Studying Black Holes on the Internet with **Hubble**

Ethan Katz-Bassett, Harsha V. Madhyastha, John P. John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson

University of Washington

August 2008

This work partially supported by Cisco, Google, NSF

Global Reachability

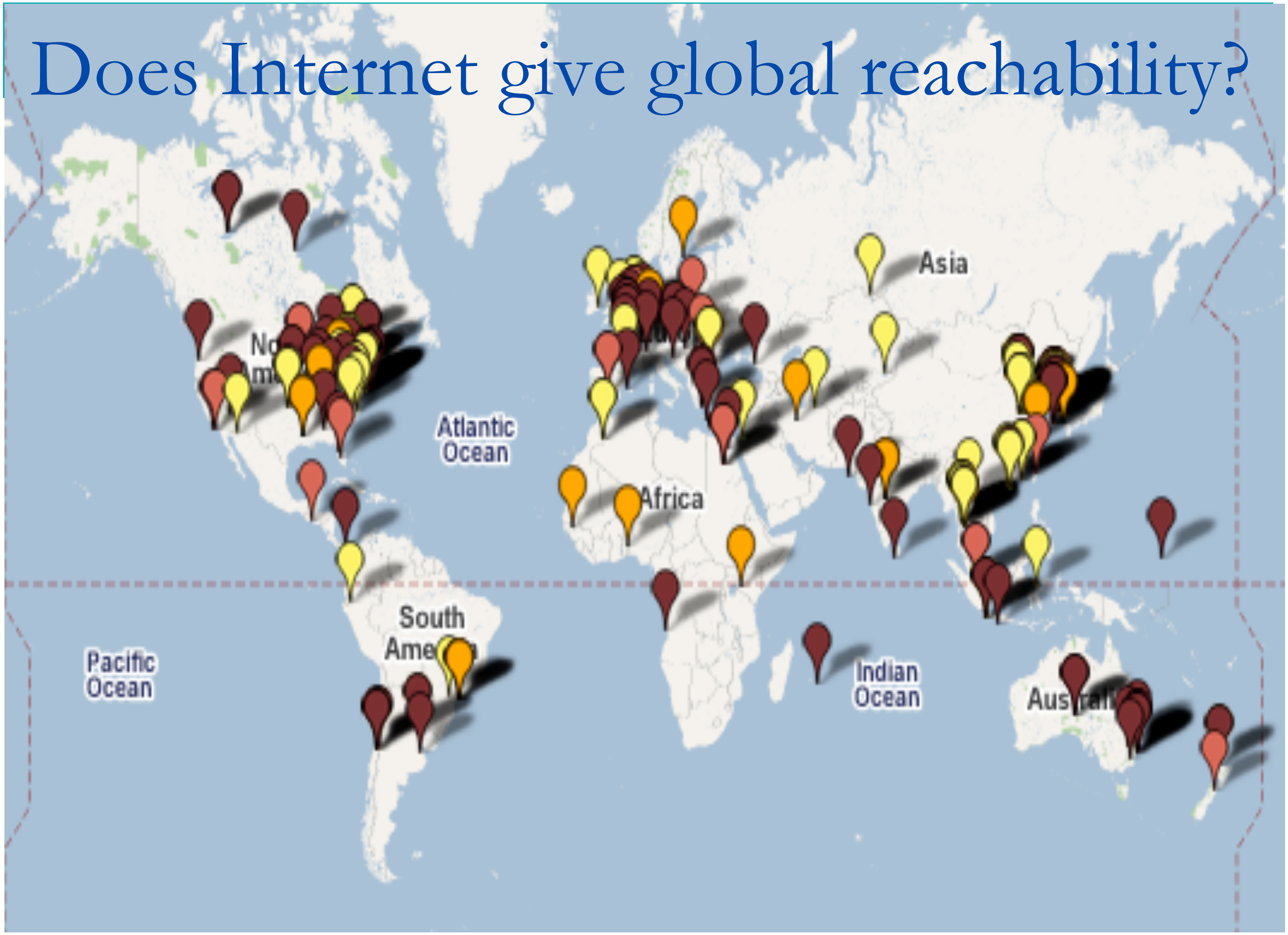
- When an address is reachable from every other address
- Most basic goal of Internet, especially BGP
  - “There is only one failure, and it is complete partition” *Clarke, Design Philosophy of the DARPA Internet Protocols*
- Physical path ⇨ BGP path ⇨ traffic reaches
- ***Black hole:*** BGP path, but traffic persistently does not reach

Does Internet give global reachability?

- From use, seems to usually work
- Can we assume the protocols just make it work?

- “Please try to reach my network 194.9.82.0/24 from your networks…. Kindly anyone assist.”
  Operator on NANOG mailing list, March 2008.

