An Empirical Evaluation of Relay Selection in Tor

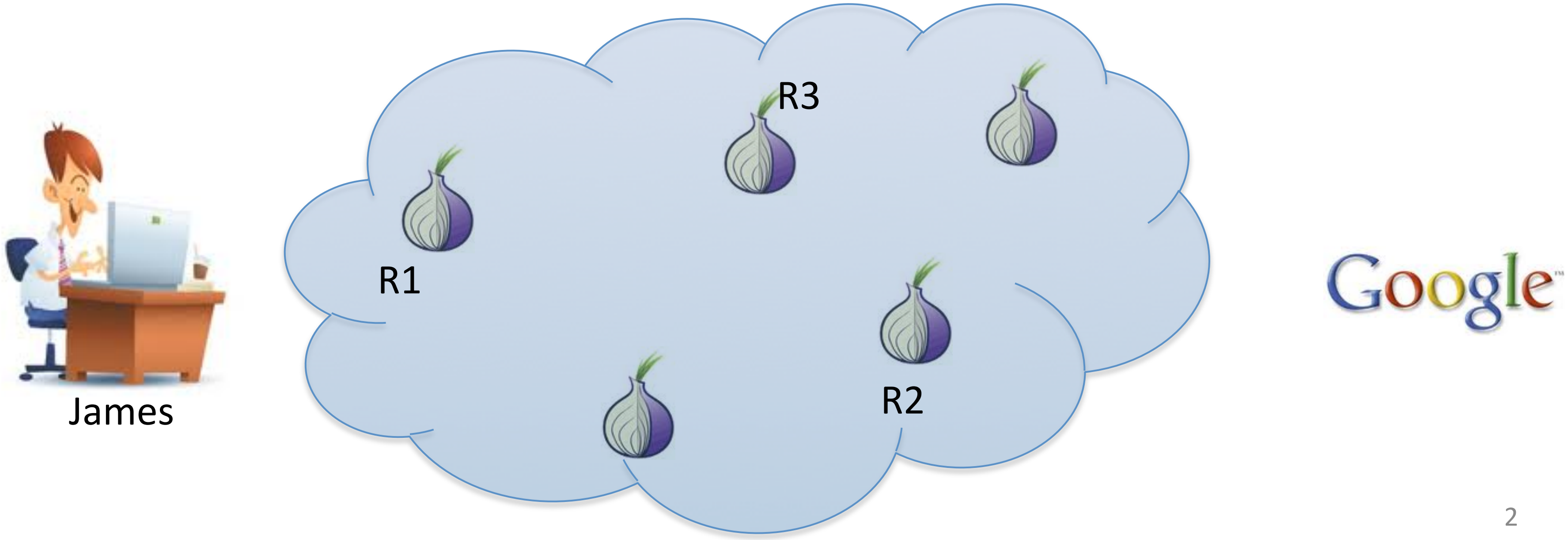
Chris Wacek Henry Tan Kevin Bauer Micah Sherr

NDSS 2013

Georgetown University

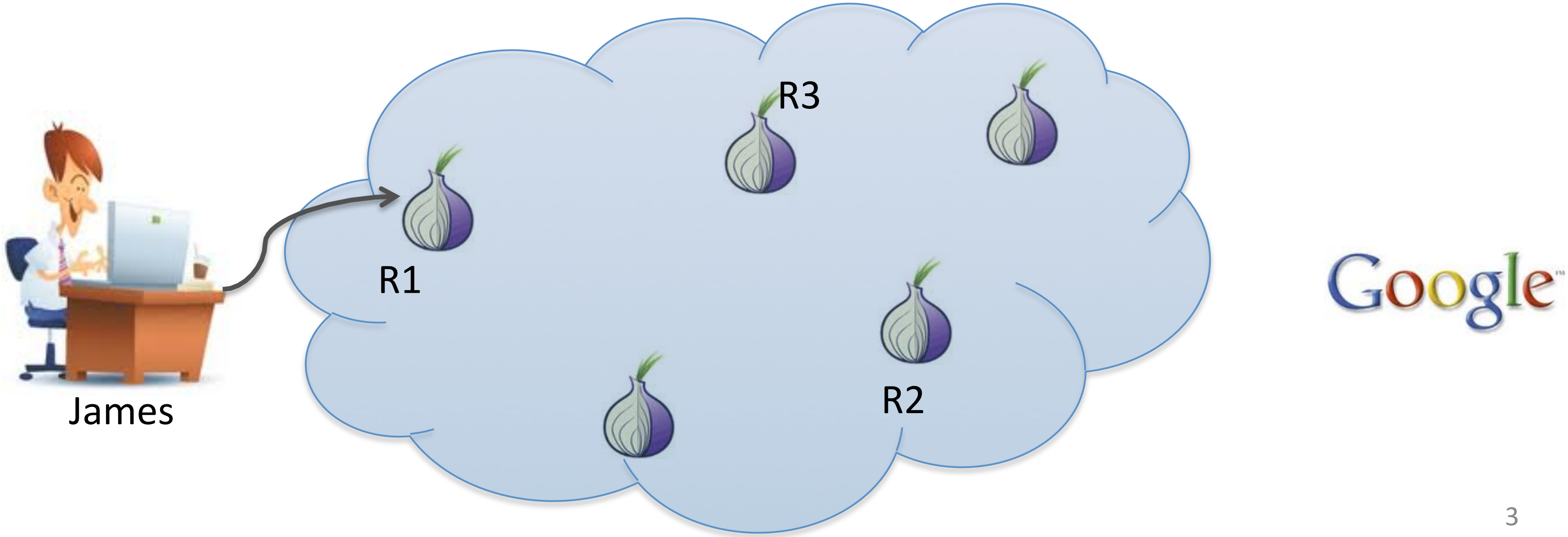
Background: What is Tor?

- Onion-routing style anonymity network
  - Anonymous *circuits* formed through set of volunteer relays.



