Anycast for Any Service

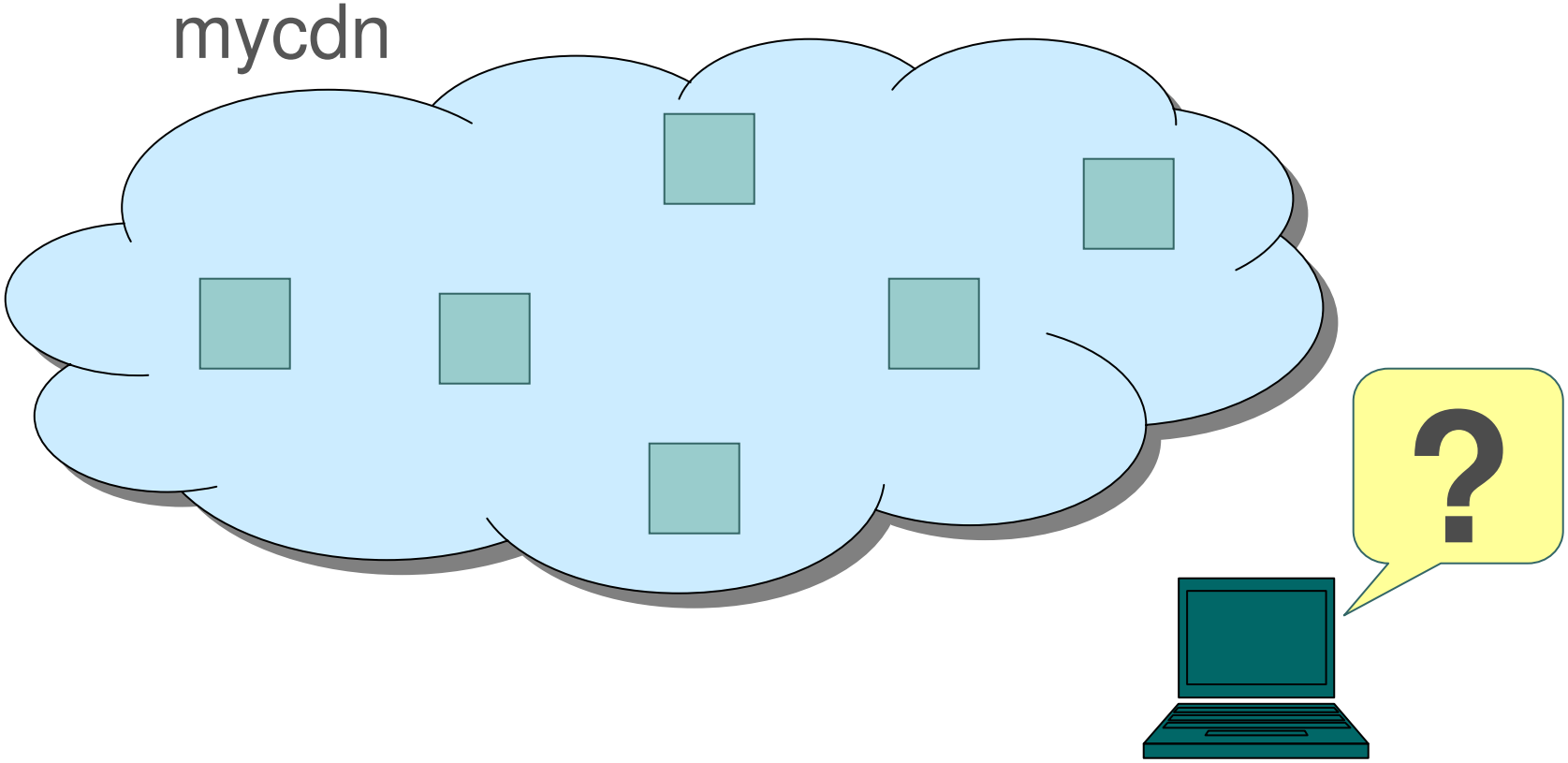
Michael J. Freedman

Karthik Lakshminarayanan