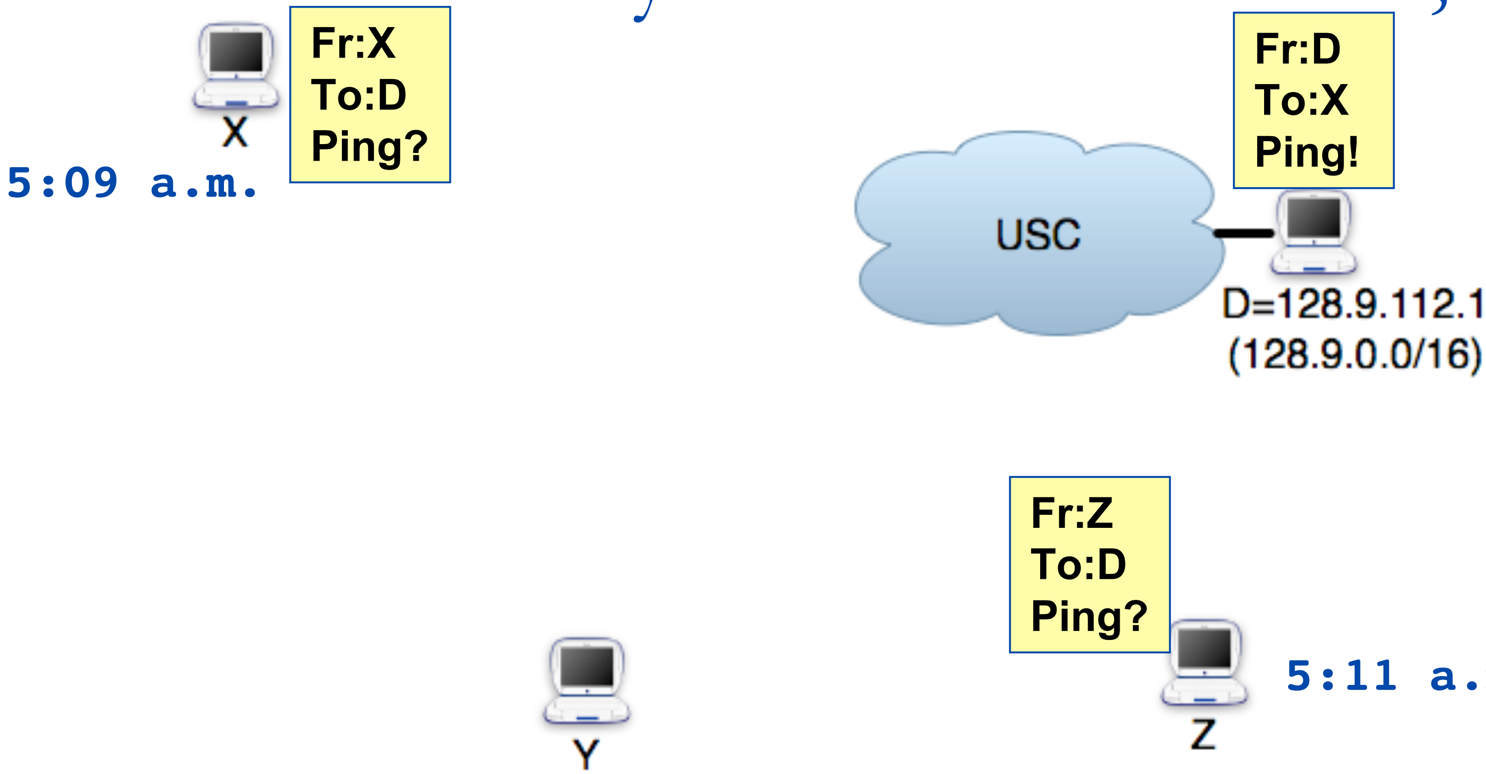
Does Internet give global reachability?



Hubble System Goal

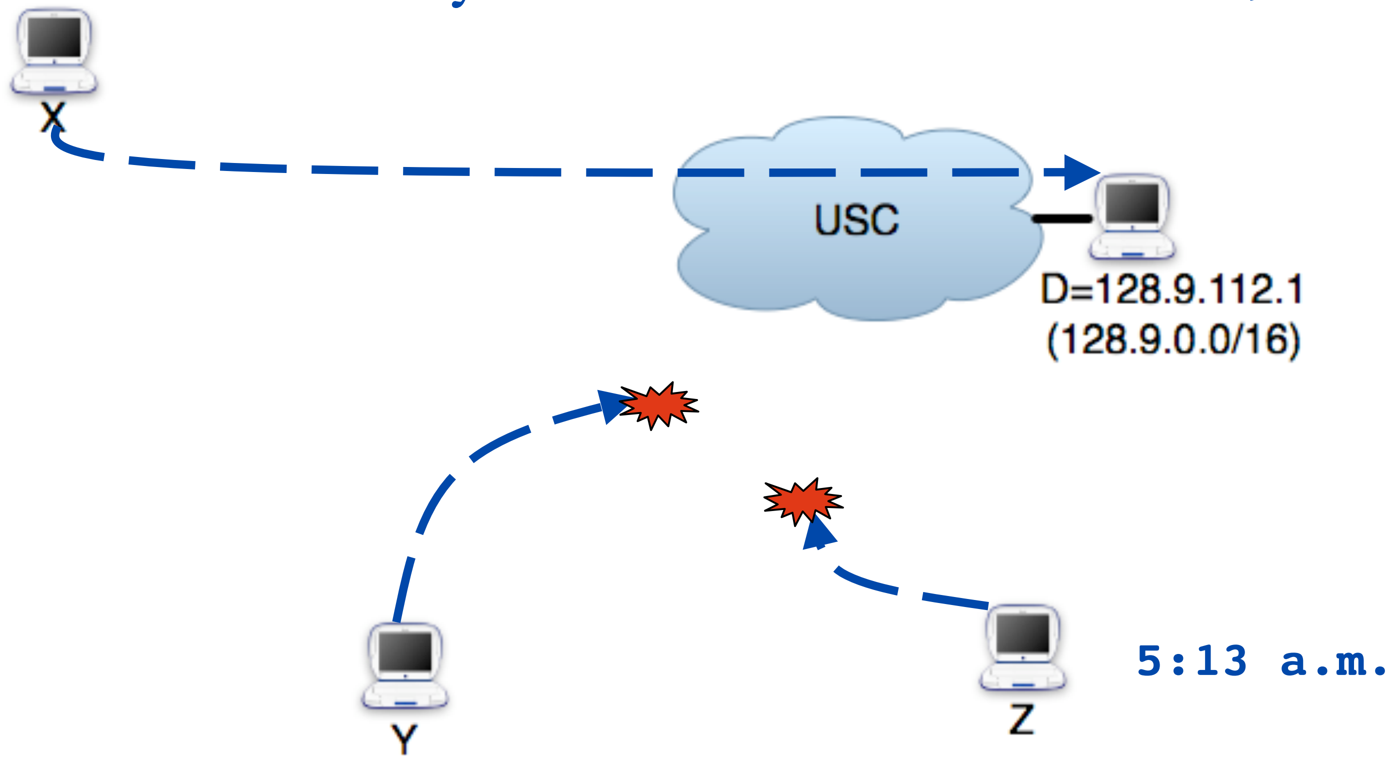
In ***real-time*** on a ***global scale***, ***automatically*** **monitor** long-lasting reachability problems and **classify** causes

