

Towards a global IP Anycast service

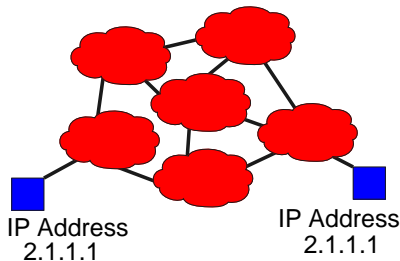
Hitesh Ballani, Paul Francis

Cornell University

ACM SIGCOMM 2005

What is IP Anycast?

One-to-Any
communication
with **no changes** to
Internet routing

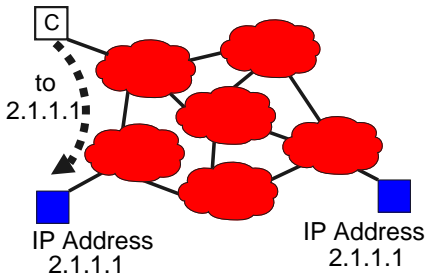


Robust and efficient service discovery

- ▶ Query-Reply Services : DNS Root-Servers etc.
- ▶ Routing Services : IPv6 transition (6to4) etc.

What is IP Anycast?

One-to-Any
communication
with **no changes** to
Internet routing

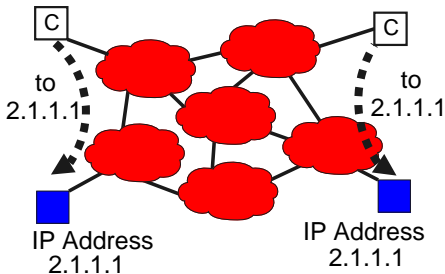


Robust and efficient service discovery

- ▶ Query-Reply Services : DNS Root-Servers etc.
- ▶ Routing Services : IPv6 transition (6to4) etc.

What is IP Anycast?

One-to-Any
communication
with **no changes** to
Internet routing

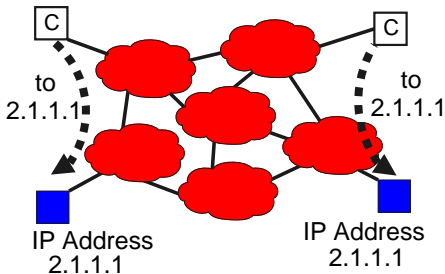


Robust and efficient service discovery

- ▶ Query-Reply Services : DNS Root-Servers etc.
- ▶ Routing Services : IPv6 transition (6to4) etc.

What is IP Anycast?

One-to-Any
communication
with **no changes** to
Internet routing



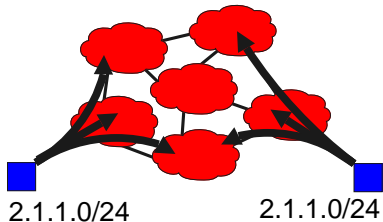
Robust and efficient service discovery

- ▶ Query-Reply Services : DNS Root-Servers etc.
- ▶ Routing Services : IPv6 transition (6to4) etc.

But its use has been limited?

Limitations of Inter-domain IP Anycast

- ▶ Wastes address space
- ▶ Does not scale by number of groups
- ▶ Difficult to deploy
 - ▶ obtain an address prefix
 - ▶ a certain level of expertise
- ▶ Is limited by IP routing
 - ▶ inability to offer load-based selection



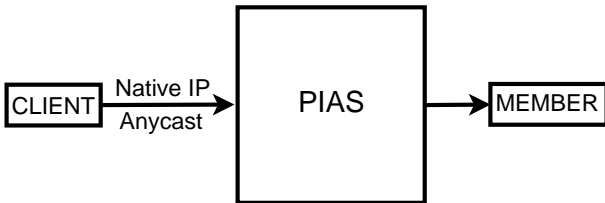
Proxy IP Anycast Service (PIAS)

What is PIAS?

A practical anycast deployment architecture

- ▶ addresses native IP Anycast limitations
- ▶ offers new features
 - ▶ opens new anycast usage avenues

Key Insight



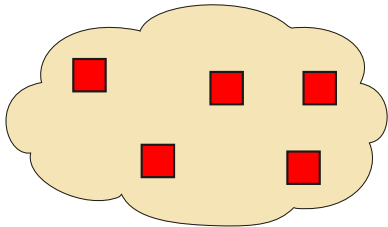
Talk outline


- ▶ PIAS : basic design
- ▶ Design challenges
- ▶ PIAS : actual design
- ▶ New anycast applications
- ▶ Unanswered questions
(simulations/measurements)

PIAS: Basic Idea

Deploy Anycast Proxies

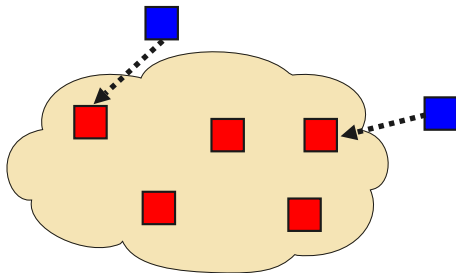
All proxies advertise the same prefix



 Anycast
Proxy

PIAS: Basic Idea

Group Members register with proxies

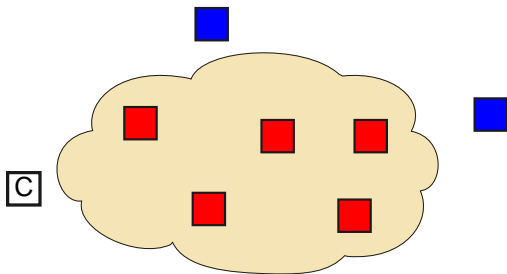


.....→ IP Anycast

 Anycast Proxy  Member (group 1)

