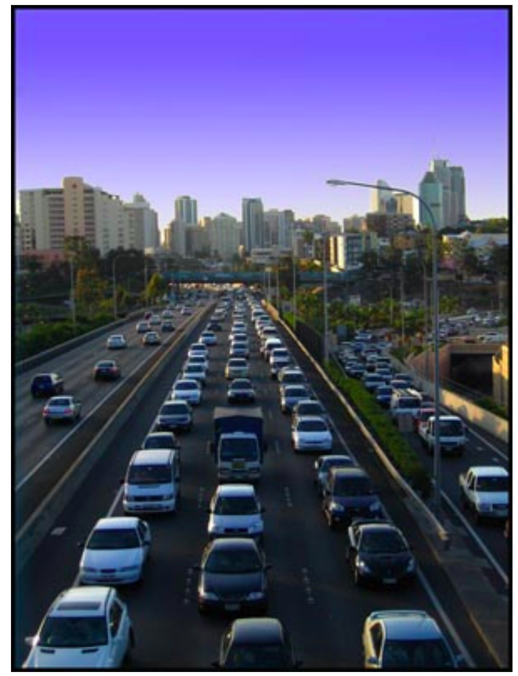
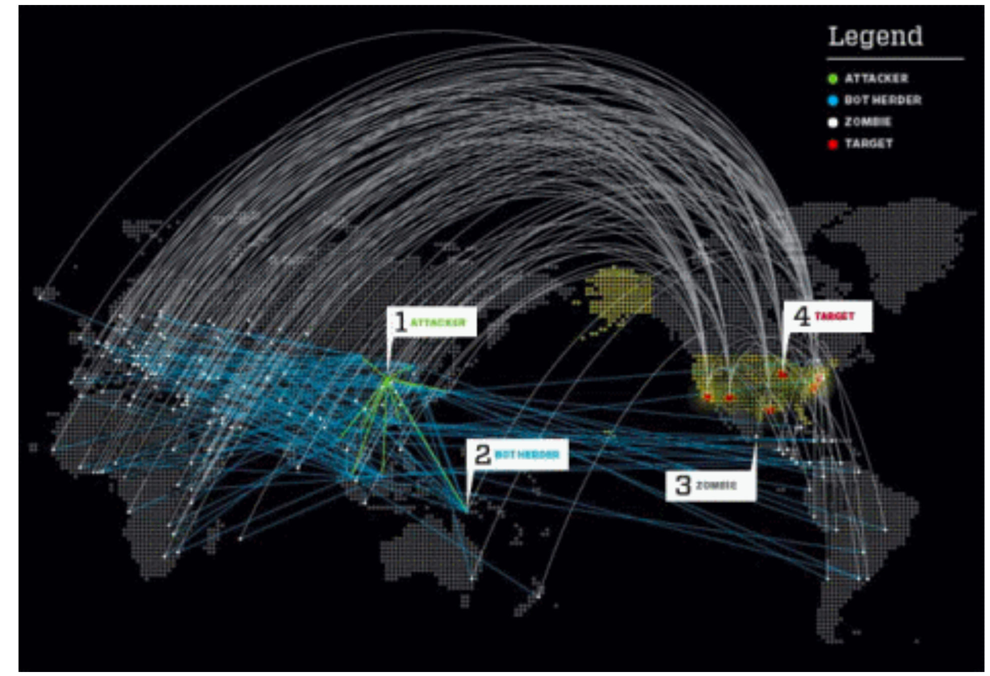




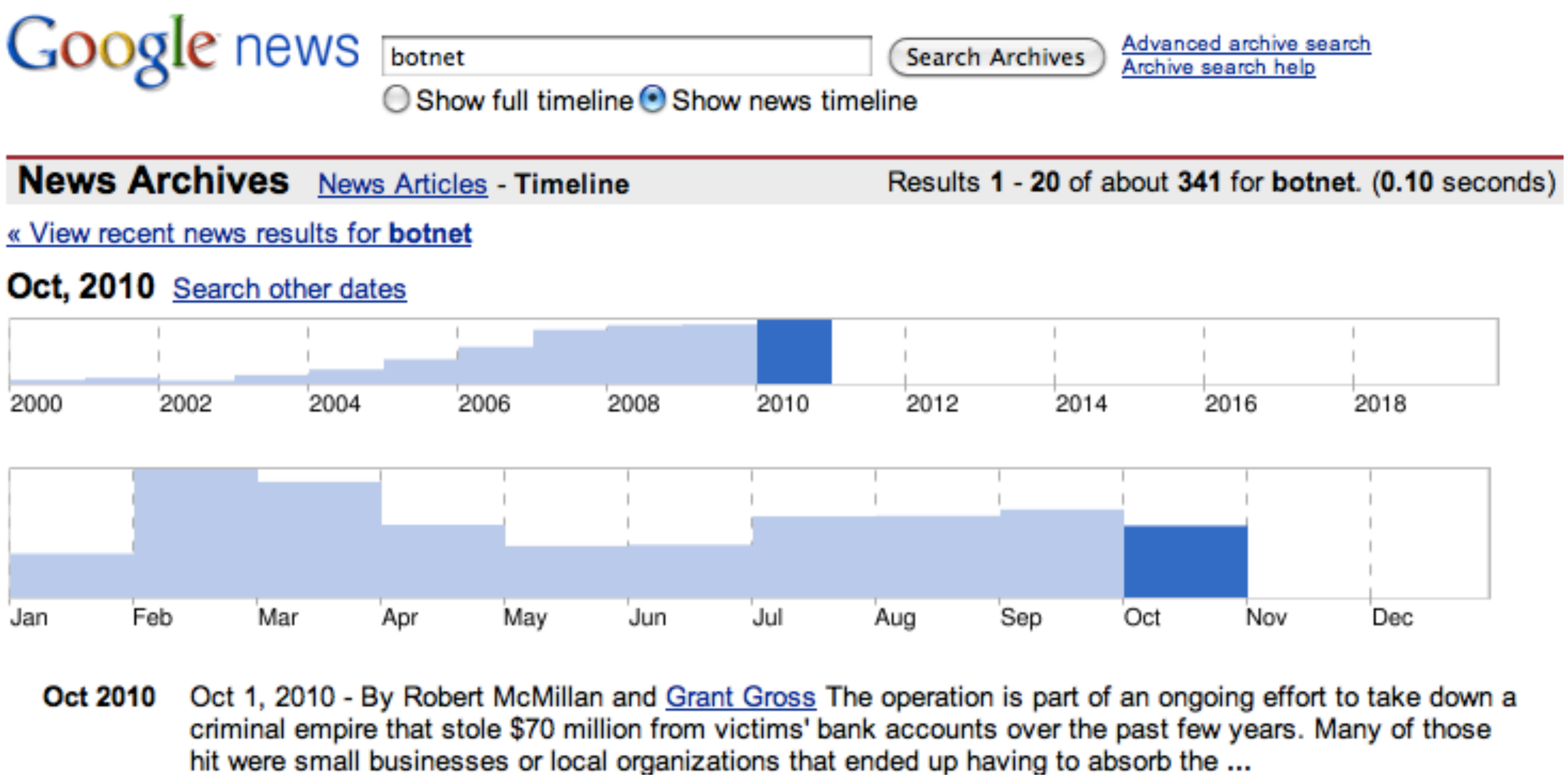
Detecting Coordinated Attacks with Traffic Analysis

Nikita Borisov, University of Illinois at Urbana-Champaign

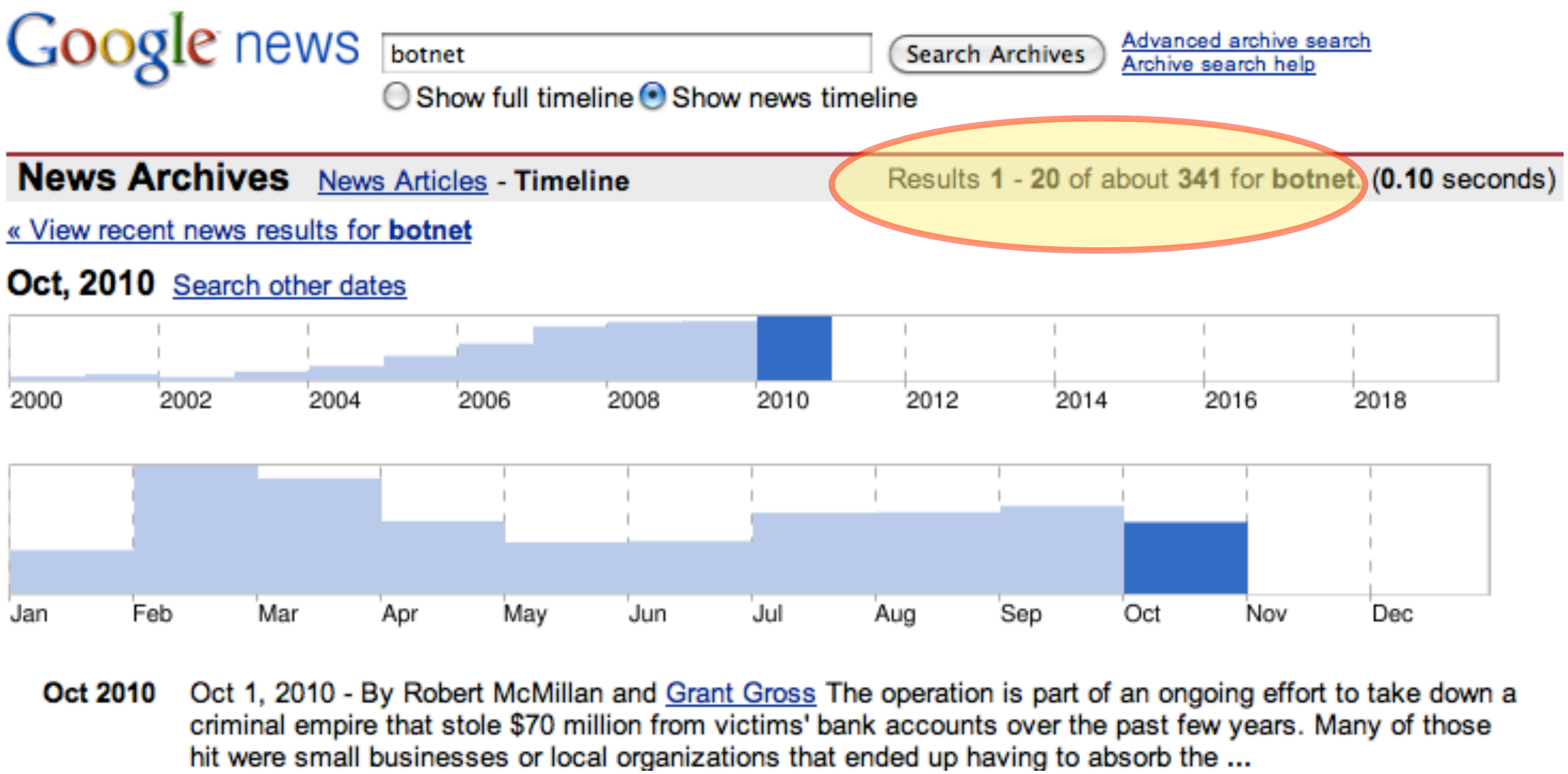
November 10, 2010

ILLINOIS

The Botnet Threat



The Botnet Threat



The Botnet Threat

The Botnet Threat

Caveat: see [Herley&Florencio, WIES'09]

The Botnet Threat

14
tweets

tweet

All 14
Influential 2

[Botnets Will Dominate Cyberattacks Through 2013: Gartner – www.esecurityplanet.com]

esecurityplanet.com/trends/article.php/3888911/Bo...l-Dominate-Cyberattacks-Through-2013-Gartner.htm - view page - cached

While not sophisticated, **botnets** are a resilient method for launching attacks against government and business, says **Gartner**.