Problem Seen by **Hubble** on Oct. 8, 2007



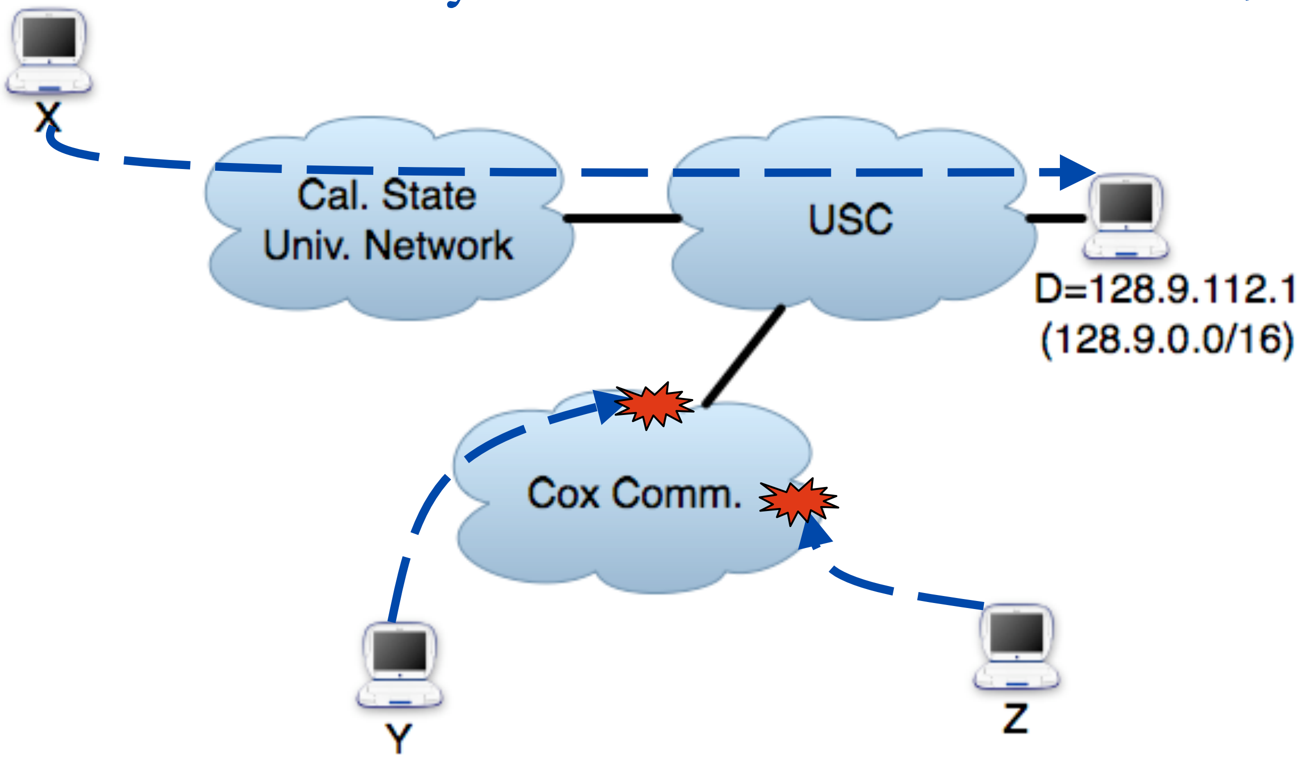
1. Target Identification – distributed ping monitors detect when the destination becomes unreachable

Problem Seen by **Hubble** on Oct. 8, 2007



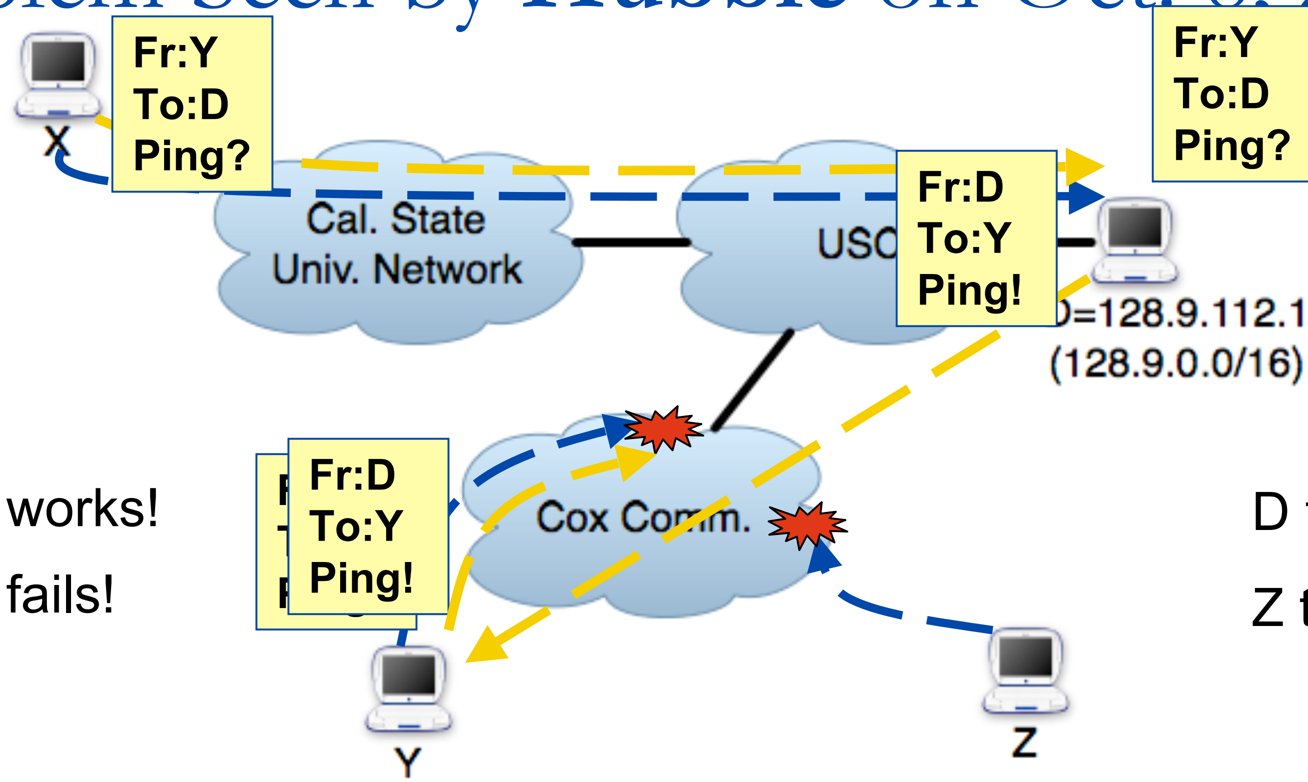
1. Target Identification – distributed ping monitors
2. Reachability analysis – distributed traceroutes determine the extent of unreachability

Problem Seen by **Hubble** on Oct. 8, 2007



1. Target Identification – distributed ping monitors
2. Reachability analysis – distributed traceroutes
3. Problem Classification
   a) group failed traceroutes

Problem Seen by **Hubble** on Oct. 8, 2007



D to Y works!