David Mazières

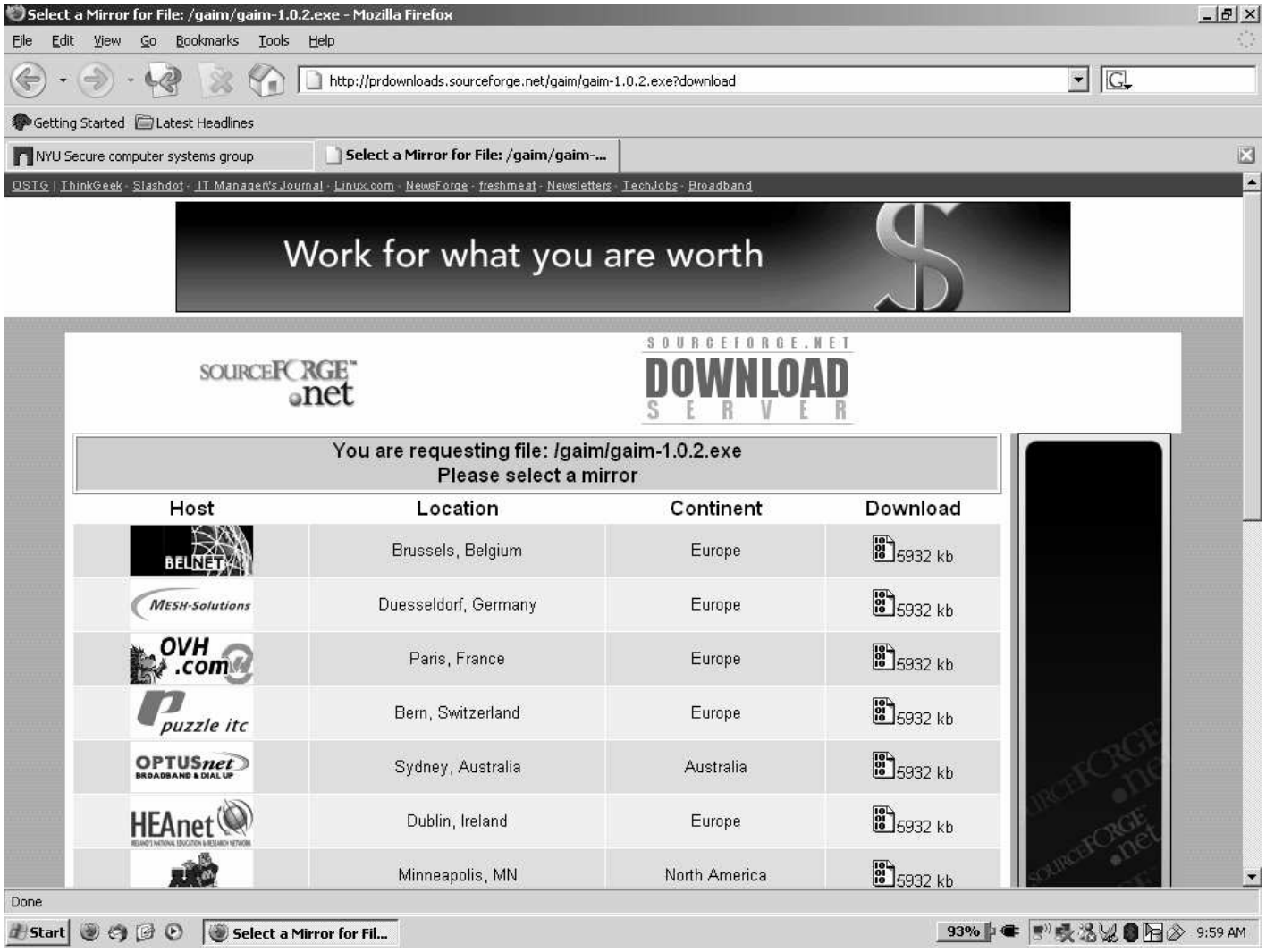
http://oasis.coralcdn.org/

What's the replica-selection problem?