Tweets about this link

RT @securitypro2009: **Botnets** Will Dominate Cyberattacks Through **2013 Gartner** www ... http://bit.ly/cD92kB
youwerewarned Jun 29, 2010 retweet

RT @securitypro2009: **Botnets** Will Dominate Cyberattacks Through **2013 Gartner** www ... http://bit.ly/cD92kB
inboxrevenge Jun 29, 2010 retweet

RT @securitypro2009: **Botnets** Will Dominate Cyberattacks Through **2013 Gartner** www ... http://bit.ly/cD92kB
cibercrime Jun 28, 2010 retweet

Botnets Will Dominate Cyberattacks Through **2013 Gartner** www ... http://bit.ly/cD92kB
securitypro2009 Highly Influential Jun 28, 2010 retweet

Detecting Botnets

- Traditional NIDS approaches
  - Signature-based
  - Anomaly detection
  - Protocol analysis

Detecting Botnets

- Traditional NIDS approaches
  - Signature-based
  - Anomaly detection
  - Protocol analysis
- Fail on modern attacks
  - Zero-day & polymorphic exploits
  - Hide anomalous activity in the crowd
  - Use encryption

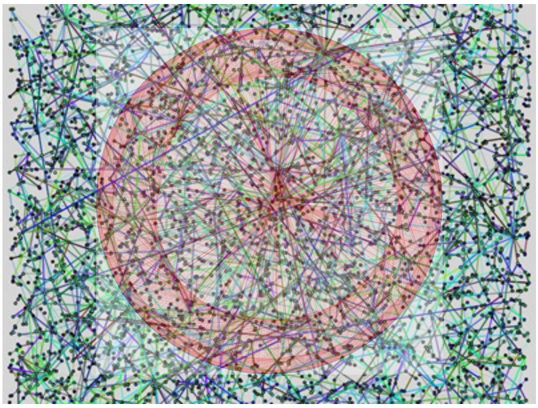
Traffic Analysis

- Analyze communication side information
  - Packet headers, sizes, counts, timings
- Derive useful information
  - Who is talking to whom
  - What kind of information traffic is carrying
  - Whether two communications are correlated

Traffic Analysis

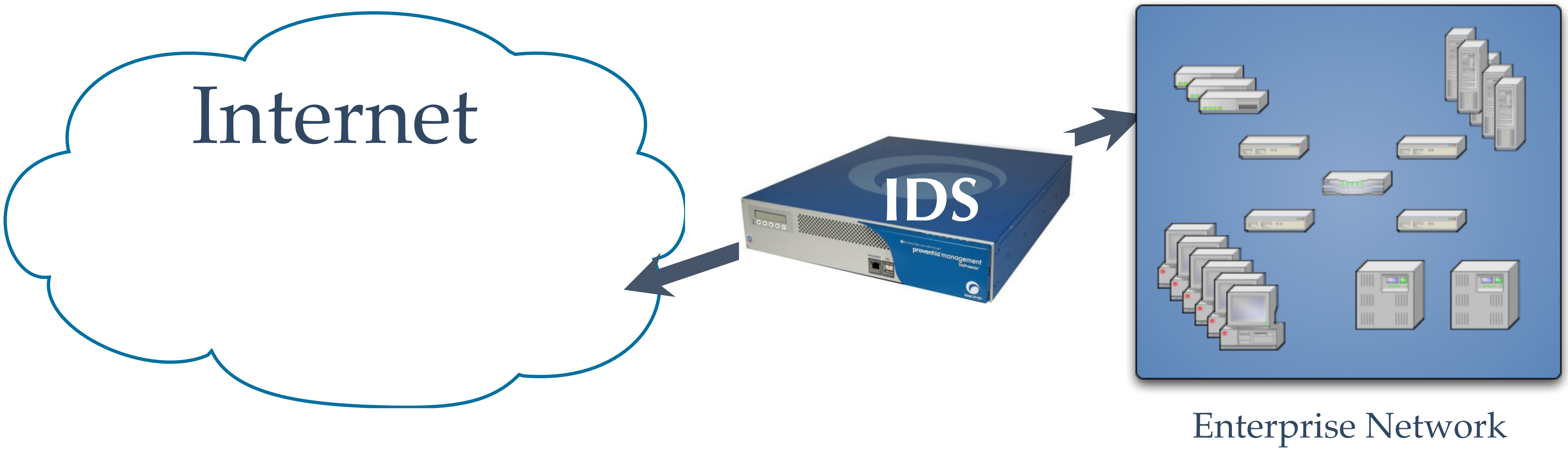
- History
  - Initially used by intelligence community (SIGINT)
  - Anonymous communication research
  - Security community

Outline

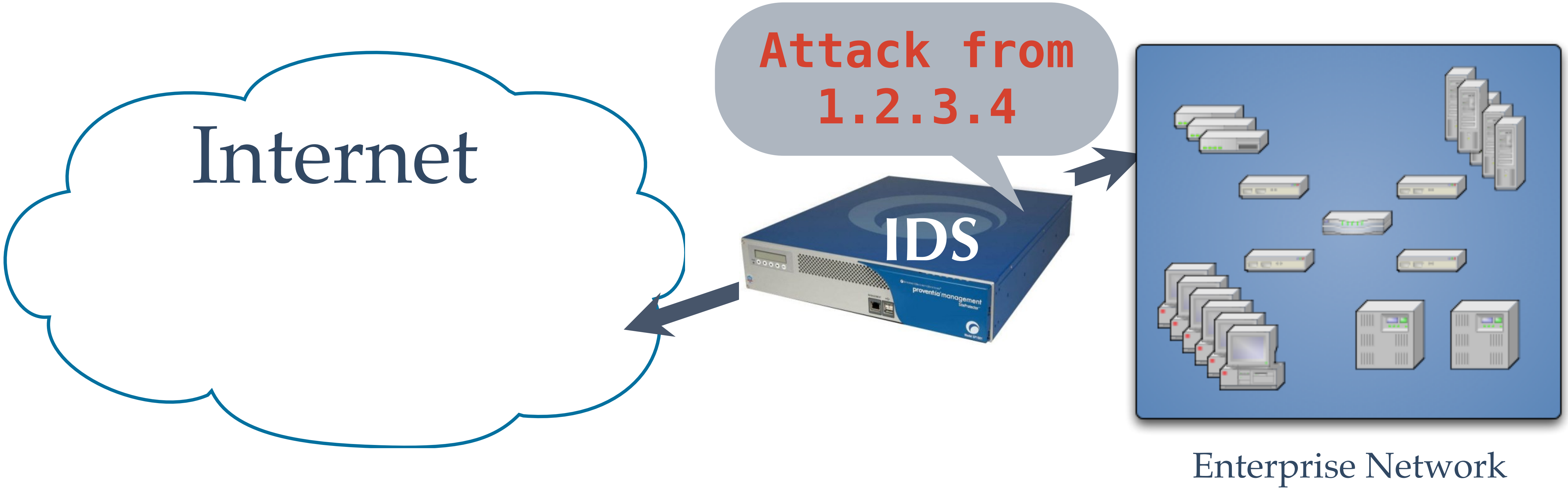


- Introduction
- Finding linked flows: stepping stones and watermarks
  - RAINBOW
  - SWIRL
- Detecting communities: isolating P2P botnets
  - BotGrep
- Conclusions