PIAS: Basic Idea

Client (C) \Rightarrow Group 1 (blue group)

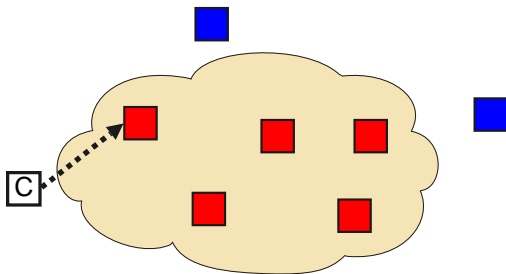


..... \rightarrow IP Anycast

 Anycast Proxy  Member (group 1)

PIAS: Basic Idea

Native IP Anycast delivers packets to proxies

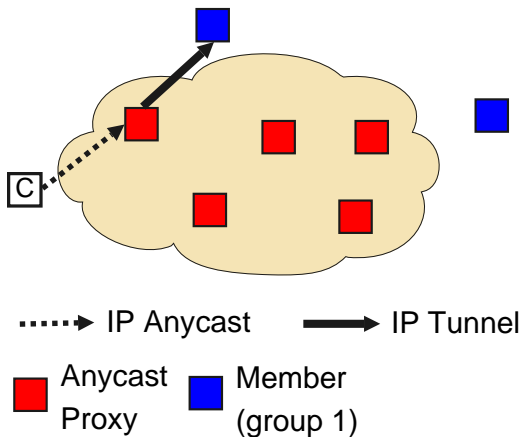


.....→ IP Anycast

 Anycast Proxy  Member (group 1)

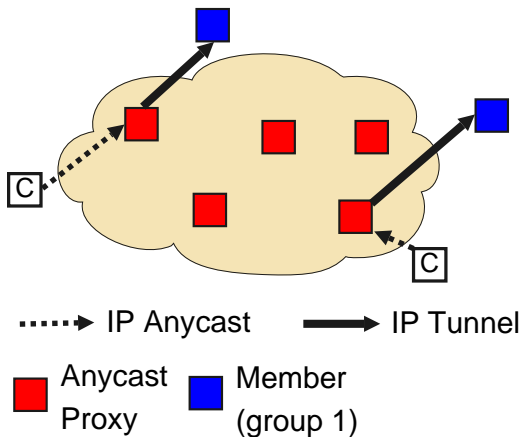
PIAS: Basic Idea

Proxies tunnel to appropriate member



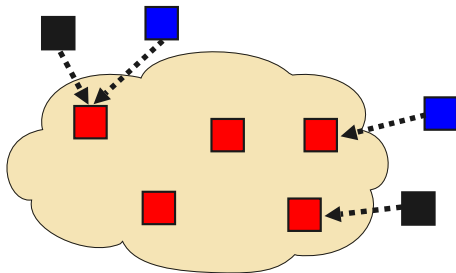
PIAS: Basic Idea

Different client might go to a different member






PIAS: Basic Idea

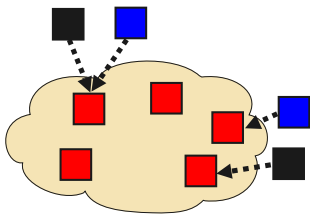
Multiple groups can register



.....➔ IP Anycast

 Anycast Proxy  Member (group 1)  Member (group 2)

What does PIAS solve?

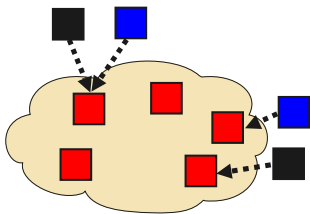


- ▶ Address Usage
- ▶ Effort Amortization
- ▶ Ease-of-Use
- ▶ Backwards Compatible
- ▶ Selection Criteria

Efficient use of address space

Thousands of groups per IP address in prefix
Group address - [IP-Address]:[Port]

What does PIAS solve?

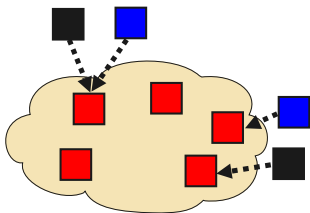


- ▶ Address Usage
- ▶ **Effort Amortization**
- ▶ Ease-of-Use
- ▶ Backwards Compatible
- ▶ Selection Criteria

Amortization of effort

Deployment effort spread across thousands of groups

What does PIAS solve?

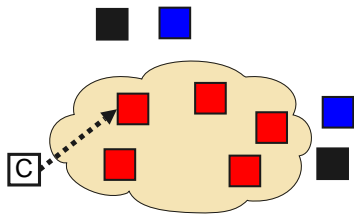


- ▶ Address Usage
- ▶ Effort Amortization
- ▶ **Ease-of-Use**
- ▶ Backwards Compatible
- ▶ Selection Criteria

Ease of join/leave

No interaction with routing

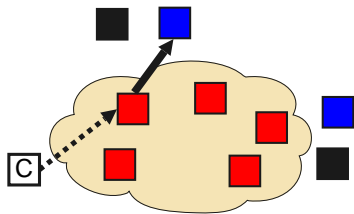
What does PIAS solve?



