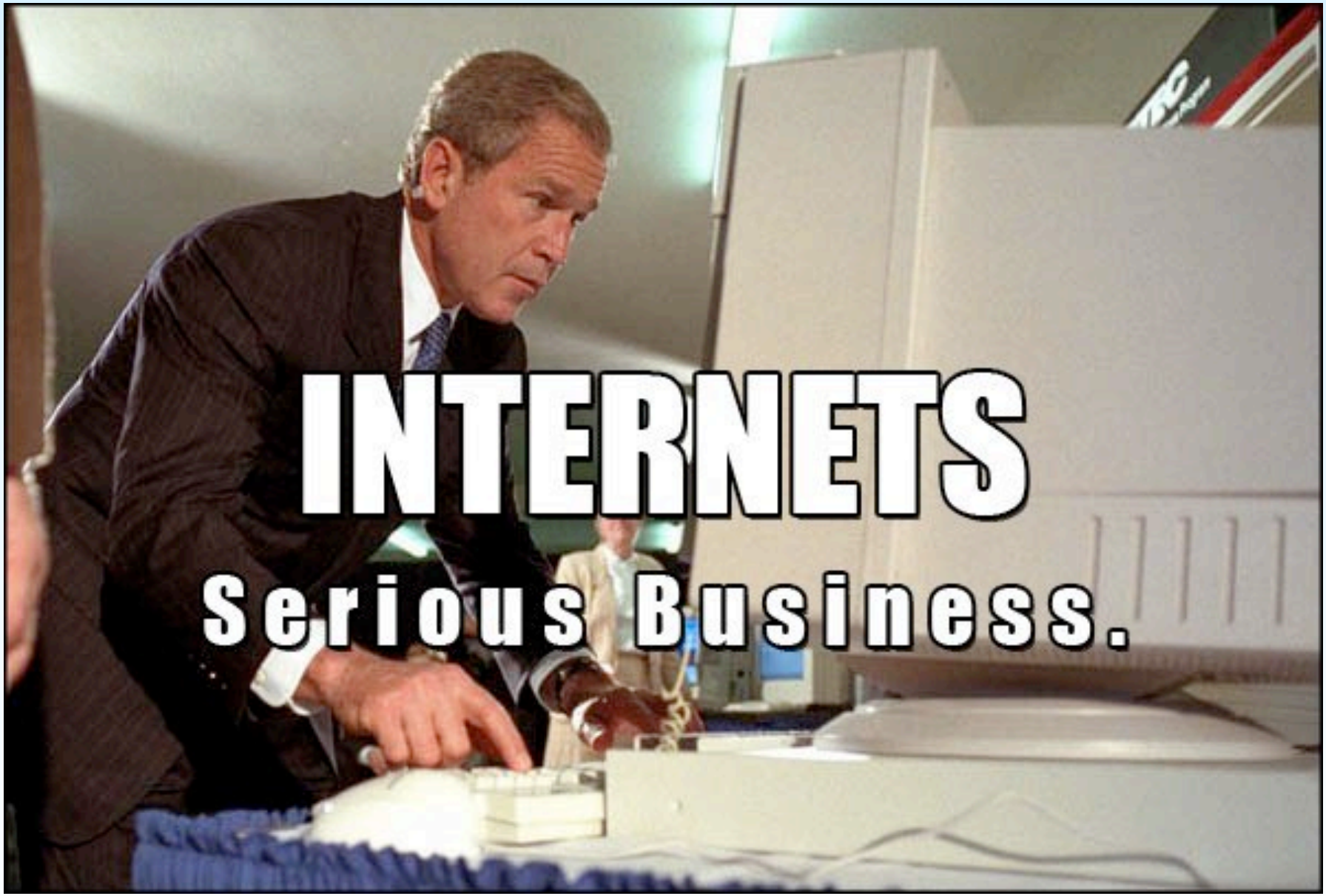
StrobeLight: Lightweight Availability Mapping and Anomaly Detection

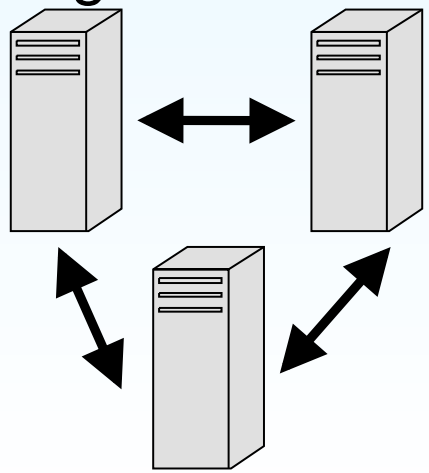
James Mickens, John Douceur, Bill Bolosky
Brian Noble


INTERNETS
Serious Business.

At any given moment, how can we tell which enterprise machines are online and network-reachable?

Customer

Mobile AJAX cloud-based social networking goodness



Who Could Give Us Availability Data?

- Best case: Zeus
- If we're lucky: the distributed system itself
  - Limited scope?
  - Doesn't scale?
  - Need to modify hosts/ routers?

Our Solution: StrobeLight

- Persistent enterprise-level monitoring
  - Track availability of 200K+ hosts
- Network-wide sweep every 30 seconds
  - Fast enough for near real-time analysis
  - Archive results for use by other services
- Doesn’t require modification to:
  - End hosts
  - Core routing infrastructure

How Would We Use This Data?

- Improve system performance
  - DHTs, Farsite: select the best storage hosts
  - Multicast trees: build more robust topologies
  - BOINC: perform smarter task allocation
- Detect system-level anomalies
  - Misconfigured routers
  - IP hijacking attacks

Outline

- Design and Implementation
- Availability Fingerprints
- Detecting IP Hijacks Using Fingerprints
- Related Work
- Conclusions

Design Goals

- Keep it simple, stupid
  - Don't modify end hosts
  - Don't change routing core