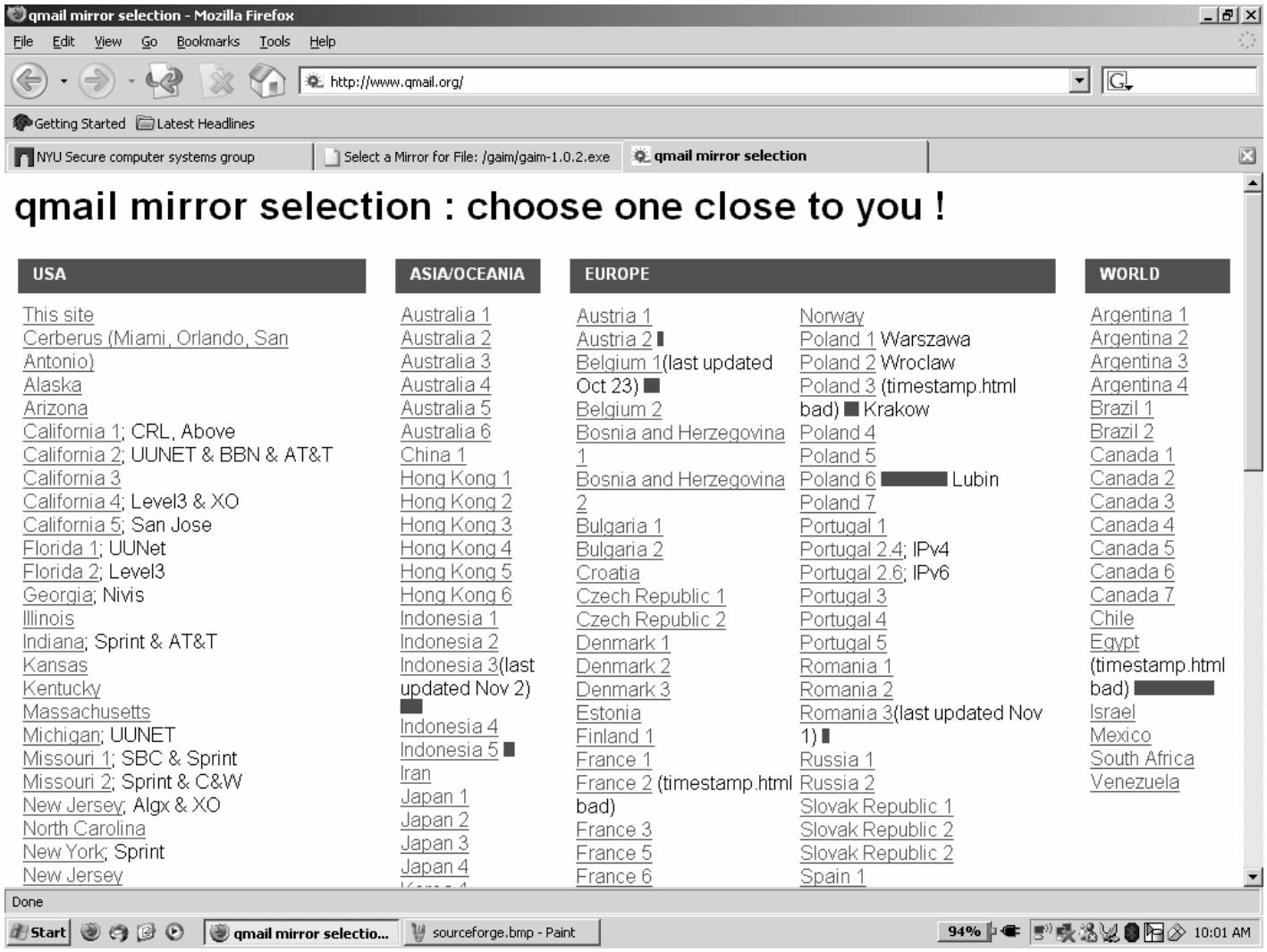


- Client needs to choose a "good" replica server
- Performance and cost dependent on replica selection