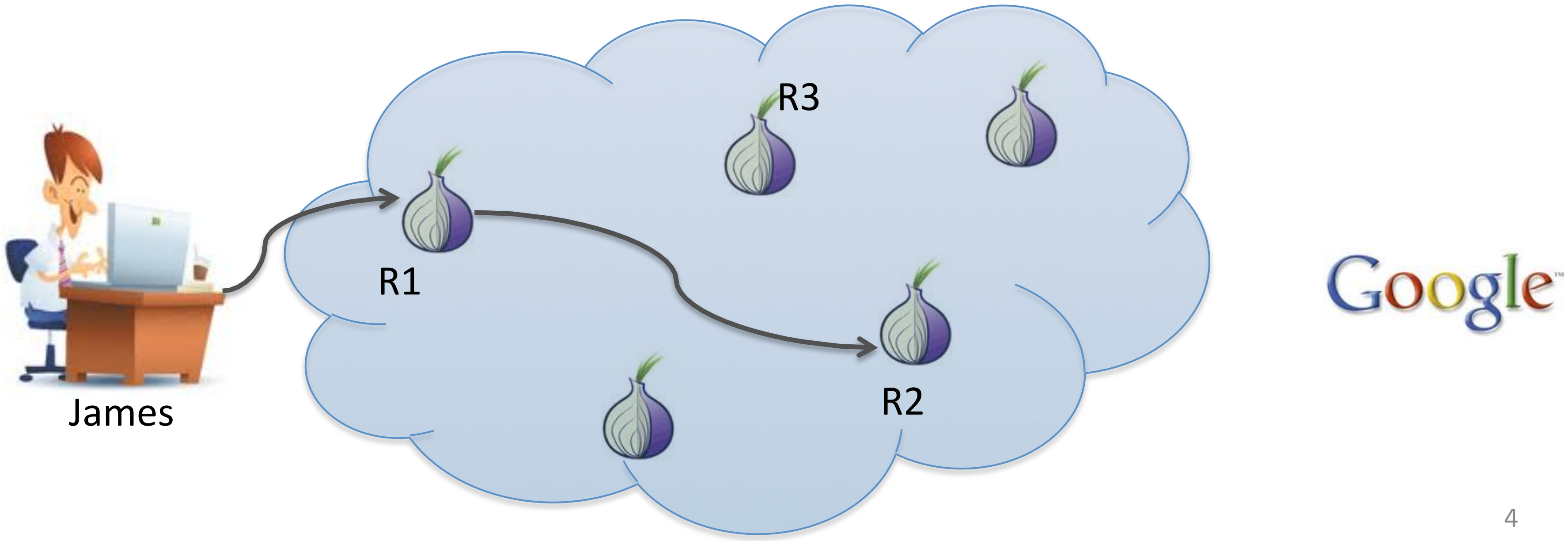
Background: What is Tor?

- Onion-routing style anonymity network
  - – Anonymous *circuits* formed through set of volunteer relays.



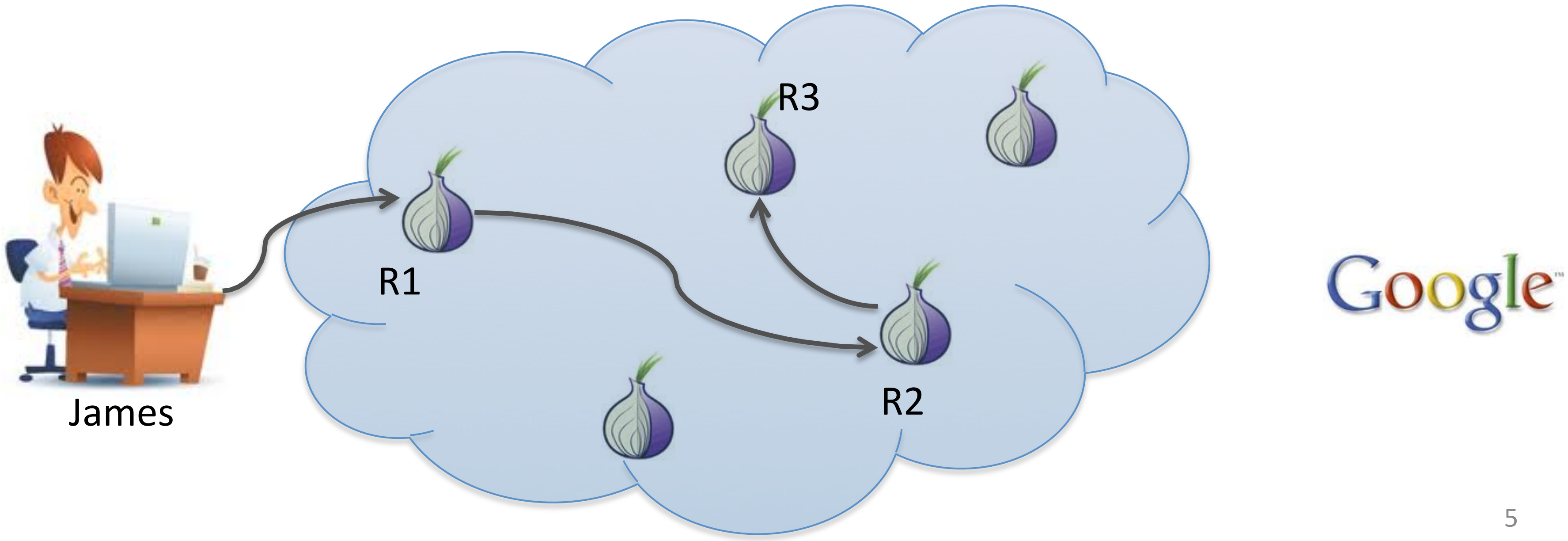
Background: What is Tor?

- Onion-routing style anonymity network
  - Anonymous *circuits* formed through set of volunteer relays.



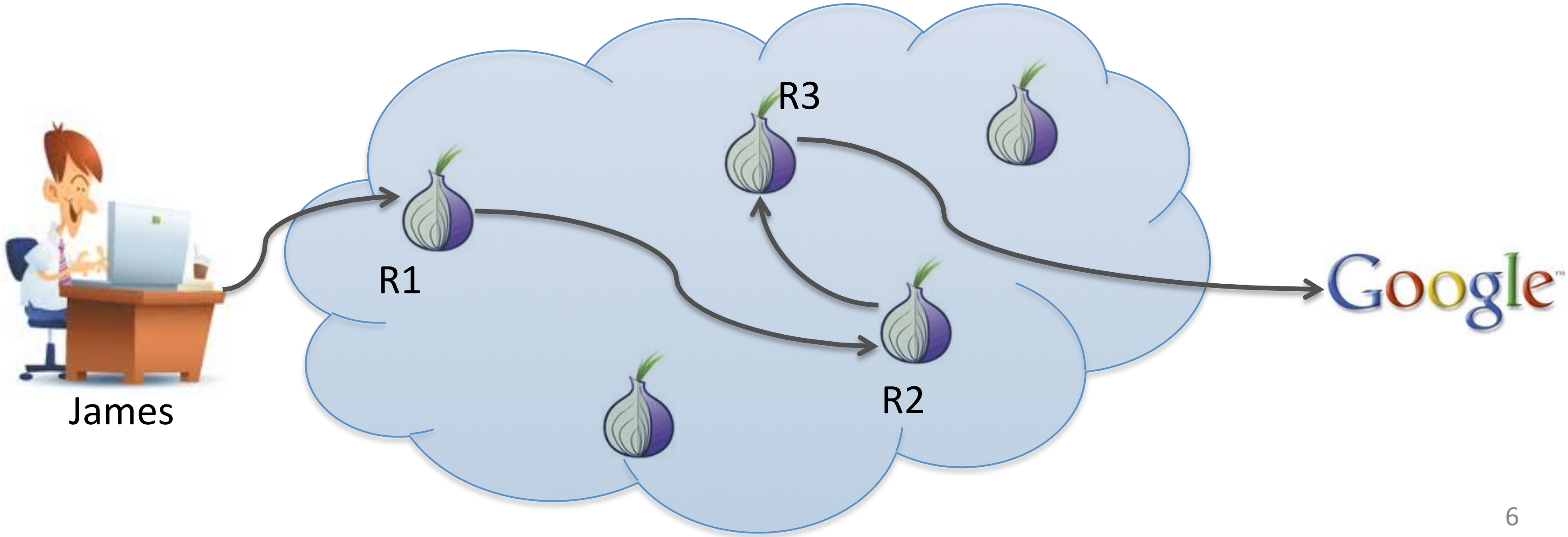
Background: What is Tor?