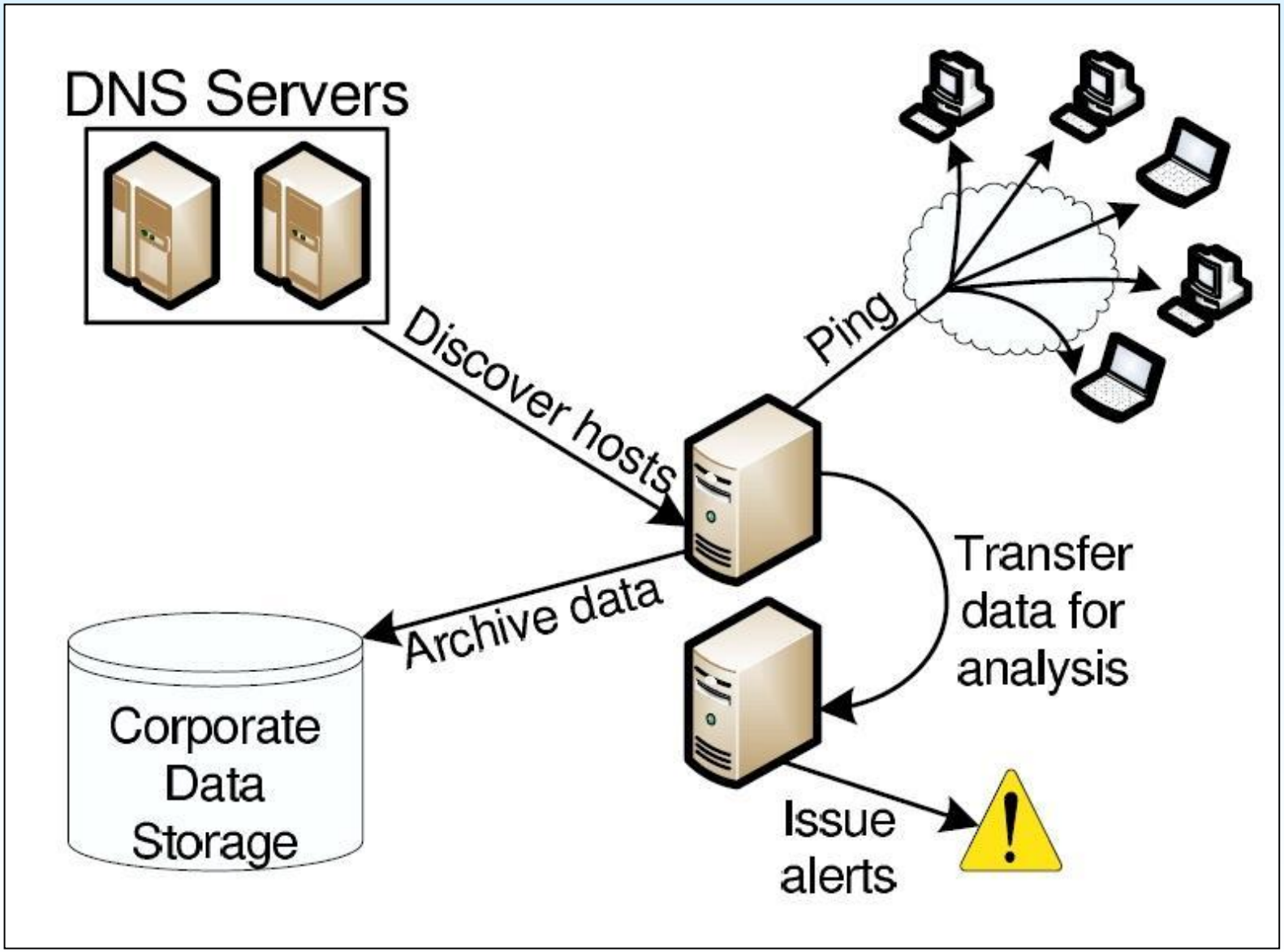
- Don't be annoying
  - Don't impact real flows

- Collect high-resolution data
  - Per-host statistics
  - Fine temporal granularity

There Were Non-goals™

- Infinite scaling: overkill in enterprise setting
  - Scaling target: hundred of thousands of hosts
  - Small number of administrative domains
  - Centralized solution might be okay
- Total address disambiguation: hard, unnecessary
  - NATs, DHCP, firewalls decouple hosts, IPs
  - We’re content to measure IP reachability

The Winning Design: StrobeLight



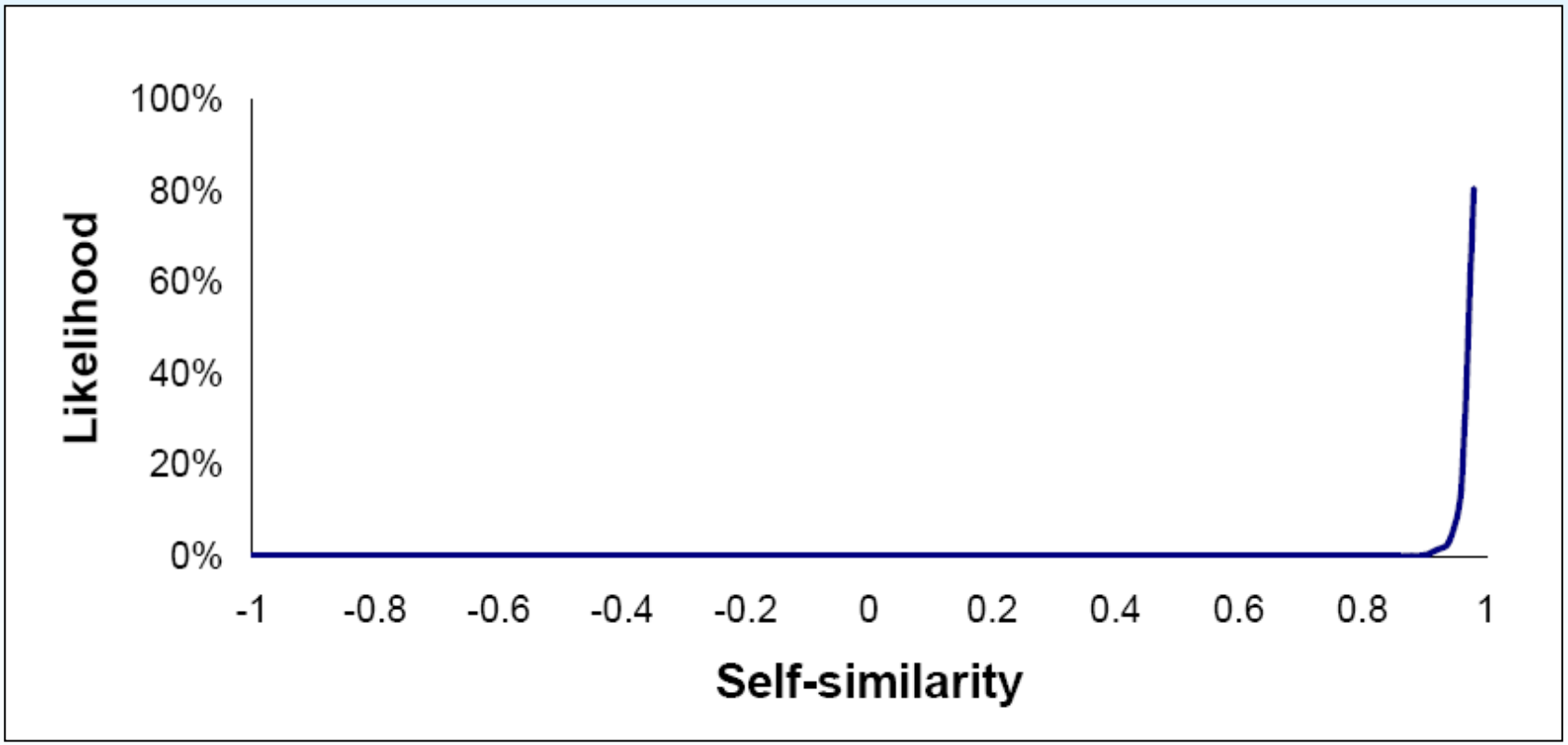
Outline

- Design and Implementation
- Availability Fingerprints
- Detecting IP Hijacks Using Fingerprints
- Related Work
- Conclusions