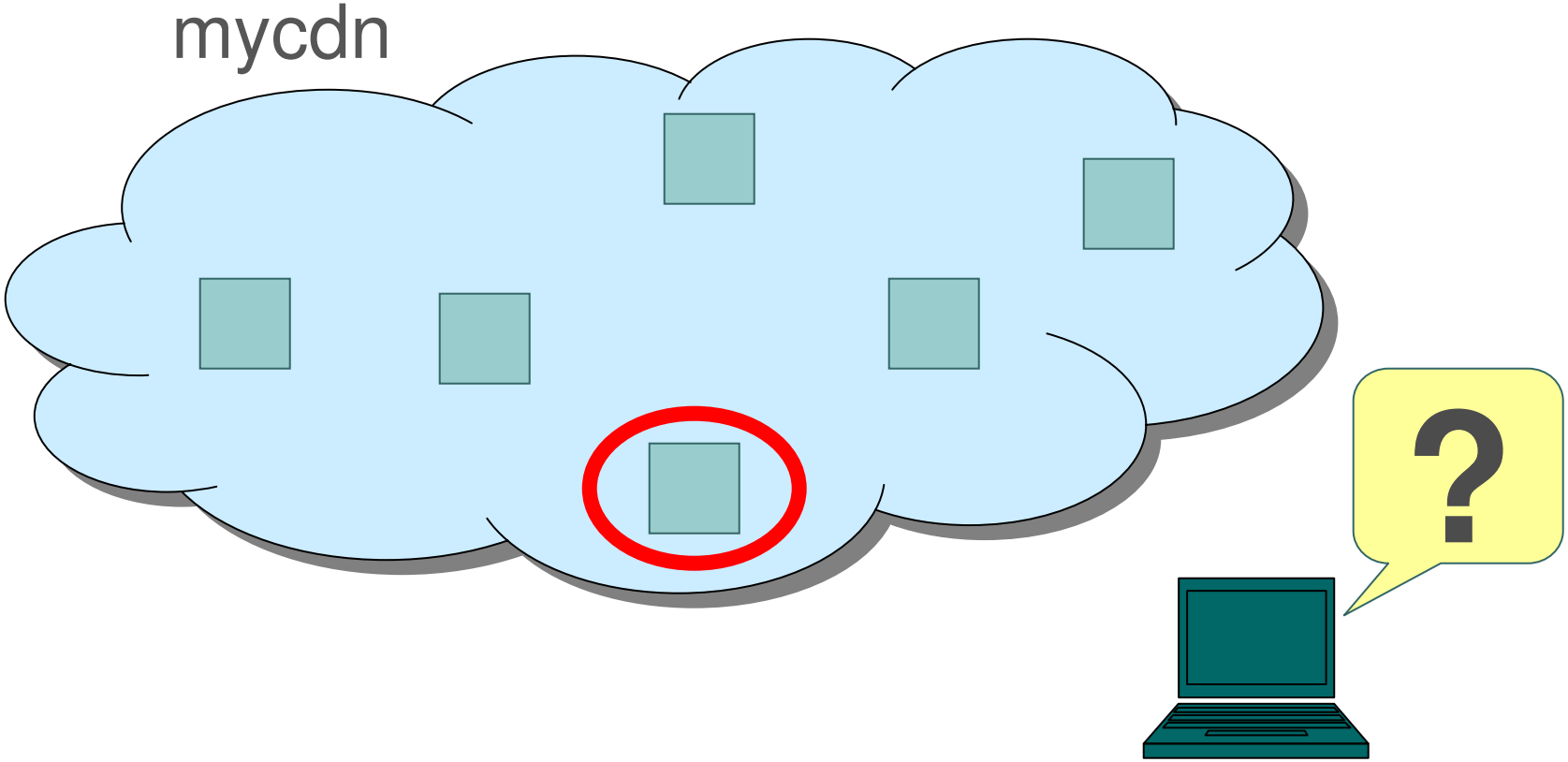
What do we currently do?



How bad can it get?