- Onion-routing style anonymity network
  - Anonymous *circuits* formed through set of volunteer relays.



Background: What is Tor?

- Onion-routing style anonymity network
  - Anonymous *circuits* formed through set of volunteer relays.

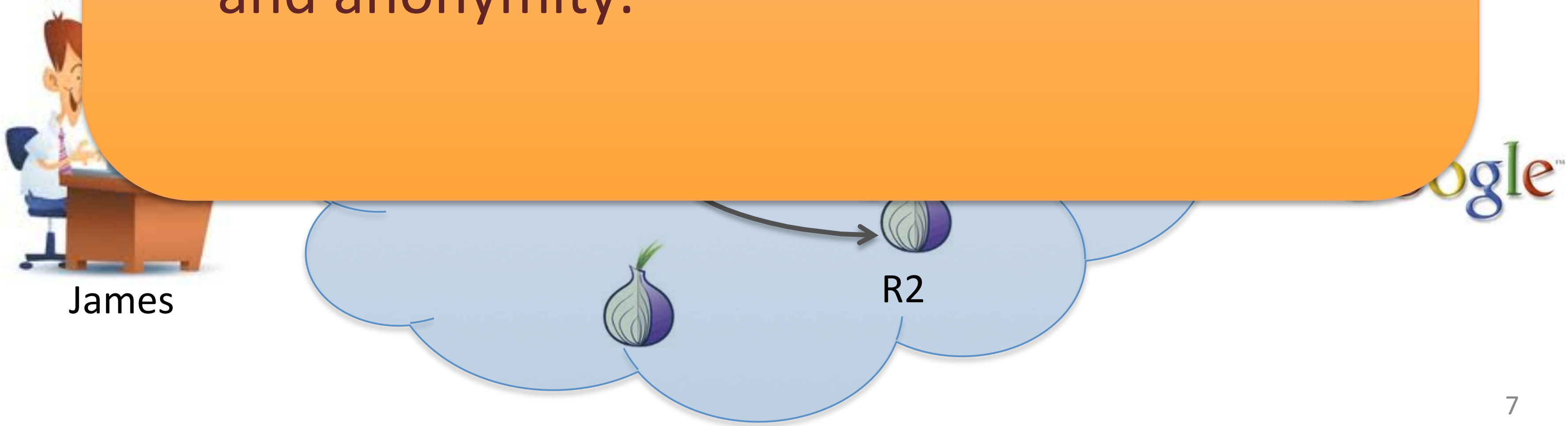


Background: What is Tor?

- Onion routing style anonymity network

Key Point:

- The client (James) chooses which relays he wants to use.
- The chosen relays affect both performance and anonymity.



Relay Selection in Tor

- Tor has a default relay selection algorithm
  - Weights towards higher bandwidth relays
  - Also weights to preserve network load balancing

- Many other strategies have been proposed:

Tunable Bandwidth Weighting
[Snader and Borisov, NDSS '08]

Geography-aware
[Akhoondi, et al., Oakland'12]

Virtual Distance-aware
[Sherr, et al., NDSS '10]

Congestion-aware
[Wang, et al., FC'12]

Evaluating Relay Selection in Tor

Goal: Effectively evaluate which relay selection strategy is the ‘best’

‘Best’ means different things to different people

- Clients have different priorities
- Large scale adoption may affect performance

Evaluating Relay Selection in Tor

How can we tell which strategy is the **best choice**?

- Evaluate each one from a security and performance perspective

Solution: Test them out in the Tor network

Evaluating Relay Selection in Tor

How can we tell which strategy is the **best choice**?

- Evaluate each one from a security and performance perspective

~~**Solution:** Test them out in the Tor network~~

Tor is a **live anonymity network**. Changing relay selection strategies on the live network without knowing the effects may have consequences for active users

What do we need from a Tor model for evaluating relay selection?

1. *Capability* for testing the effectiveness of new protocols if adopted across the network

2. *Confidence* that evaluation results will translate to real-world Tor

3. *Metrics* to understand anonymity and performance implications

Selecting a platform that enables realistic experimentation

CAPABILITY

Capability: Full Network Emulation

Emulate the Tor network, rather than operating on the live Tor network.

ExperimenTor [Bauer, et al ., CSET '11] is a large scale network emulation framework.

Bandwidth and latency characteristics can be applied to network links.

Capability: Full Network Emulation

Benefits:

- Emulates all portions of the Tor network, including clients, relays and destinations.
- Runs the actual unmodified Tor binaries
- Allows evaluation of changes in how clients select relays.
- Enables testing strategies that require protocol changes.

Capability: Full Network Emulation

Benefits:

- Emulates all portions of the Tor network, including clients, relays and destinations.
- Runs the actual unmodified Tor binaries
- Allows evaluation of changes in how clients select relays.
- Enables testing strategies that require protocol changes.

Disadvantages:

- Scalability – *ExperimenTor* can't handle a network the size of the full Tor network (~3500 relays / 500000+ clients)

Building a believable network model

CONFIDENCE

Confidence: Model the actual Internet

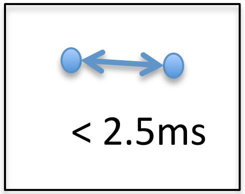
- Existing Internet “maps” lack sufficient granularity
  - Desire inter-host latency, AS membership, and other granular characteristics.
- We build a model at the granularity of a **point-of-presence**
  - Represents an access point on the internet.

Confidence: Model the actual Internet

- Building **point-of-presence** graph:
  - Using CAIDA traceroute data, we build a graph of connected IP addresses
- Heuristically group IPs into **points-of-presence**

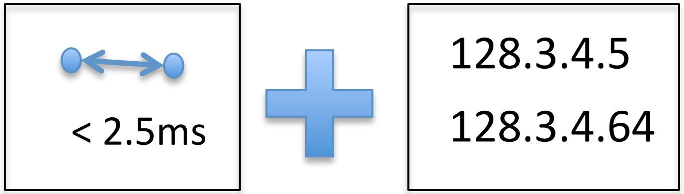
Confidence: Model the actual Internet

- Building **point-of-presence** graph:
  - Using CAIDA traceroute data, we build a graph of connected IP addresses
- Heuristically group IPs into **points-of-presence**



Confidence: Model the actual Internet

- Building **point-of-presence** graph:
  - Using CAIDA traceroute data, we build a graph of connected IP addresses
- Heuristically group IPs into **points-of-presence**