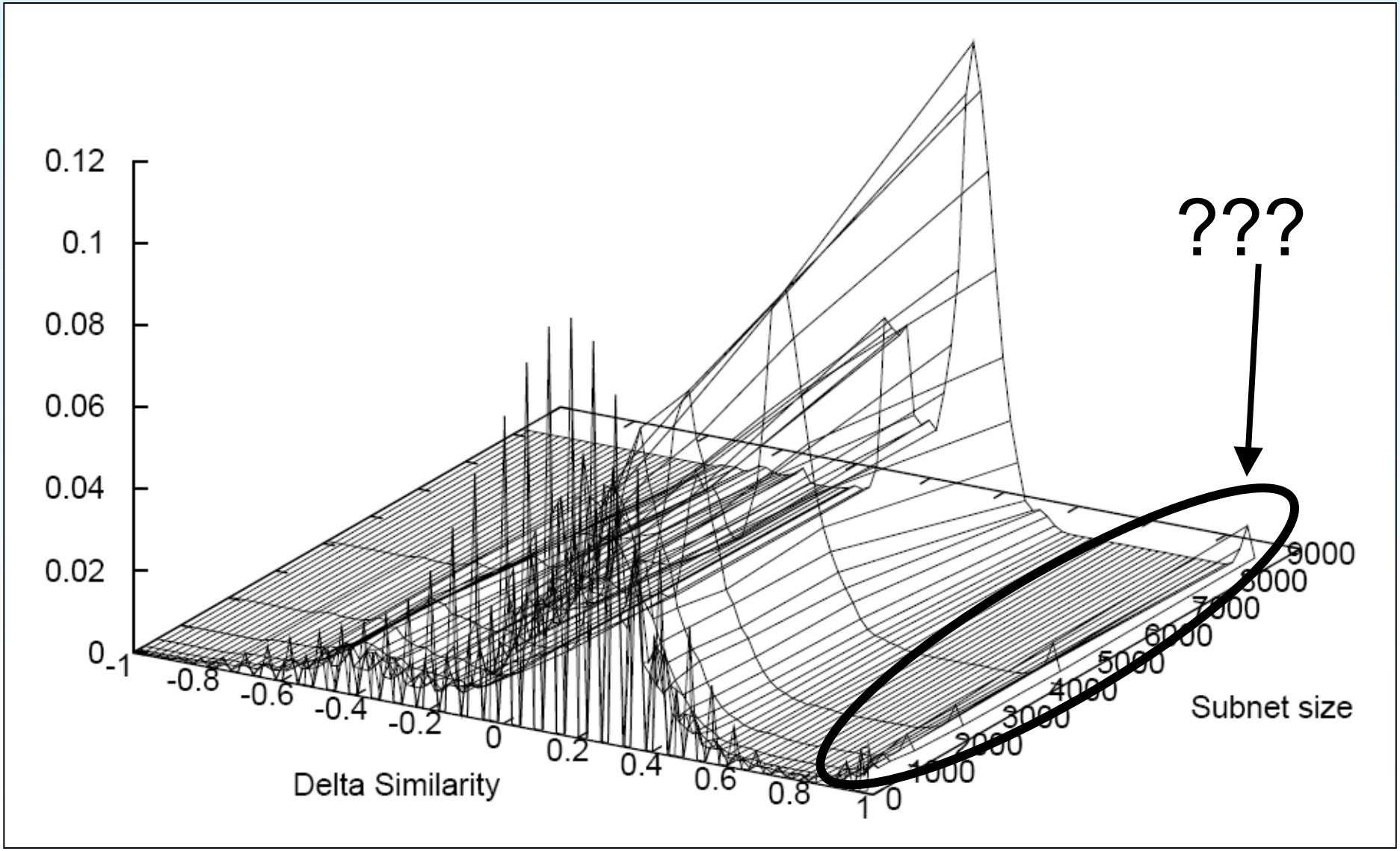
Availability Fingerprint

- Instantaneous snapshot of subnet availability
  - Bit vector: $b_h = 1$ iff host h responded to probe
- Similarity metric: # of equivalent bit positions
  - Normalize to the range [-1,1]
- What does fingerprint similarity look like . . .
  - Within a single subnet across time?
  - Between different subnets at a given moment?

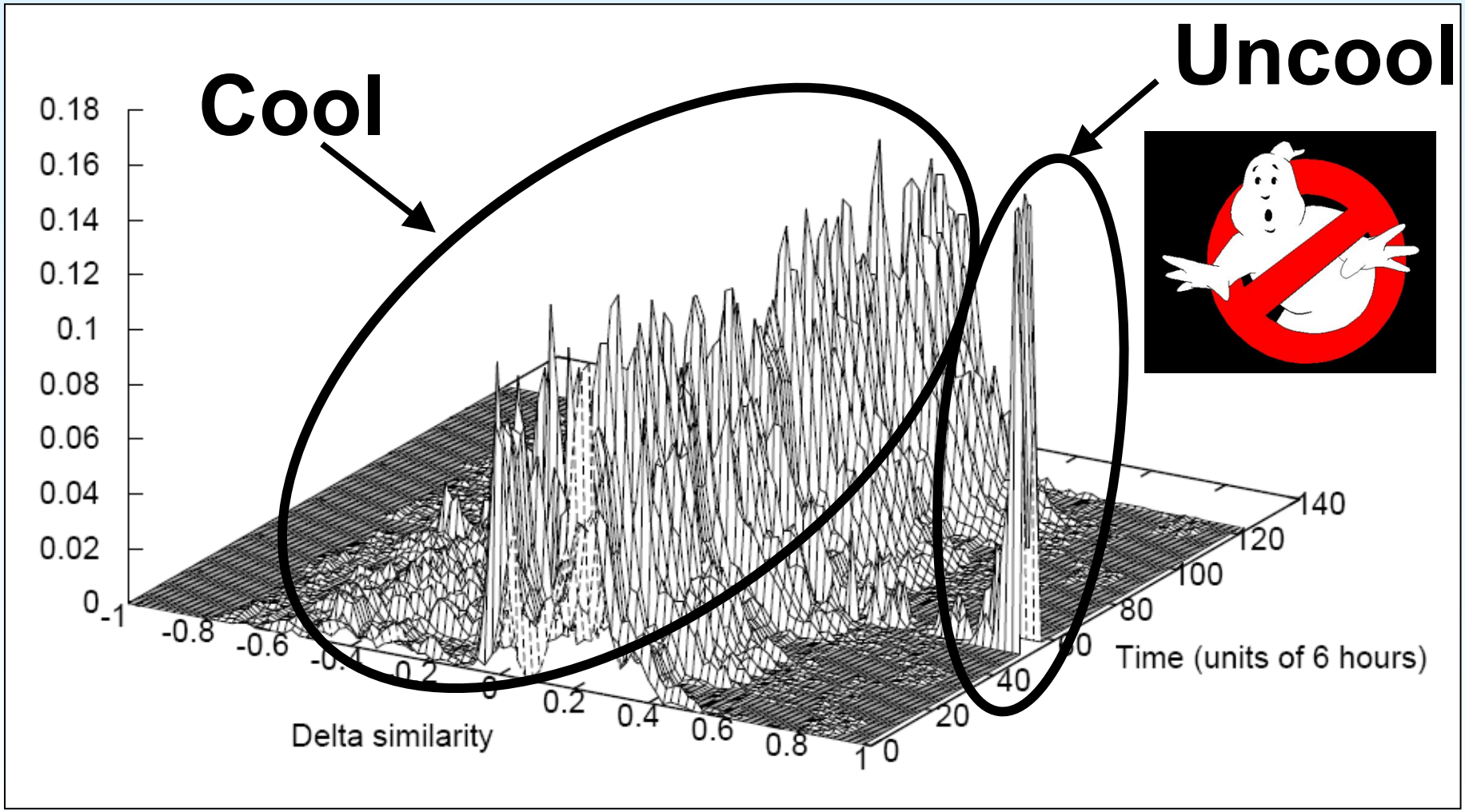
Self-similarity: 15 minute intervals (256-host subnets)