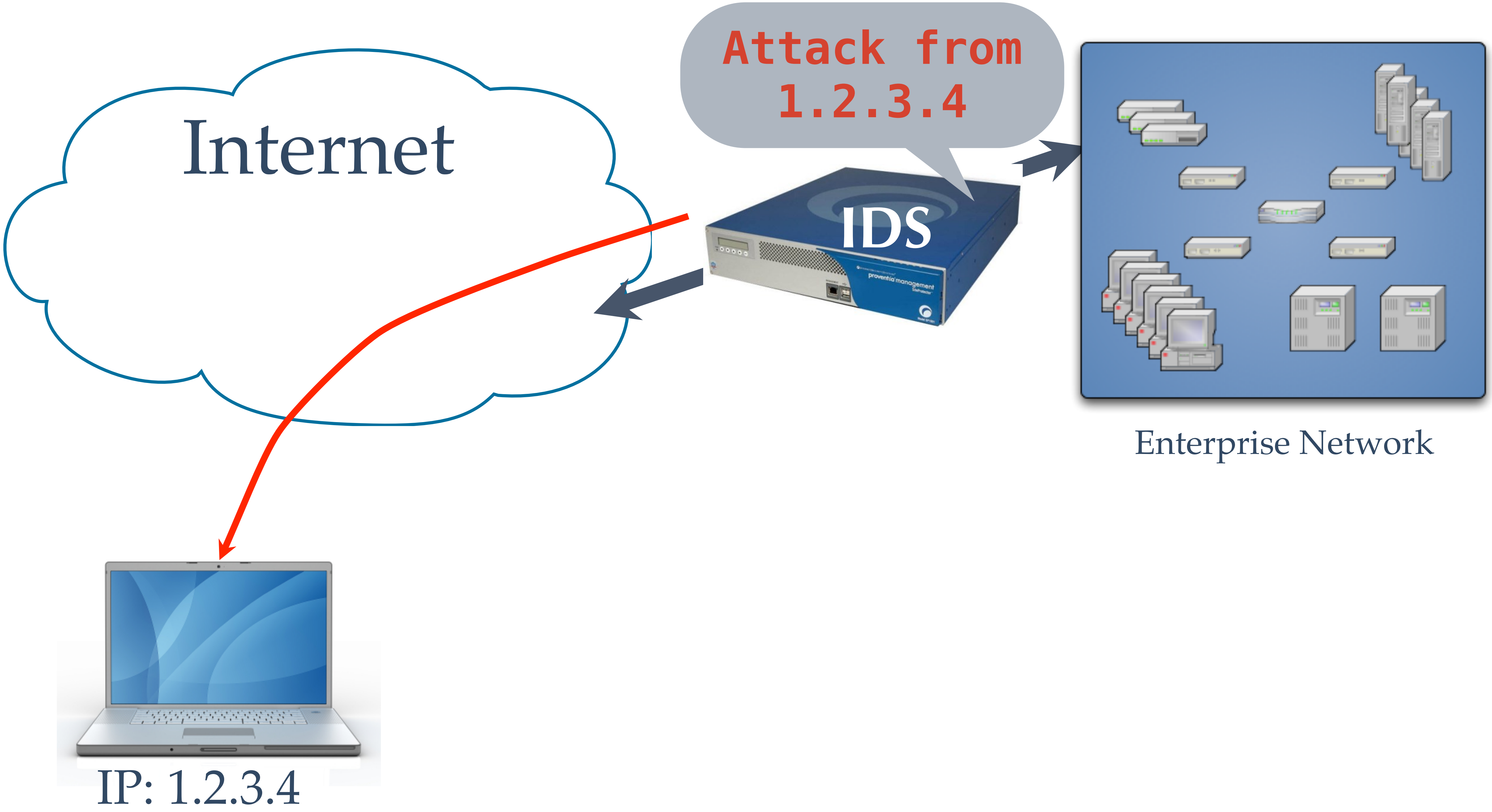
Attack Attribution



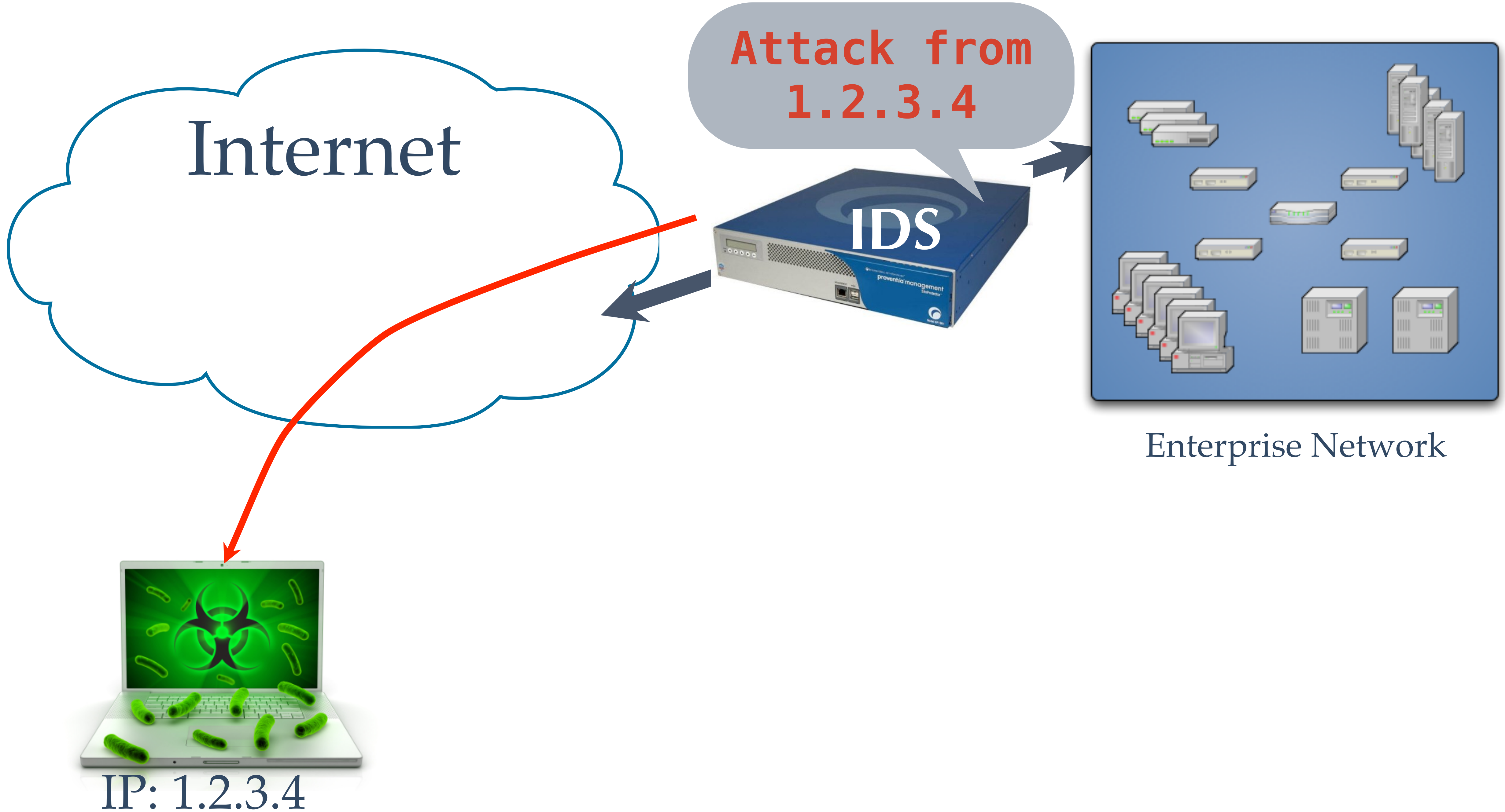
Attack Attribution



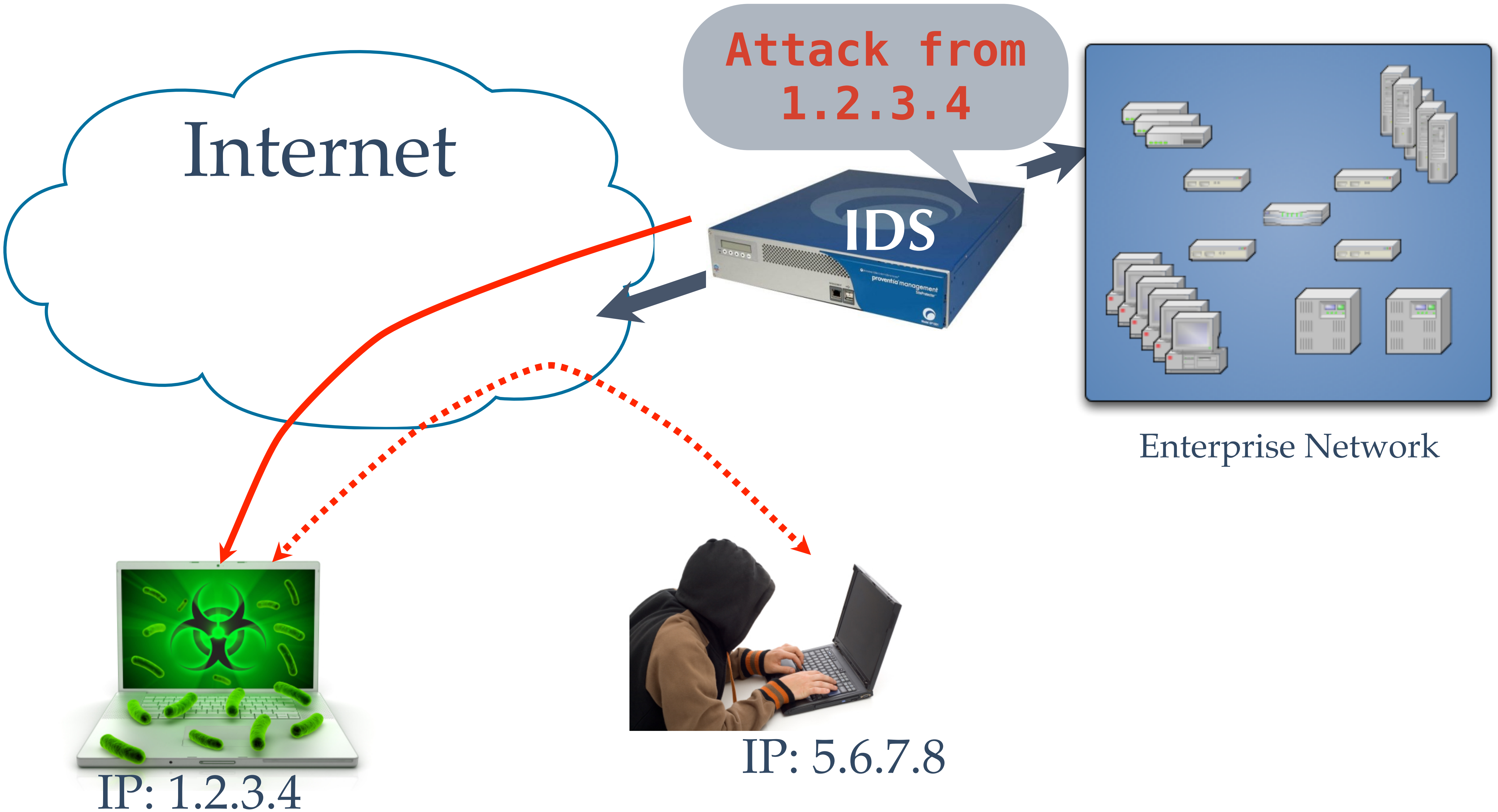
Attack Attribution



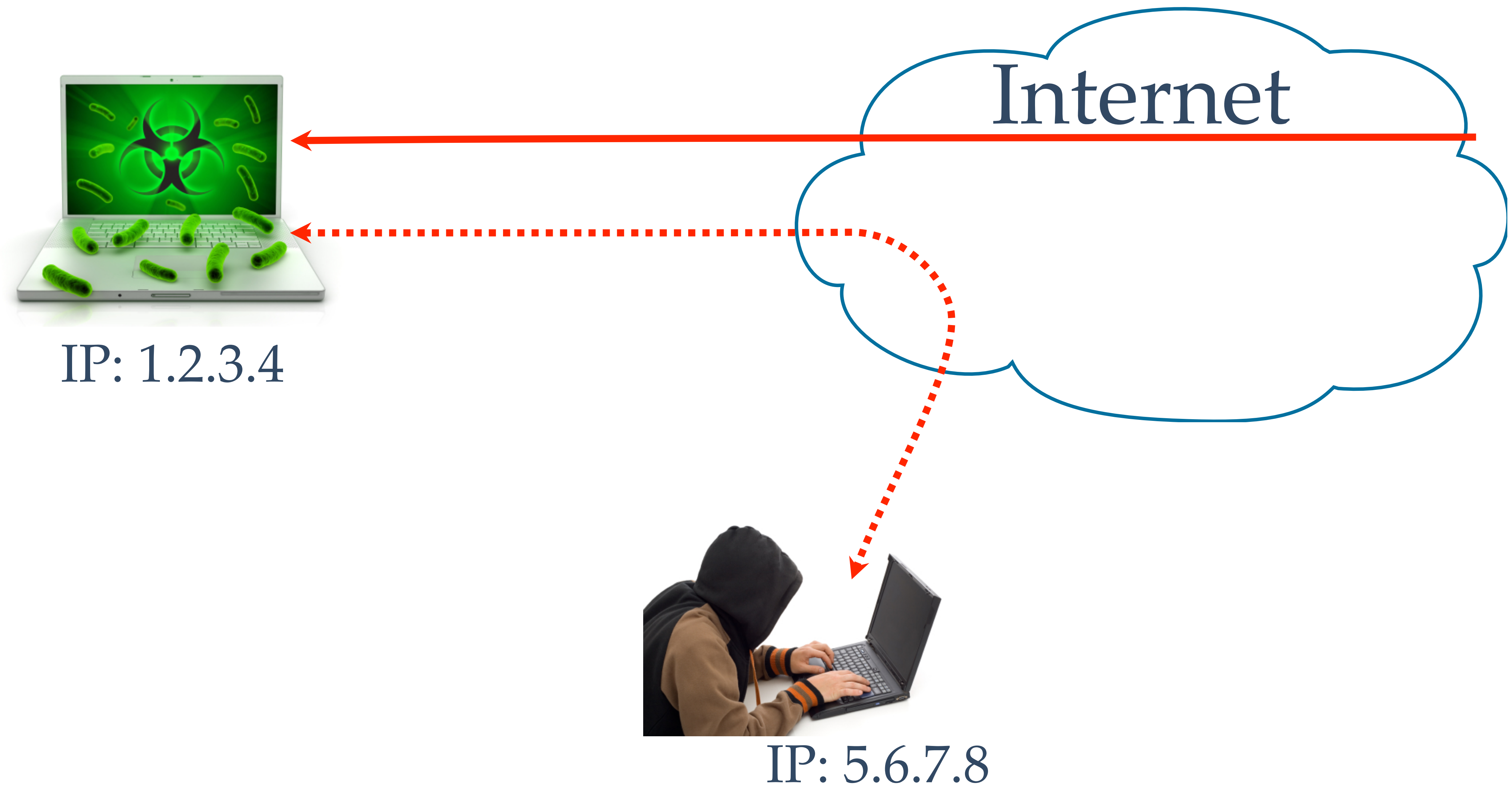
Attack Attribution



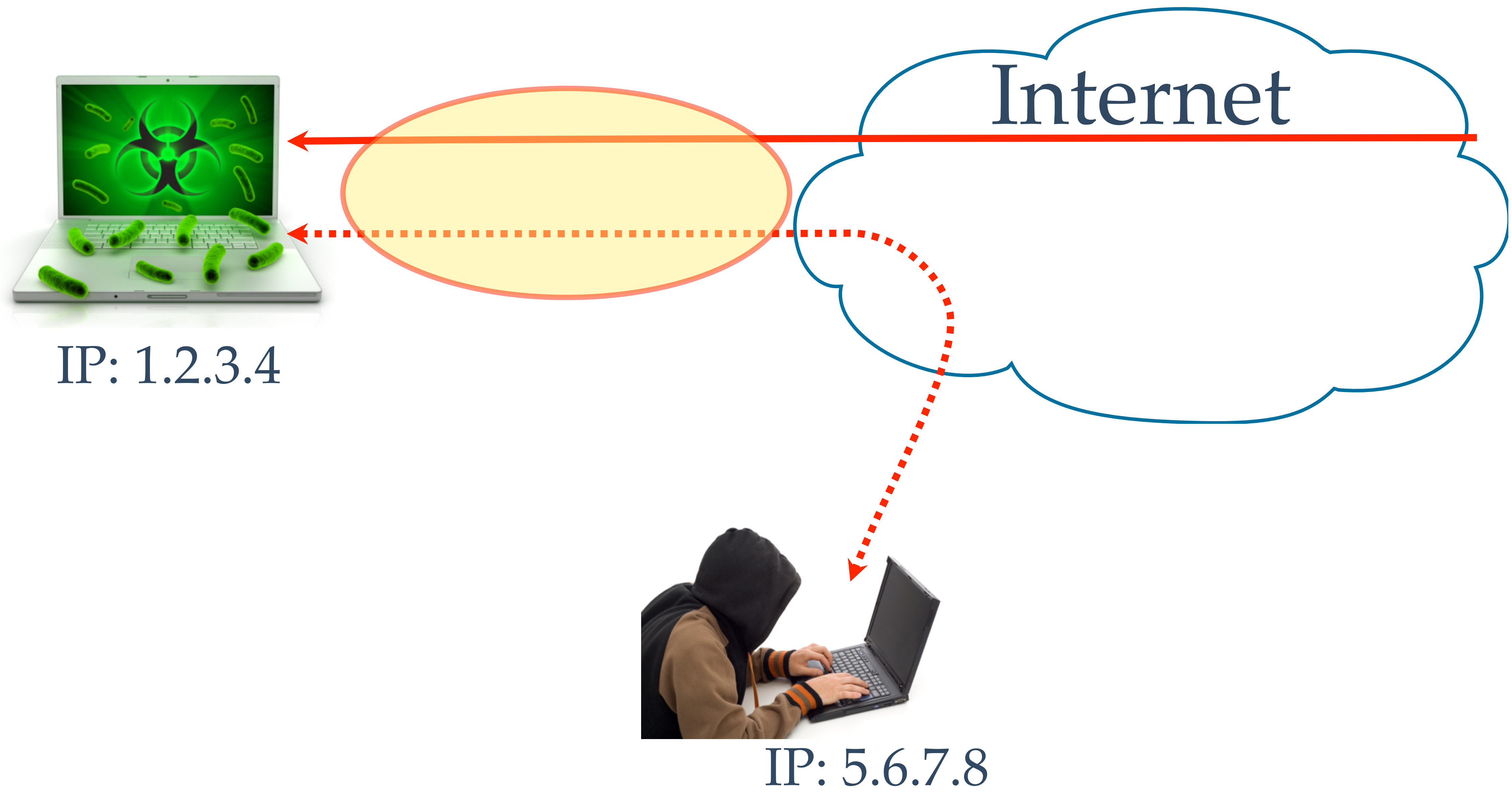
Attack Attribution



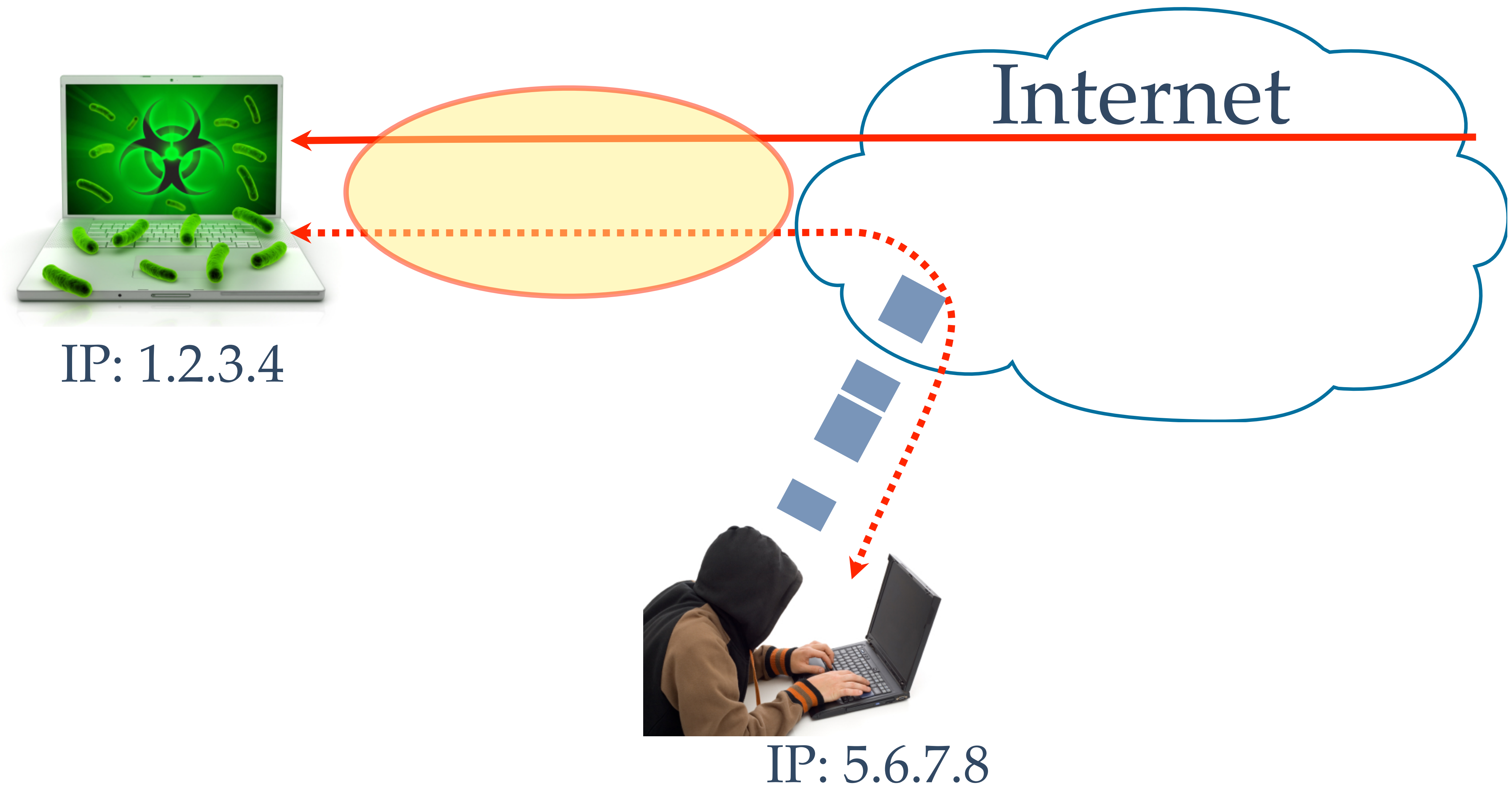
Stepping Stone Detection



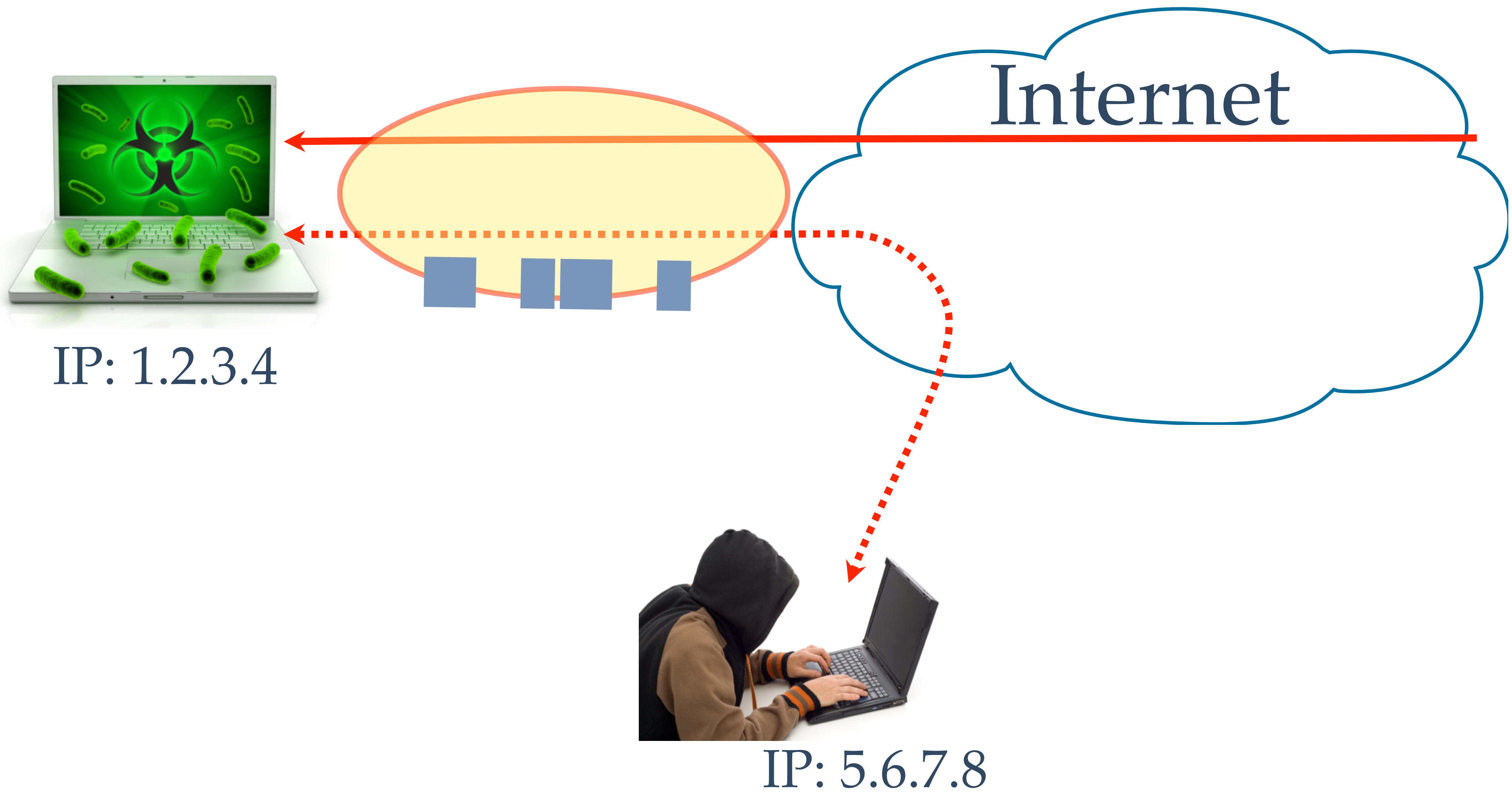
Stepping Stone Detection



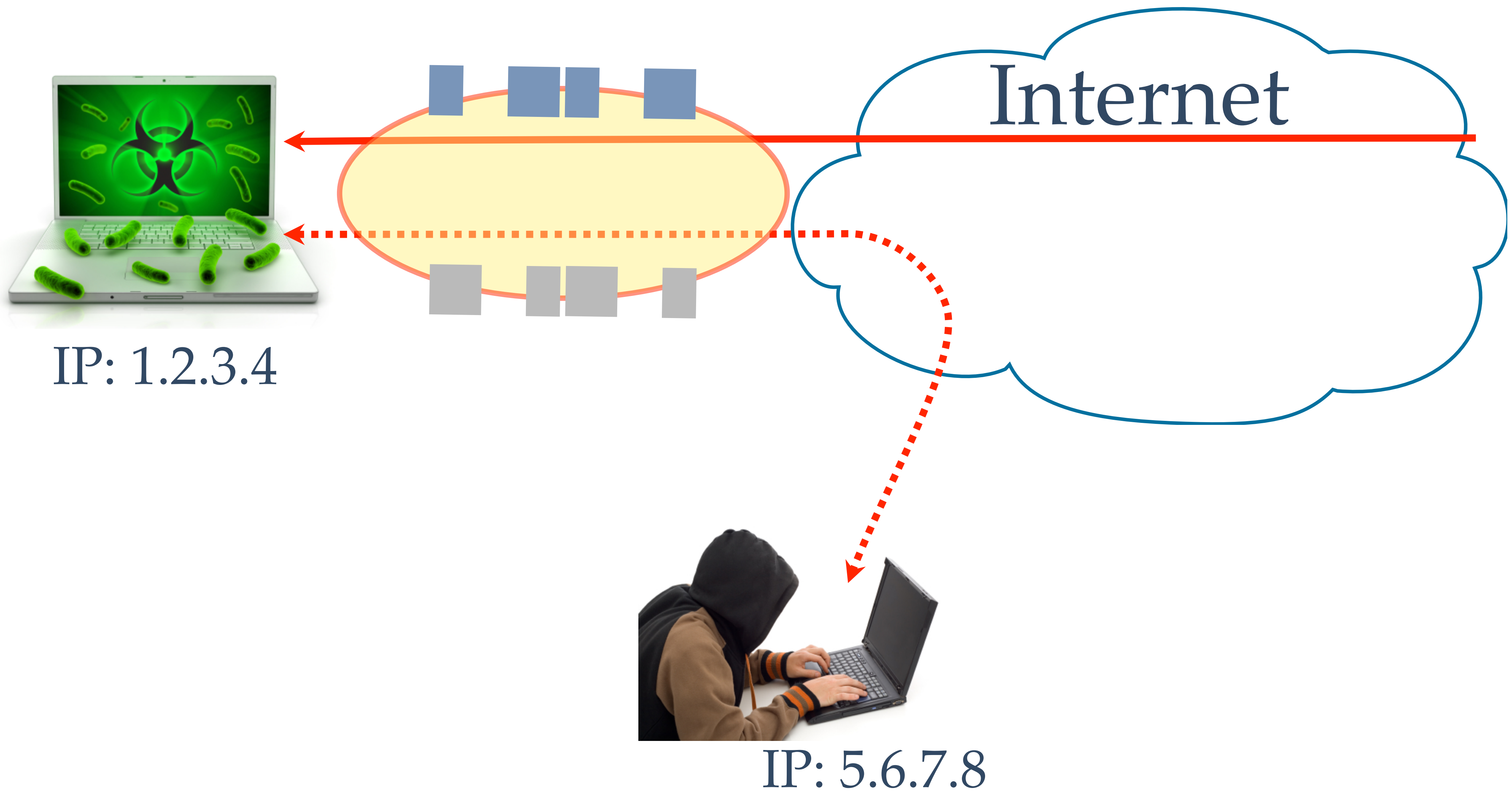
Stepping Stone Detection



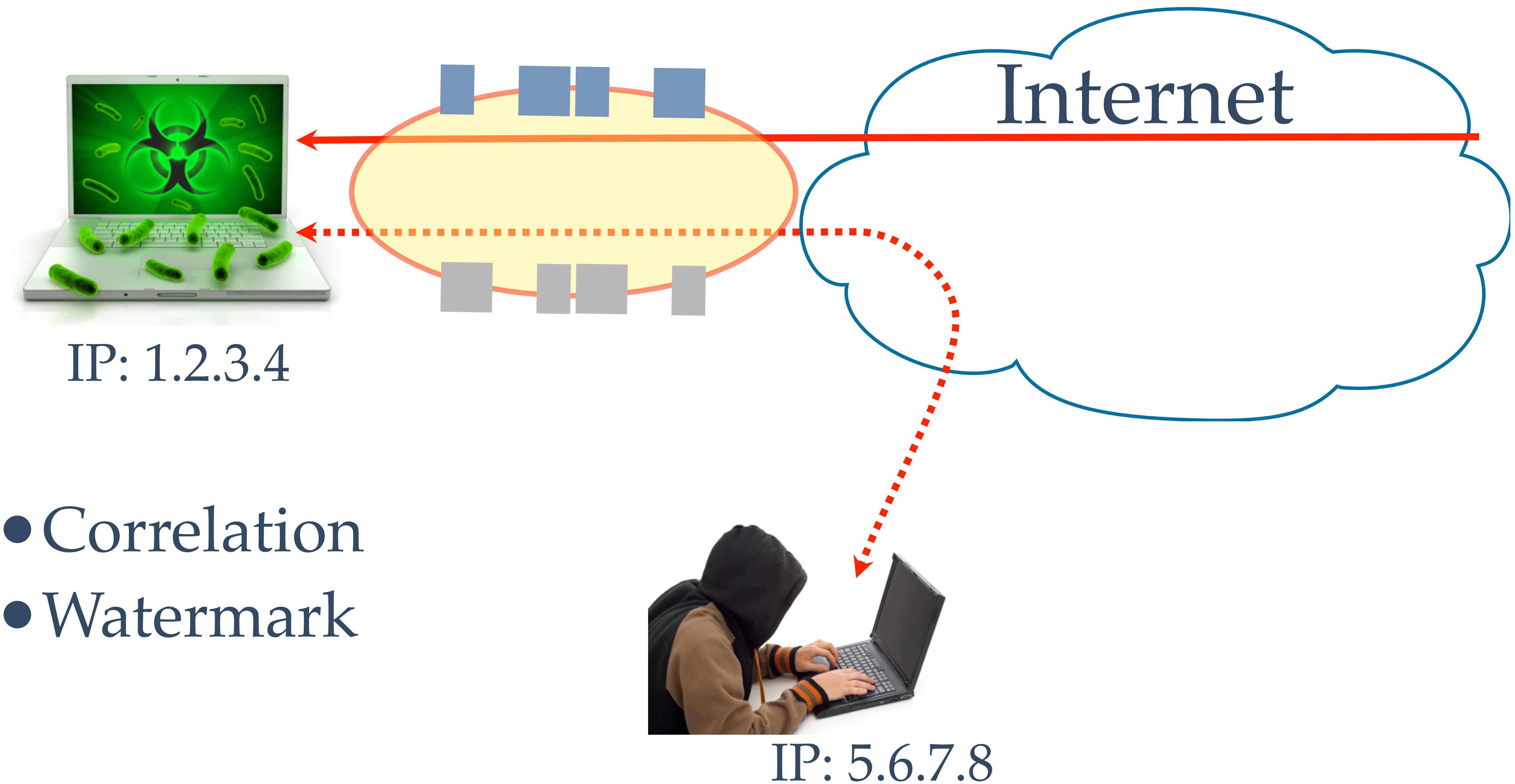
Stepping Stone Detection



Stepping Stone Detection



Stepping Stone Detection



- Correlation
- Watermark

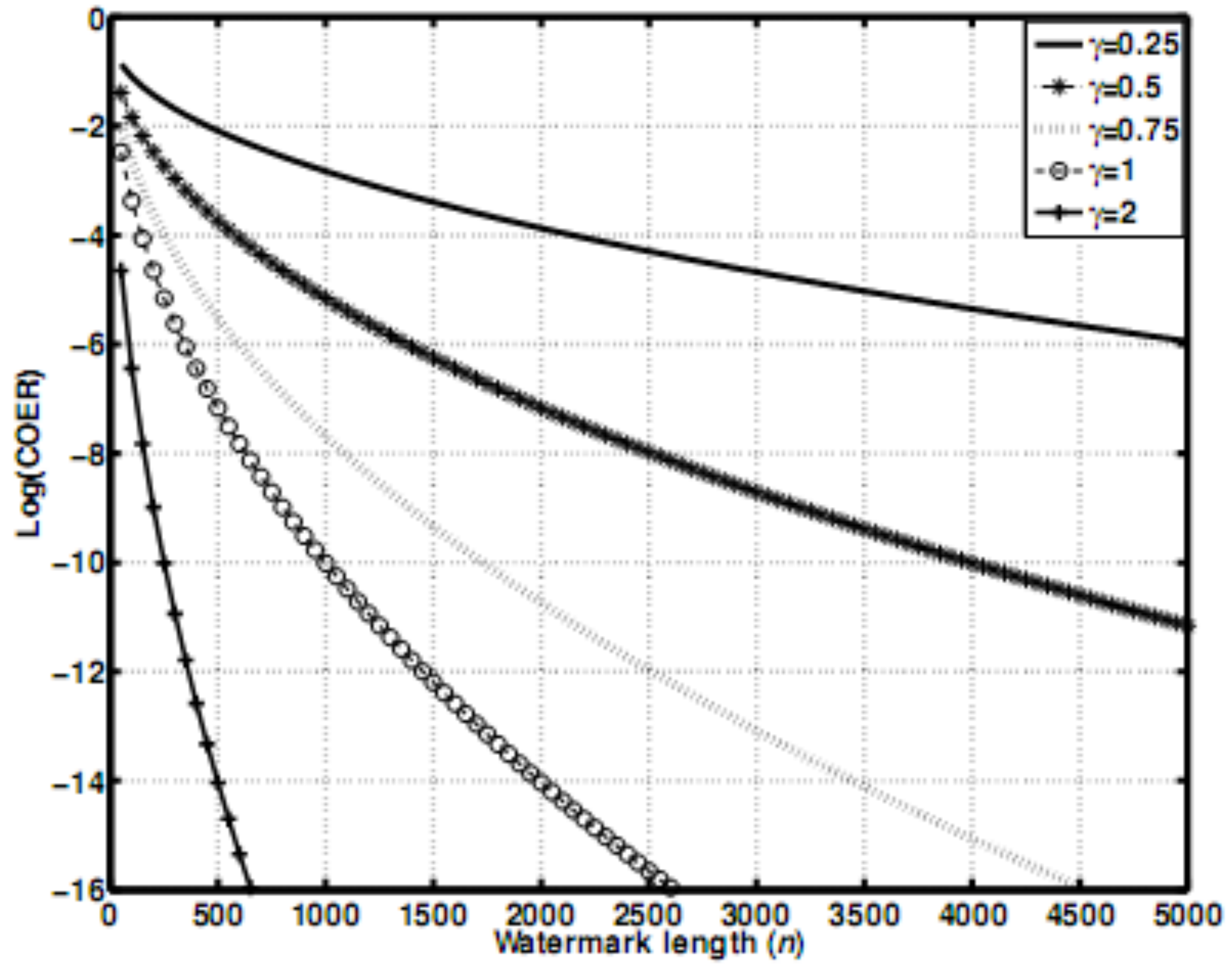
RAINBOW [NDSS'09]

- Goal:
  - *Low distortion* – do not interfere with legitimate users
