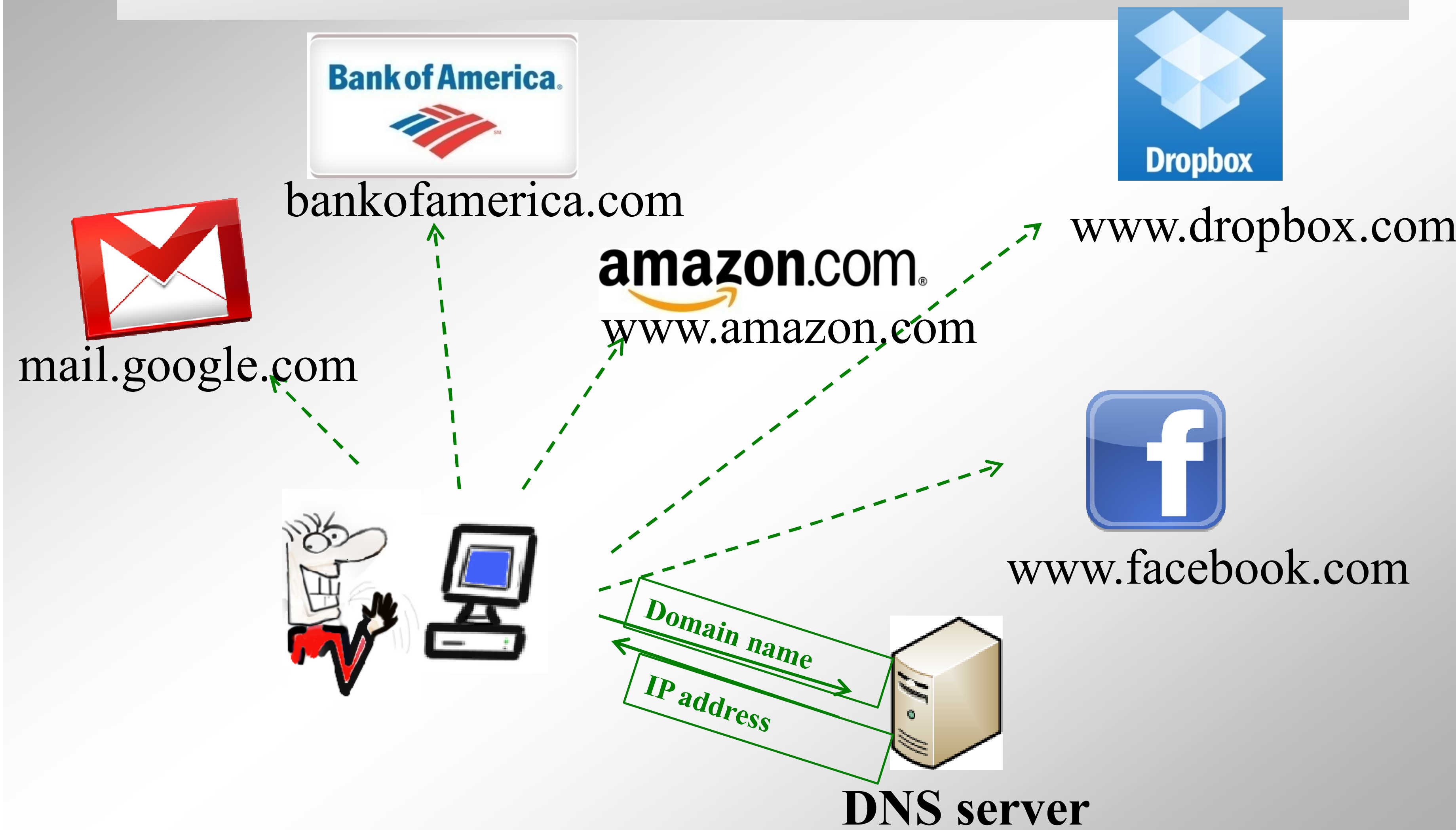
Evaluating Anycast in the Domain Name System

John Heidemann
joint work with Xun Fan and Ramesh Govindan
University of Southern California / Information Sciences Institute and Comp. Sci. Dept.
L-root evaluation with Joe Abley, ICANN

12 May 2013
DNS-OARC, Dublin, Ireland

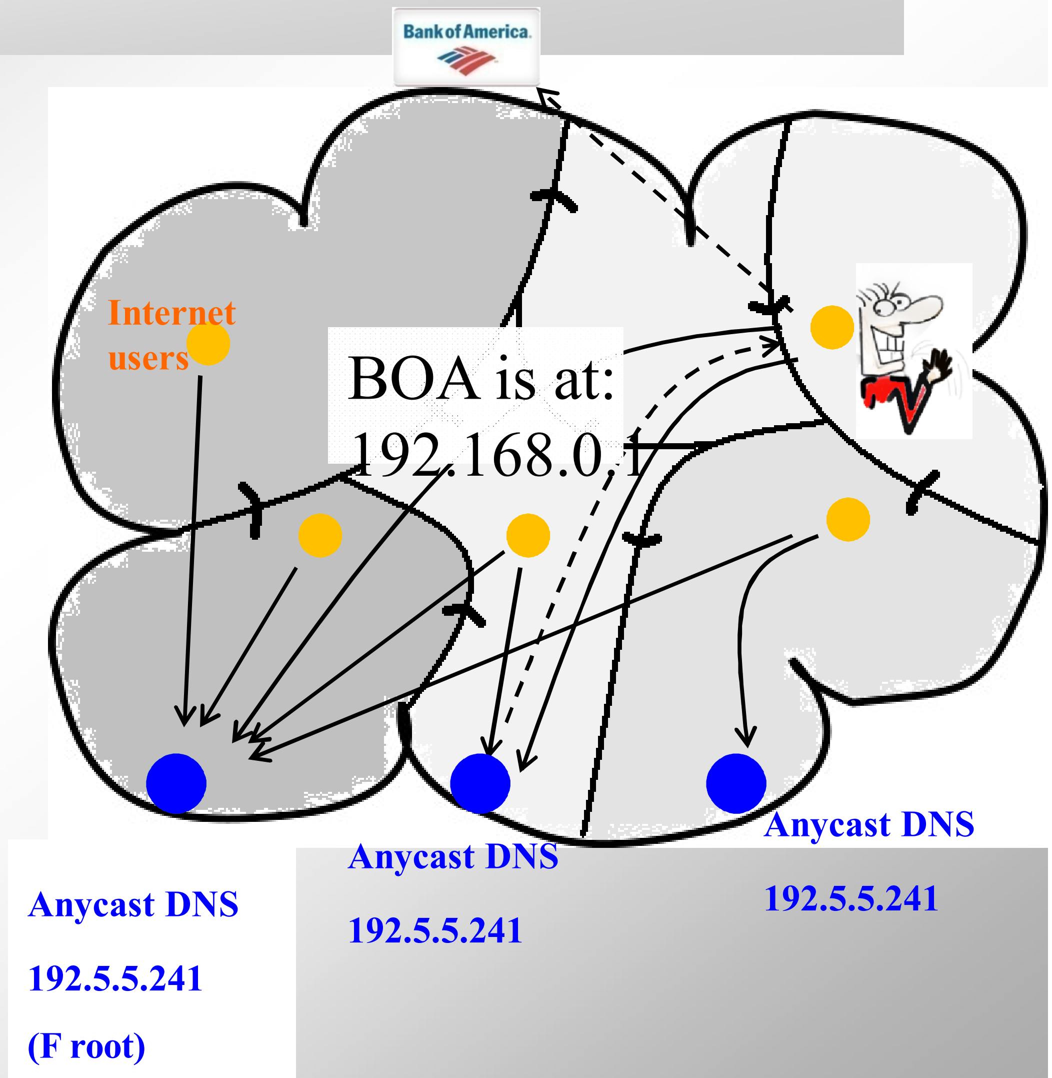
work supported by DHS S&T, Cyber Security Division
This research is sponsored by the Department of Homeland Security (DHS) Science and Technology Directorate, HSARPA, Cyber Security Division, BAA 11-01-RIKA and Air Force Research Laboratory, Information Directorate under agreement number FA8750-12-2-0344, and contract number D08PC75599. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views contained herein are those of the authors and do not necessarily represent those of DHS or the U.S. Government.