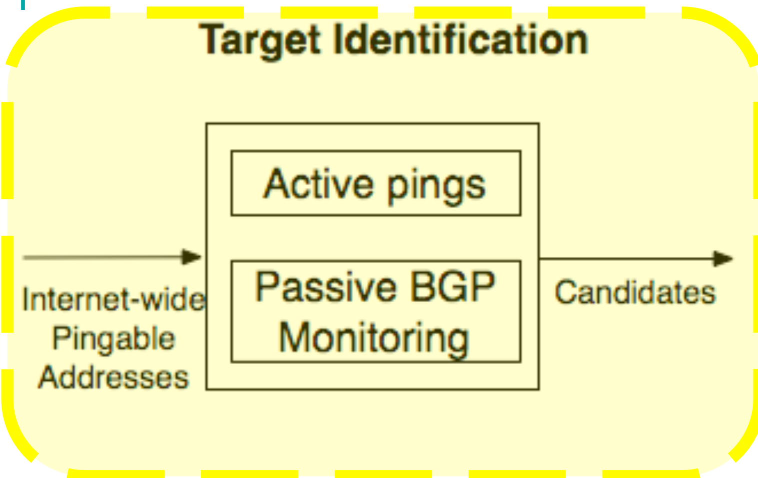
Y to D fails!

D to Z works!

Z to D fails!

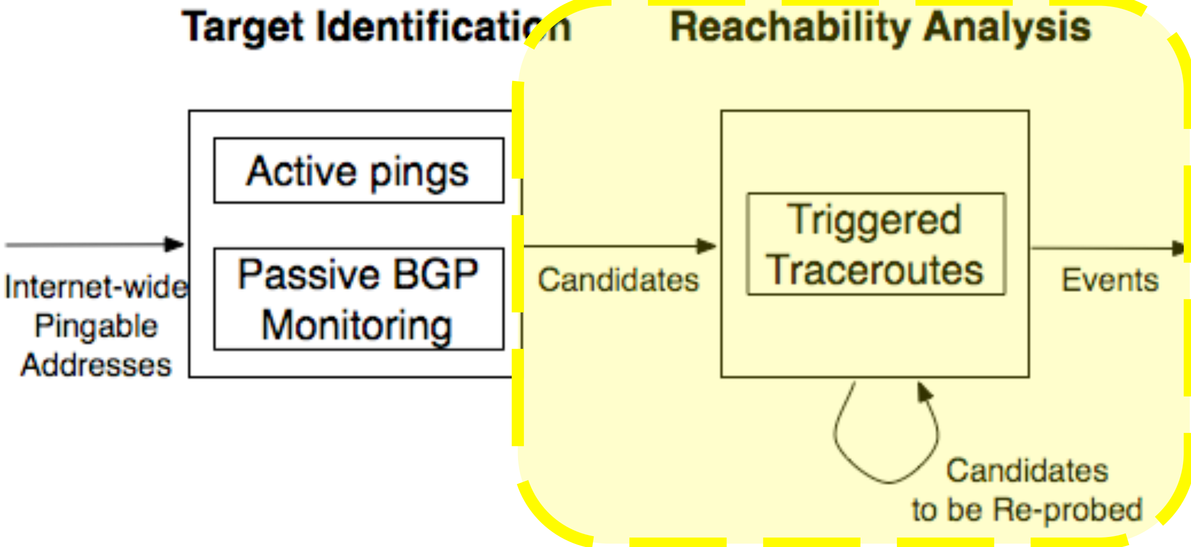
1. Target Identification – distributed ping monitors
2. Reachability analysis – distributed traceroutes
3. Problem Classification
   a) group failed traceroutes
   b) spoofed probes to isolate direction of failure

Architecture: Detect Problem



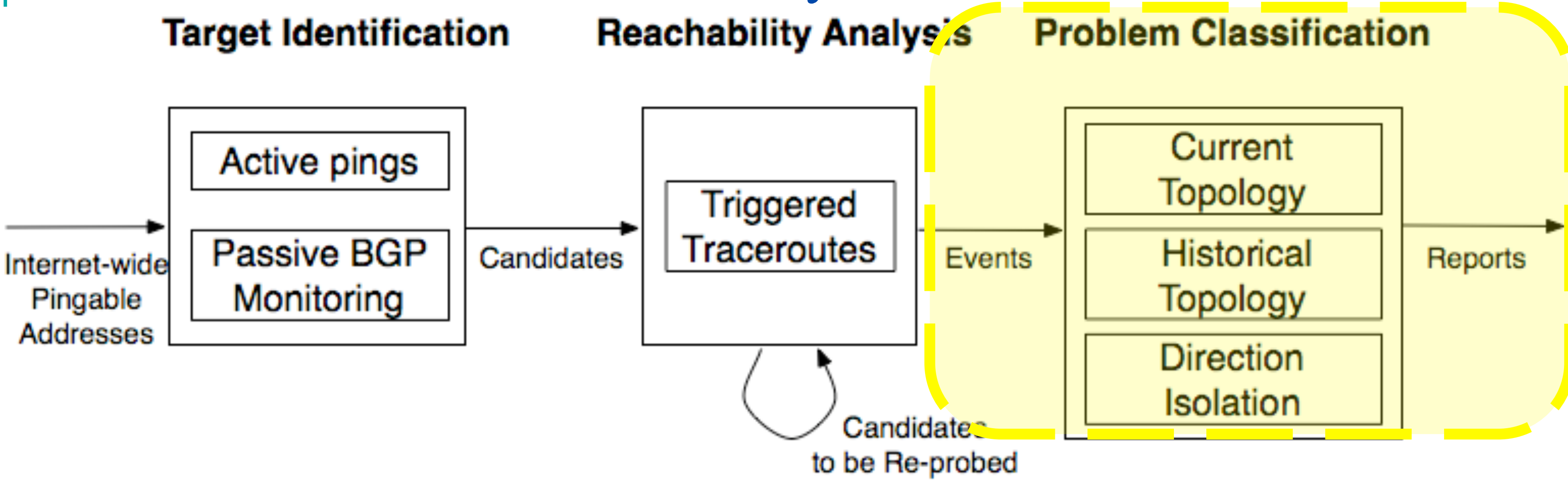
- Ping prefix to check if still reachable
  - Every 2 minutes from PlanetLab
  - Report target after series of failed pings
- Maintain BGP tables from RouteViews feeds
  - Allows IP ⇒ AS mapping
  - Identify prefixes undergoing BGP changes as targets