- ▶ Address Usage
- ▶ Effort Amortization
- ▶ Ease-of-Use
- ▶ **Backwards Compatible**
- ▶ Selection Criteria

No changes to clients
just as native IP Anycast

What does PIAS solve?

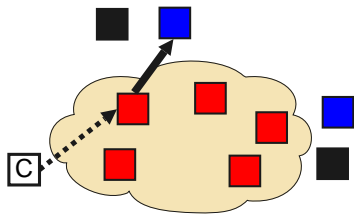


- ▶ Address Usage
- ▶ Effort Amortization
- ▶ Ease-of-Use
- ▶ Backwards Compatible
- ▶ **Selection Criteria**

Multiple selection criteria

for example, load balance, proximity

What does PIAS solve?



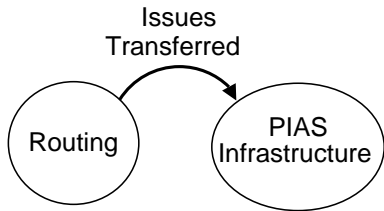
- ▶ Address Usage
- ▶ Effort Amortization
- ▶ Ease-of-Use
- ▶ Backwards Compatible
- ▶ Selection Criteria

Multiple selection criteria

for example, load balance, proximity

Group members can be clients for the group!

What does PIAS solve?



- ▶ Address Usage
- ▶ Effort Amortization
- ▶ Ease-of-Use
- ▶ Backwards Compatible
- ▶ Selection Criteria

All this just by proxying?

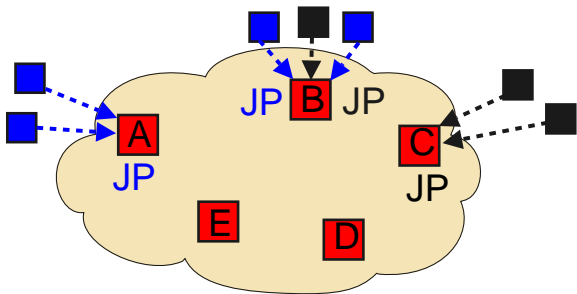
- ▶ **decoupled issues from routing**
- ▶ can be easily addressed in proxy infrastructure

PIAS : design challenges

- ▶ Scalability by
 - ▶ no. of groups, group size/dynamics
 - ▶ no. of proxies
- ▶ Robustness and fast-failover

PIAS : design challenges

- ▶ Scalability by
 - ▶ no. of groups, group size/dynamics
 - ▶ no. of proxies
- ▶ Robustness and fast-failover

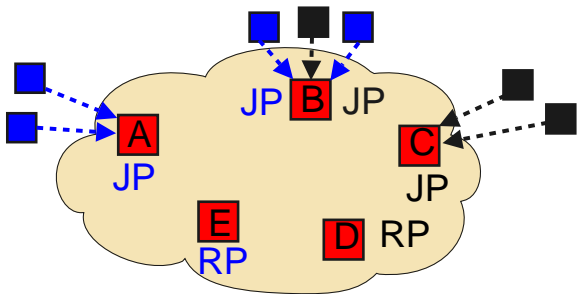


Members register with *Join Proxies (JP)*

Registration involves member authentication

PIAS : design challenges

- ▶ Scalability by
 - ▶ no. of groups, group size/dynamics
 - ▶ no. of proxies
- ▶ Robustness and fast-failover

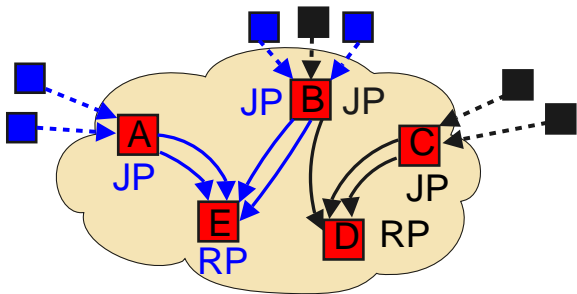


Rendezvous Proxies (RP) : keep group state

group address mapped to RP using consistent hash

PIAS : design challenges

- ▶ Scalability by
 - ▶ no. of groups, group size/dynamics
 - ▶ no. of proxies
- ▶ Robustness and fast-failover

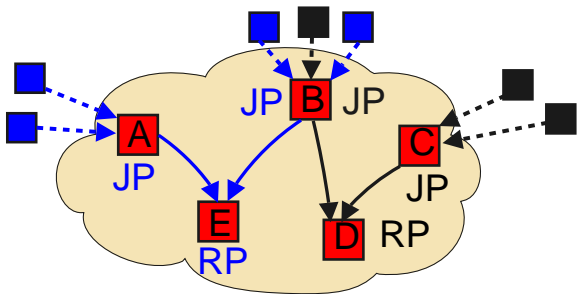


Rendezvous Proxies (RP) : keep group state

JPs inform RPs of member leave/joins

PIAS : design challenges

- ▶ Scalability by
 - ▶ no. of groups, **group size/dynamics**
 - ▶ no. of proxies
- ▶ Robustness and fast-failover