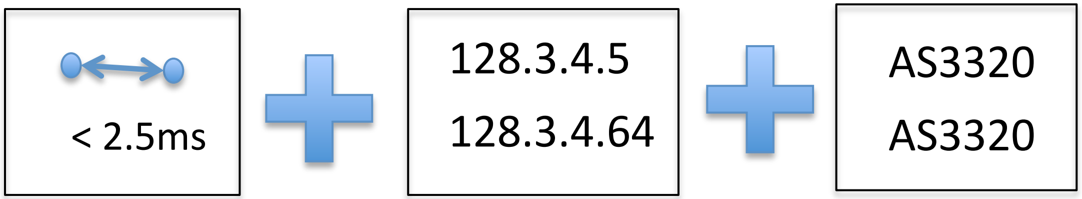
< 2.5ms

128.3.4.5

128.3.4.64

Confidence: Model the actual Internet

- Building **point-of-presence** graph:
  - Using CAIDA traceroute data, we build a graph of connected IP addresses
- Heuristically group IPs into **points-of-presence**



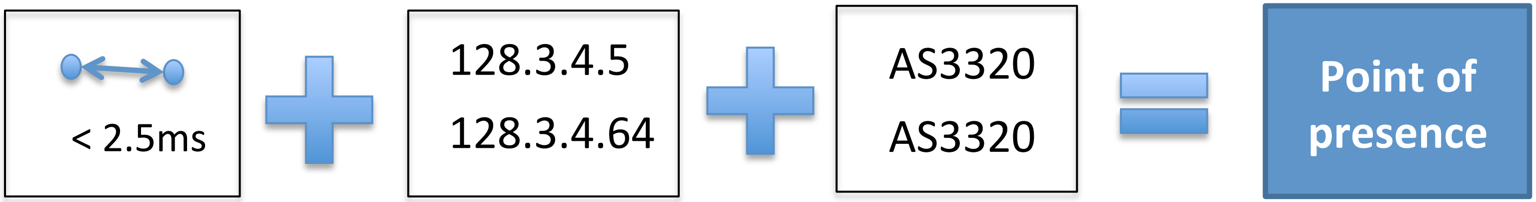
< 2.5ms

128.3.4.5
128.3.4.64

AS3320
AS3320

Confidence: Model the actual Internet

- Building **point-of-presence** graph:
  - – Using CAIDA traceroute data, we build a graph of connected IP addresses
- Heuristically group IPs into **points-of-presence**



< 2.5ms

128.3.4.5

128.3.4.64

AS3320

AS3320

Point of presence

Confidence: Model the actual Internet

Vertices are **points-of-presence** in the Internet with associated IP addresses

Edges represent links between **points-of-presences** as present in traceroute data

Edge weights are latencies from traceroutes

Confidence: Model the actual Internet

Vertices are **points-of-presence** in the Internet with associated IP addresses

Edges represent links between **points-of-presences** as present in traceroute data

Edge weights are latencies from traceroutes

What about Tor?

Confidence: Model the actual Internet

- Attach Tor relays to the network graph:
  - Match Tor relay IP addresses to IP addresses in the graph
  - Allow matches at the /24 level.

- Allows us to attach 1524 distinct Tor relays.

Confidence: Model the actual Internet

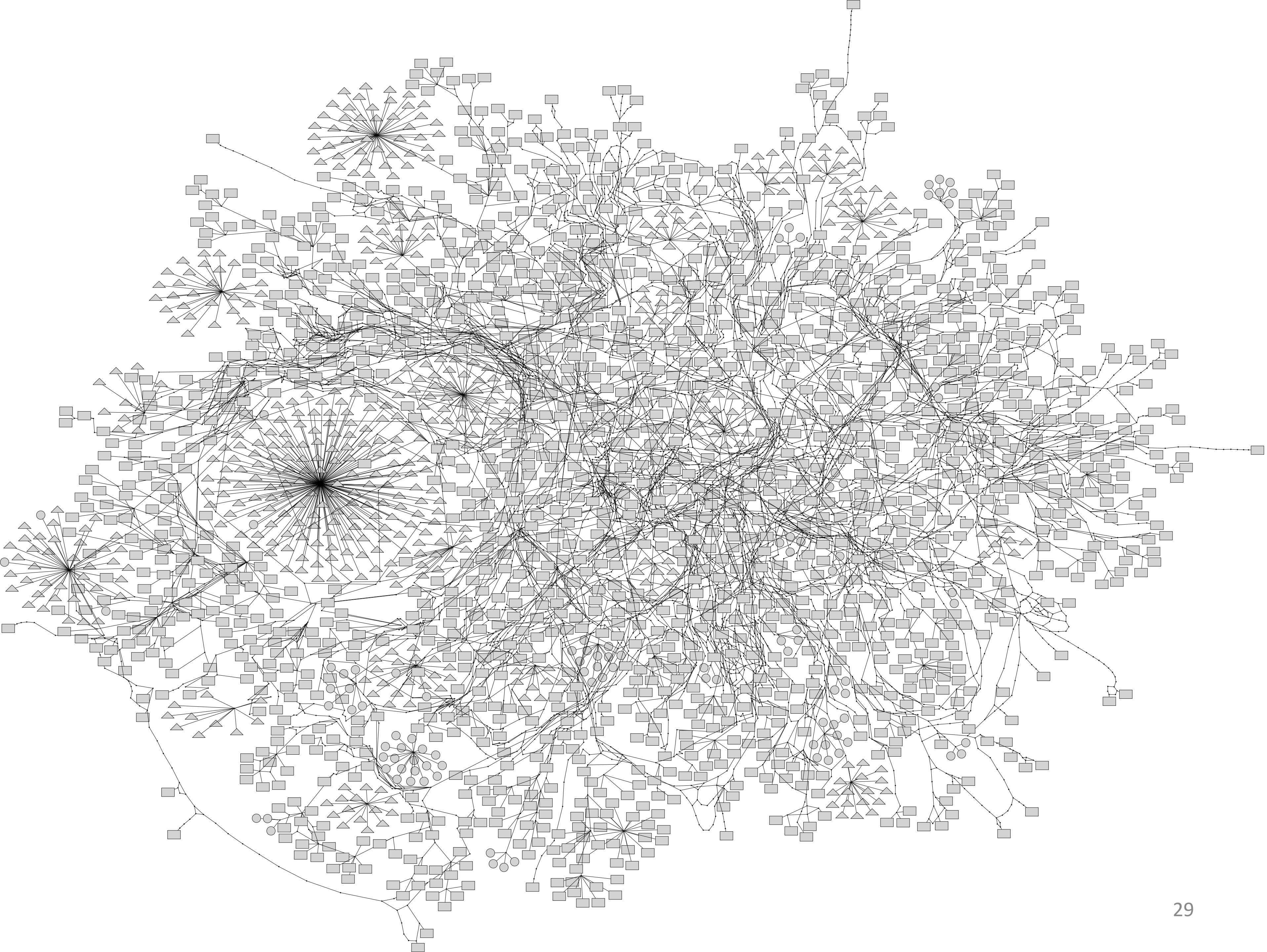
- Attach Tor relays to the network graph:
  - Match Tor relay IP addresses to IP addresses in the graph
  - Allow matches at the /24 level.