DNS is fundamental



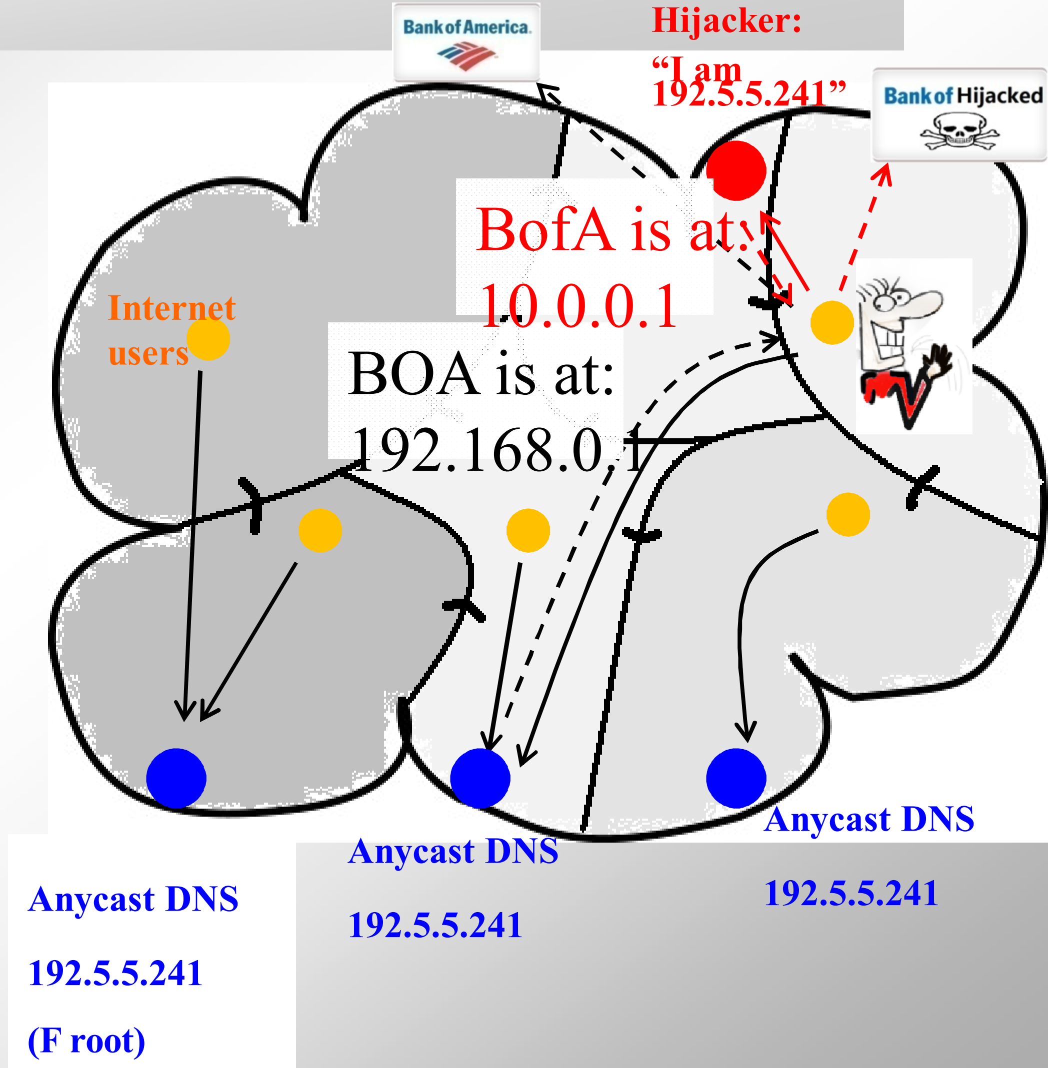
Many DNS services use anycast

- Previously: Unicast
- Anycast
  - Share one address (***anycast address***)
  - Available in multiple, locations (***anycast nodes***)
  - Each node has ***cachement area***



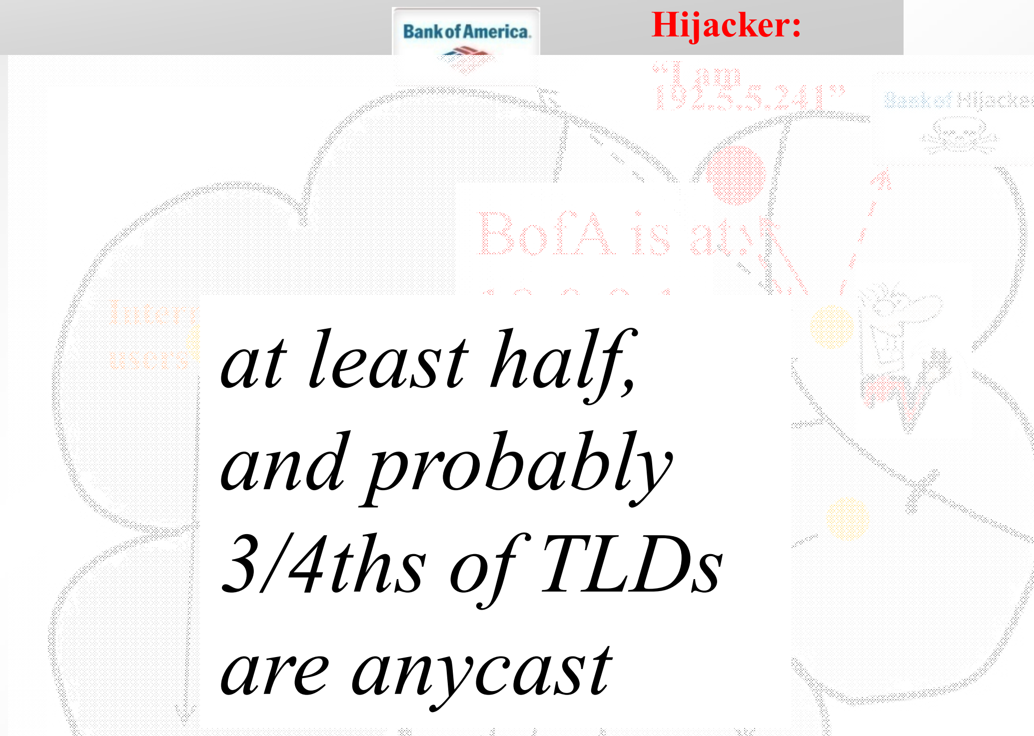
Many DNS services use anycast

- Previously: Unicast
- Anycast
  - Share one address (***anycast address***)
  - Available in multiple, locations (***anycast nodes***)
  - Each node has ***cachement area***
- Vulnerable to hijack



Many DNS services use anycast

- Previously: Unicast
- Anycast
  - Share one address (***anycast address***)
  - Available in multiple, locations (***anycast nodes***)
  - Each node has ***cachement area***
- Vulnerable to hijack
- Used in many DNS services
  - Root, TLD, public resolvers