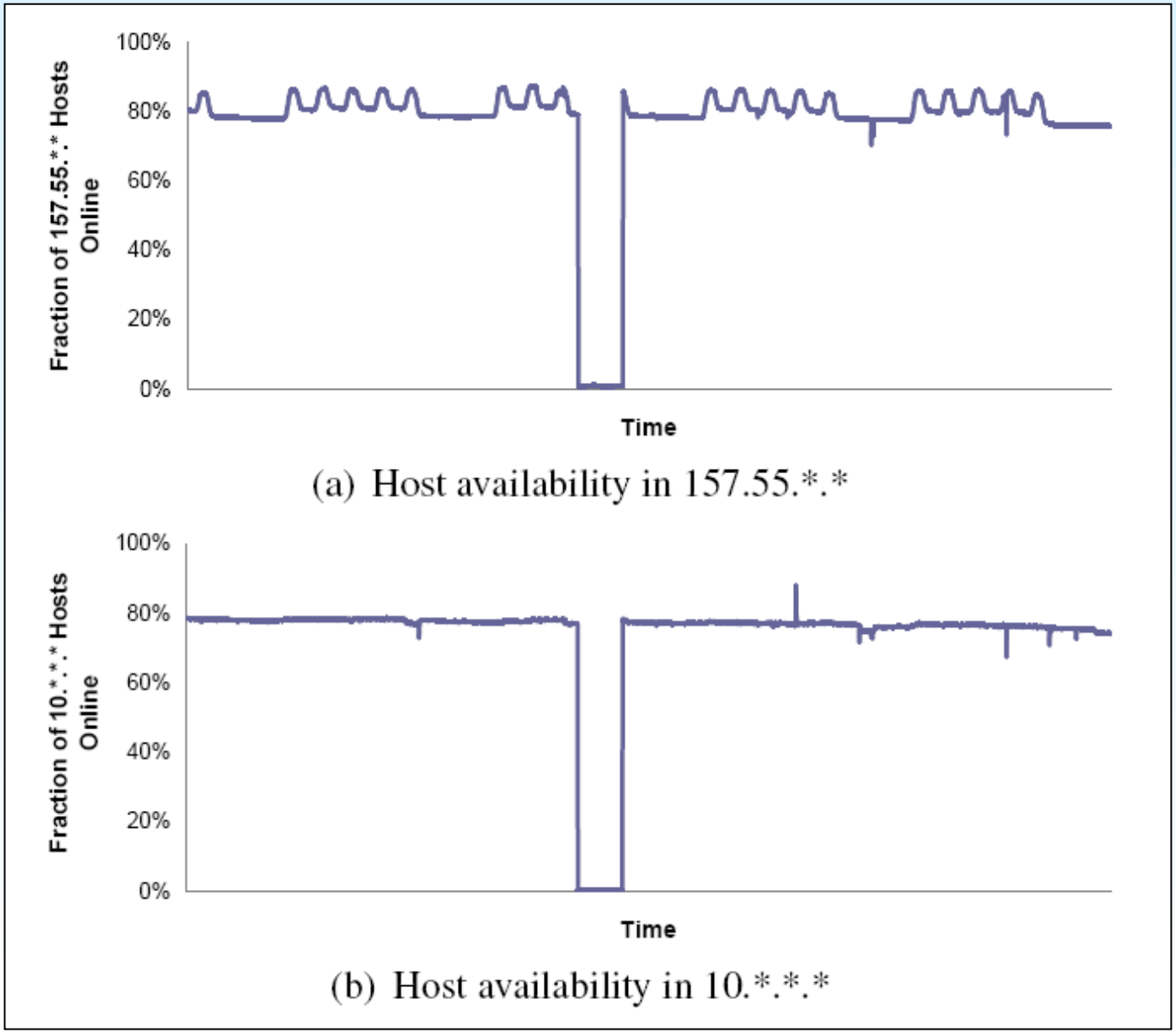
Instantaneous Cross-subnet Similarity



Cross-subnet similarity vs. Time

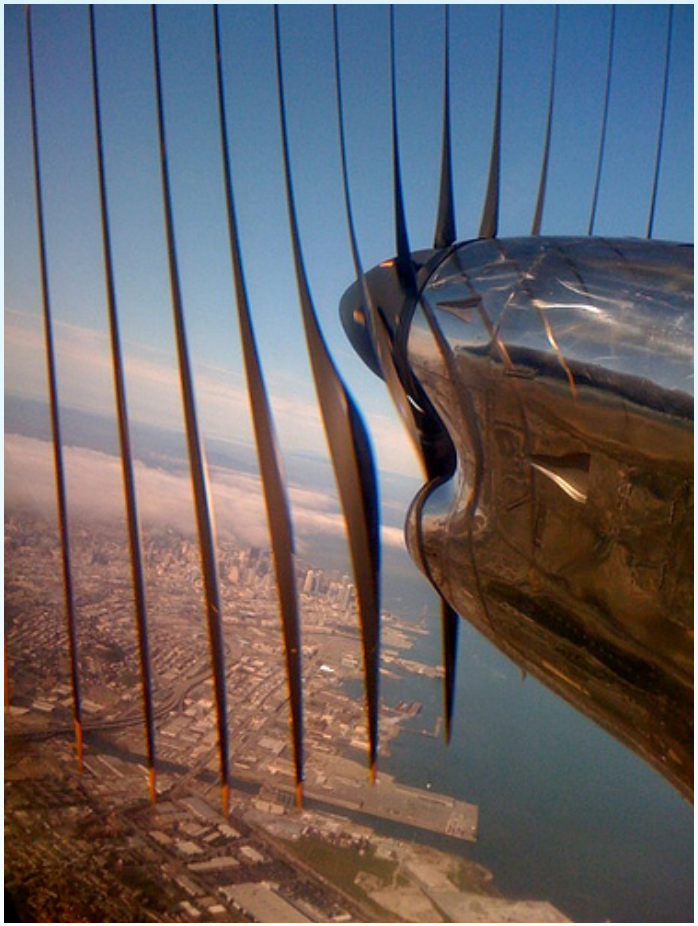


Ghosts Were Not To Blame



One Use For StrobeLight

Outline



- Design and Implementation
- Availability Fingerprints
- Detecting IP Hijacks Using Fingerprints
- Related Work
- Conclusions