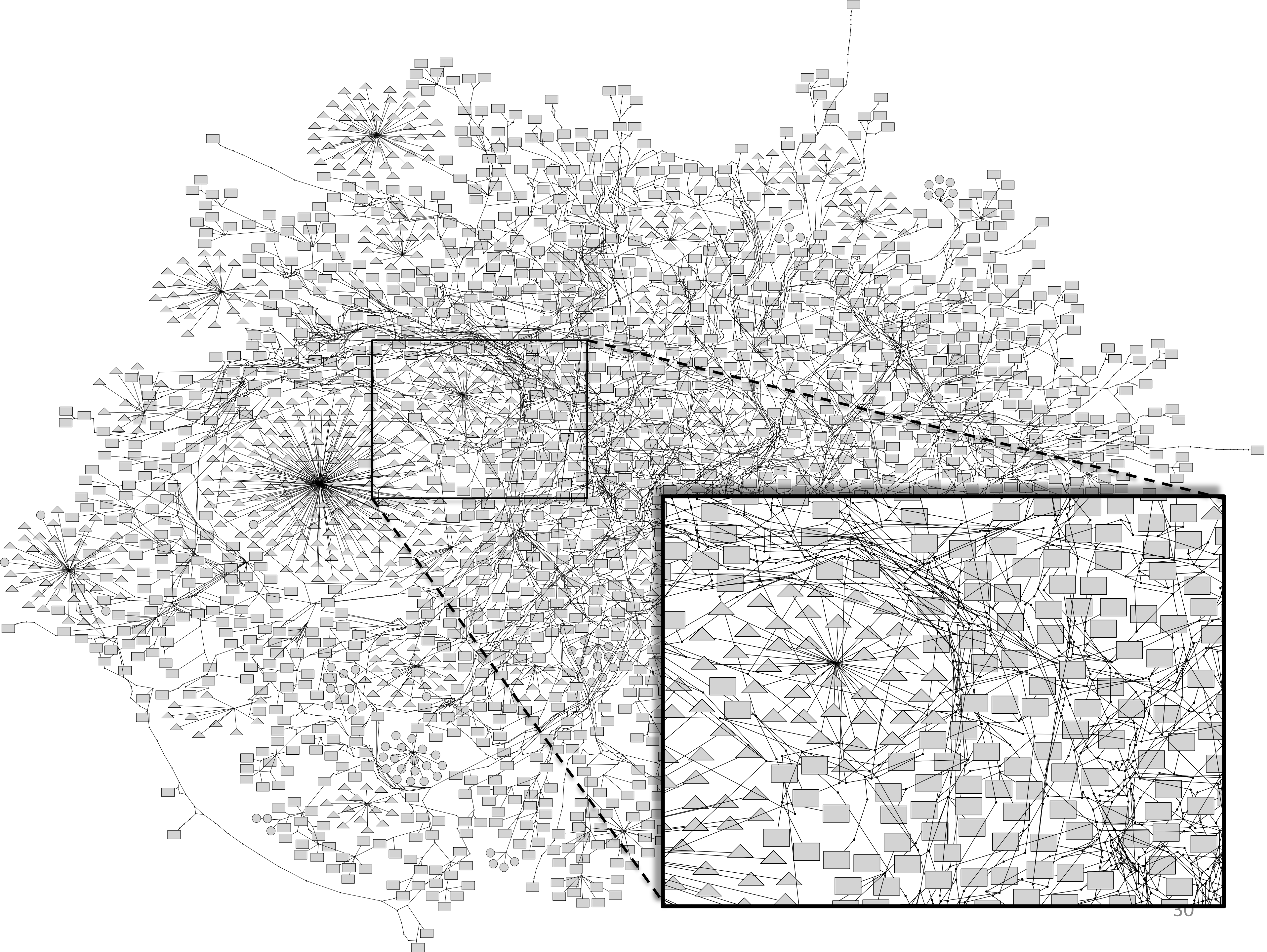
- Allows us to attach 1524 distinct Tor relays.

- And clients and destinations?

Confidence: Model the actual Internet

- Attach clients and destinations to the largest **point-of-presence** for an AS, assigning more clients and destinations to the more popular ASes
- Use data about the 25 most popular Tor client and destination ASes from 2009 [Edman and Syverson, CCS'09]





Confidence: Verify our topologies represent the Tor network

- These topologies:
  - Don’t contain every relay
  - Make some simplifying assumptions

- To have confidence in our model, we compare some high level characteristics.
  - Sampled relay bandwidth distribution
  - Percentage of relay types

Confidence: Verify our topologies represent the Tor network

- These topologies:
  - Don’t contain every relay
  - Make some simplifying assumptions

- To have confidence in our model, we compare some high level characteristics.
  - Sampled relay bandwidth distribution ✓
  - Percentage of relay types

Confidence: Verify our topologies represent the Tor network

- These topologies:
  - Don't contain every relay
  - Make some simplifying assumptions

- To have confidence in our model, we compare some high level characteristics.
  - Sampled relay bandwidth distribution ✓
  - Percentage of relay types ✓

Applying the model

RESULTS

Metrics: Understanding Evaluation Results

Performance	Throughput
	Time to first Byte
	Ping Round Trip Time
Anonymity	Gini Coefficient
	Entropy
	AS Presence

Applying the Model: Selection Strategies

- **Tor**
- Unweighted
- LASTor
- Coordinate
- Tor+Coordinate
- Congestion-Aware

The default Tor strategy.

Bias relay selection proportionally to relays' reported bandwidth.

Assign special weights to guard and exit relays.

Designed to achieve good load balancing.

Applying the Model: Selection Strategies

- Tor
- **Unweighted**
- LASTor
- Coordinate
- Tor+Coordinate
- Congestion-Aware

No bandwidth bias. Relays selected uniformly at random

Applying the Model: Selection Strategies

- Tor
- Unweighted
- **LASTor**
- Coordinate
- Tor+Coordinate
- Congestion-Aware

Variant of LASTor.

[Akhoondi, et al., Oakland 2012]

Use geographic distances to estimate latencies. Cluster relays into grid squares, and choose path of grid squares to minimize latency. For each grid square in path, choose relay at random.

Applying the Model: Selection Strategies

- Tor
- Unweighted
- LASTor
- **Coordinate**
- Tor+Coordinate
- Congestion-Aware

Use Vivaldi virtual coordinate embedding system to estimate latencies [Sherr, et al., NDSS 2010] [Dabek, et al., SIGCOMM 2004]

Only consider latency between relays

Generate 3 anonymous paths using **no bandwidth bias**;
Select the path with the lowest estimated latency.

Applying the Model: Selection Strategies

- Tor
- Unweighted
- LASTor
- Coordinate
- **Tor+Coordinate**
- Congestion-Aware

Bandwidth and latency-aware selection [Sherr, et al., NDSS 2010]

Use Vivaldi virtual coordinate embedding system to estimate latencies [Dabek, SIGCOMM'04]

Generate 3 anonymous paths using **Tor's bandwidth-weighted strategy**;
Select the path with the lowest estimated latency.