Anycast is the solution



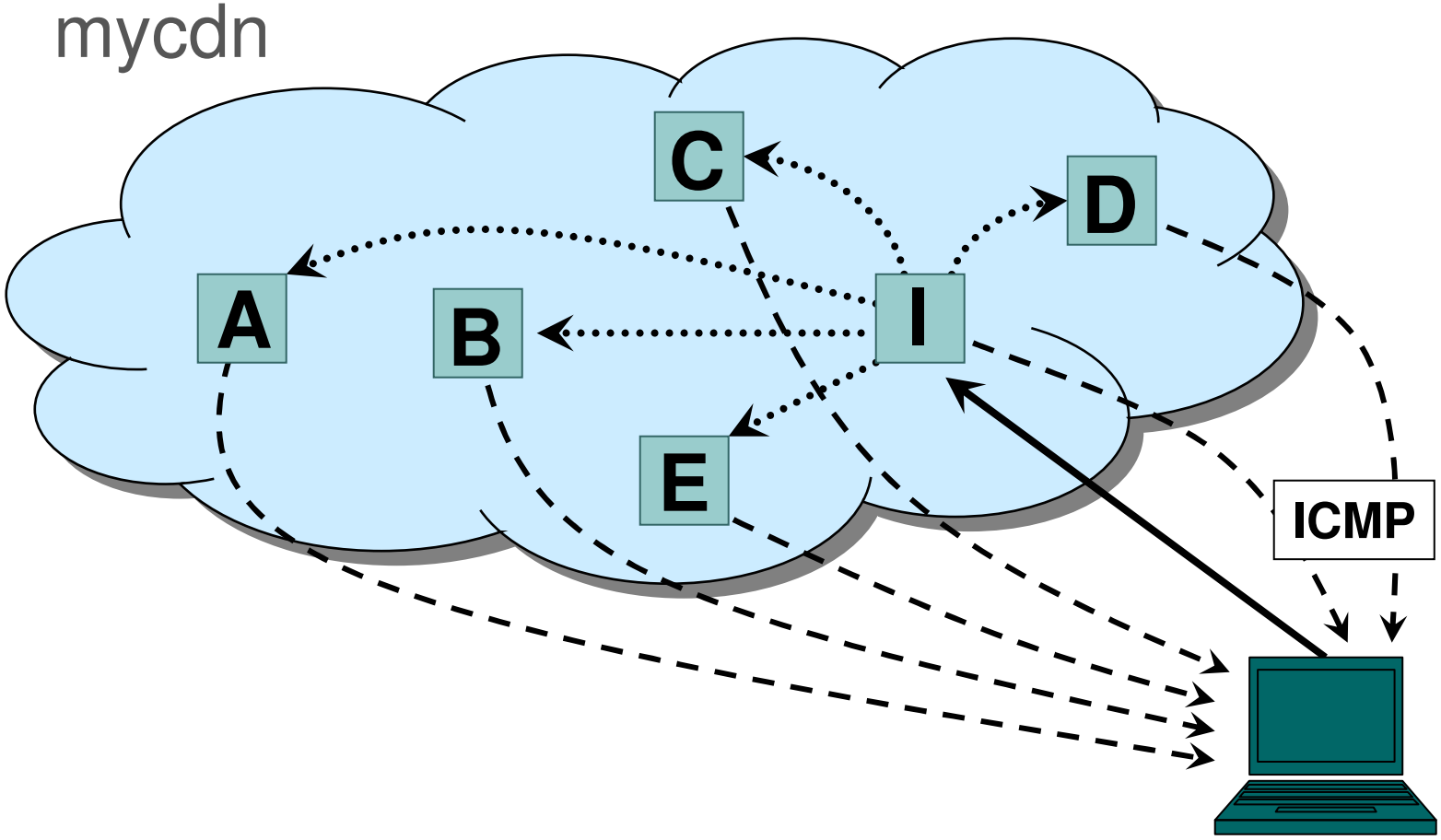
- Anycast = automated “good” replica selection
- OASIS is a flexible anycast system for multiple services