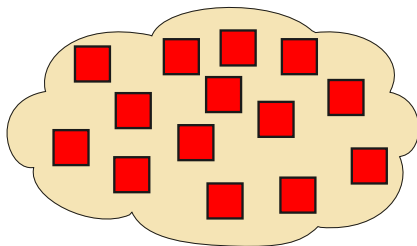


Hierarchy : reduce load on **RPs**

RPs track JPs, JPs track members

PIAS : design challenges

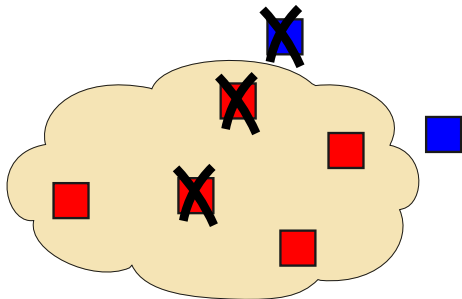
- ▶ Scalability by
 - ▶ no. of groups, group size/dynamics
 - ▶ no. of proxies
- ▶ Robustness and fast-failover



Overlay and Routing issues

PIAS : design challenges

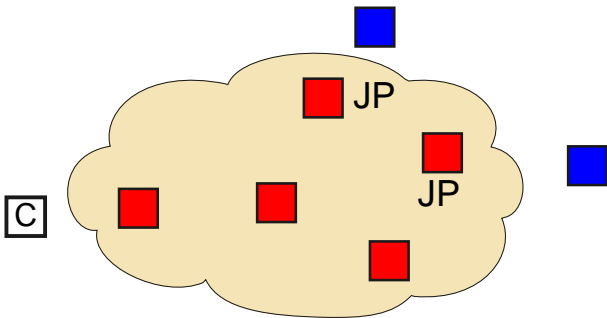
- ▶ Scalability by
 - ▶ no. of groups, group size/dynamics
 - ▶ no. of proxies
- ▶ Robustness and fast-failover





Proxy and Member failures

PIAS : putting it all together

Anycast : Client (C) to Group 1 (blue)

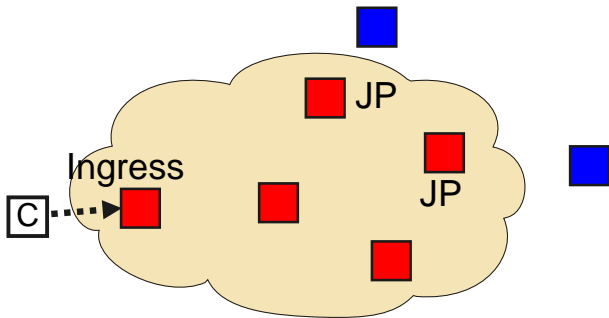


 Anycast
Proxy


 Member
(group 1)


PIAS : putting it all together

$C \Rightarrow$ *Ingress Proxy*



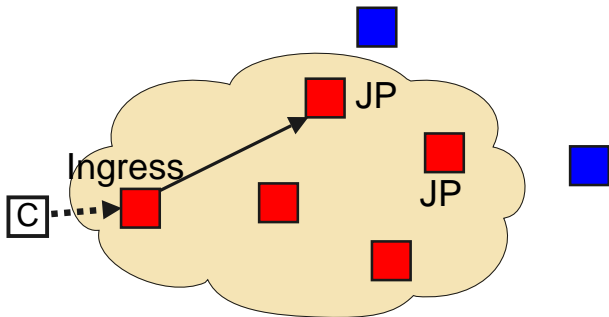
.....➔ IP Anycast

 Anycast
Proxy


 Member
(group 1)


PIAS : putting it all together

Ingress Proxy \Rightarrow Join Proxy



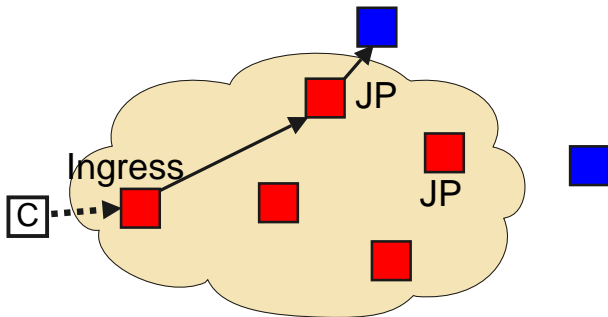
..... \rightarrow IP Anycast \longrightarrow IP Unicast

 Anycast
Proxy


 Member
(group 1)


PIAS : putting it all together

Join Proxy \Rightarrow Member



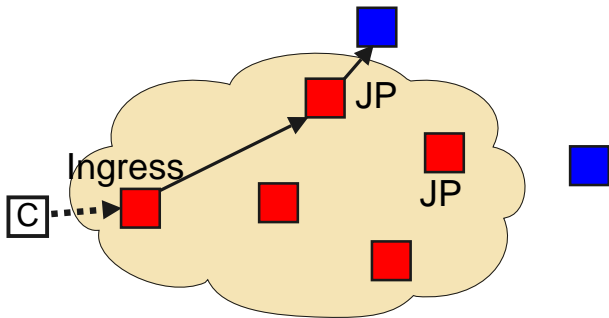
..... \rightarrow IP Anycast \longrightarrow IP Unicast