Architecture: Assess Extent of Problem



- Traceroutes to gather topological data
  - Keep probing while problem persists
  - Every 15 minutes from 35 PlanetLab sites
- Analyze which traceroutes reach
  - BGP table to map addresses to ASes
  - Alias information to map interfaces to routers

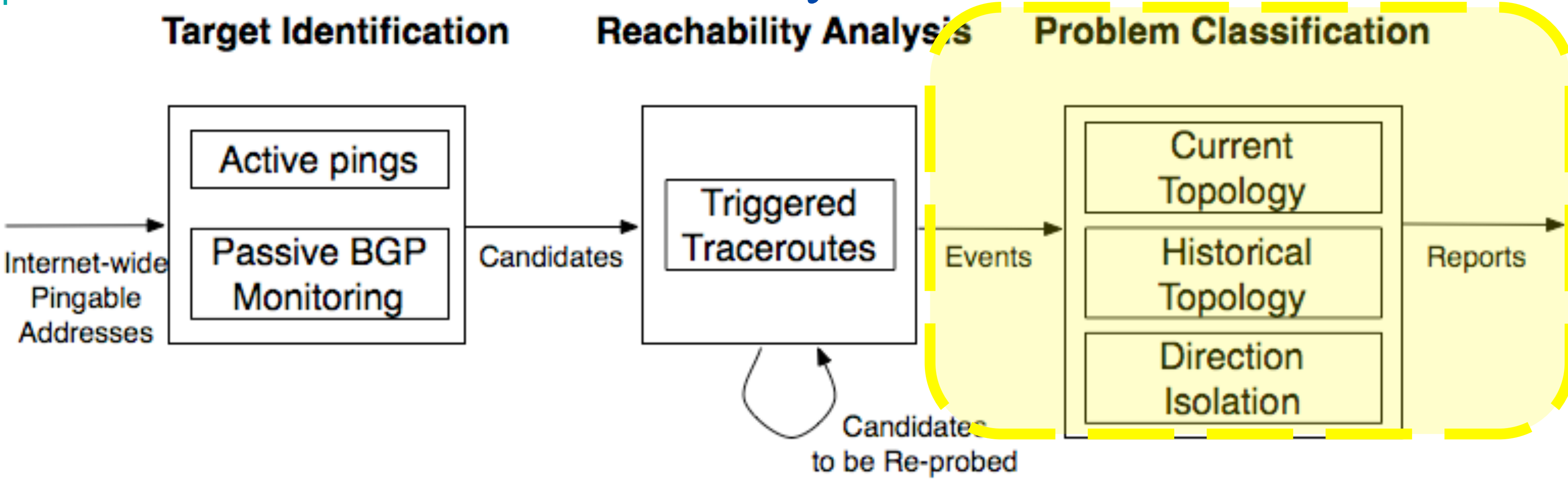
Architecture: Classify Problem



To aid operators in diagnosis and repair:

- Which ISP contains problem?
- Which routers?
- Which destinations?

Architecture: Classify Problem



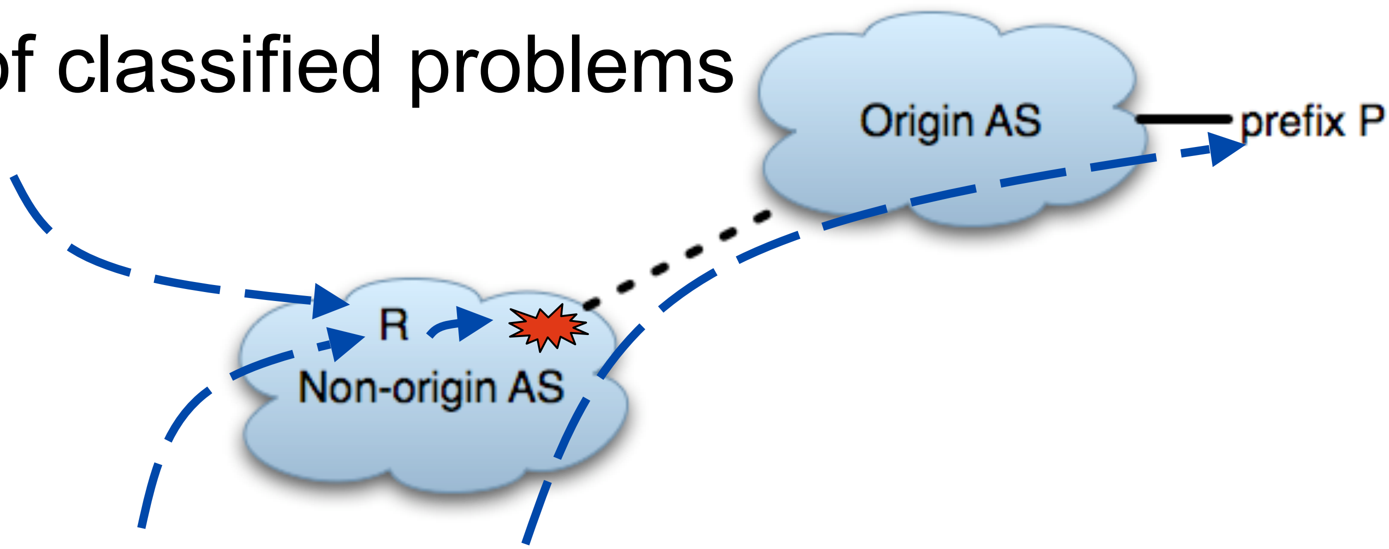
- Real-time, automated classification
- Find common entity that explains substantial number of failed traceroutes to a prefix
- Does not have to explain all failed traceroutes
- Not necessarily pinpointing exact failure

Classifying with Current Topology

- Group failed/successful traceroutes by last AS, router

Example: Router problem

- No probes reach ***P*** through router ***R***
- Some reach through ***R***'s AS
- 28% of classified problems