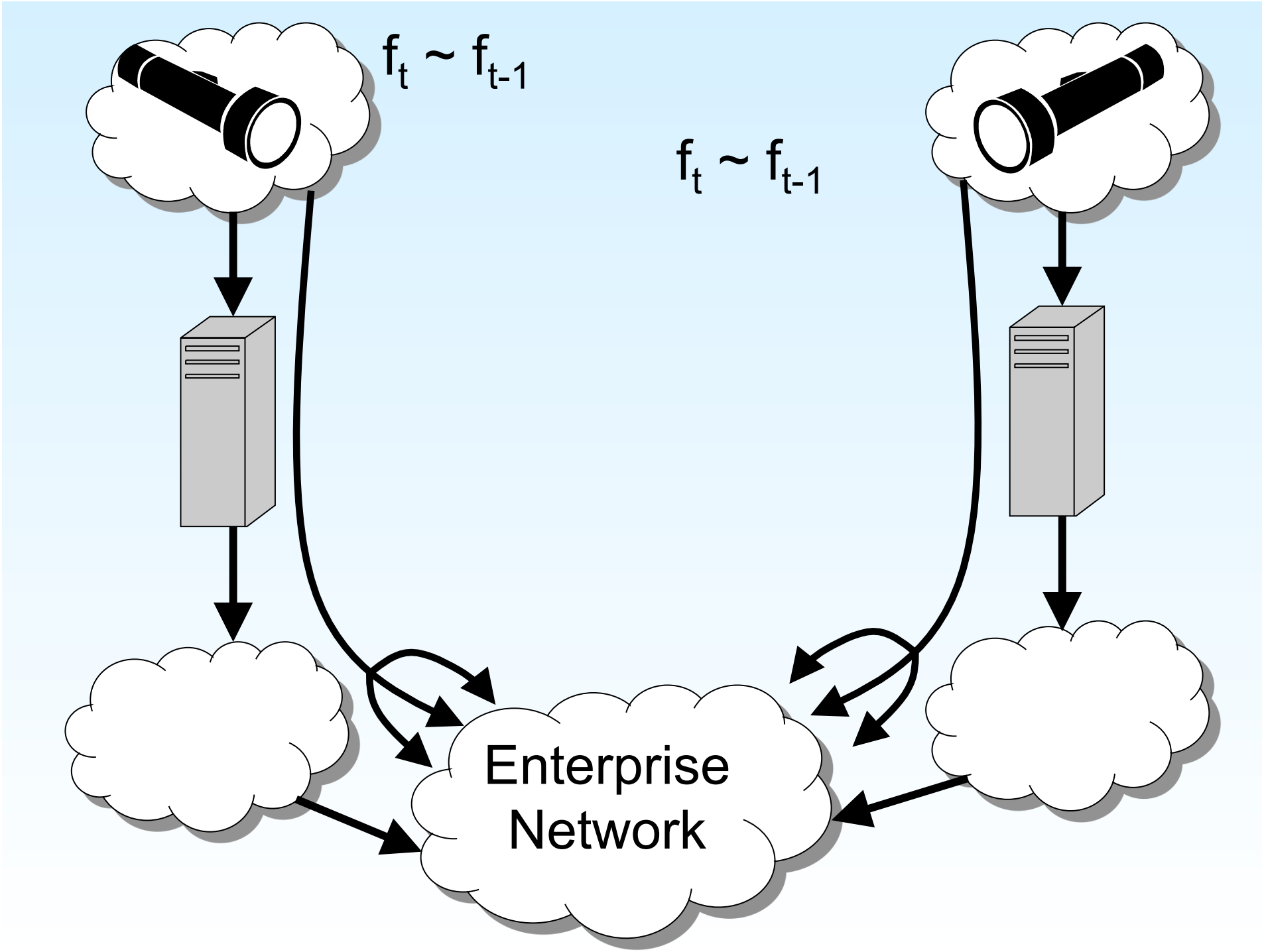
IP Hijacking

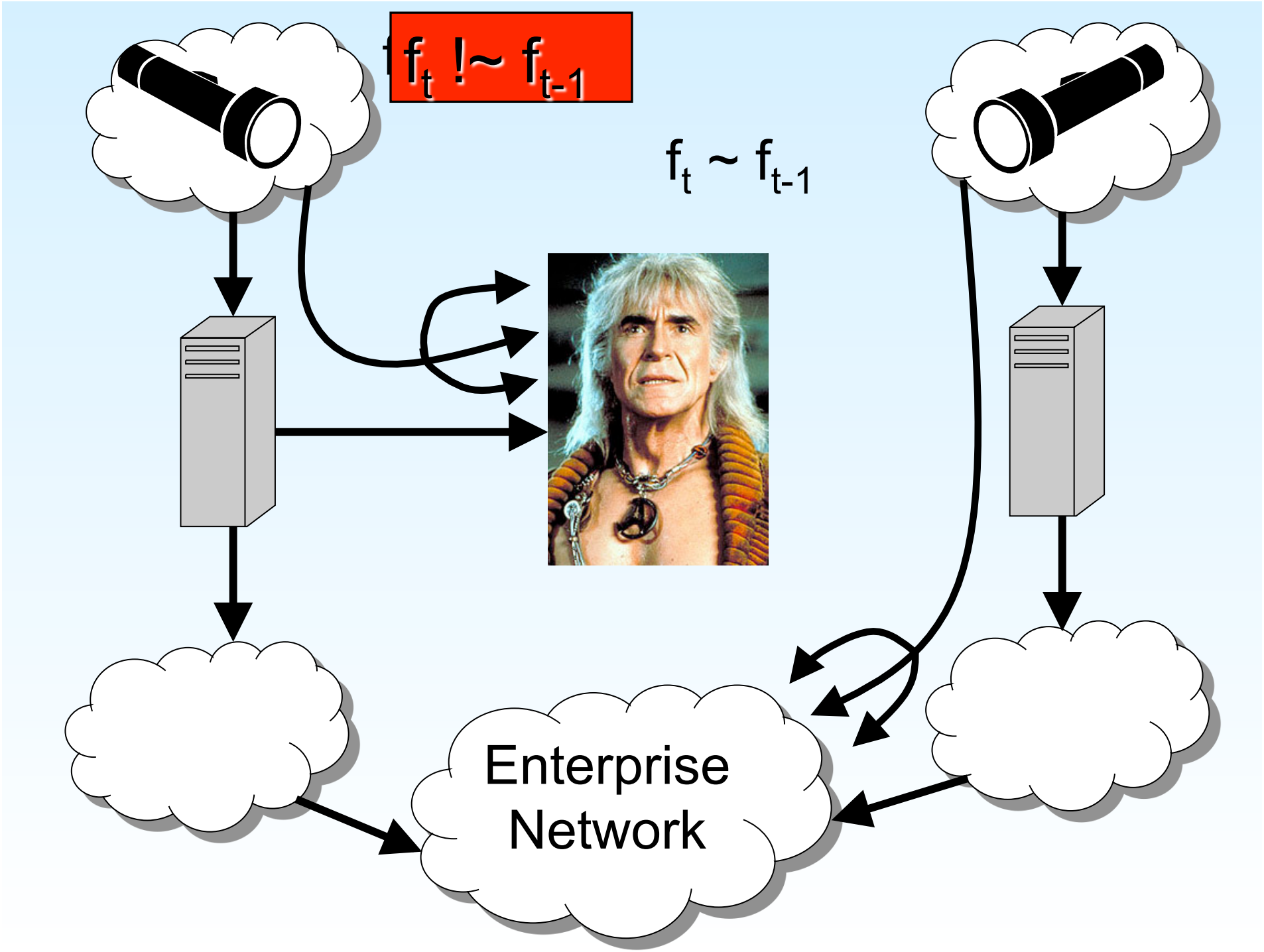
- Internet: a collection of autonomous systems
- BGP protocol stitches ASes together
  - ASes announce prefix ownership, path lengths
  - No authentication of announcements!
- Hijack attack: disrupt routing to target prefix
  - Announce ownership of/short route to prefix
  - Some routers may not be affected (location matters)