 Anycast
Proxy


 Member
(group 1)


PIAS : putting it all together

Client \Rightarrow Ingress P. \Rightarrow Join P. \Rightarrow Member



..... \rightarrow IP Anycast \longrightarrow IP Unicast

 Anycast
Proxy

 Member
(group 1)

New anycast applications

Anycast service offered by PIAS

- ▶ practical
- ▶ easy-to-use
- ▶ scales by group number/size/dynamics
- ▶ group members can be clients too

Applications

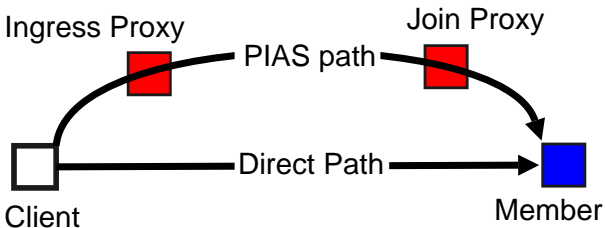
- ▶ Peer discovery : network games, p2p applications etc.
- ▶ Reaching an overlay network : querying **OpenDHT**, global **RON**, **i3** etc.

PIAS : possible problems

- ▶ Stretch
 - ▶ Affinity
 - ▶ Proximity
-

PIAS : possible problems

- ▶ **Stretch**
 - ▶ Affinity
 - ▶ Proximity
-



$\text{Stretch} = \text{PIAS path len.} / \text{Direct path len.}$

What is the stretch imposed by PIAS?

PIAS : possible problems

- ▶ **Stretch** : simulation
 - ▶ Affinity
 - ▶ Proximity
-

Topology

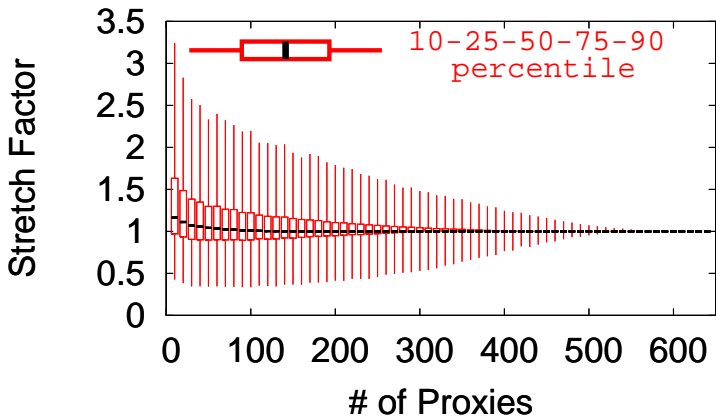
- ▶ POP-level topology for tier-1 ISPs (**Rocketfuel**)
- ▶ 22 ISPs, 687 POPs, 2825 inter-POP links
- ▶ Annotated links with actual distance (kms)

Simulation

- ▶ **SSFNET** for BGP route calculation

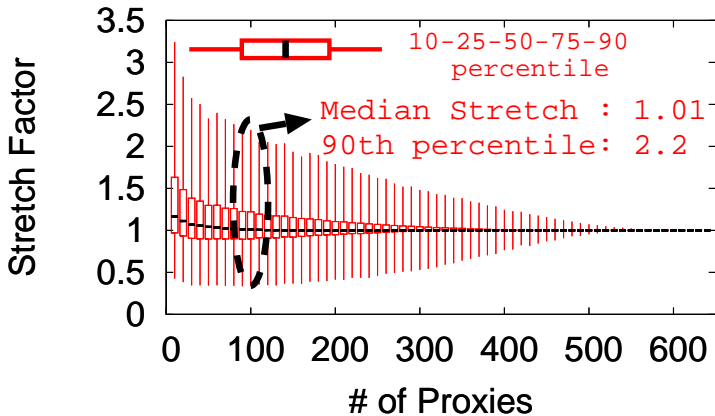
PIAS : possible problems

- ▶ Stretch : simulation
 - ▶ Affinity
 - ▶ Proximity
-



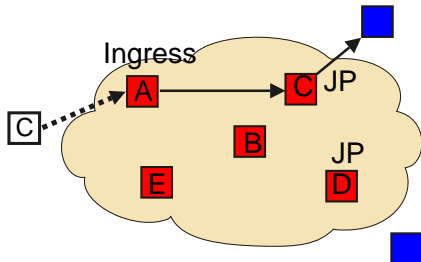
PIAS : possible problems

- ▶ Stretch : simulation
 - ▶ Affinity
 - ▶ Proximity
-



PIAS : possible problems

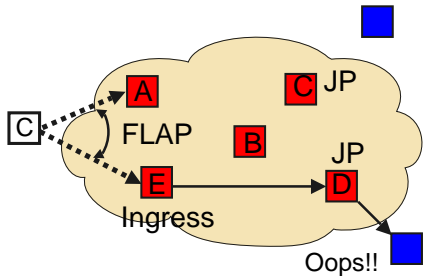
- ▶ Stretch
 - ▶ **Affinity**
 - ▶ Proximity
-



Affinity : same client to same ingress

PIAS : possible problems

- ▶ Stretch
 - ▶ **Affinity**
 - ▶ Proximity
-



Affinity : same client to same ingress

What is the affinity offered by native IP Anycast?