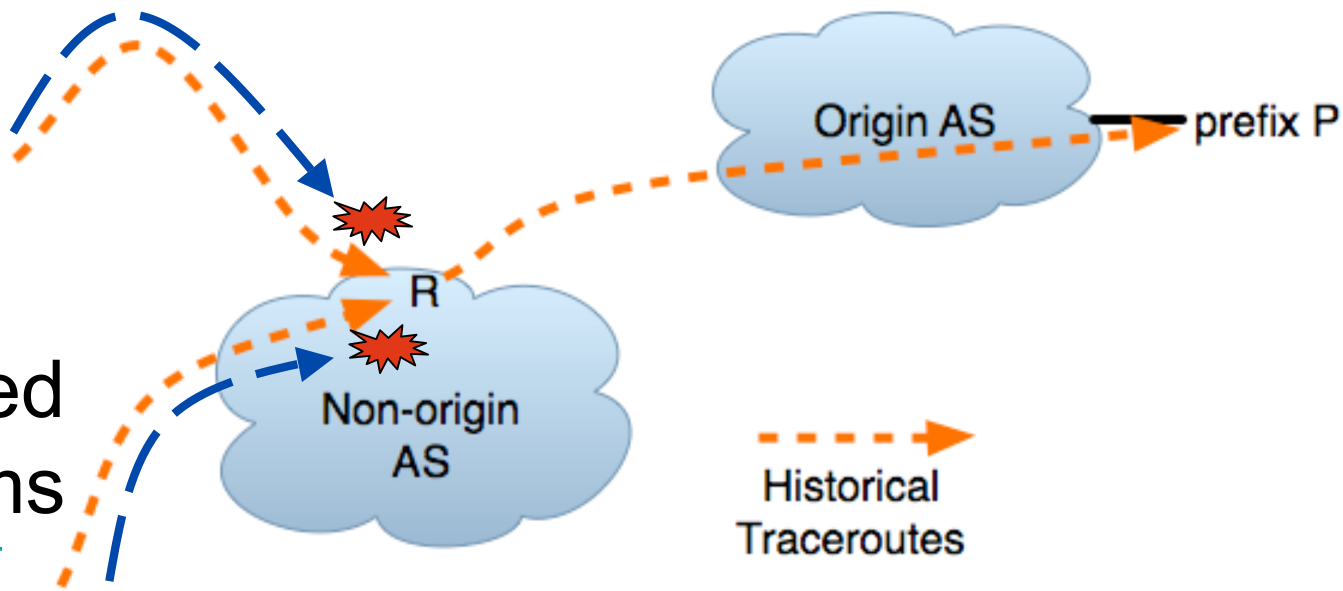


Classifying with Historical Topology

- Daily probes from PlanetLab to all prefixes
- Gives baseline view of paths before problems

Example: "Next hop" problem

- Paths previously converged on router ***R***
- Now terminate just before ***R***



- 14% of classified problems

Classifying with Direction Isolation

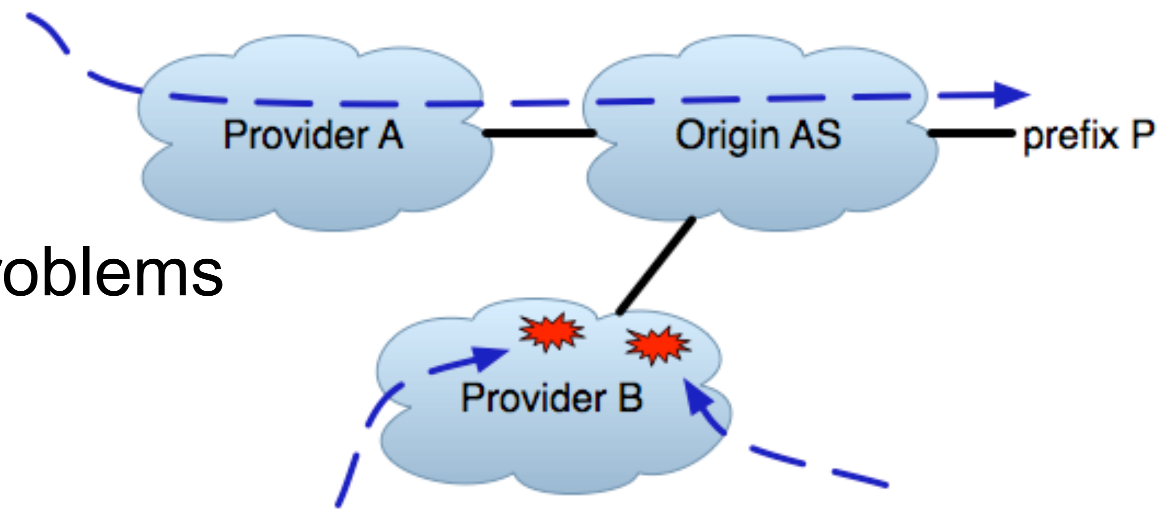
- Traceroutes only return routers on forward path
  - Might assume last hop is problem
  - Even so, require working reverse path
  - Hard to determine reverse path
- Internet paths can be asymmetric
- Isolate forward from reverse to test individually
- Without node behind problem, use spoofed probes
  - Spoof ***from S*** to check forward path from ***S***
  - Spoof ***as S*** to check reverse path back to ***S***

Classifying with Direction Isolation

- **Hubble** deployment on RON employs spoofed probes
  - 6 of 13 RON permit source spoofing
  - PlanetLab does not allow source spoofing