IP Hijacking

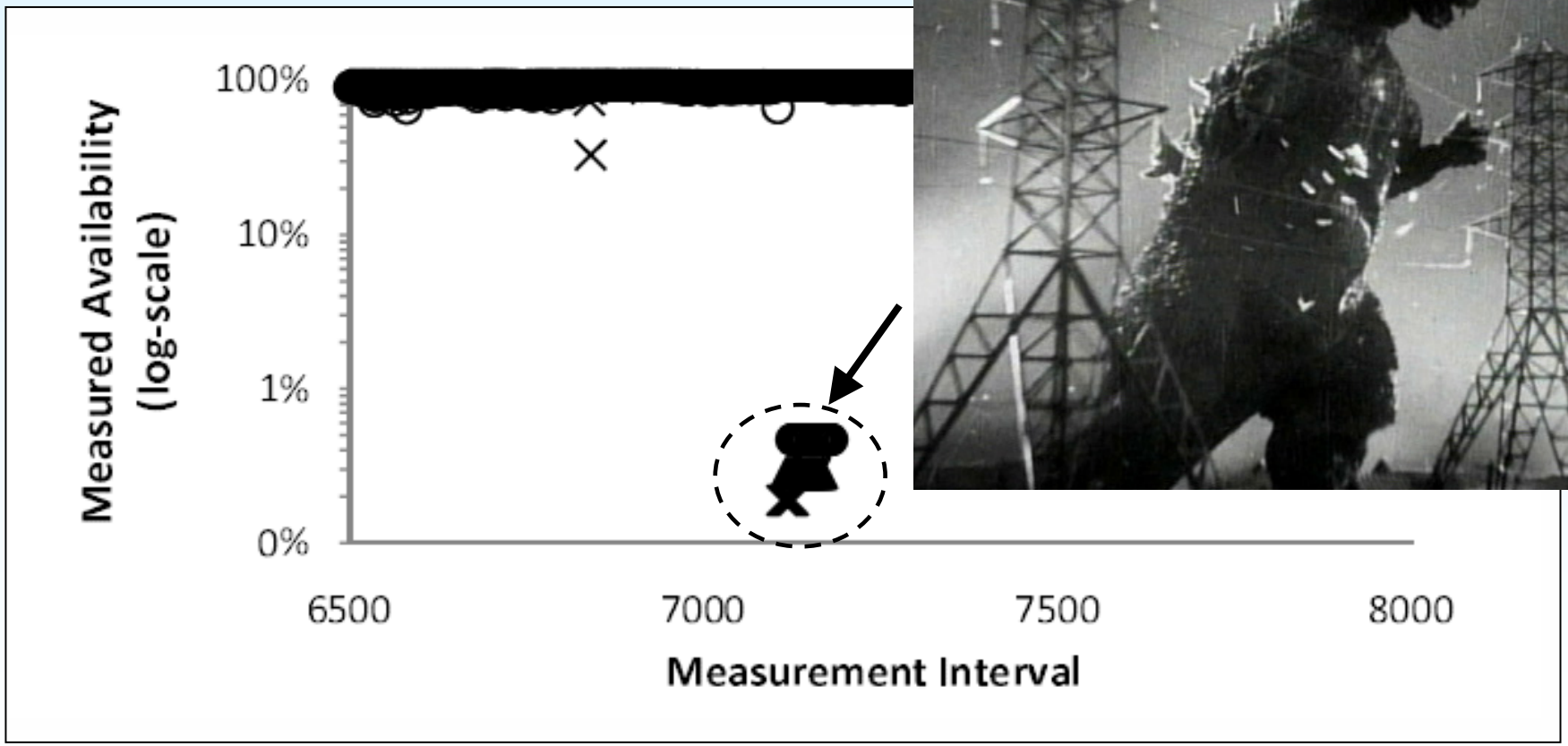
1) Blackhole attack: drop all traffic
2) Imposture attack: impersonate target prefix
3) Interception attack: inspect/modify traffic

- First two should cause fingerprint anomalies!