  - *Invisibility* – resist detection
  - *Robustness* – survive network jitter, repacketization, …
- Non-goals:
  - *Active robustness* – survive attacker countermeasure
  - *Blind detection* – no communication b/w watermarker & detector

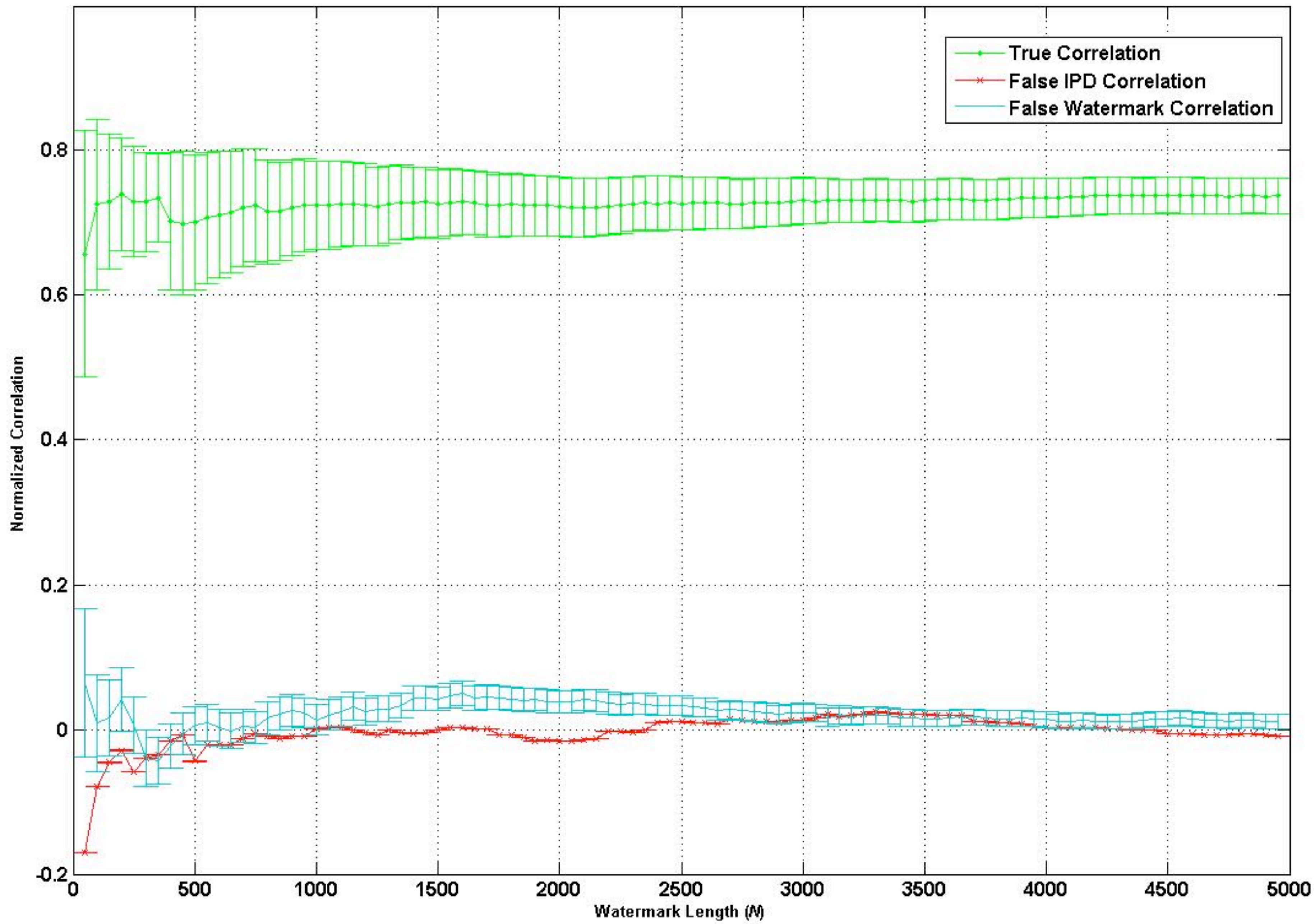
RAINBOW approach

- Generate a pseudo-random sequence
  - b = 0,1,0,0,1,1,1,0,1,0,...
  - Based on a secret seed *s*
- Adjust inter-packet delays by:
  - $w_i = +a$ if $b_i = 1$
  - $w_i = -a$ if $b_i = 0$
- Sequence is biased such that $$\forall i, \left| \sum_{j=1}^{i} (-1)^{b_i} \right| \leq 5$$
- *a* is comparable to network jitter (5-10ms)

Performance (analysis)



Implementation Results



RAINBOW Applications

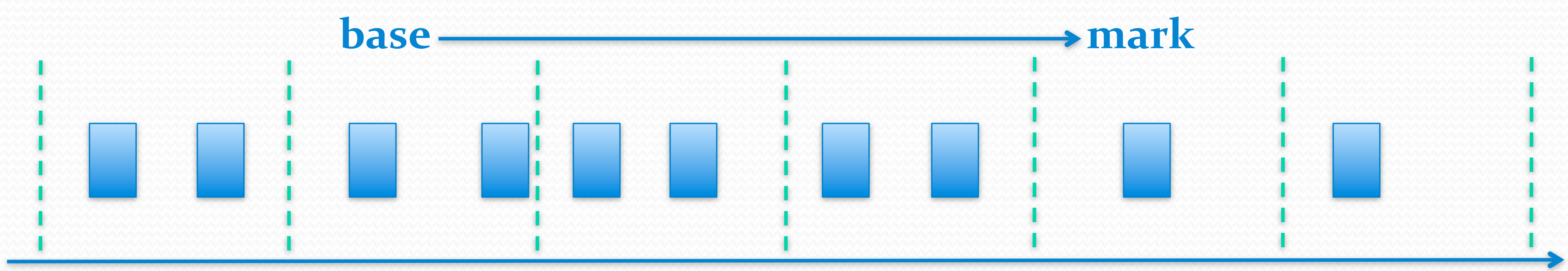