PIAS : possible problems

- ▶ Stretch
 - ▶ **Affinity** : measured anycasted DNS root-servers
 - ▶ Proximity
-

Traceroute-Servers

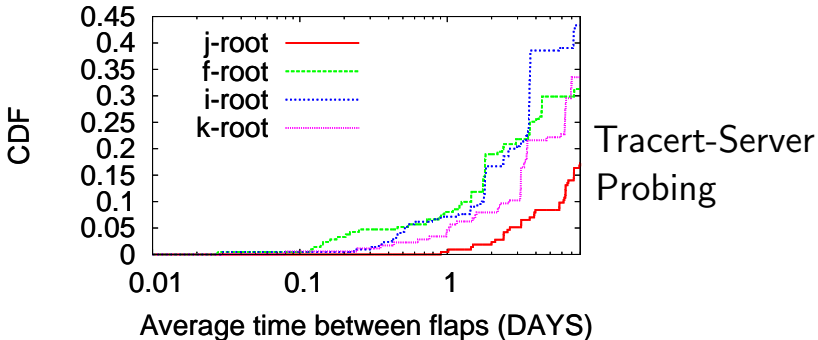
- ▶ 244 vantage points
- ▶ Duration : 7 days
- ▶ Europe-centric distribution

Planetlab

- ▶ 163 Planetlab sites
- ▶ Duration : 3 months (Dec'04-Mar'05)
- ▶ US-centric distribution

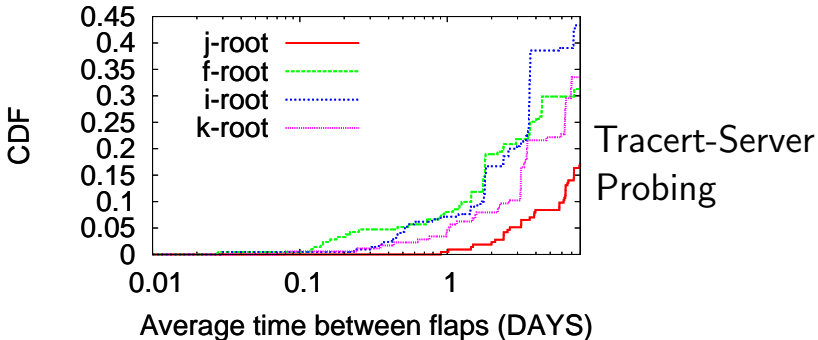
PIAS : possible problems

- ▶ Stretch
 - ▶ **Affinity** : measured anycasted DNS root-servers
 - ▶ Proximity
-



PIAS : possible problems

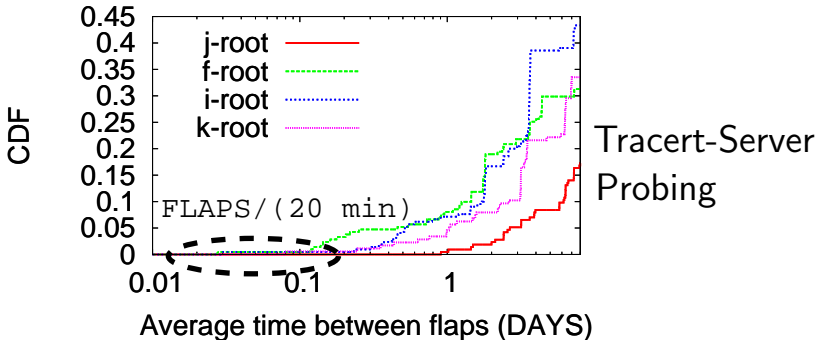
- ▶ Stretch
 - ▶ **Affinity** : measured anycasted DNS root-servers
 - ▶ Proximity
-



Less than 1 flap per day for $\sim 95\%$ of nodes

PIAS : possible problems

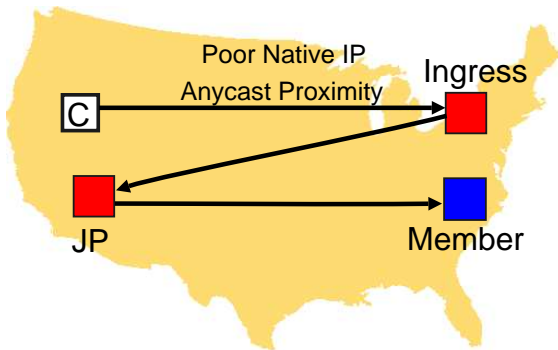
- ▶ Stretch
 - ▶ **Affinity** : measured anycasted DNS root-servers
 - ▶ Proximity
-



Less than 1 flap per day for $\sim 95\%$ of nodes

PIAS : possible problems

- ▶ Stretch
 - ▶ Affinity
 - ▶ Proximity
-



Is native IP anycast based proximity useful?

PIAS : possible problems

- ▶ Stretch
 - ▶ Affinity
 - ▶ Proximity : measuring proximity
-

Does IP Anycast offer latency-based proximity?

- ▶ measured the proximity offered by root-server anycast deployments
- ▶ from ~ 40000 clients

PIAS : possible problems

- ▶ Stretch
 - ▶ Affinity
 - ▶ Proximity : measuring proximity
-

Does IP Anycast offer latency-based proximity?