Applying the Model: Selection Strategies

- Tor
- Unweighted
- LASTor
- Coordinate
- Tor+Coordinate
- **Congestion-Aware**

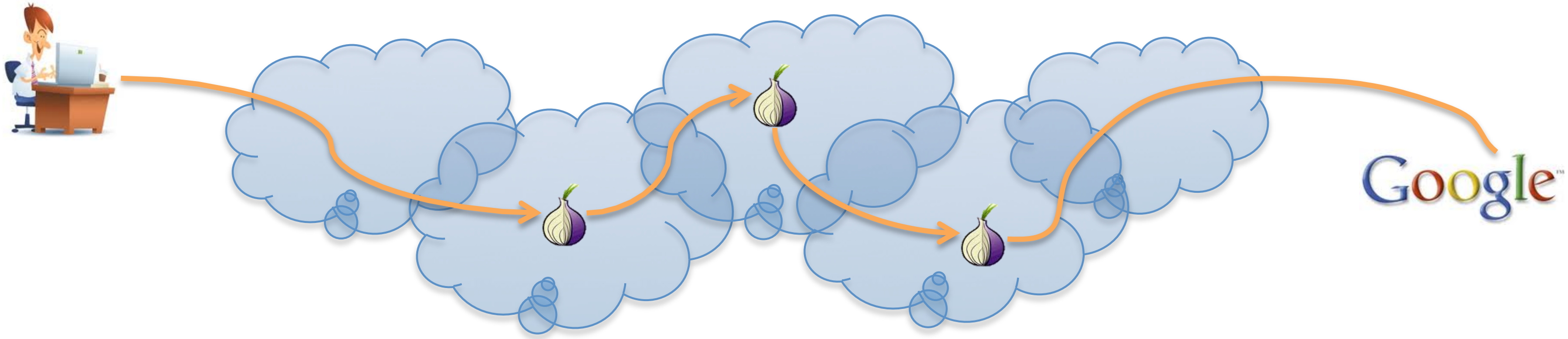
Uses normal Tor selection strategy

Actively measures constructed circuits, and discards them if they appear congested
[Wang ,FC '12]

Orthogonal to other strategies

Applying the Model: Path Selection Simulations

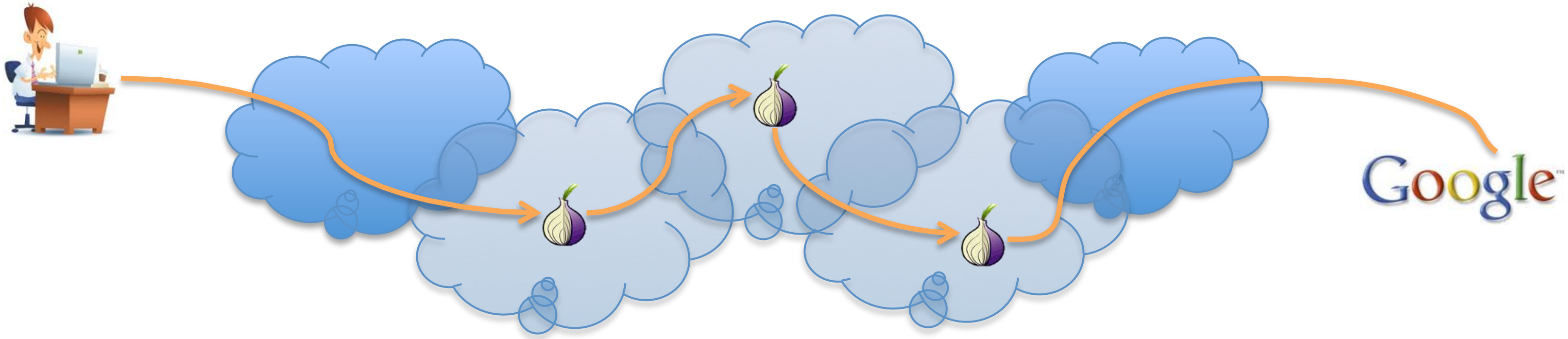
- Built thousands of simulated paths from the relays in the 1524-relay model



- Can give insight into ASes that pose anonymity concerns

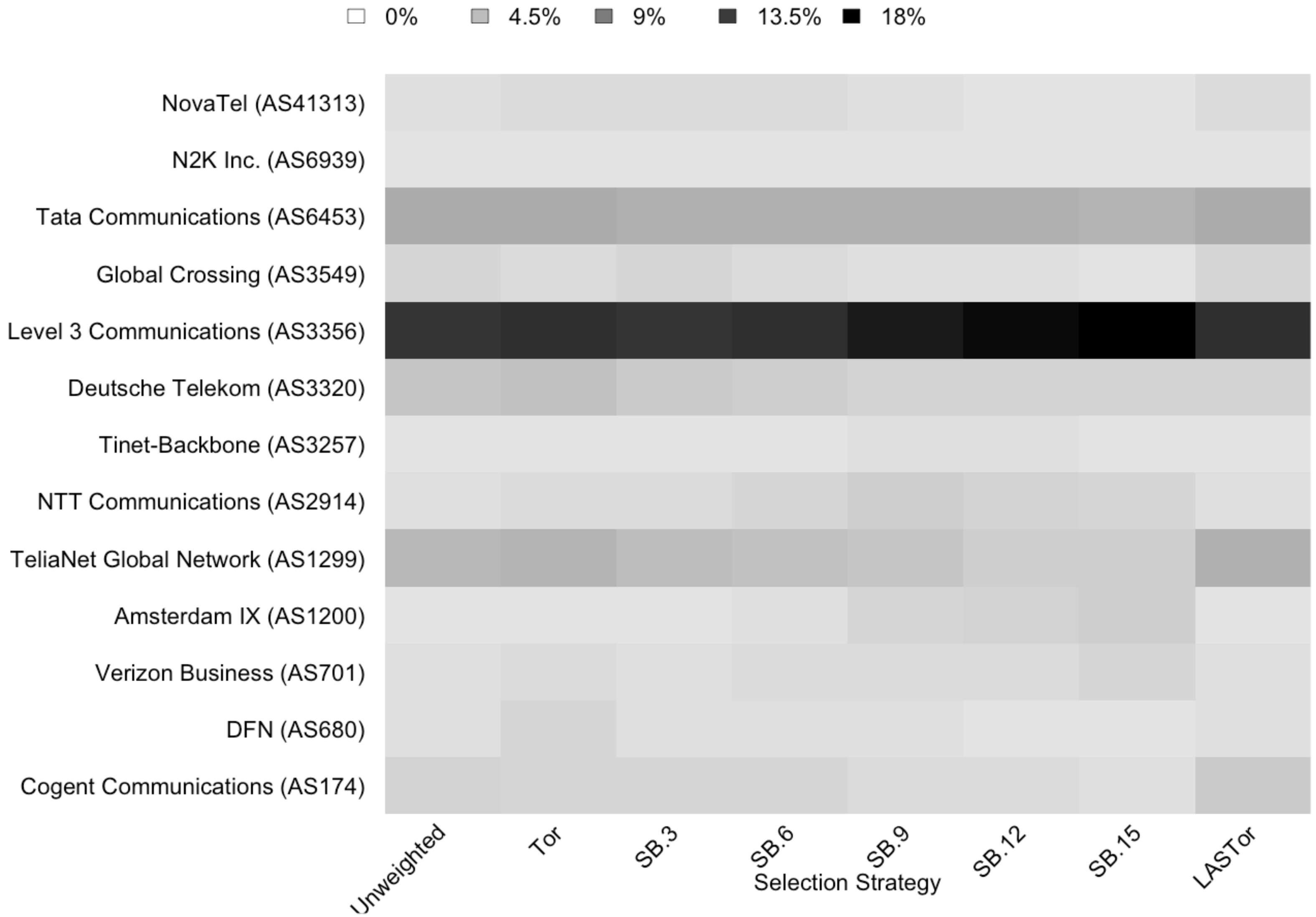
Applying the Model: Path Selection Simulations

