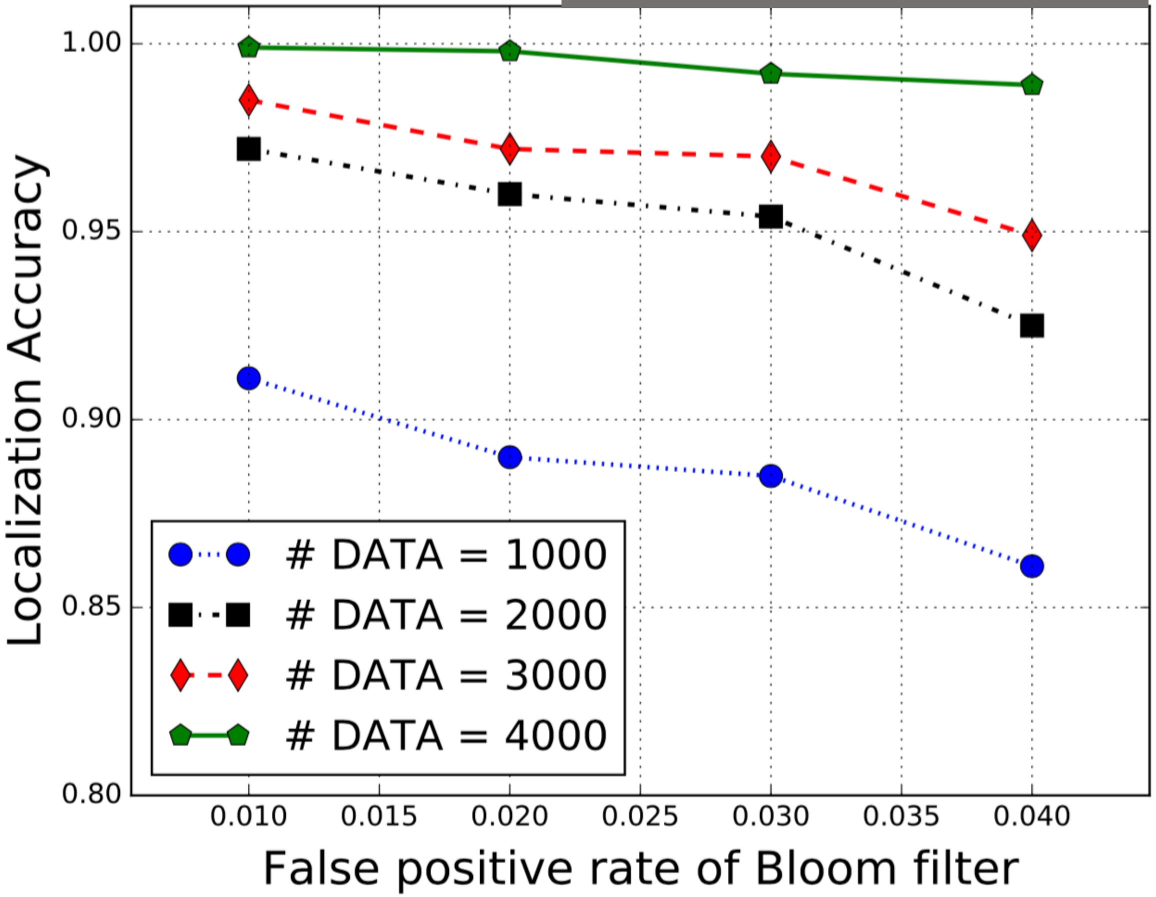
Pros & Cons

- **Low storage, but a higher communication overhead**
- **Paths symmetric or significantly overlapping**
- **Delay localization requires time synchronization between nodes**
- **Secure against sophisticated attackers**