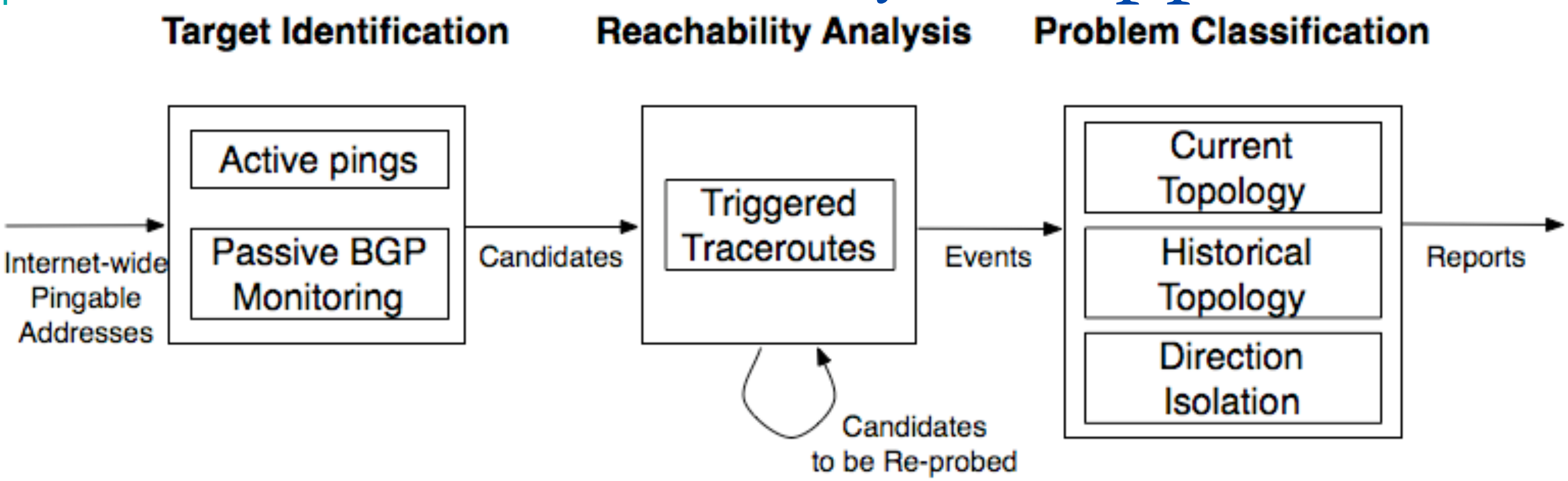
Example: Multi-homed provider problem

- Probes through Provider ***B*** fail
- Some reach through Provider ***A***
- Like Cox/USC



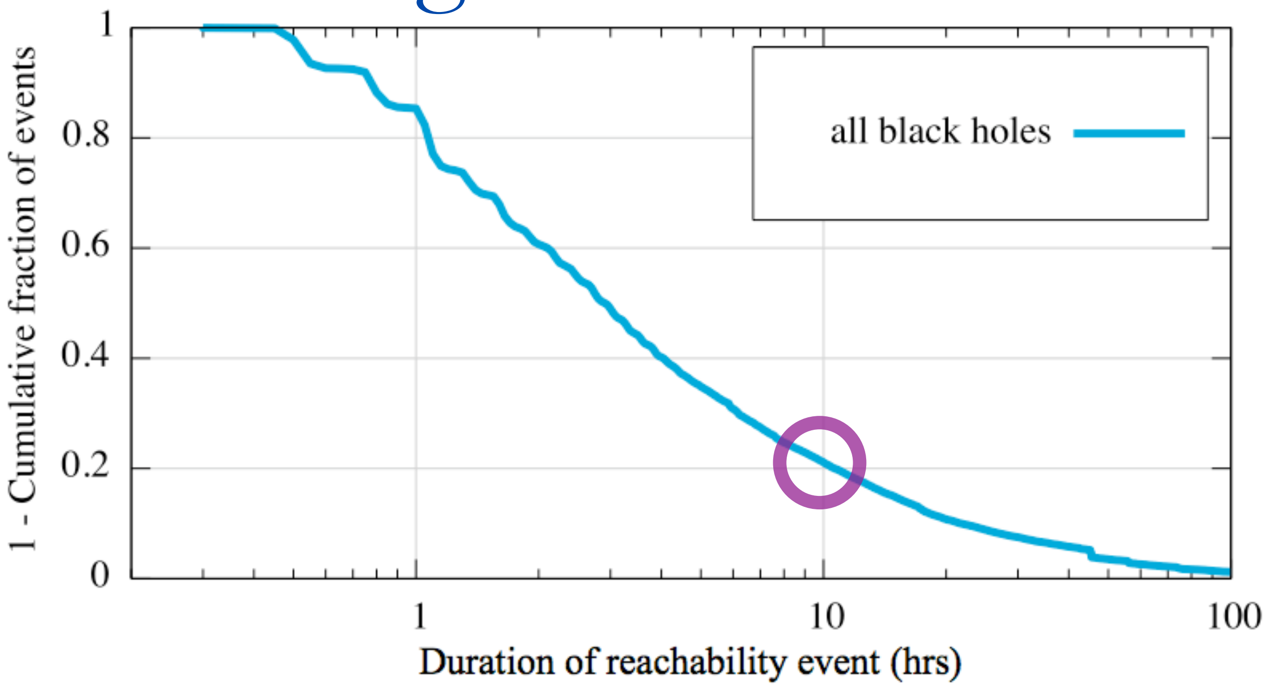
- 6% of classified problems

Architecture: Summary of Approach



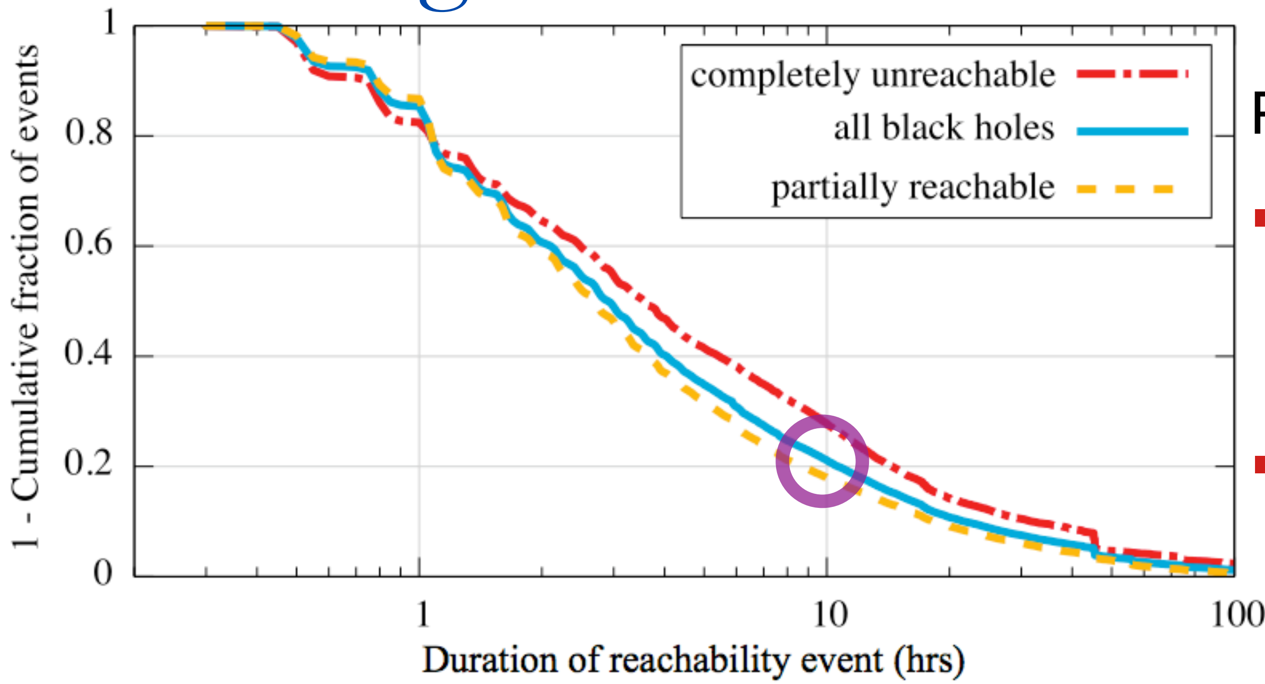
- Synthesis of multiple information sources
  - Passive monitoring of route advertisements
  - Active monitoring from distributed vantage points
- Historical monitoring data to enable troubleshooting
- Topological classification and spoofing point at problem