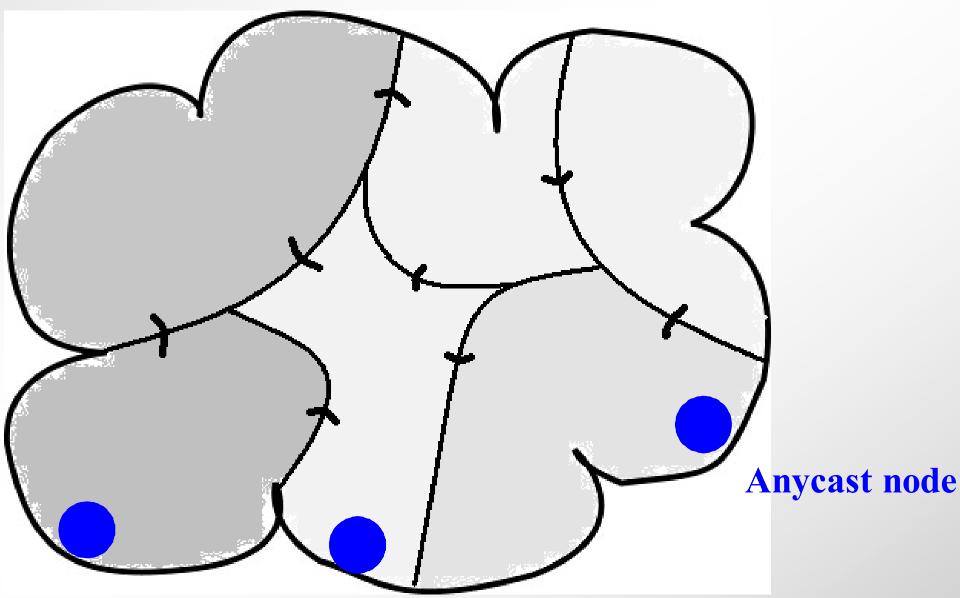
at least half, and probably 3/4ths of TLDs are anycast

Number of TLD names	definite anycast	possible anycast	higher bound
314 (100%)	**177 (56%)**	48	**225 (72%)**

TABLE VIII: *Anycast services discovered for TLD names*

Anycast Enumeration

- Which node responds to DNS query?
- How many **anycast nodes** are there?



  – No way to answer for whole Internet now

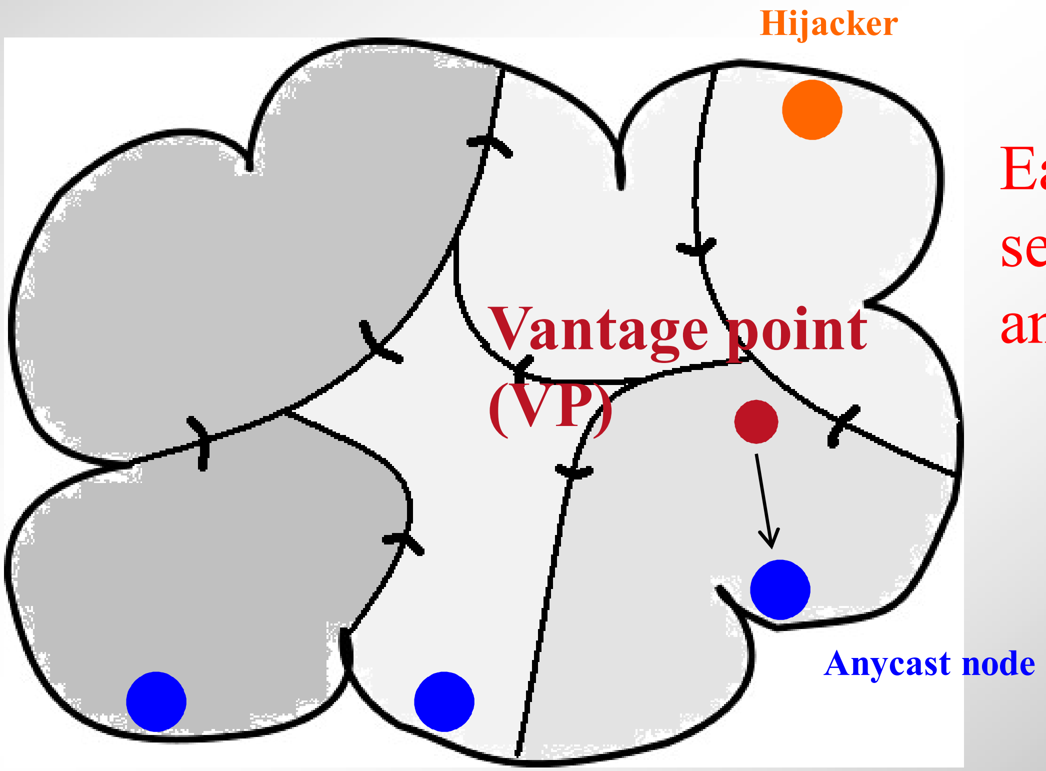
Many people care

- Anycast service providers
  - A “client-eye’s” view of the service
  - Any masquerading or hijacking?
- Purchaser of anycast services
  - Audit: “does the service I bought really have 60 nodes”?

Outline

- Methodology
- Validation
- Evaluation
- Conclusion

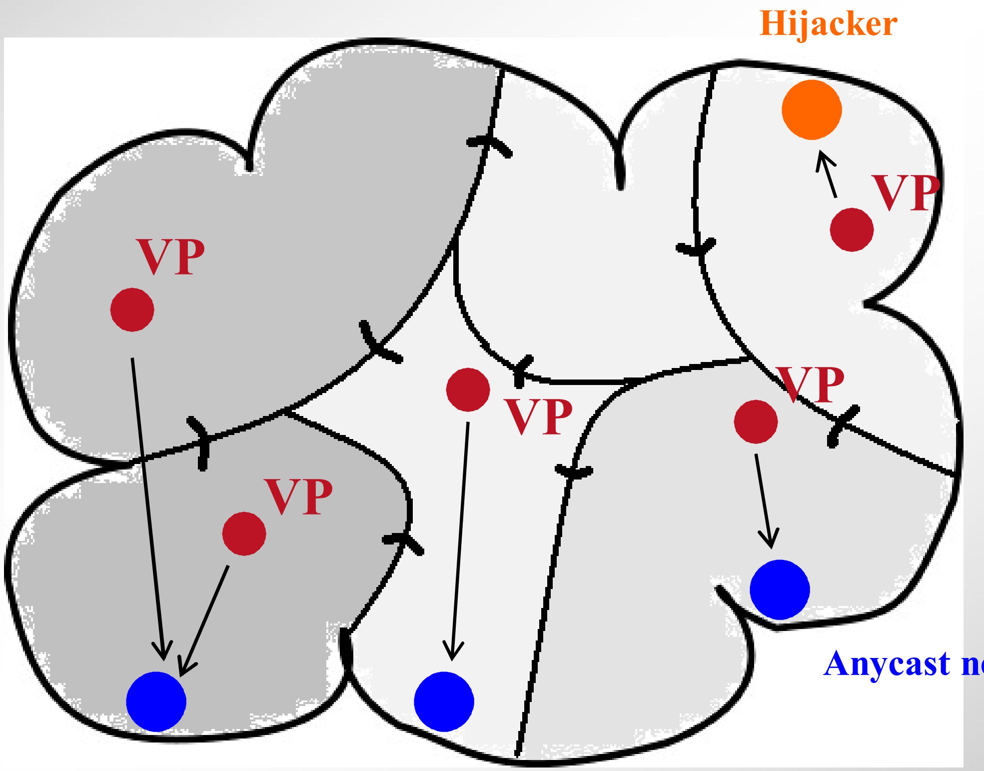
Enumeration challenges



Each VP can see only one anycast node

Approach: Active probes from multiple Vantage Points (VP)

VPs may find a same anycast node



In order to find all anycast nodes, we need ***multiple VPs*** and at least ***one VP for each node's catchment area***.