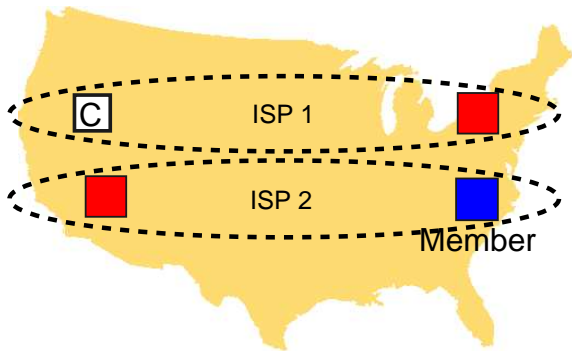
- ▶ measured the proximity offered by root-server anycast deployments
- ▶ from ~ 40000 clients

Results (details in technical report)

- ▶ No (for a naive deployment)
 - ▶ 5-6 times the ideal proximity was common

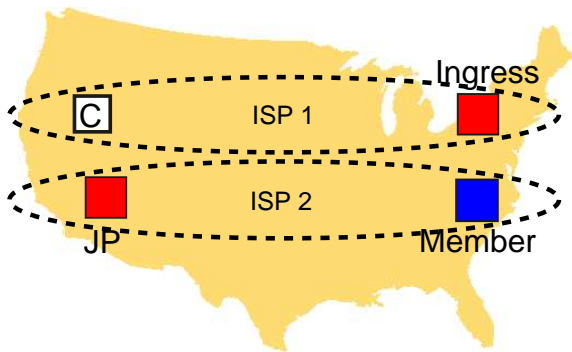
PIAS : possible problems

- ▶ Stretch
 - ▶ Affinity
 - ▶ Proximity : example of poor proximity
-



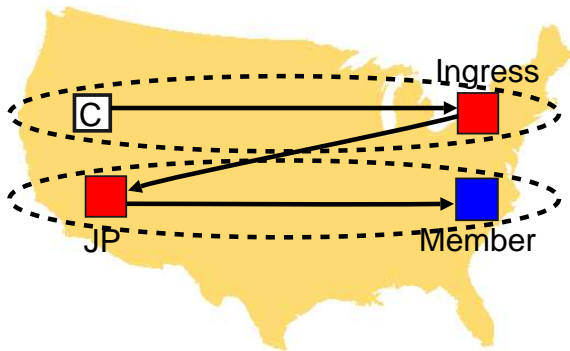
PIAS : possible problems

- ▶ Stretch
 - ▶ Affinity
 - ▶ Proximity : example of poor proximity
-



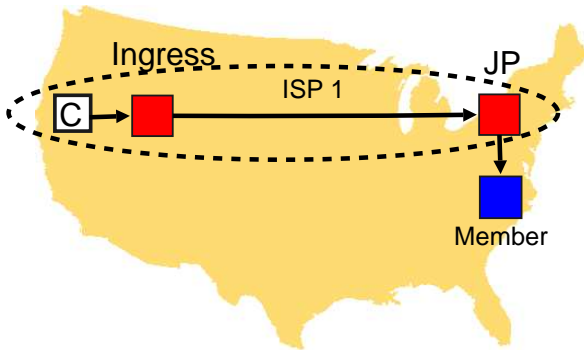
PIAS : possible problems

- ▶ Stretch
 - ▶ Affinity
 - ▶ Proximity : example of poor proximity
-



PIAS : possible problems

- ▶ Stretch
 - ▶ Affinity
 - ▶ Proximity : a simple alleviative
-



Planned deployment to attain proximity

Why bother?

Application-layer anycast is already out there

Why bother?

Application-layer anycast is already out there

Advantages of PIAS ...

- ✓ use for low-level protocols
- ✓ proximity is a lot easier
 - ✓ easier management
- ✓ faster failover
- ✓ no extra round-trip
 - ✗ the overhead of proxy traversal

Summary

Proxy IP Anycast Service

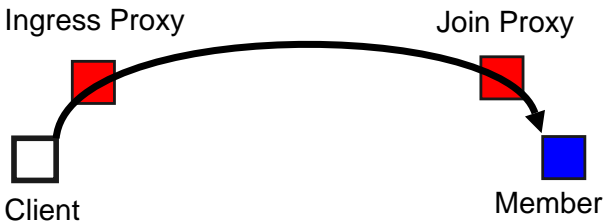
- ▶ **practical** anycast deployment architecture
- ▶ addresses native IP and application-layer anycast limitations
- ▶ opens new usage avenues

Anycast for the network community

- ▶ currently deploying PIAS
- ▶ publicly usable in the near future

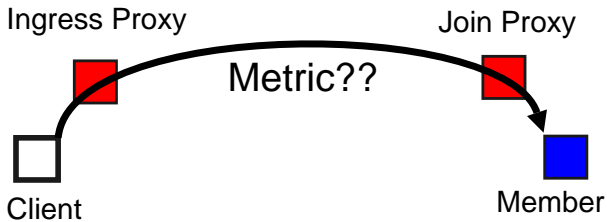
<http://pias.gforge.cis.cornell.edu>

PIAS : the metrics that can be offered



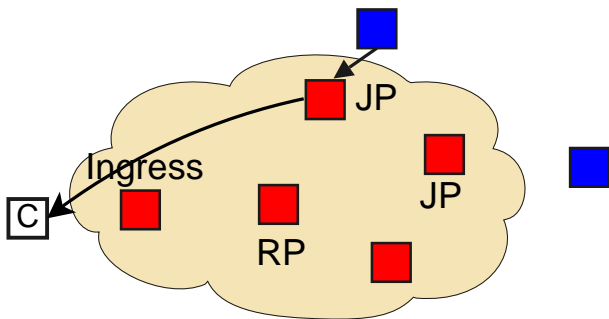
$C \Rightarrow \text{Member}$ is about the same as $\text{IngressP.} \Rightarrow \text{JoinP.}$