- Built thousands of simulated paths from the relays in the 1524-relay model



- Can give insight into ASes that pose anonymity concerns

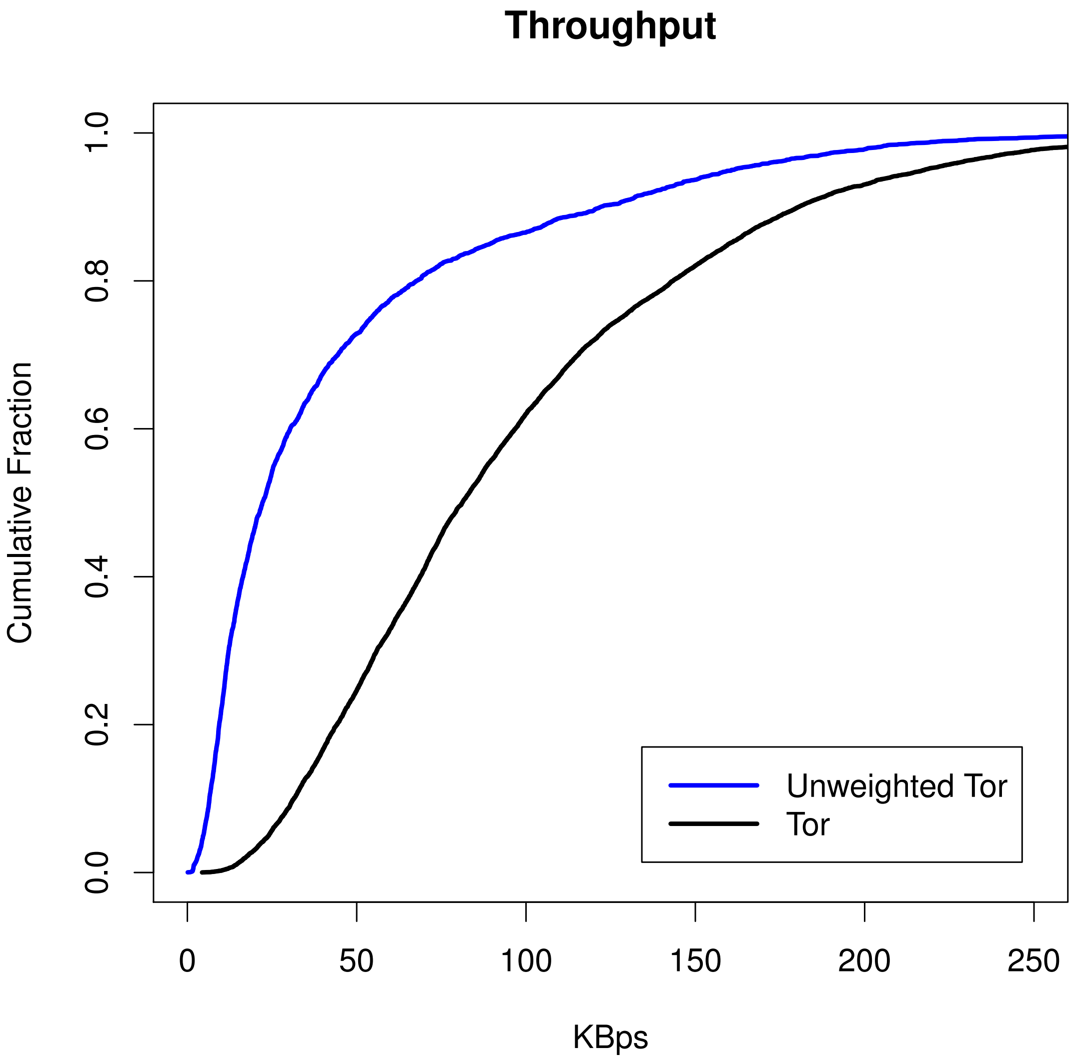
Applying the Model in Simulation



Applying the Model: Performance and Anonymity Evaluation

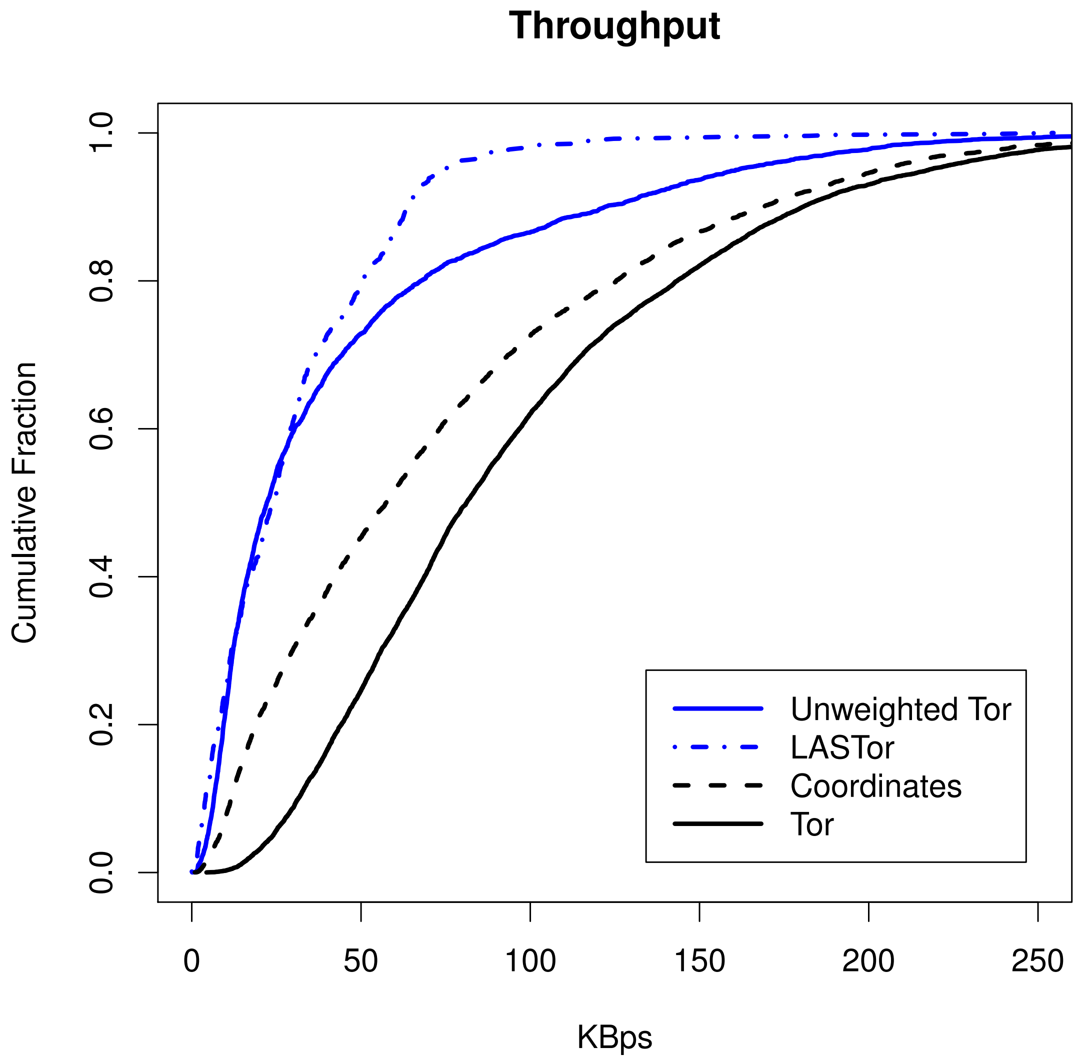
- Emulated our ‘scaled’ Tor network with 50 relays using *ExperimenTor* as a platform
  - Inter-host latencies given by network model
  - Tor relay bandwidths configured according to real-world Tor