Our approach

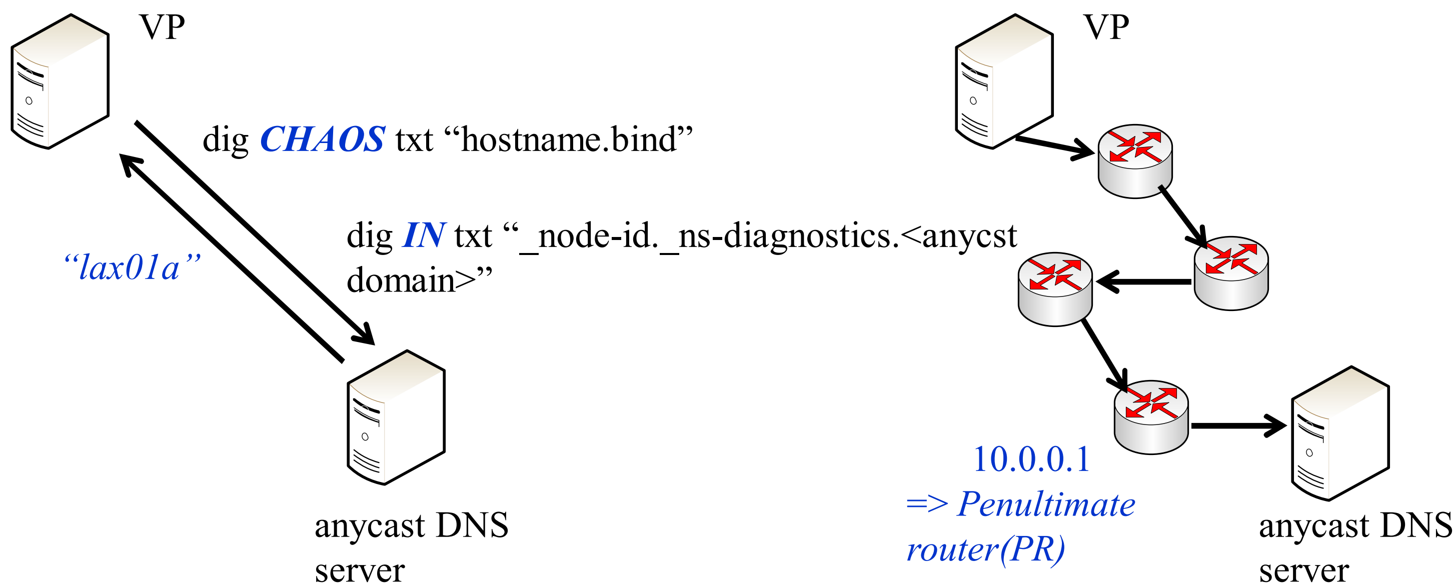
- Active query
  - Two existing mechanisms: DNS CHAOS query and traceroute
  - Our proposed method: DNS IN query
- Vantage points (VPs)
  - PlanetLab
  - User's browser
  - Open recursive name servers (rDNS)

Our approach

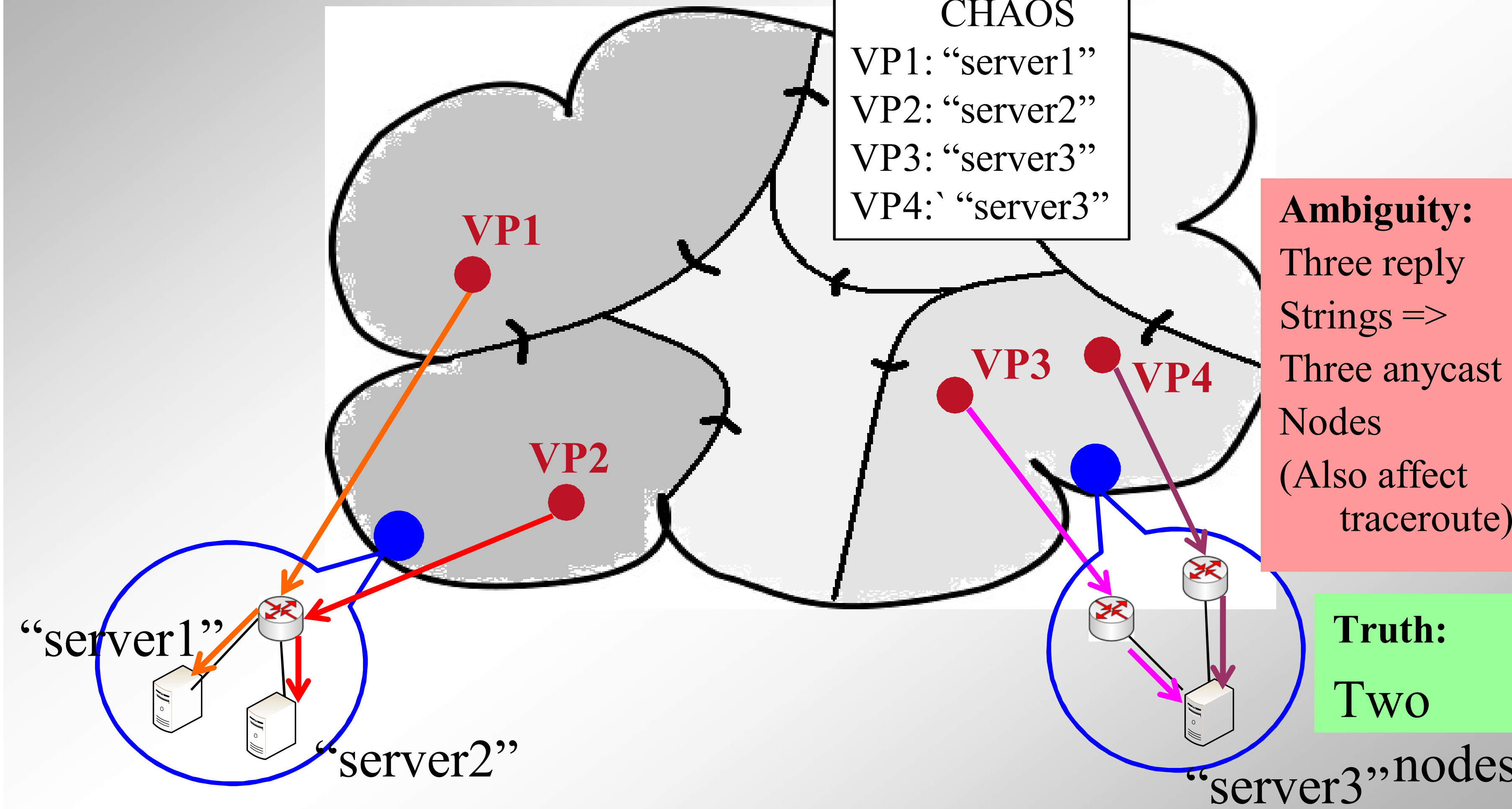
- Active query
  - Two existing mechanisms: DNS CHAOS query and traceroute
  - Our proposed method: DNS IN query
- Vantage points (VPs)
  - PlanetLab
  - User's browser
  - Open recursive name servers (rDNS)

Three kinds of active queries

Active query	DNS CHAOS query	Traceroute	DNS IN query
	• Existing mechanism, widely supported • Response not standardized => ambiguity	• Ambiguity problem • Combine with CHAOS query to solve ambiguity • Work with limited VPs	• Our proposed method • Standardize resrponse • Need support from DNS server-side • Work with many VPs (rDNS)