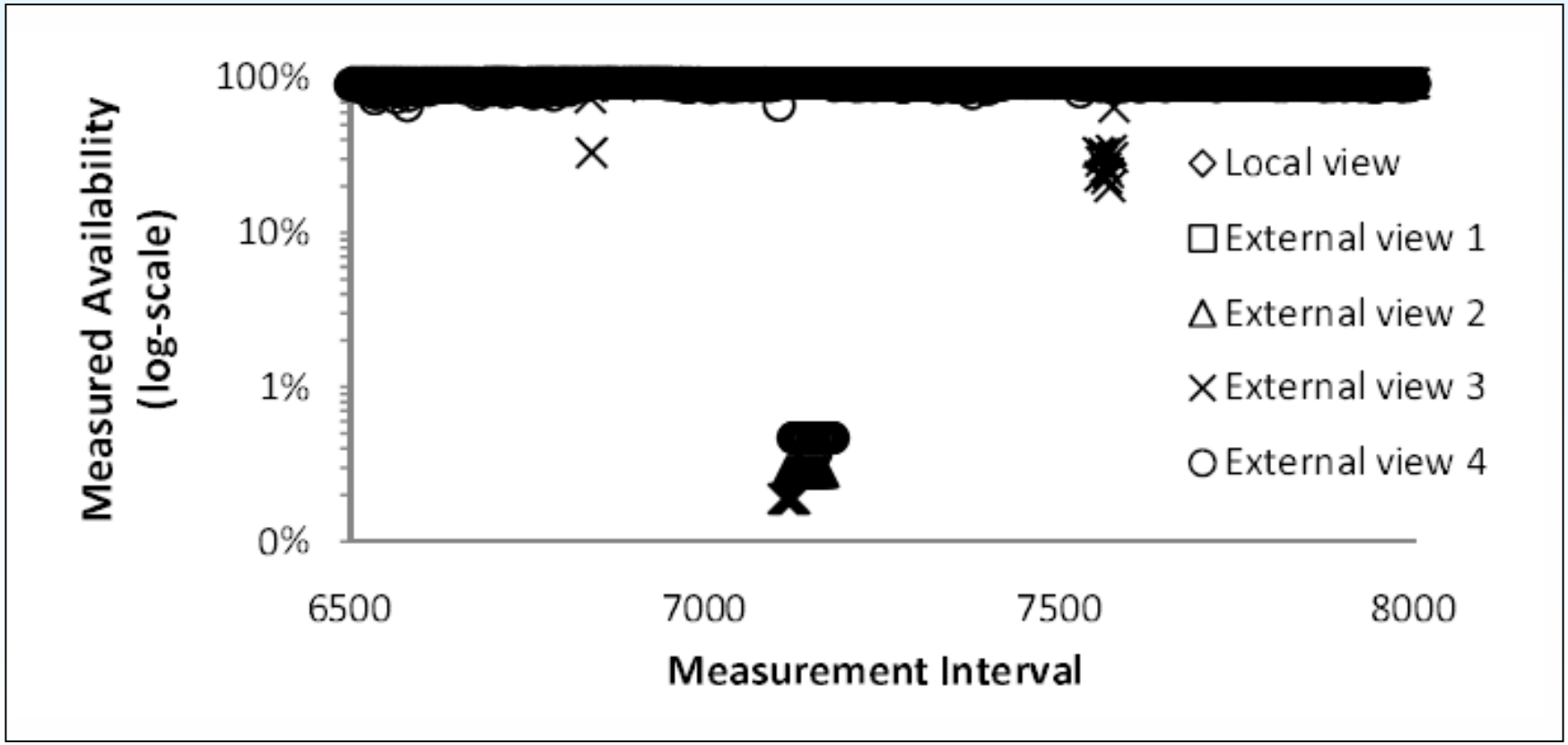

$f_t \sim f_{t-1}$
$f_t \sim f_{t-1}$
Enterprise
Network


f_t !~ f_{t-1}
f_t ~ f_{t-1}
Enterprise Network

Does WAN Distort Our Probes?



Does WAN Distort Our Probes?



Spectrum Agility Hijacks

- Short-lived manipulation of BGP state
  - Hijack /8 prefix
  - Send spam from random IP addresses
  - Withdraw BGP advertisement a few minutes later
- Assume attacker subnet has random fingerprint

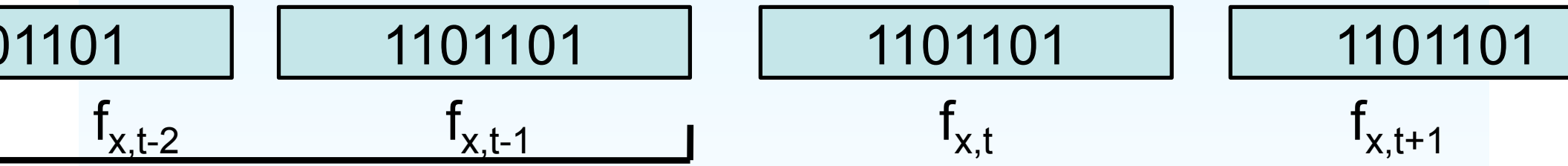
Spectrum Agility Hijacks

- Simulation setup
  - Slide window through MSR trace
  - For each subnet x, test two similarities

Spectrum Agility Hijacks

- Simulation setup
  - Slide window through MSR trace
  - For each subnet x, test two similarities

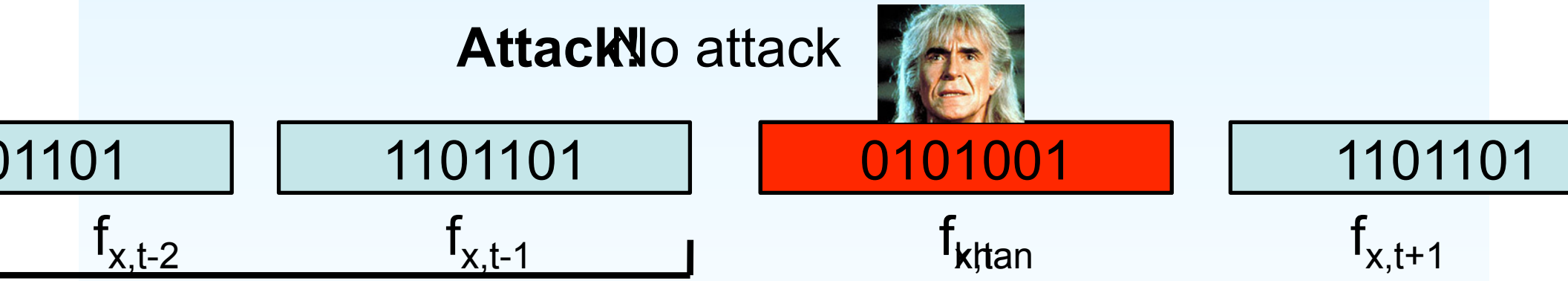
No attack



True negative: $sim(f_{x,t}, f_{x,t-1}) \geq c$
False positive: $sim(f_{x,t}, f_{x,t-1}) < c$

Spectrum Agility Hijacks

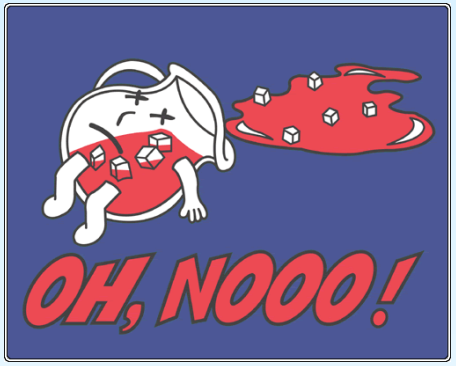
- Simulation setup
  - Slide window through MSR trace
  - For each subnet x, test two similarities