ACCURACY AND THROUGHPUT

Localization accuracy

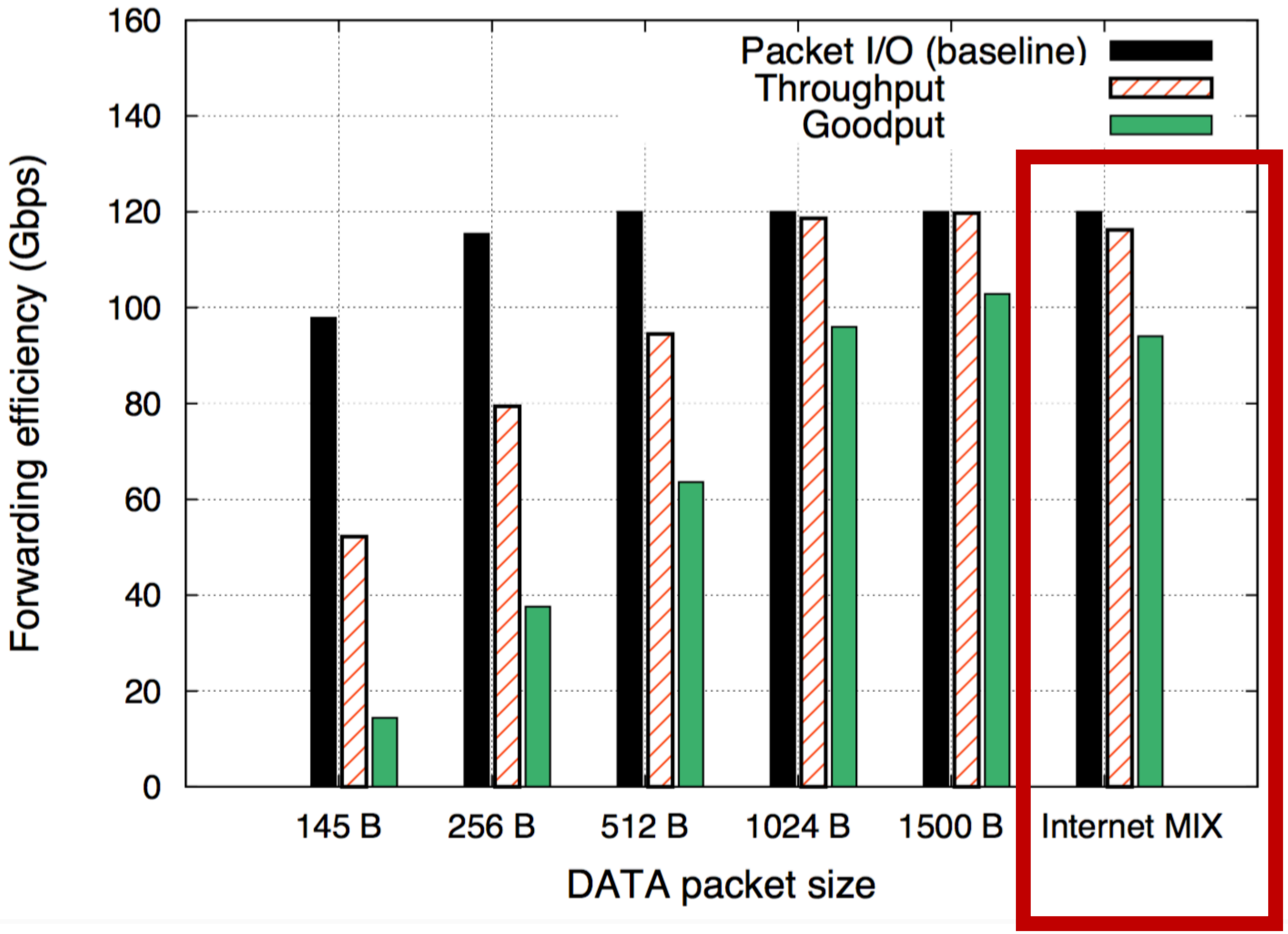
- One malicious node, at random locations on path
- Path length 5 ASes, link natural packet loss 0.1%



Corruption score accuracy

Throughput and Goodput

- **Commodity server** as Faultprints router receiving traffic at 120 Gbps



- Sampling rate 10%
- Bloom filter false positive rate 0.02
- Path length 5 ASes

Conclusion

- **Faultprints localizes Internet-wide packet drop, delay, and modification**
- **Low storage requirements: ~46 MB for 10 Gbps traffic rate**
- **Secure against storage exhaustion attacks and framing attacks**
- **Real-world traffic forwarded on commodity server at ~117 / 120 Gbps**