- Enterprise with one border gateway
  - IPD database can fit into a few GB of RAM
- Enterprise with several border gateways
  - Must communicate IPDs (in near real-time)
  - Overhead can be high
- Large ISP?
  - Not practical

SWIRL

- **Blind** watermark
- Approach:
  - Interval-based watermarks for blind detection, robustness
  - Data-dependent watermark to avoid multi-flow attacks

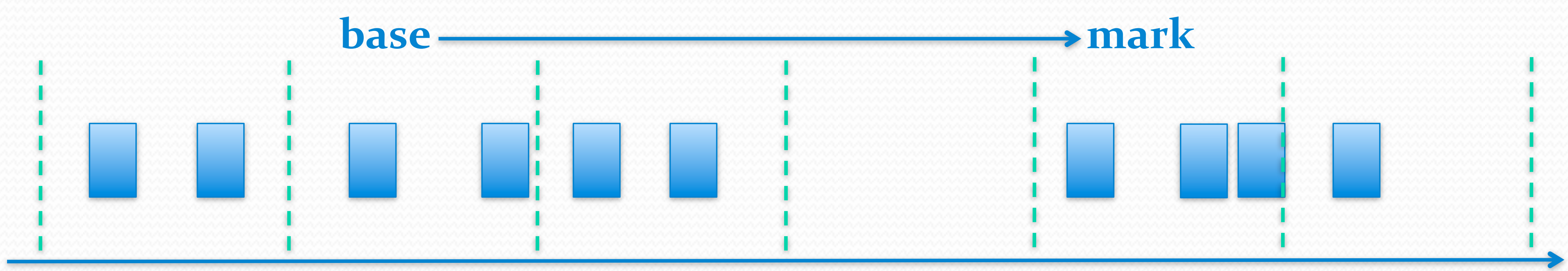
Set up

- For each watermark bit, define *base* and *mark* intervals
- Base interval is left unmodified
- Mark interval is watermarked, using a pattern dependent on base interval



Set up

- For each watermark bit, define *base* and *mark* intervals
- Base interval is left unmodified
- Mark interval is watermarked, using a pattern dependent on base interval