CHAOS query leads to ambiguity

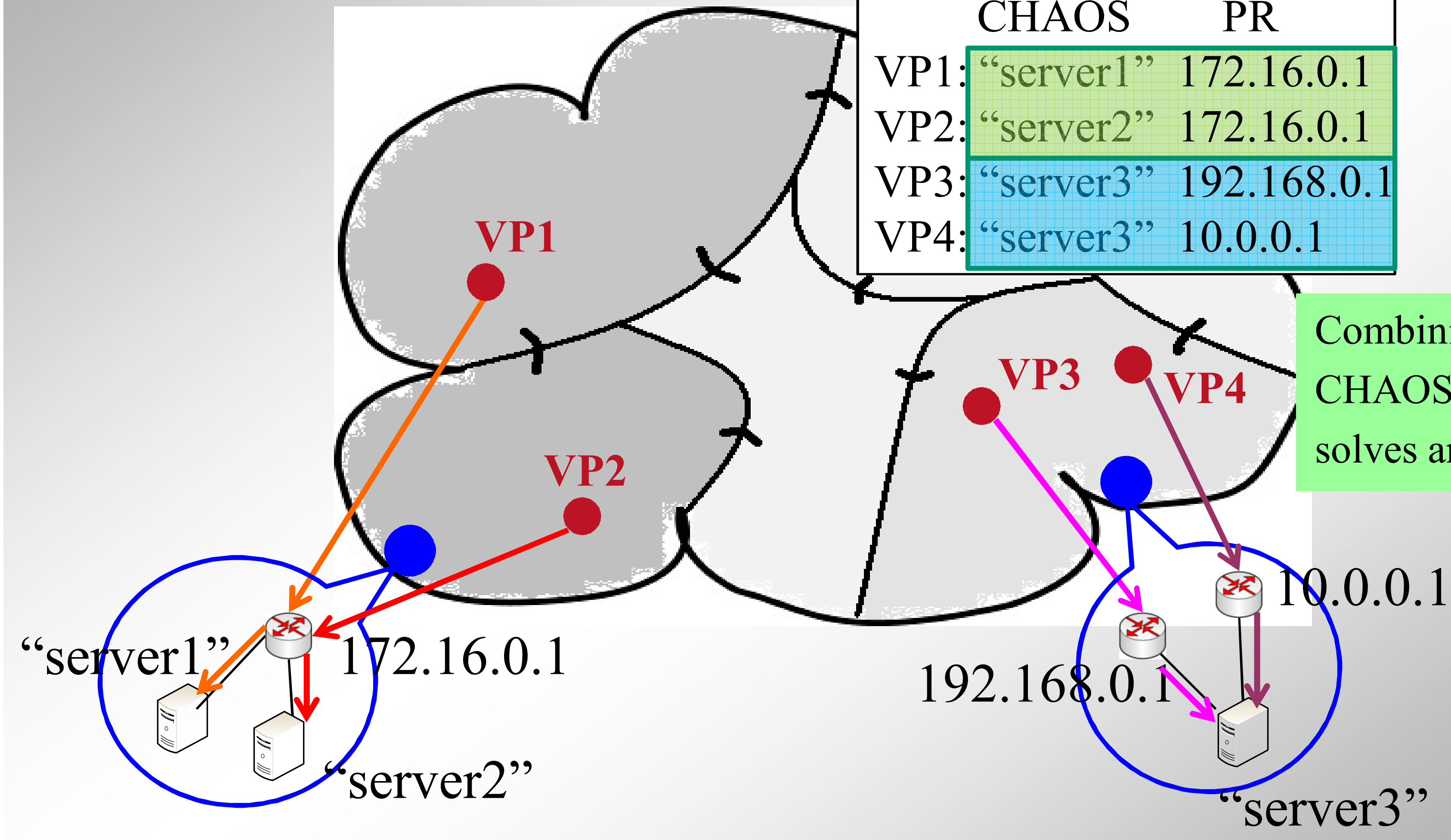


CHAOS
VP1: "server1"
VP2: "server2"
VP3: "server3"
VP4:` "server3"

Ambiguity:
Three reply Strings =>
Three anycast Nodes
(Also affect traceroute)

Truth:
Two

Combined method *solve* ambiguity



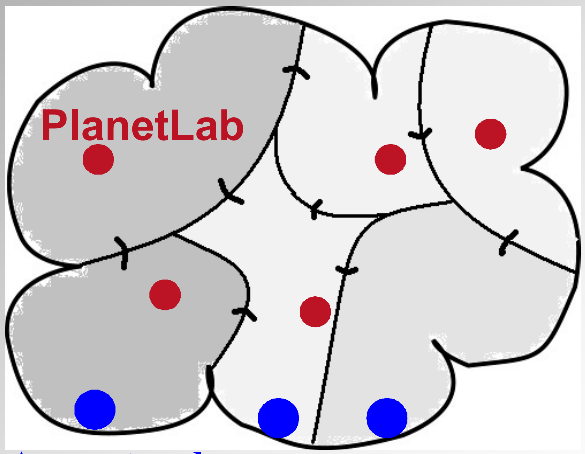
	CHAOS	PR
VP1:	"server1"	172.16.0.1
VP2:	"server2"	172.16.0.1
VP3:	"server3"	192.168.0.1
VP4:	"server3"	10.0.0.1

Combining CHAOS and PR solves ambiguity

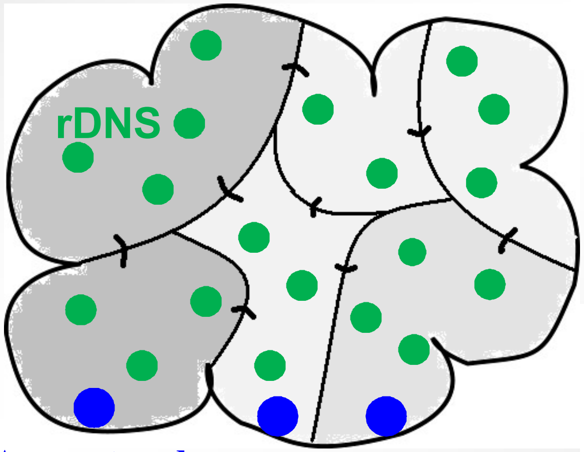
Our approach

- Active query
  - Two existing mechanisms: DNS CHAOS query and traceroute
  - Our proposed method: DNS IN query
- Vantage points (VPs)
  - PlanetLab
  - User's browser
  - Open recursive name servers (rDNS)

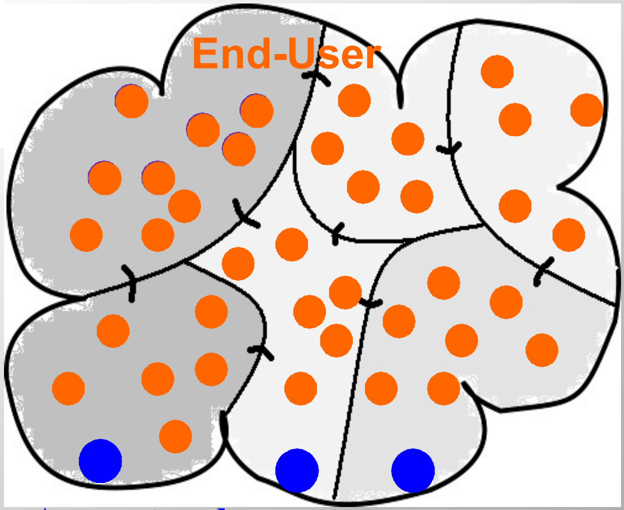
Three kinds of VPs



Anycast node



Anycast node



Anycast node

Internet end-users

- Pro: many users
  - We know 60k
  - Potential billions
- Con: Measurements depend on user action