PIAS : the metrics that can be offered



$C \Rightarrow$ Member is about the same as $\text{IngressP.} \Rightarrow \text{JoinP.}$
can support metrics such as latency, prop. delay etc.
scalably

PIAS : reverse path

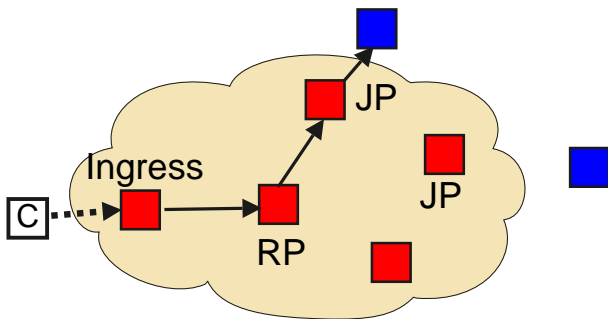


.....➔ IP Anycast ➔ IP Unicast


■ Anycast
Proxy


■ Member
(group 1)

PIAS : the real picture

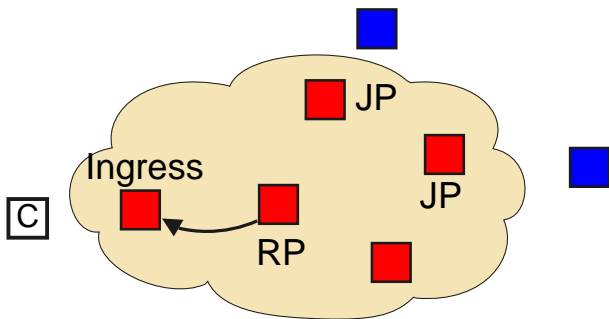


.....➔ IP Anycast ➔ IP Unicast

 Anycast
Proxy

 Member
(group 1)

PIAS : the real picture

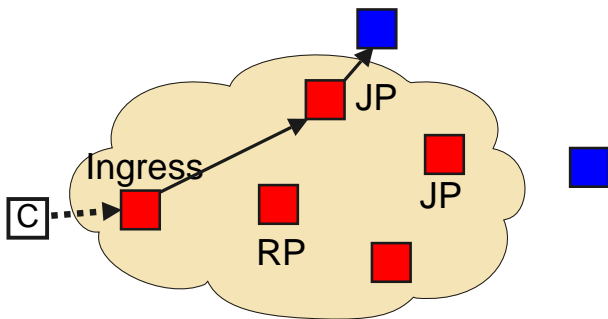


.....➔ IP Anycast ➔ IP Unicast


■ Anycast
Proxy


■ Member
(group 1)

PIAS : the real picture



.....➔ IP Anycast ➔ IP Unicast

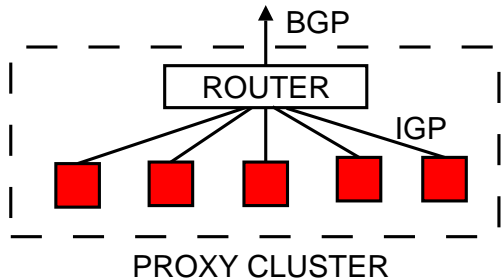
 Anycast
Proxy

 Member
(group 1)

PIAS : engineering issues

Scalability by no. of proxies

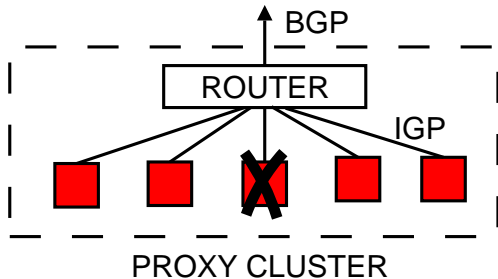
- ▶ a clustered deployment model
- ▶ decouples proxy dynamics from inter-domain routing



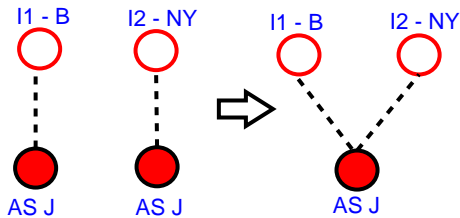
PIAS : engineering issues

Failures

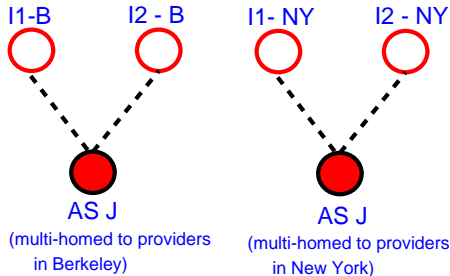
- ▶ no impact on inter-domain routing



Inter-domain routing and Anycast!!



Anycast'ed AS appears similar to a multihomed AS



But is different from typical multihoming!