Base interval

- Compute packet *centroid*

$$C = \frac{\sum t_i - t_0}{n}$$

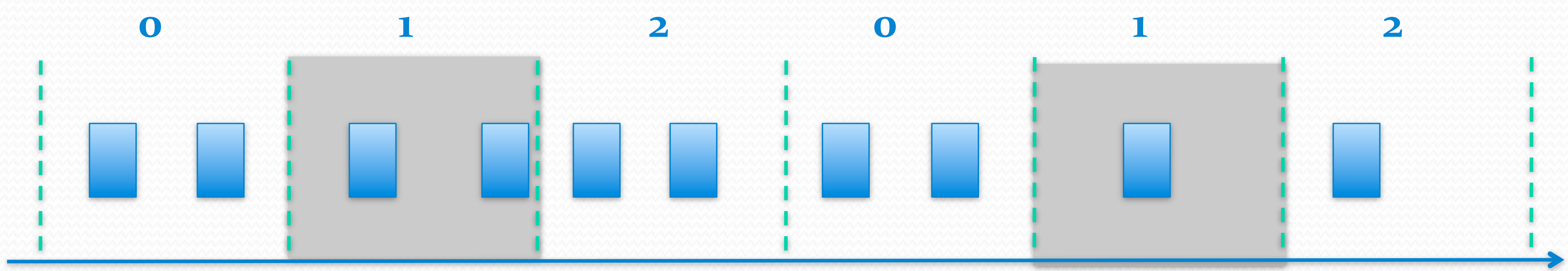
- Convert it to a number in [0,n-1]

$$b = \lfloor CDF(C) \rfloor * n$$

- Use shared secret key *s* to compute mark index *m*
  - $m = b*s \pmod{n}$
  - $\gcd(s,n) = 1$

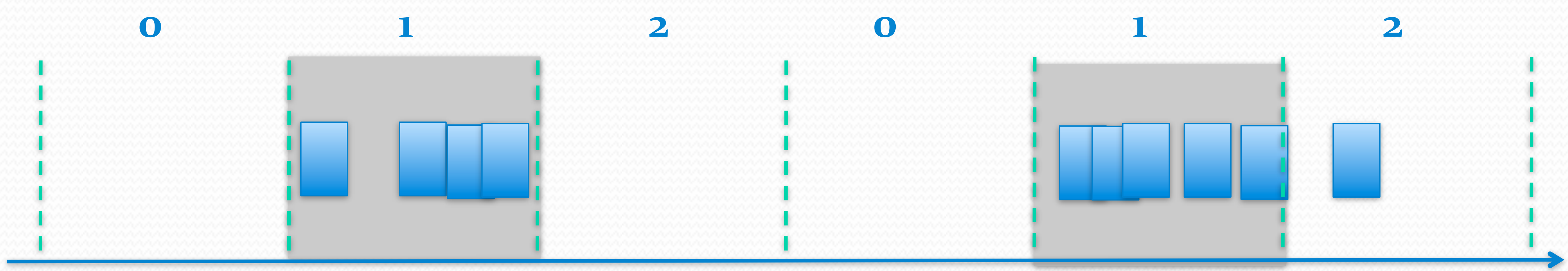
Mark Intervals

- Split into *r***n* sub-intervals
- All traffic moved to intervals $k = \underline{m} \pmod{n}$
  - Note: only works with sparse traffic



Mark Intervals

- Split into $r*n$ sub-intervals
- All traffic moved to intervals $k = \underline{m} \pmod{n}$
  - Note: only works with sparse traffic

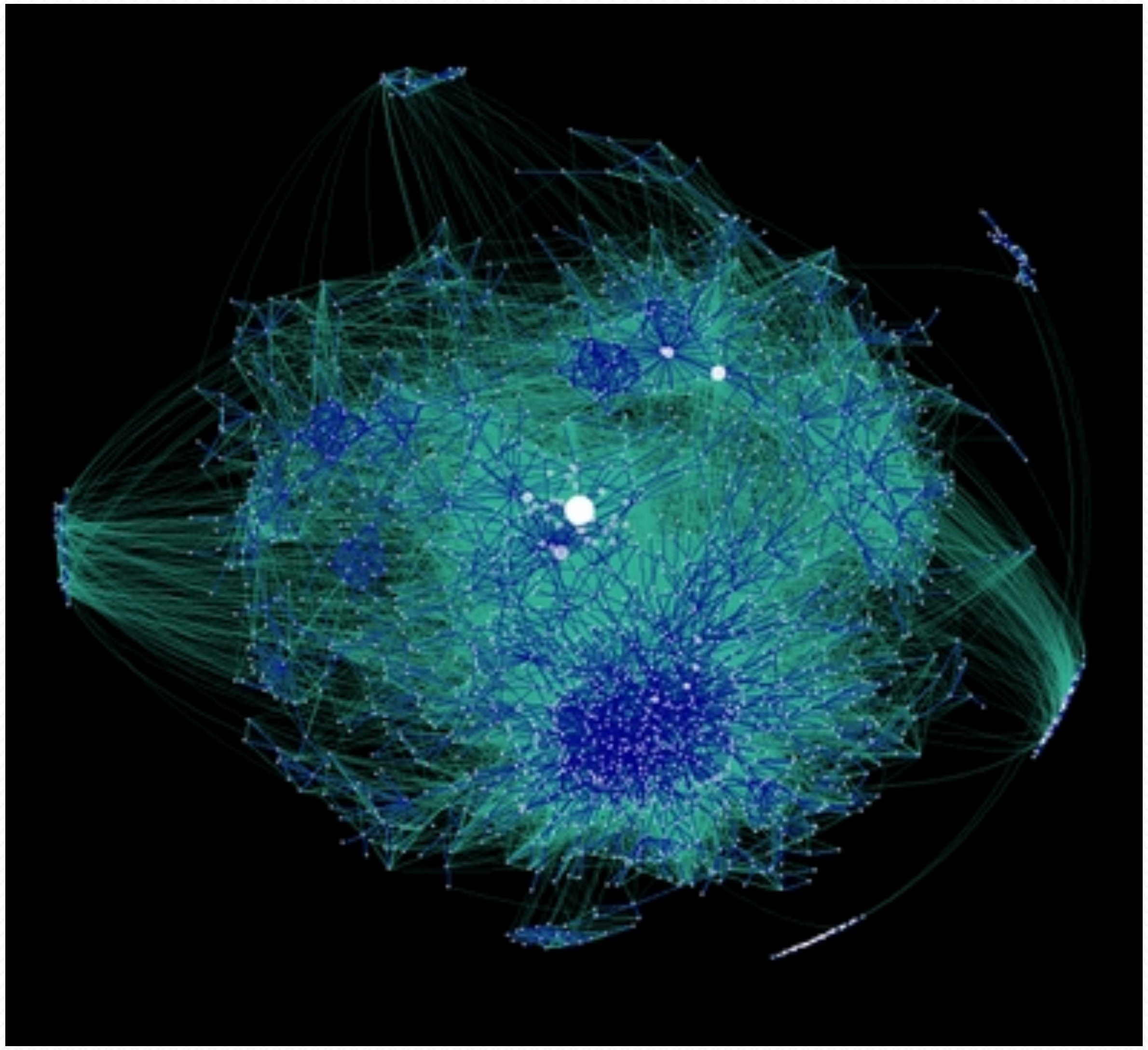


Comparison with RAINBOW

- Similar:
  - Resilience to robustness
  - Error rates
- Different:
  - Needs larger flows to watermark (about 10x)
  - Much faster detection ($O(n)$ instead of $O(n^2)$)
  - $O(1)$ communication

Botnets

- Coordinated attack platforms
- Thousands to millions(?) of nodes
- Source of most spam
- Also DDoS, ...



Botnet detection

- Misuse detection
  - 0-day, partitioned misuse
- Anomaly detection
  - Each individual bot can fly under the radar
- Clustering
  - Find similar, suspicious behavior among hosts
- **Communication**

P2P Communication

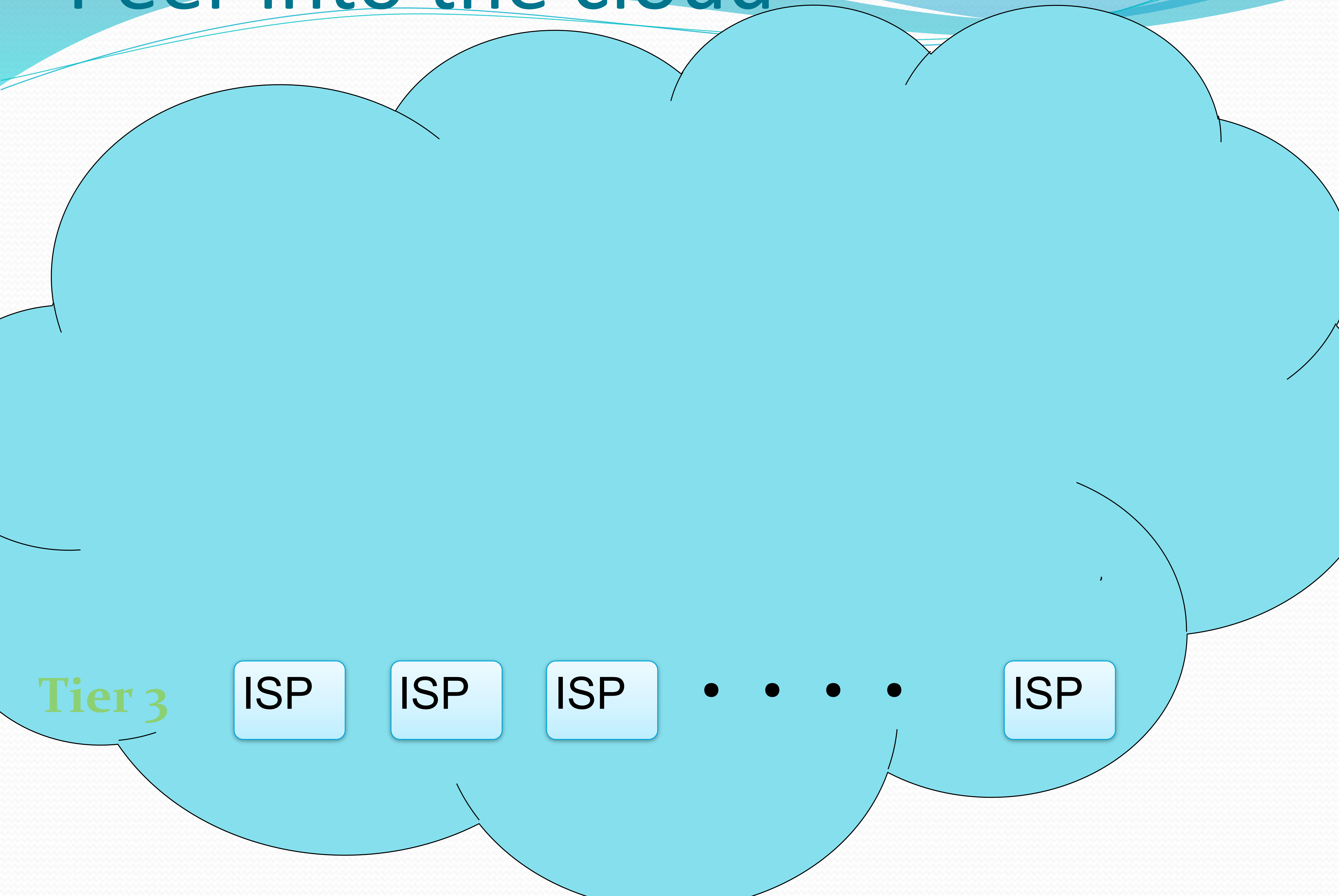
- Botnets are going P2P
  - No central nodes to find, attack
  - Efficient communication from any point to another
  - Resilient to churn
- Structured P2P networks (e.g., Chord, Kademlia)
  - Low node degree
  - High expansion
- Detection
  - Local behavior unremarkable
  - Global communication patterns detectable