How long do black holes last?



- 3 week study starting September 17, 2007
- 31,000 black holes involving 10,000 prefixes
- 20% lasted at least 10 hours!
- 68% were cases of partial reachability

How long do black holes last?



Partial reachability:

- Can't be just hardware failure
- Configuration/ policy

- 3 week study starting September 17, 2007
- 31,000 black holes involving 10,000 prefixes
- 20% lasted at least 10 hours!
- 68% were cases of partial reachability

Other Measurement Results

- Can't find problems using only BGP updates
  - Only 38% of problems correlate with RouteViews updates
- Multi-homing may not give resilience against failure
  - 100s of multi-homed prefixes had provider problems like COX/USC, and **ALL** occurred on path **TO** prefix
- Inconsistencies across an AS
  - For an AS responsible for partial reachability, usually some paths work and some do not
- Path changes accompany failures
  - 3/4 router problems are with routers **NOT** on baseline path

Summary

- **Hubble**: working real-time system
- Lots of reachability problems, some long lasting
- Baseline/ fine-grained data enable classification

http://hubble.cs.washington.edu

Uses iPlane, MaxMind, Google Maps

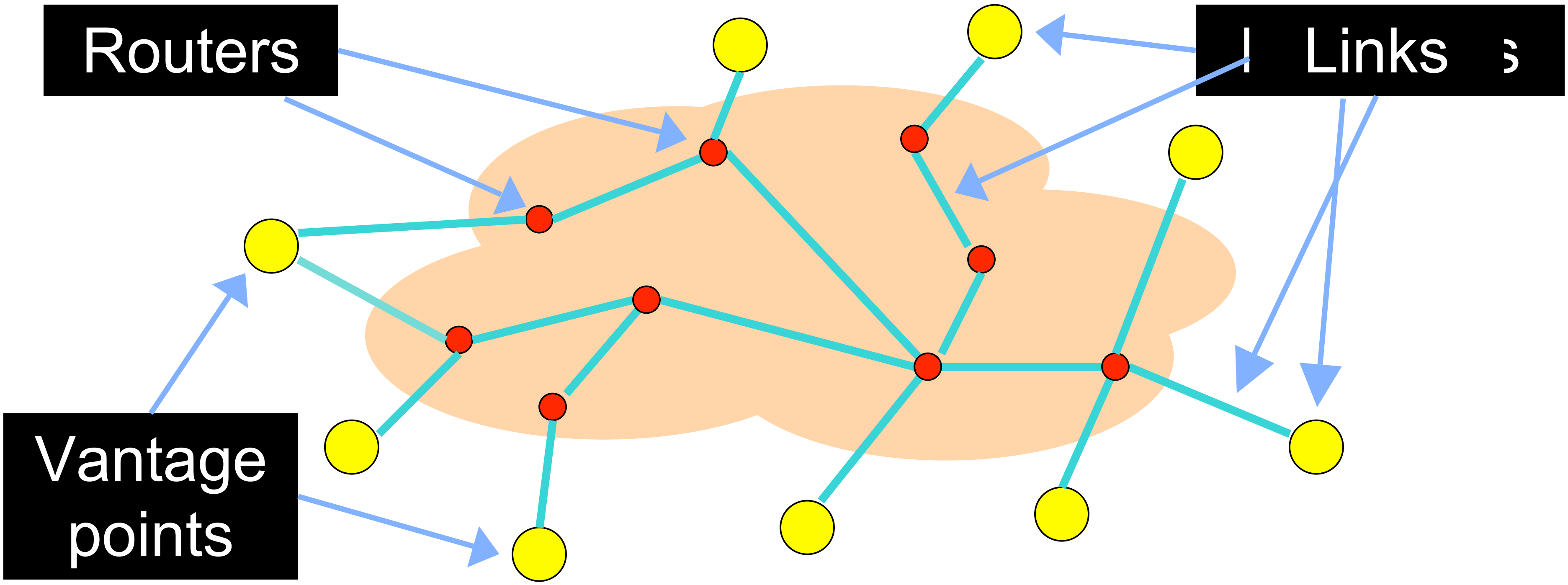
Beyond **Hubble**

- ***iPlane** overview*
  - Providing Internet path and path property predictions
  - Sibling/ parent to **Hubble**
    - Real Internet-scale measurement-based systems
- Ongoing work

iPlane Motivation and Goals

- Lots of distributed applications need path information
  - Google, Akamai, Amazon, BitTorrent, Skype, …
  - All need properties of Internet paths
- Every application measures the Internet independently
- Our goal: To understand how to predict path info
  - **Reusable**: across applications
  - **Scalable**: Internet-wide
  - **Efficient**: minimize measurements

iPlane: Building Internet Atlas



- Construct an “atlas” of the Internet topology
- Use the atlas to predict paths and path properties
- Think “Google Maps” for the Internet

iPlane Summarized

- Running as a real system for ~2 years
- Key pieces:
  - Structural approach: Enables predictions of multiple metrics
  - Path composition: Predict paths by composing observed path segments
  - Clustering: Internet-scale predictions by measuring at right granularity
  - Path selection: Infer routing policy from observed paths
  - Link measurement: Account for routing asymmetry
- Demonstrated utility of iPlane in helping distributed applications deliver better performance