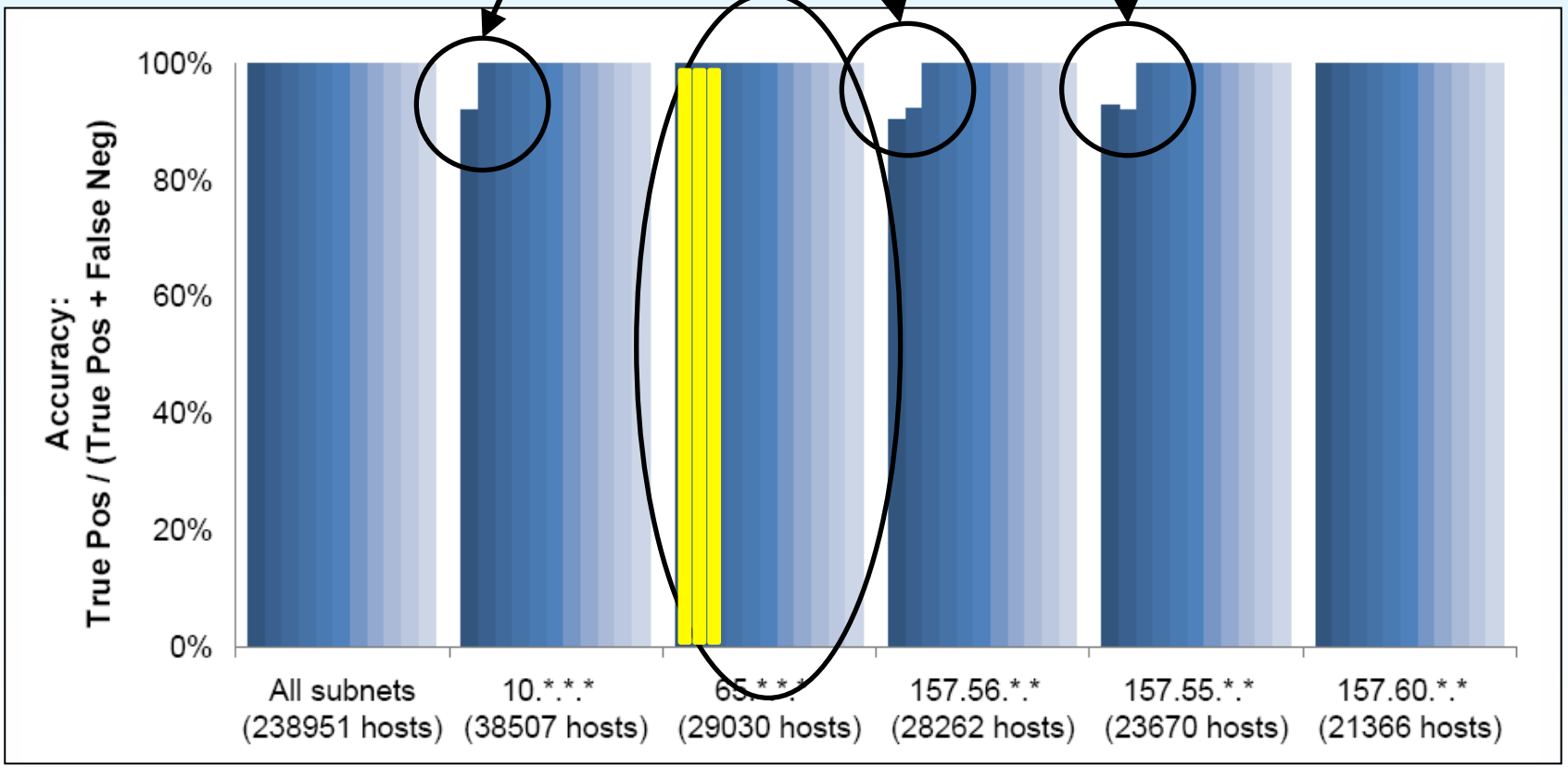
True positive: $sim(f_{khan}, f_{x,t-1}) < c$

False negative: $sim(f_{khan}, f_{x,t-1}) \geq c$

Detecting Spectrum Attacks: c=0.78



DNS failure: StrobeLight thinks hosts have died



Outline

- Design and Implementation
- Availability Fingerprints
- Detecting IP Hijacks Using Fingerprints
- Related Work
- Conclusions

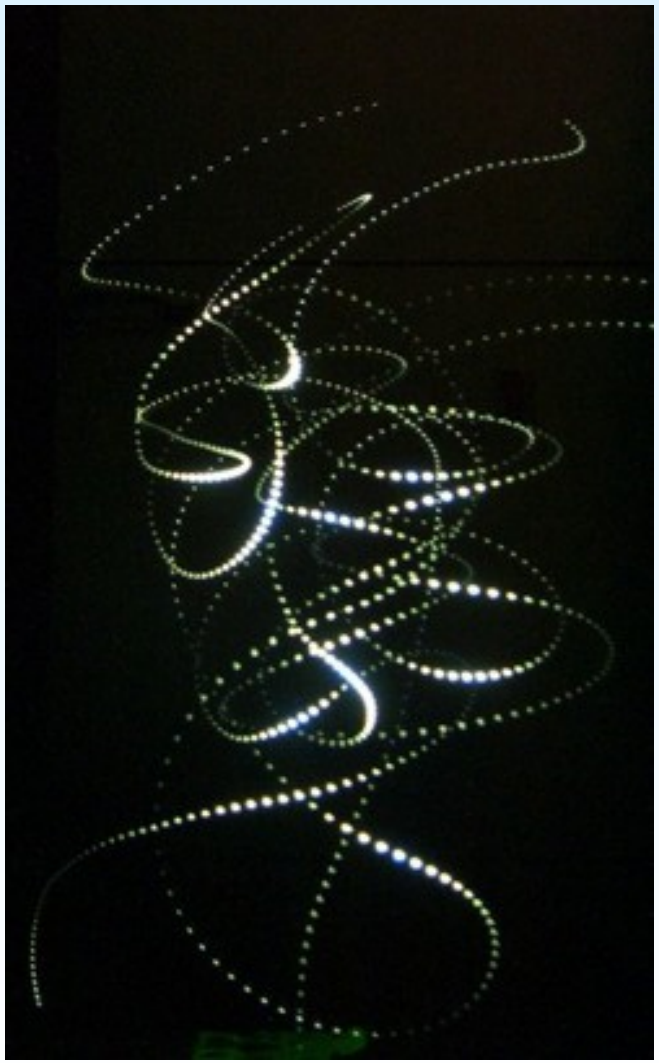
Availability Monitoring

- Academic network path monitors
  - CoMon, iPlane, RON
  - Don’t scale to enterprise/don’t track per-host stats
- Commercial monitoring tools
  - Pro: Richer set of statistics
  - Cons: More difficult to deploy, slower refresh

Detecting IP Hijacking

- Modify BGP/push crypto into routing core
  - Aiello 2003, Hu 2004, Zhao 2002, etc.
- Passive monitoring of routing state
  - Find anomalies in RouteViews, IRR
- Data plane fingerprints (Hu and Mao 2006)
  - Monitor live BGP for suspicious updates
  - Scan target prefix with nmap, IP ID probes
  - Raise alarm if different views are inconsistent

Conclusion



- StrobeLight: enterprise-level availability monitor
  - End hosts/routers unchanged
  - Real-time feeds, archival data
- Example of StrobeLight client: Hijack detector
  - Uses availability fingerprints to find routing anomalies
  - Anomaly detection is fast and accurate
  - Don’t need to modify BGP/push crypto into routers

Thanks!