Peer into the cloud

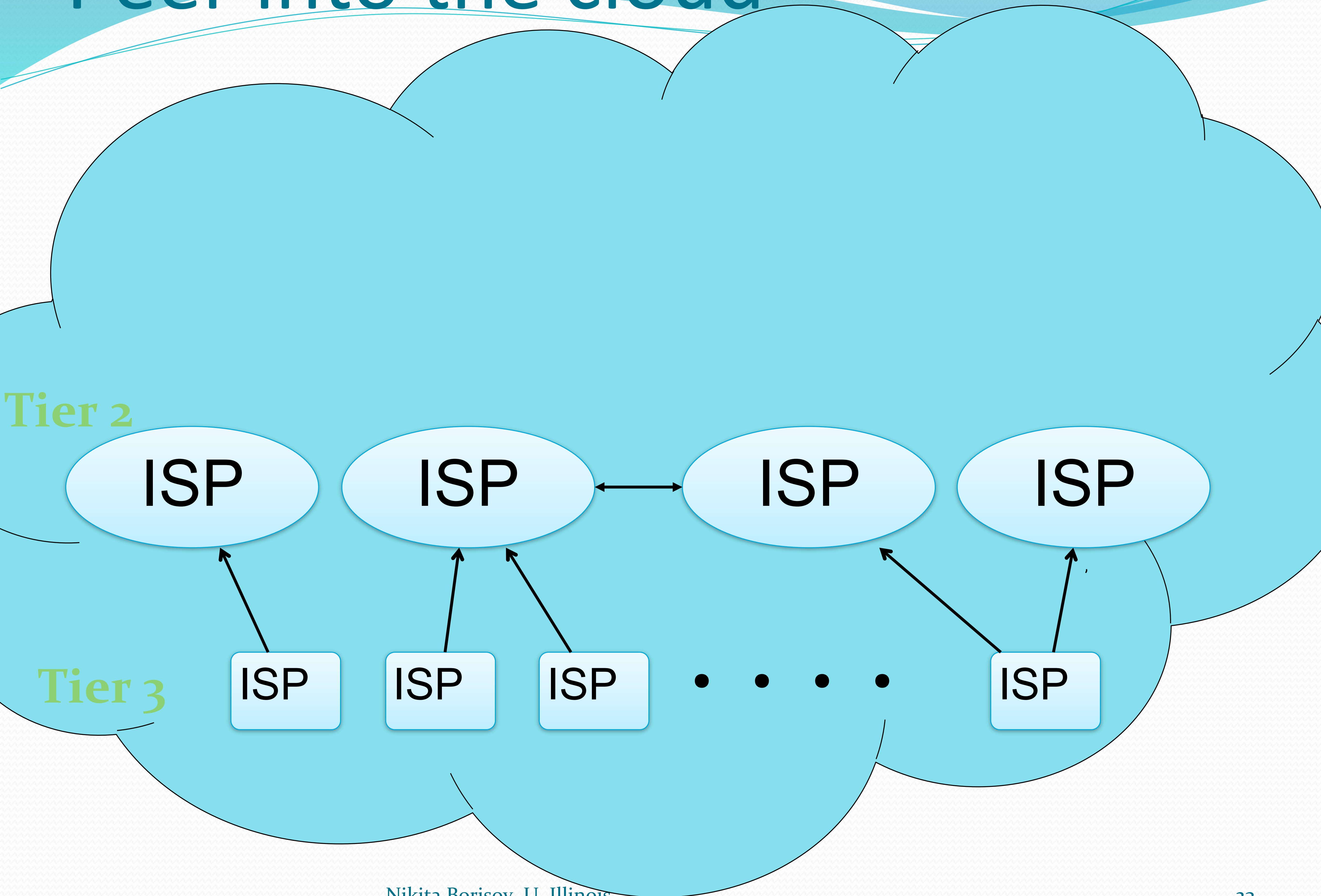
,

Peer into the cloud

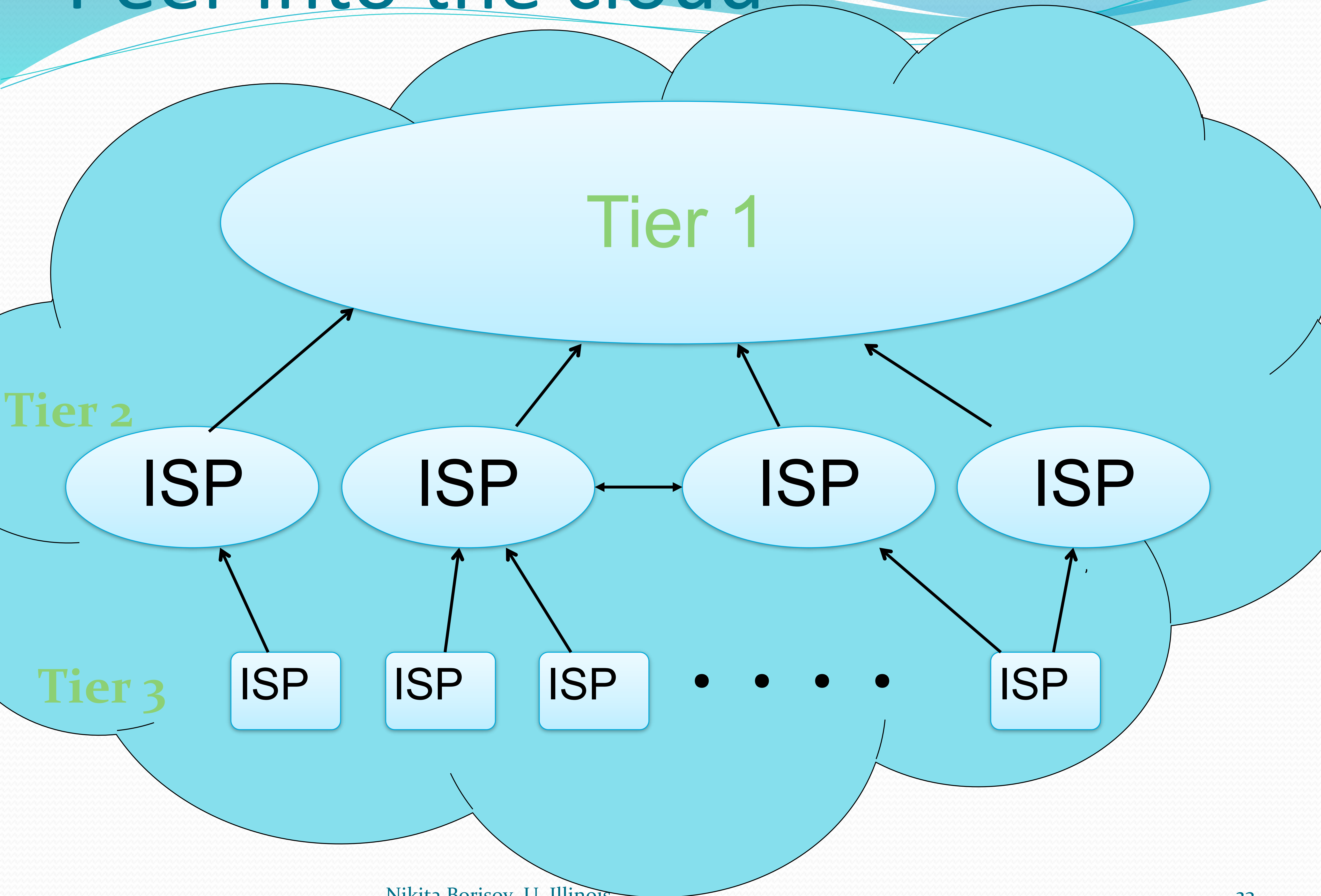
Peer into the cloud



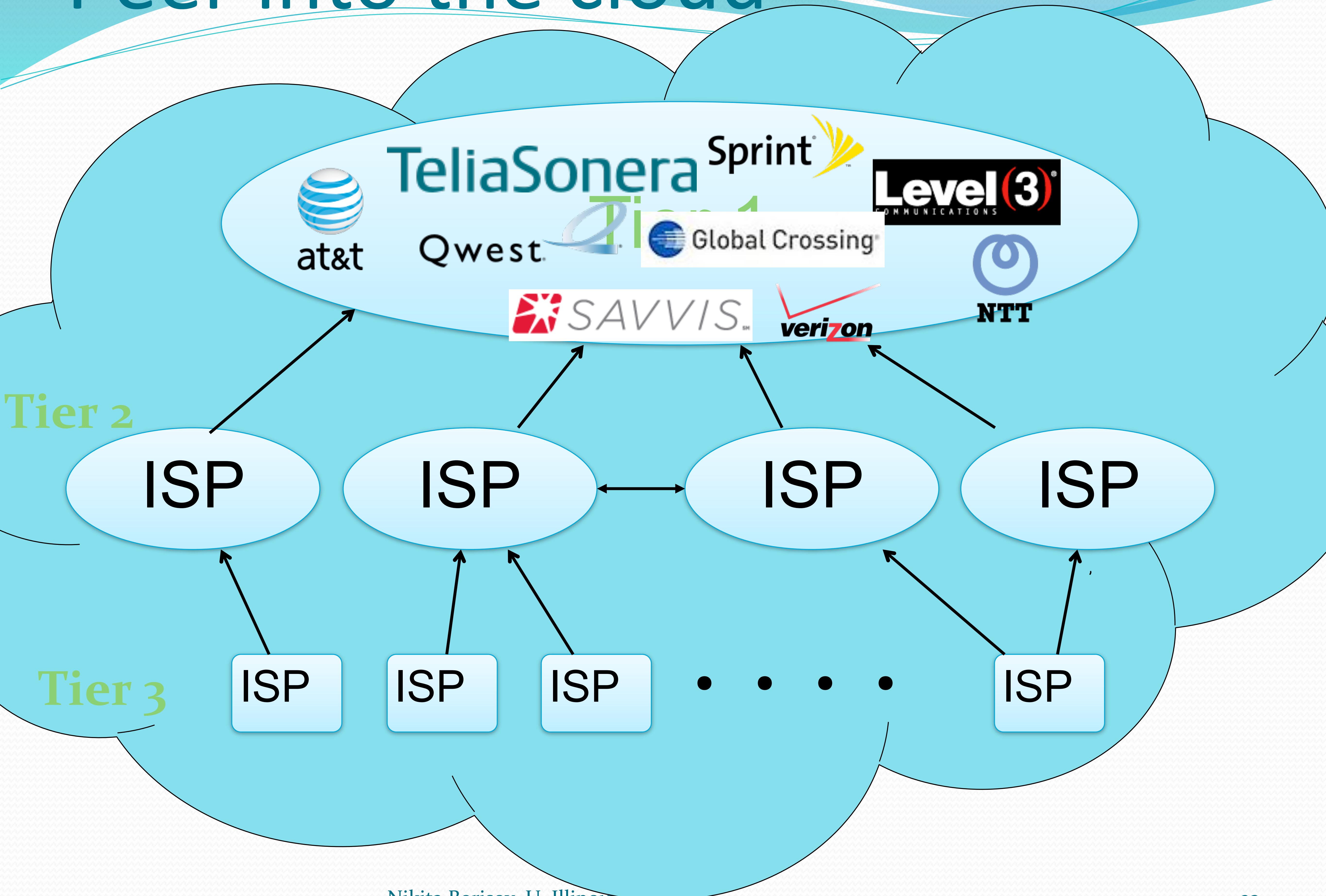
Peer into the cloud



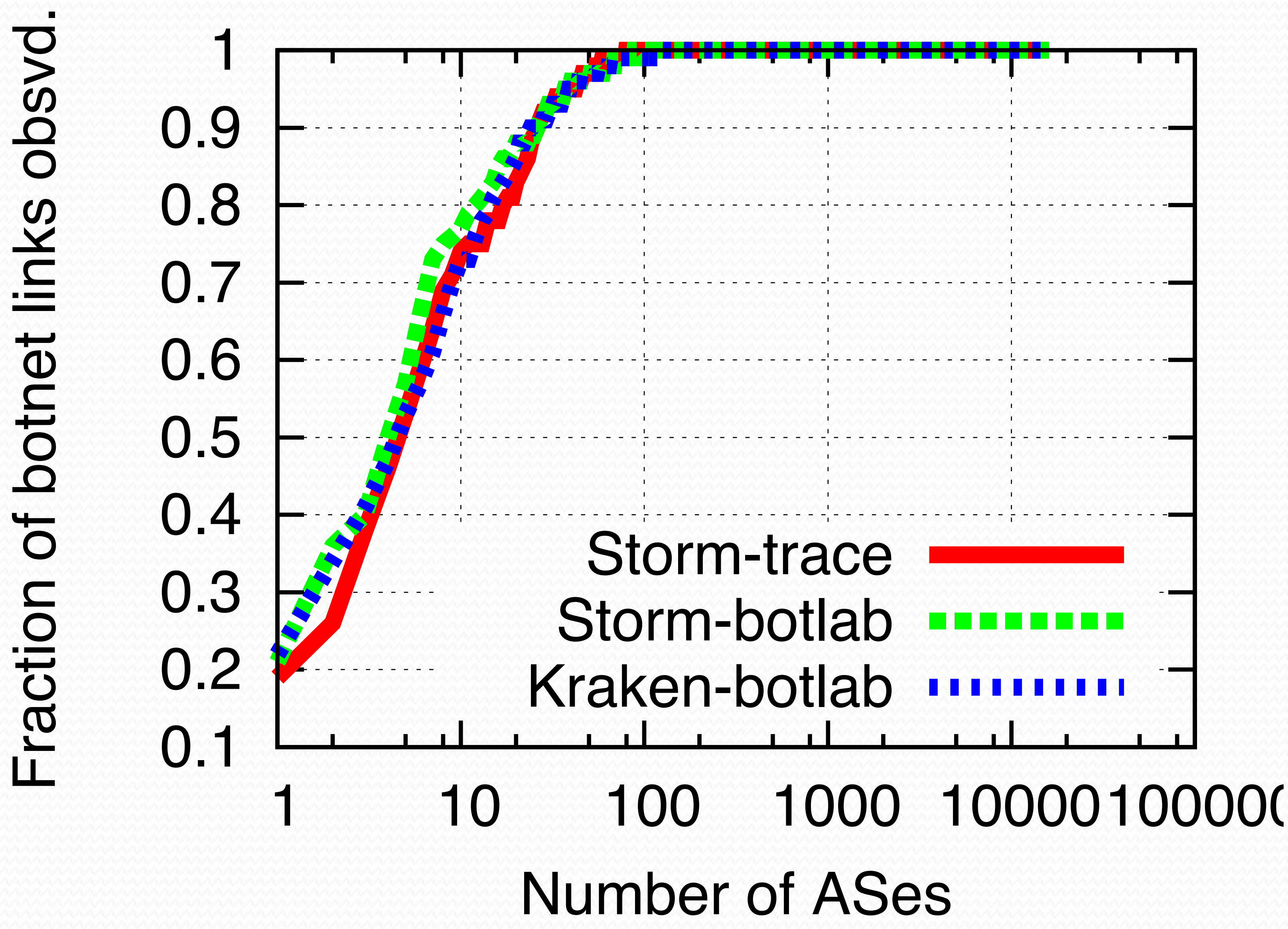
Peer into the cloud



Peer into the cloud



ISP visibility



Mixing Times

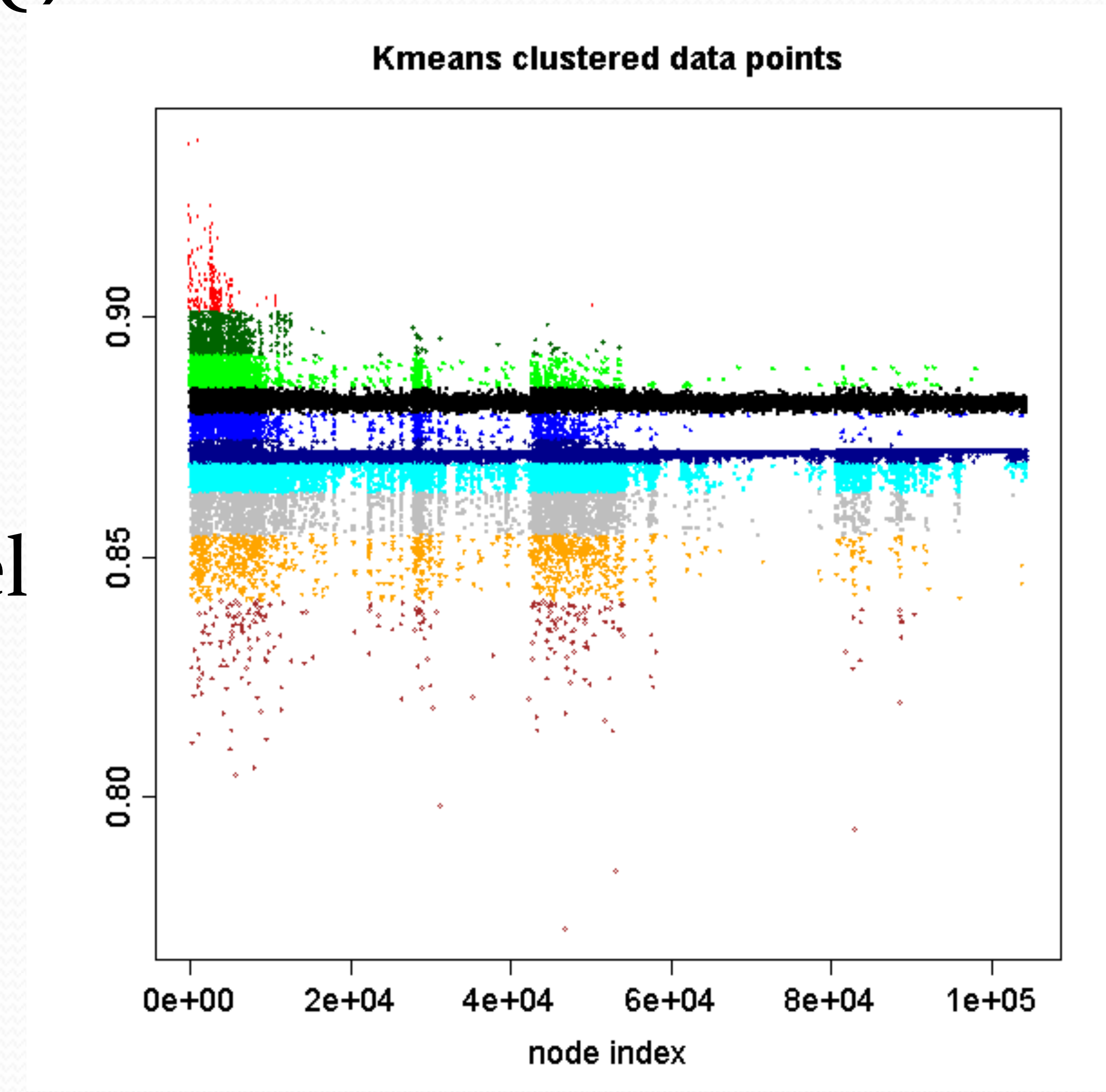
- Can model random walk on a graph as Markov chain
  - p_i = probability of being at node I
  - **p**' = **Tp**, where **T** is the Laplacian of the graph
  - $\lim_{n \to \infty} T^n p = \pi$ – stationary distribution
- Mixing time: speed of convergence to stationary distribution
  - (most) Structured P2P networks have fast mixing times
  - Mixing time related to conductance, bisection width (lack of a small cut) – desirable properties

Graph Search

- Goal: find subgraph G' of G that is fast-mixing
  - |G'| = 1K to 1M, |G| = 100M+
  - Must use sampling
- Initial pre-filtering step
  - Clustering of similar nodes (regular patterns)
- Recursive partitioning
  - Low-conductance cuts
  - Use Markov-chain Monte Carlo sampling techniques from SybilLimit [NDSS'08]

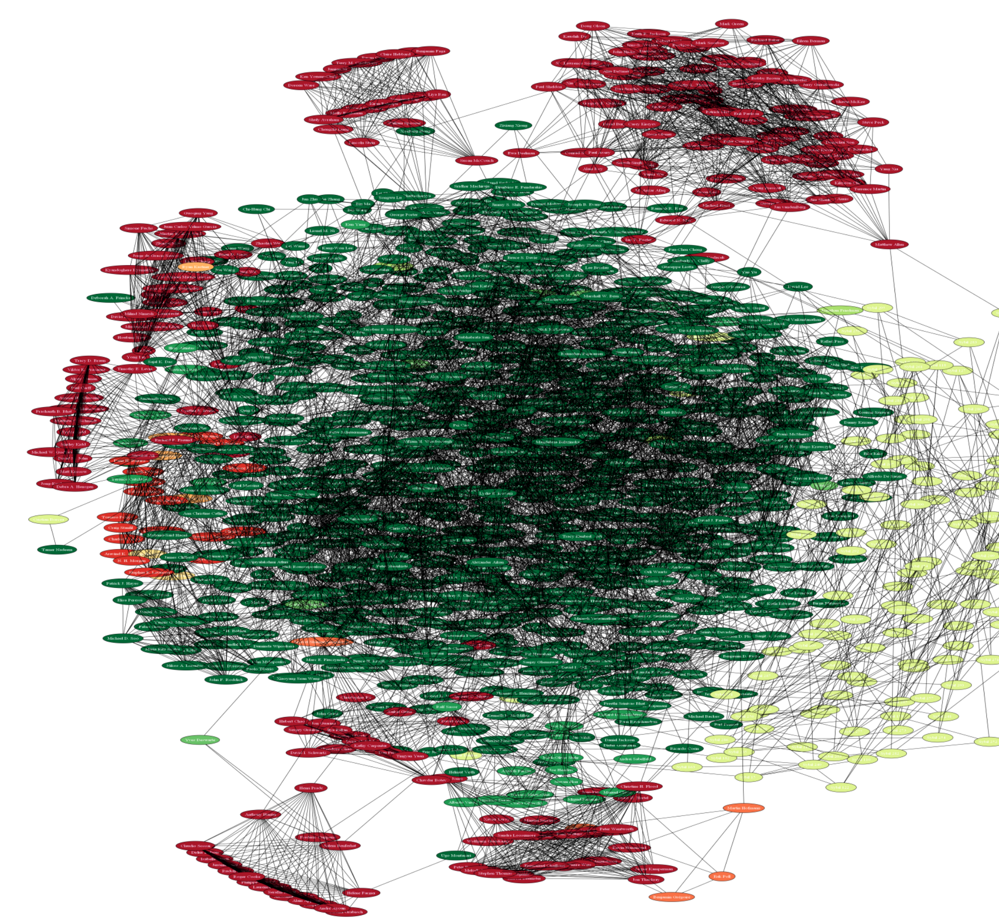
Pre-filtering

- Perform short random walk (O (log n))
- Cluster nodes by probability
  - Normalize by node degree
  - Use k-means (or X-means)
- Analyze each cluster separatel



Partitioning

- Find a low-conductance cut
  - Sample partitions X and X^c with $|X| > |X^c|$
  - Use MCMC to find low-conductance cut
    - $P(X, X^c) = \text{conductance}(X, X^c)$
    - Generate samples according to P
  - Find marginal probability v in X^c