The need for anycast

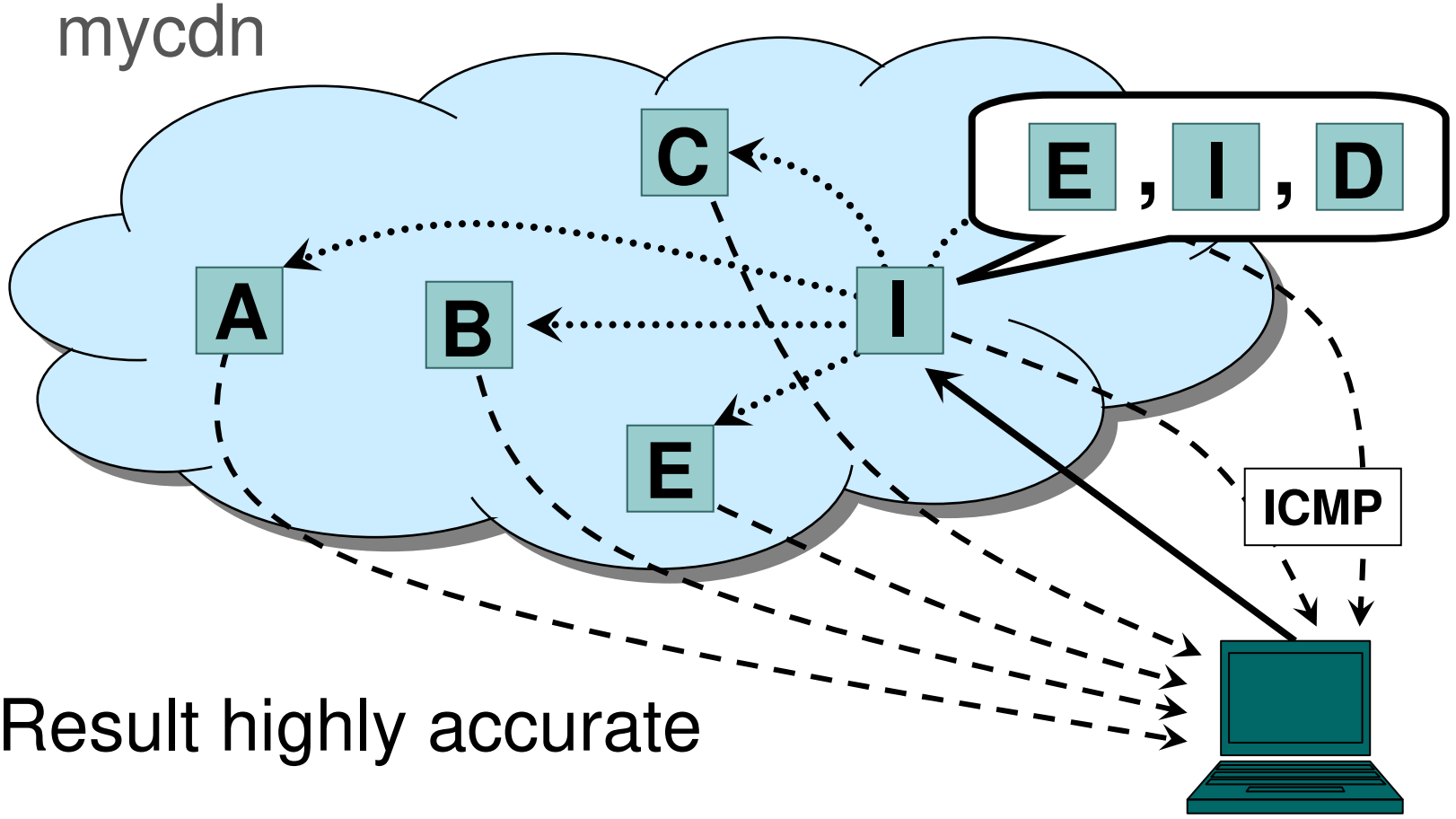
- Internet systems rely on replicated content and services
  - Distributed mirrors: Web servers, FTP servers, …
  - Content Distribution Networks: Akamai, CoralCDN, …
  - Internet Naming Systems: DNS, SFR, DOA, …
  - Distributed File Systems: CFS, Shark, ….
  - Routing Overlays: RON, Detour, i3, …
  - Distributed Hash Storage Systems: OpenDHT, …
- All could benefit from anycast service

How should one implement anycast?