PlanetLab

- Pro: run any query
- Con: few sites
  - We use 240
  - Potential 500 available sites

Recursive name servers

- Pro: many site
  - We known 320k
  - Potential 27M!
- Con: Only work with IN queries

Summary of approaches

Active probe	*DNS queries (CHAOS or IN) and traceroute*	*DNS queries (IN)*	*DNS queries (CHAOS)*
Source (Vantage points)	public research infrastructure (PlanetLab)	public operational infrastructure (recursive name servers: rDNS)	Clients' browser

- Applies to most anycast DNS services
- Now in operation
- Moderate recall

- Applies to specific anycast DNS services
- Good recall
- Plan to push to DNS community, positive feed back (ISC, PCH and AS112)

- Applies to most anycast DNS
- Good recall
- Depends on user activity

Outline

- Methodology
- Validation
- Evaluation
- Conclusion

Validation: metrics

- *Precision*: when we answer, is it true?
- *Recall*: how much of the truth do we find?
  - true positive/(true positive +false negative)

Validation with PlanetLab

- Target services: F-root and PCH
  - Large operational services with many nodes
  - Willing to share ground truth
- Results

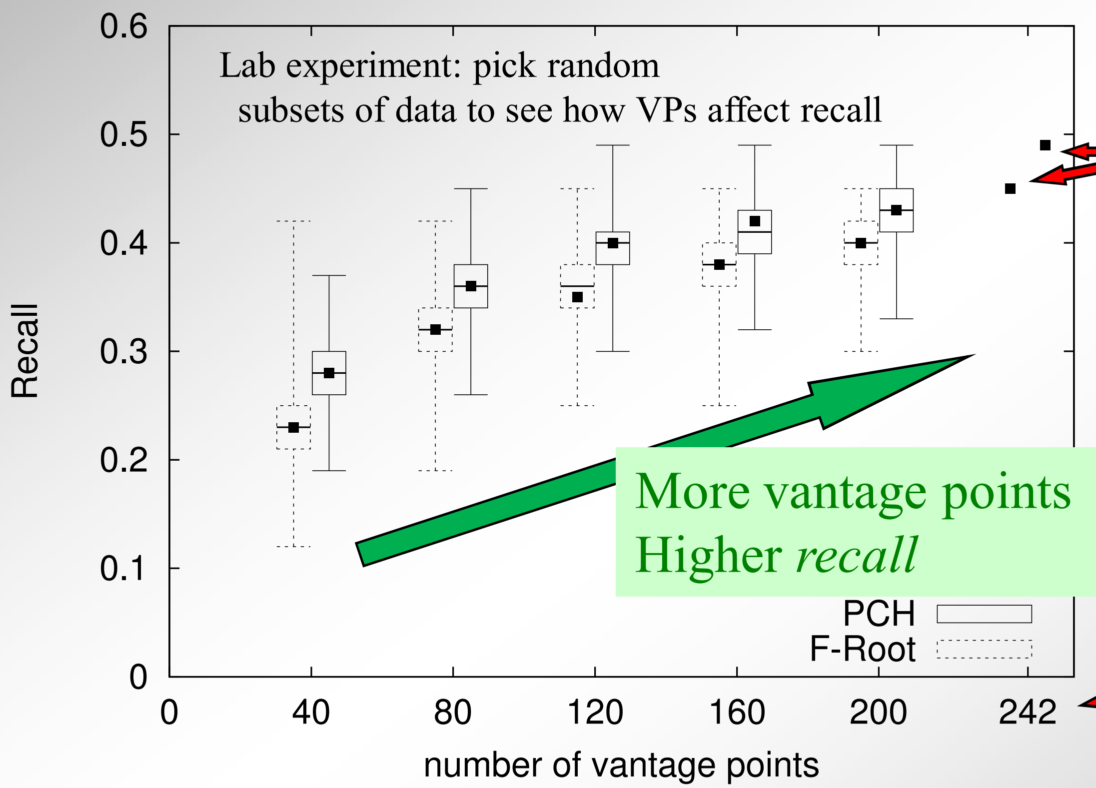
	CHAOS DNS query only		Traceroute only		Combined	
	Precision	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*
F-root	64%	45%	58%	38%	88%	45%
PCH	100%	49%	79%	49%	100%	49%

Modest Recal

Good precision: combination works!

USC Viterbi School of Engineering

More vantage points => More Recall



recall limited by number of vantage points

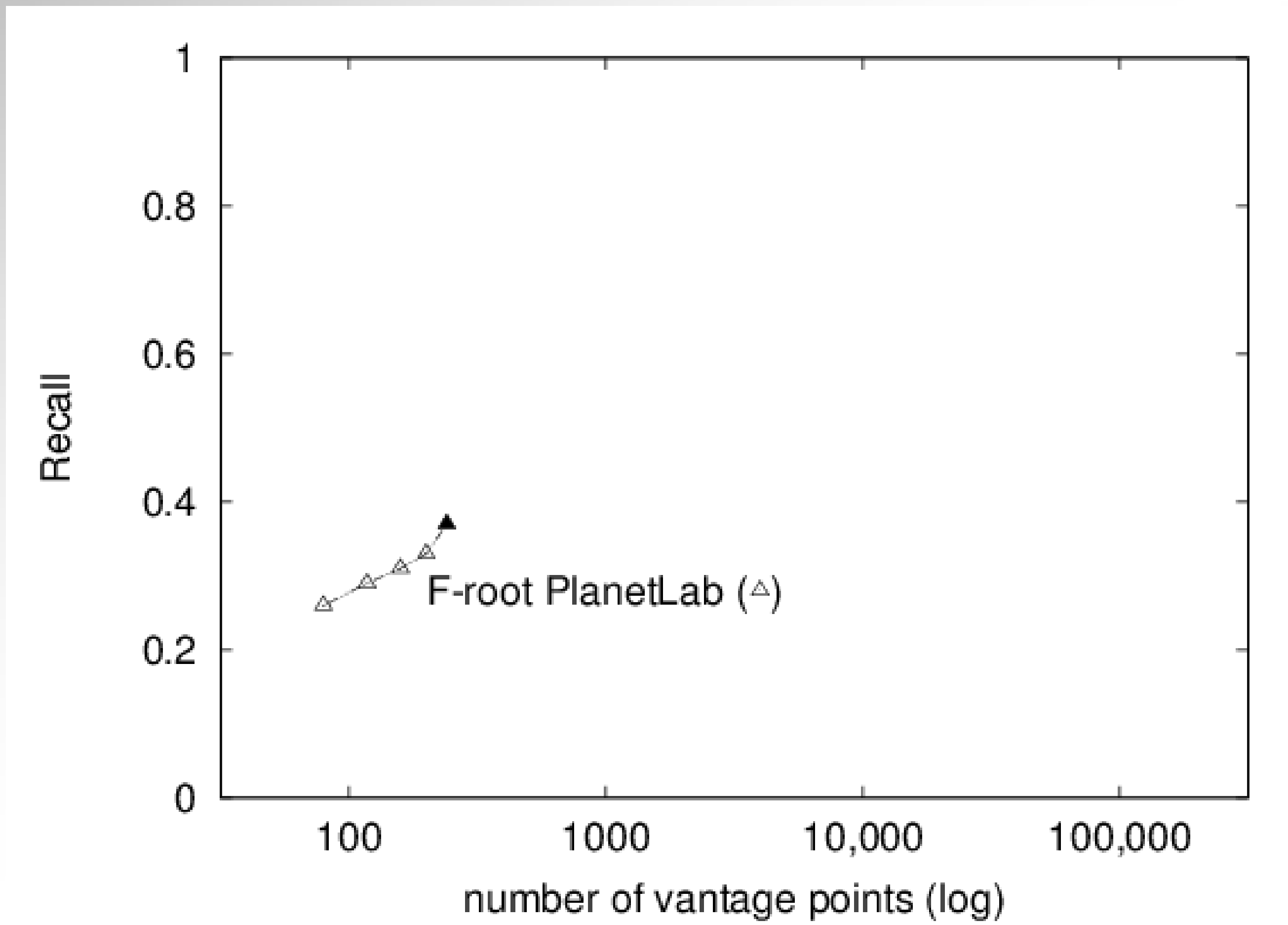
only ~240 vantage points in PlanetLab

⇒ to increase recall, need many vantage points!