Applying the Model in Emulation



Weighting for bandwidth makes a significant difference

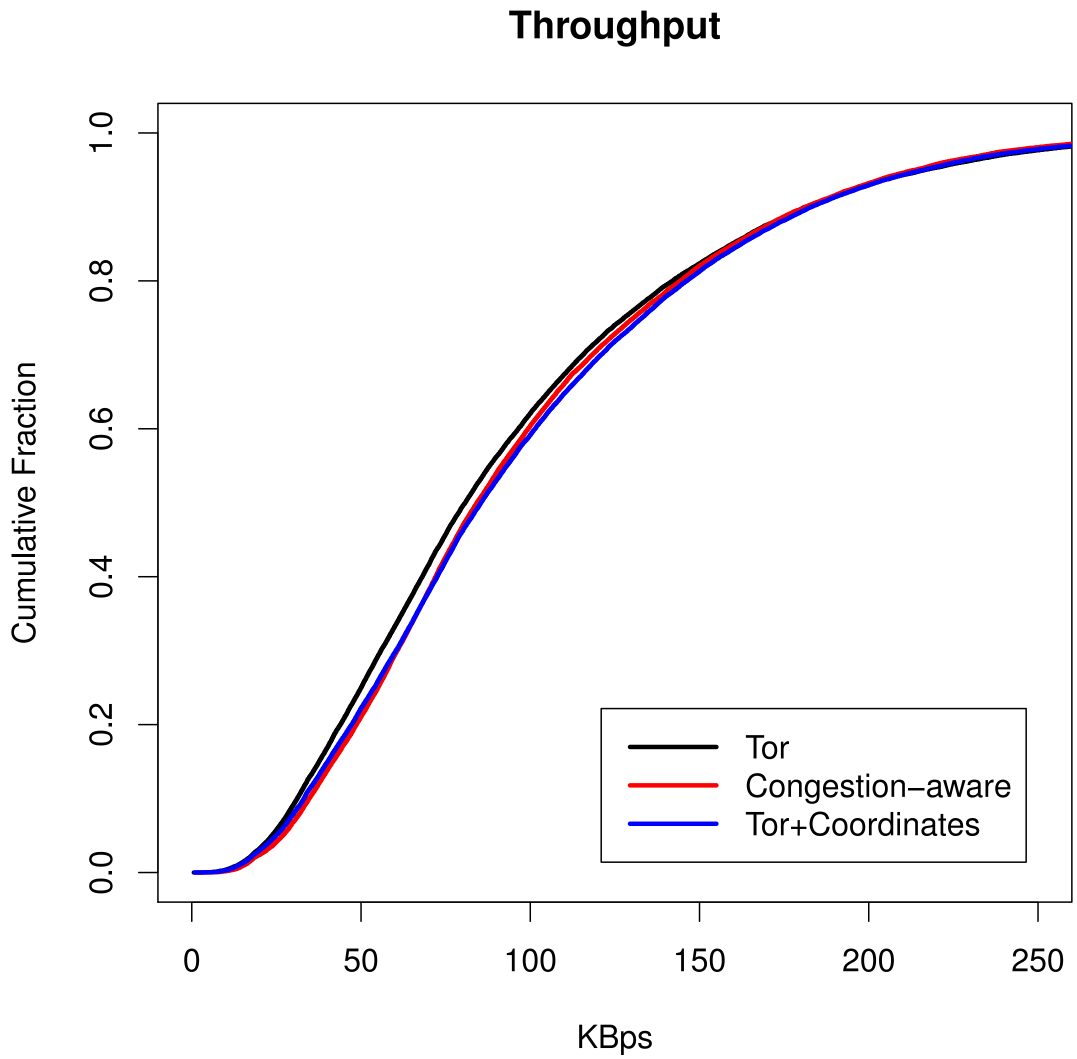
Applying the Model in Emulation



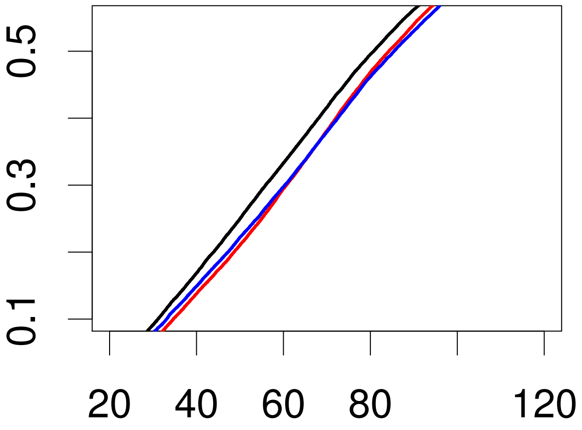
Strategies that don't account for bandwidth perform poorly

Geographic selection in particular doesn't work very well

Applying the Model in Emulation



Layering strategies over bandwidth weighting can provide incremental improvements



Applying the Model in Emulation

Selection Strategy	Gini Coefficient
Tor + Coordinates	0.77
Tor	0.71
Congestion-aware	0.61
Coordinates	0.56
Unweighted Tor	0.53
LASTor	0.50

More concentrated

Anonymity goes down as strategies become more selective

More evenly distributed

In Conclusion: Results

- We confirmed that load balancing is the most important aspect for Tor
  - Strategies that do not account for available bandwidth will perform poorly
- There is potential for improving performance by layering strategies
  - Bandwidth weighting combined with latency or congestion aware strategies can be successful

In Conclusion: Modeling

- We can build a network model for evaluating the Tor network that is grounded in concrete network measurements.

- Armed with this model, we can use emulation and simulation platforms to evaluate relay selection (and other things!) in the Tor network in a rigorous manner.

QUESTIONS?