Strawman: probe & find nearest

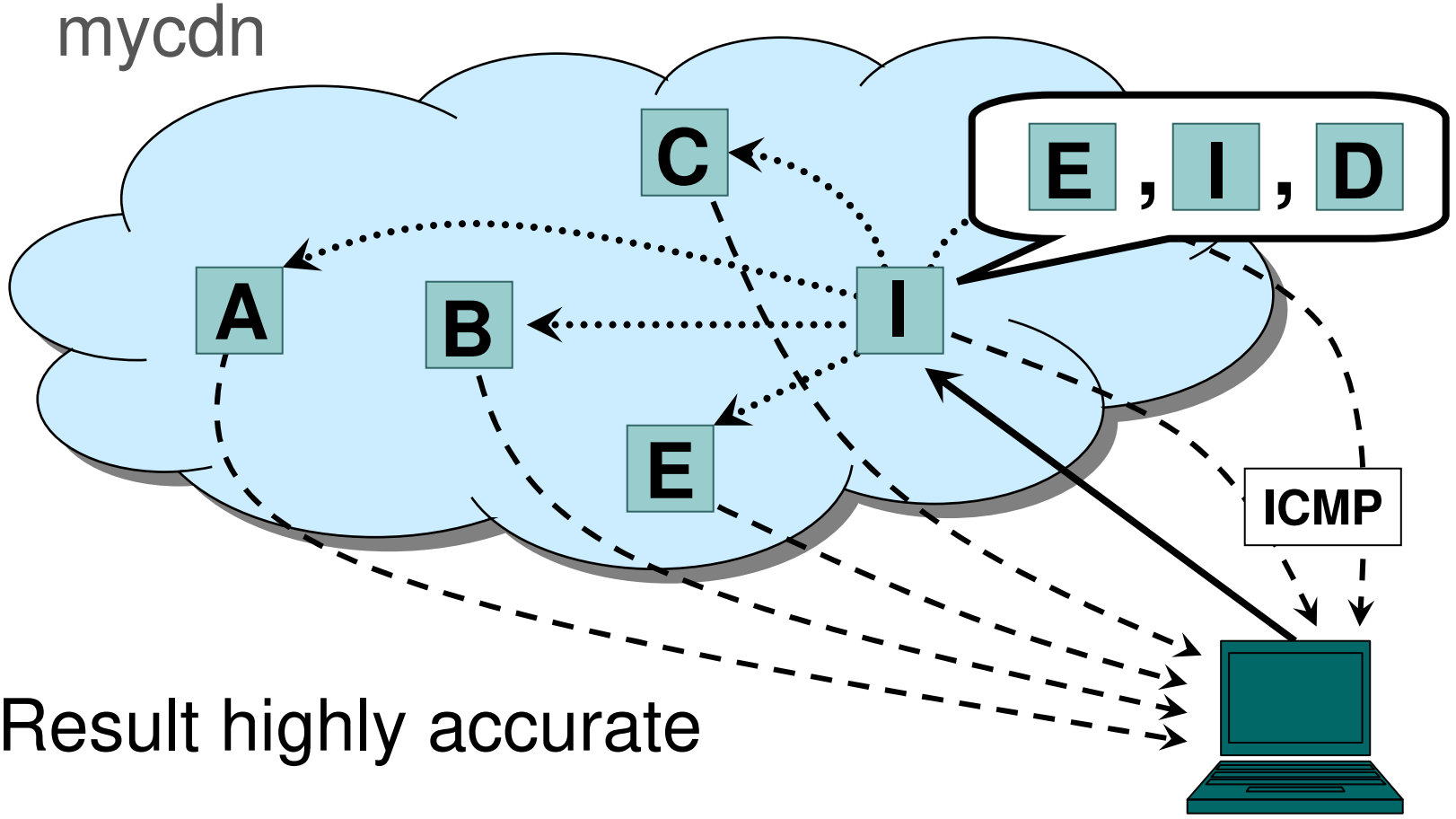


Strawman: probe & find nearest



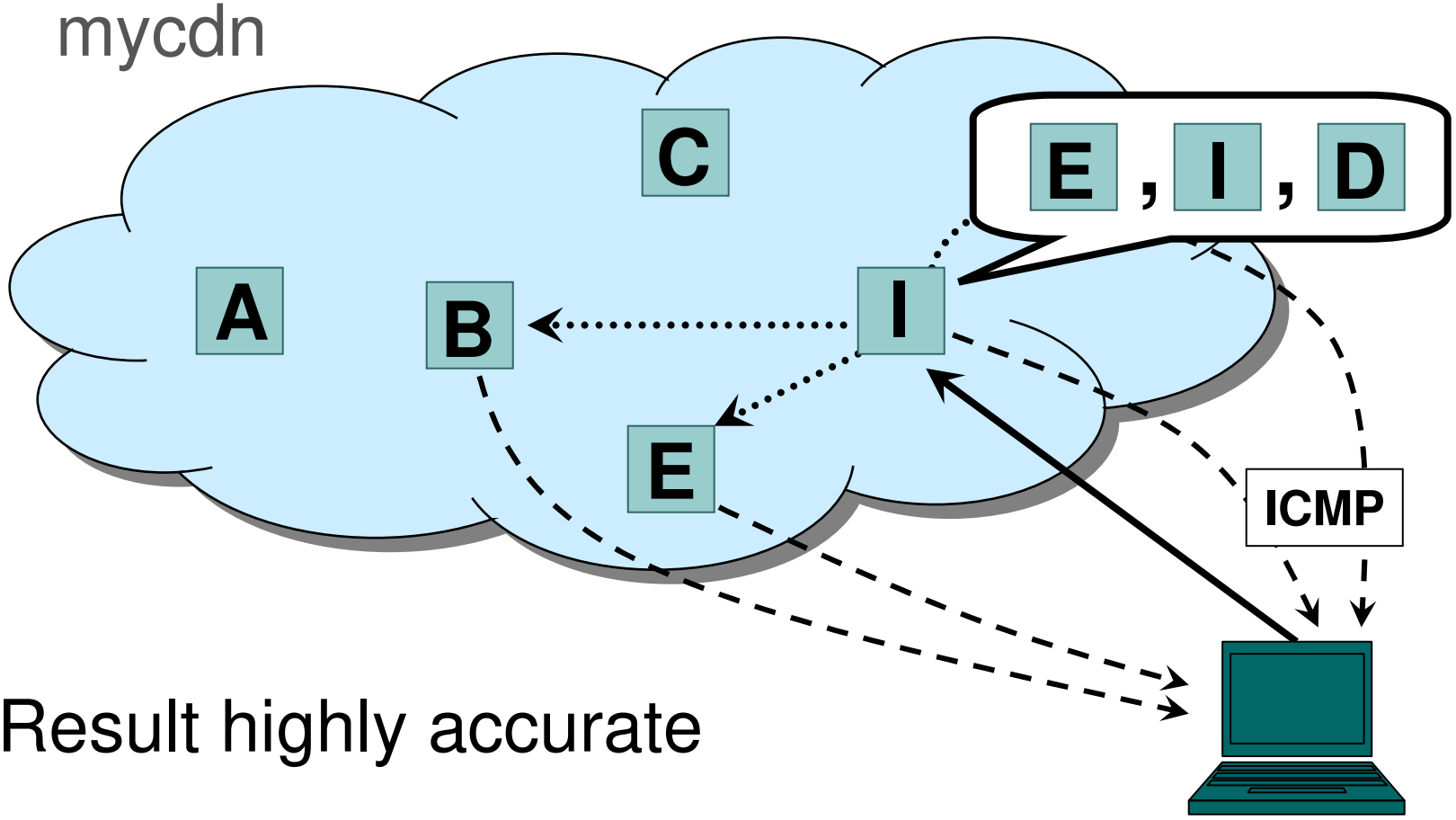
- ✓ Result highly accurate
- ✗ Lots of probing
- ✗ Slow to compute