  - Make cut based on probability thre



Validation Tests

- Heuristics to decide when we're done
  - Conductance of cut too high
- Heuristics to decide if partition is P2P
  - Degree homogeneity
  - Fast mixing

Results, Tier-1 ISPs

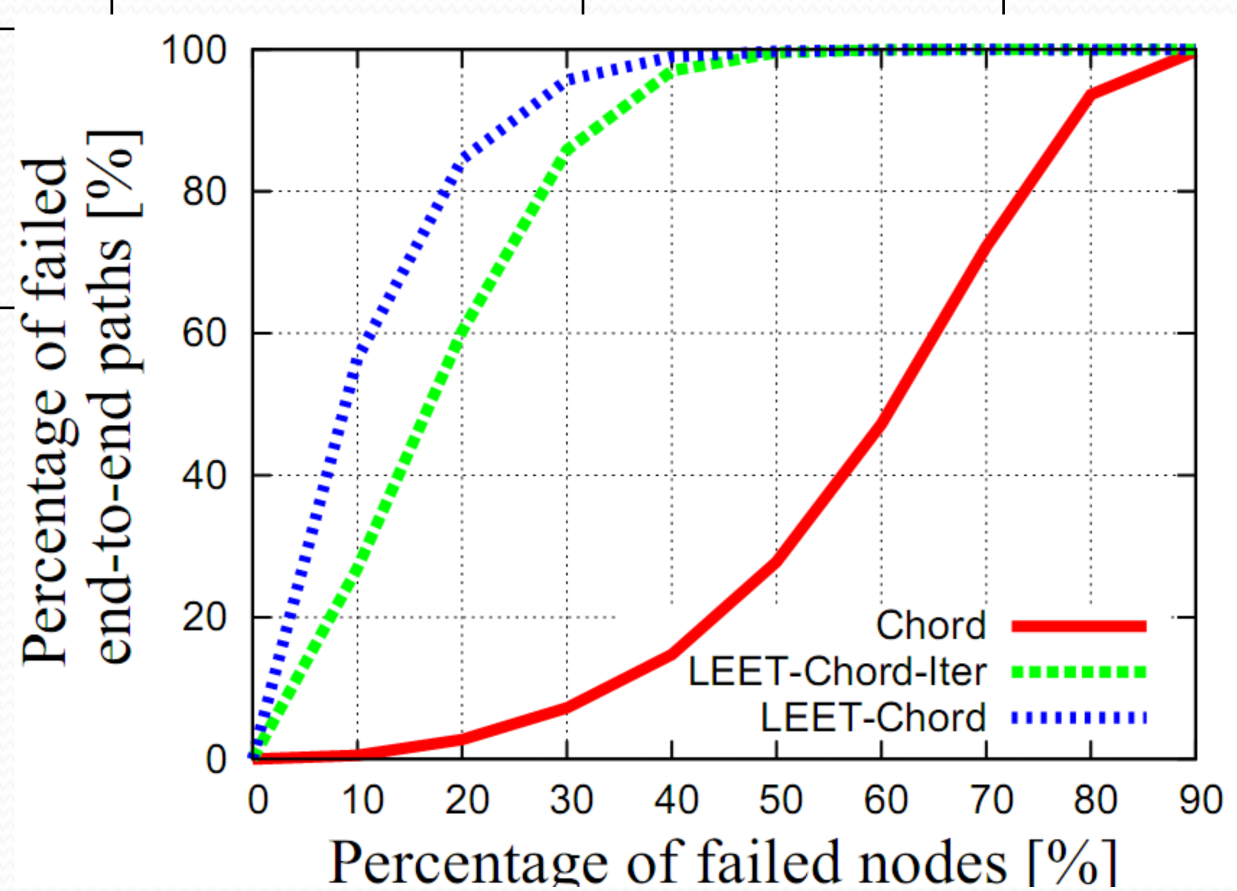
Topology	|V|	%FP	%Detected
deBruijn	1000 10000 100000	0.00 0.00 0.12	97.30 95.78 98.26
Kademlia	1000 10000 100000	0.00 0.01 0.09	99.50 99.70 99.47
Chord	1000 10000 100000	0.00 0.01 0.06	99.60 99.35 94.64

Stealth Approaches

Topology	|V|	%FP	%Detected
Chord	1000 10000 100000	0.00 0.01 0.08	97.80 97.68 98.06
LEET-Chord	1000 10000 100000	0.00 0.03 0.42	97.00 98.40 99.00

Stealth Approaches

Topology	|V|	%FP	%Detected
Chord	1000	0.00	97.80
	10000	0.01	97.68
	100000	0.08	98.06
LEET-Chord			97.00
			98.40
			99.00



Privacy-preserving algorithms

- Central algorithm: random walk
  - Can be modeled as a multiplication of vector by (sparse) matrix
  - Use Paillier homomorphism to keep vector encrypted and multiply encrypted vector (**p**) by plaintext matrix (**T**)
- Performance
  - O(|E|) homomorphic encryptions
  - Approx. 10M CPU-seconds for given parameters
  - Embarassingly parallel
  - Can use faster algorithms for deltas

Recognizing Misbehavior

- Start with a honeynet seed
  - Identify P2P network containing honeynet nodes
- Use anomaly / misbehavior detection
  - Statistical significance test

Conclusions

- Traffic analysis can be a useful security tool
  - Watermarks for stepping stone detection
  - Community detection for botnets
- In-network defenses open new possibilities
  - May be the *only* way to defend against current attacks