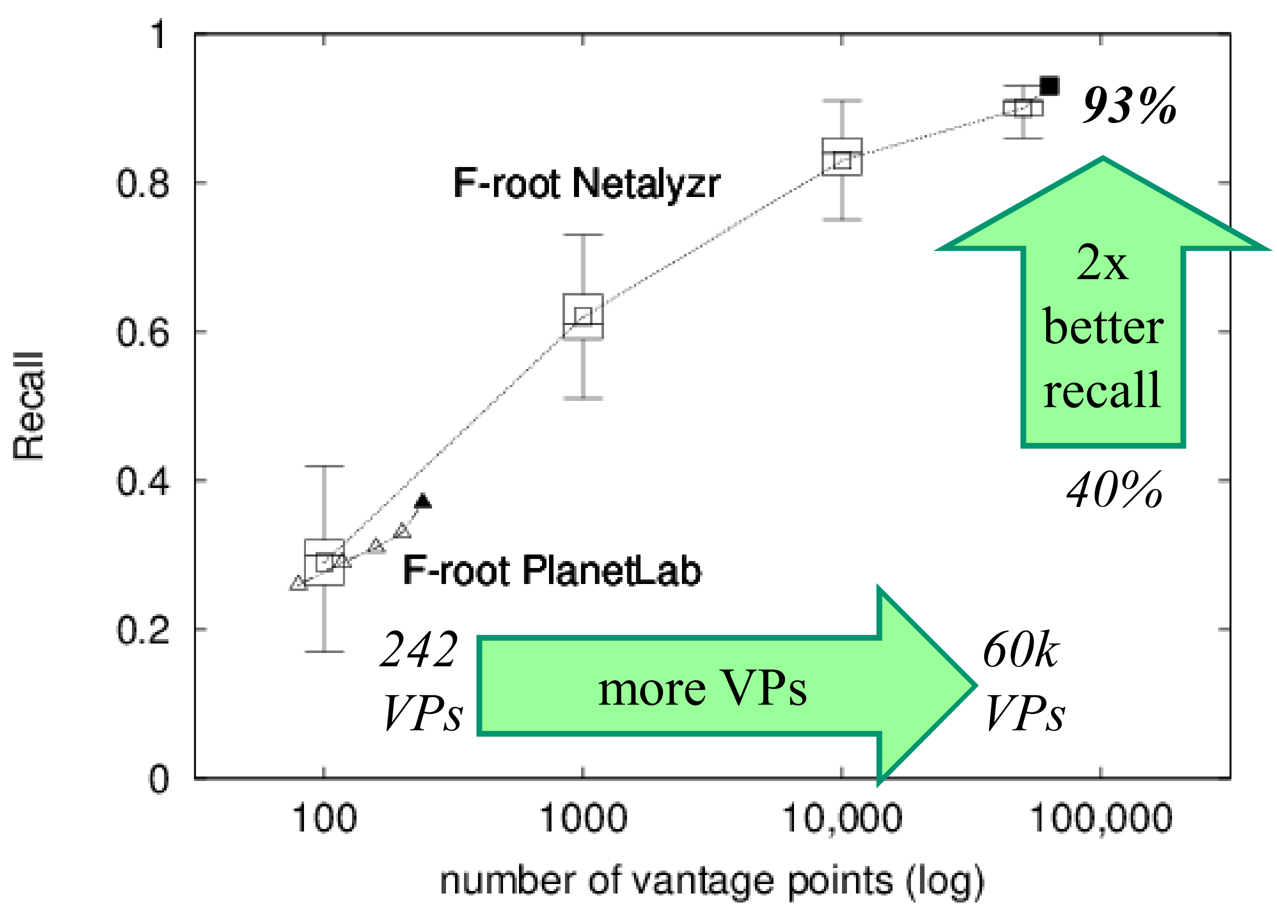
VPs from end-users: CHAOS queries from Netalyzr

- ICSI Netalyzr: a network debugging tool
  - inspired by our work,
    Nick Weaver added CHAOS queries to Netalyzr
  - they shared 4 months of Netalyzr data (thanks!)
- 61,914 Vantage Points
  - each a unique IP address
  - 164 countries, 4153 ASes
  - many users (not just geeks; likely unbiased)
- Long collection time:
  2011-11-30 to 2012-04-01

CHAOS queries with end users: improved recall



CHAOS queries with end users: improved recall



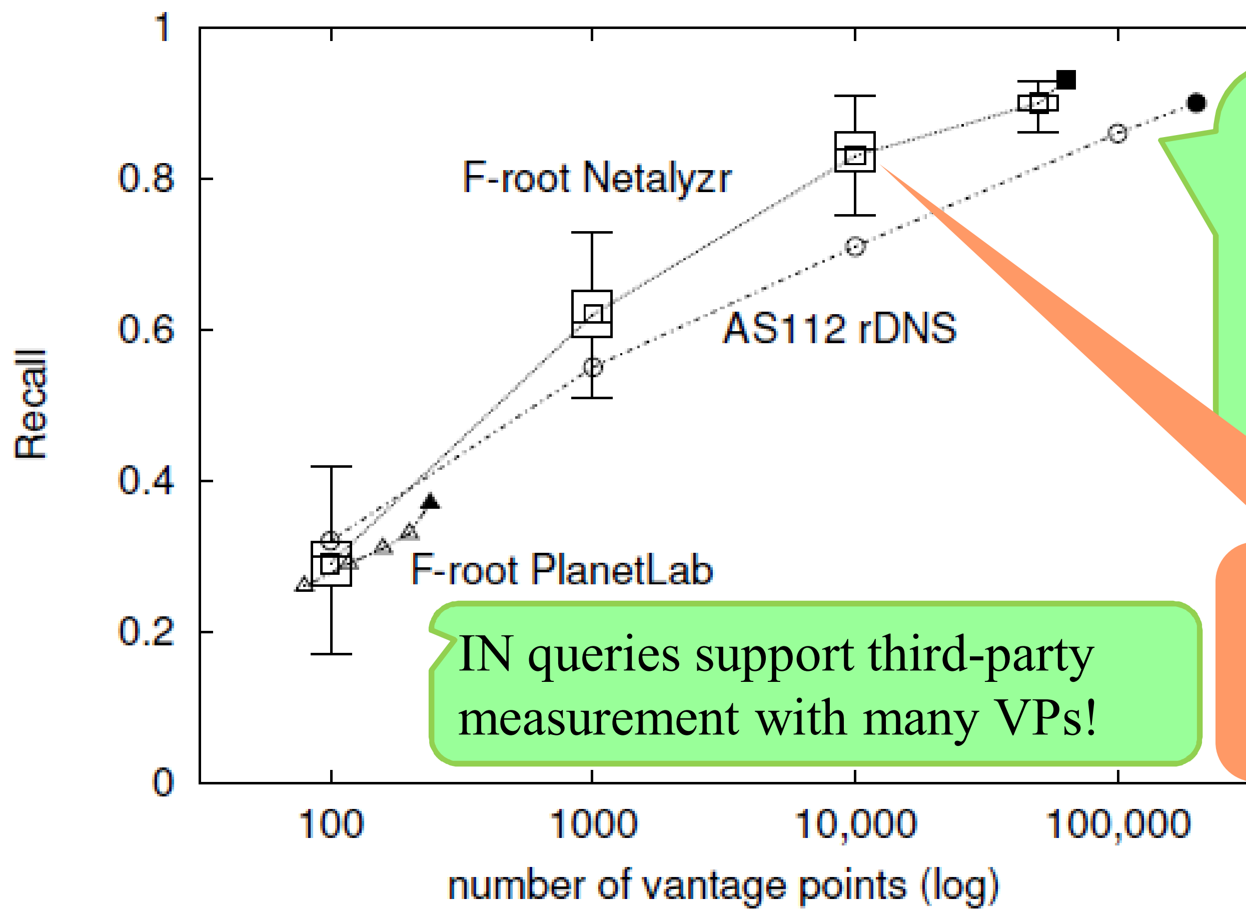
Recursive DNS: Improving on CHAOS

- problems with CHAOS
  - no standard response => interpretation is ambiguious
  - requires direct DNS queries (not recursive) => VPs hard to get
    - ~240 with PlanetLab
    - 60k with Netalyzr
  - end-user queries cannot be done on demand: Netalyzr takes 4 months
- proposal: new type of IN DNS query
  - new TXT record
  - standard name and reponse
- benefits:
  - works with rDNS => millions of potential VPs
  - on-demand rapid => one hour to query 200k rDNS

How to Validate IN Queries?

- Problem: no anycast does our IN approach today
- Solution:
  - study AS112 DNS service (reverse DNS for private addrs)
  - it implements something close to our scheme
  - serves as proxy for our approach
- Details
  - test with 320k recursive DNS as VPs
    - 220 countries/regions
    - 15,210 ASes
  - compare to published AS112 server list as ground truth
  - data taken January 2012

IN queries with rDNS: good recall



Same recall as CHAOS records (difference because of target: F-root vs. AS112)

10,000 VPs are needed to find 80% of nodes

IN queries support third-party measurement with many VPs!

Outline

- Methodology
- Validation
- Evaluation
  - Find masquerader
  - Published vs. measured: TLDs and AS112
  - Potential anycast use in TLDs
  - L-Root Analysis
- Conclusion

Evaluation: found masquerader

- Approach: CHAOS query + traceroute with PlanetLab
- Found a masquerading F-root node in CERNET, China
  - CHAOS record: none
  - Traceroute (last router): 202.112.36.246
  - Not malicious, within CERNET's right,
    but may be surprising to their users
- Confirmed with ISC: not one of theirs
  - they know masquerading happens
  - but no way to systematically track (until our work!)

Evaluation: Published vs. Measured Anycast

- AS112 and Root DNS publish lists
- questions:
  - how complete are they? (what they miss)
  - help understand our method? (what we miss)
  - how inaccurate are they? (what they shouldn't have)

- root DNS data as of April 2012 (May 2011 data is similar)
- AS112 data as of January 2012

AS112: Published vs. Measured

what they miss:
35 nodes: manual lists are often incomplete

what we miss:
7 of 70 nodes: we need many VPs

what they shouldn't have: 18 of 70 nodes no longer reply

				authority	**rDNS**
Found by rDNS, but not in ground truth	35			missing	**new**
Operator list (authority truth)	70	100%		both known	
node alive	37	53%			
found by BGP information (and not rDNS)	7			known	*missing*
found by rDNS	30	42%		both known	
found by PlanetLab	14	20%		both known	
node down	18	26%		out-of-date	**corrected**
hard to judge	15	21%		interpretation uncertain	
Conservative ground truth (37 + 15 + 35)	87		100%		
found by rDNS (30 + 35)	65	(Conservative recall)	75%		
Realistic ground truth (37 + 35)	72		100%		
found by rDNS (30 + 35)	65	(Realistic recall)	90%		

TABLE I: *Evaluation of IN queries coverage compared to the AS112 providers list as ground truth.*

DNS Roots: Published vs. Measured

DNS root servers	measured		published	found
A (Verisign)	2	<	**6**	33%
B (ISI)	1	=	1	100%
C (Cogent)	6	=	6	100%
D (Univ. of Maryland)	1	=	1	100%
E (NASA)	9	>	1	900%
F (ISC)	**53**	>	49	108%
G (DISA)	6	=	6	100%
H (U.S. ARL)	**3**	>	2	150%
I (Automica)	**39**	>	38	103%
J (Verisign)	59	<	**70**	84%
K (RIPE)	17	<	**18**	94%
L (ICANN)	78	<	**107**	73%
M (WIDE)	6	=	6	100%

TABLE V: *Comparing measured against published numbers of anycast nodes for all anycast root servers.*

what they miss:
4 operators have deployments no listed at root-servers.org

us wrong one case:
H-root ops: 3 instances at 1 node (we need traceroute, not just CHAOS)

what we miss:
incomplete in 4 cases (but usually >70%)

Evaluation: anycast in TLDs

- Method: CHAOS query + traceroute with PlanetLab
- Target 314 top level domains (CCTLD+GTLD)

Number of TLD names	definite anycast	possible anycast	higher bound
314 (100%)	**177 (56%)**	48	**225** (72%)

Possibly 72% of TLDs are using anycast. (As of May 2012)

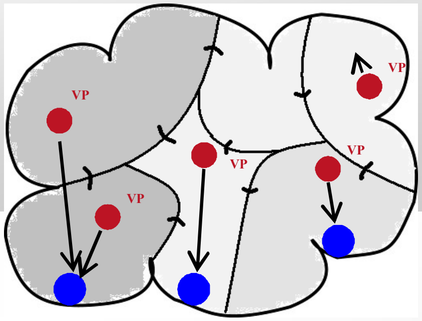
Evaluation: L-Root

- help from Joe Abley to study of L-Root
- L-Root has IN-records
  - TXT and A for identity.l.root-servers.org @beacon.l.root-servers.org
  - implemented as 2nd server in l-root prefix (same anycast)
  - details in I-D *draft-jabley-dnsop-anycast-mapping-01*
- findings
  - 237 of 273 **(87%)** with 200k rDNS VPs in Jan. 2013
- implications:
  - confirms many VPs help recall
  - example of parallel architecture to support diagnosis
  - boy L-Root is building out their infrastructure :-)

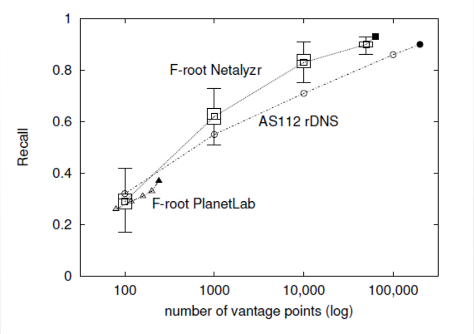
Where From Here?

- we'd love feedback about this work
- and about possible next steps
  - interest in standardizing IN records?
  - need operator control of enumeration?
    - we have some ideas to control access

Conclusions



- New approaches to enumerate anycast
  - good recall
  - new method improves recall vs. prior methods
- Evaluation of current anycast deployments
- Feedback about where next?



- more info:
  - peer-reviewed paper: Fan et al. "Evaluating Anycast in the Domain Name System", INFOCOM 2013, at http://www.isi.edu/~johnh/PAPERS/Fan13a.html
  - more detail in Tech Report: ftp://ftp.isi.edu/isi-pubs/tr-681.pdf
  - dataset: http://www.isi.edu/ant/traces/anycast/index.html