Strawman: probe & find nearest



- ✓ Result highly accurate
- ✗ ~~Lots of probing~~
- ✗ Slow to compute

Avoid probing on-demand

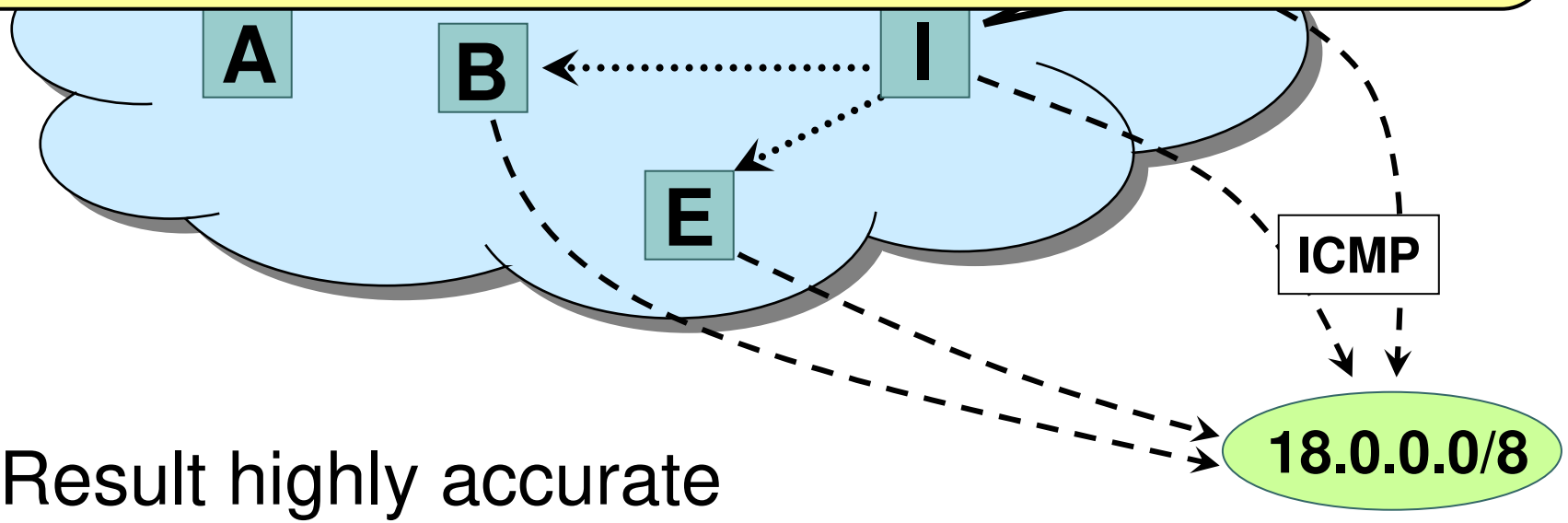


- ✓ Result highly accurate
- ✗ ~~Lots of probing~~
- ✗ ~~Slow to compute~~

Avoid probing on-demand

[IMC05] shows IP prefixes often preserve locality

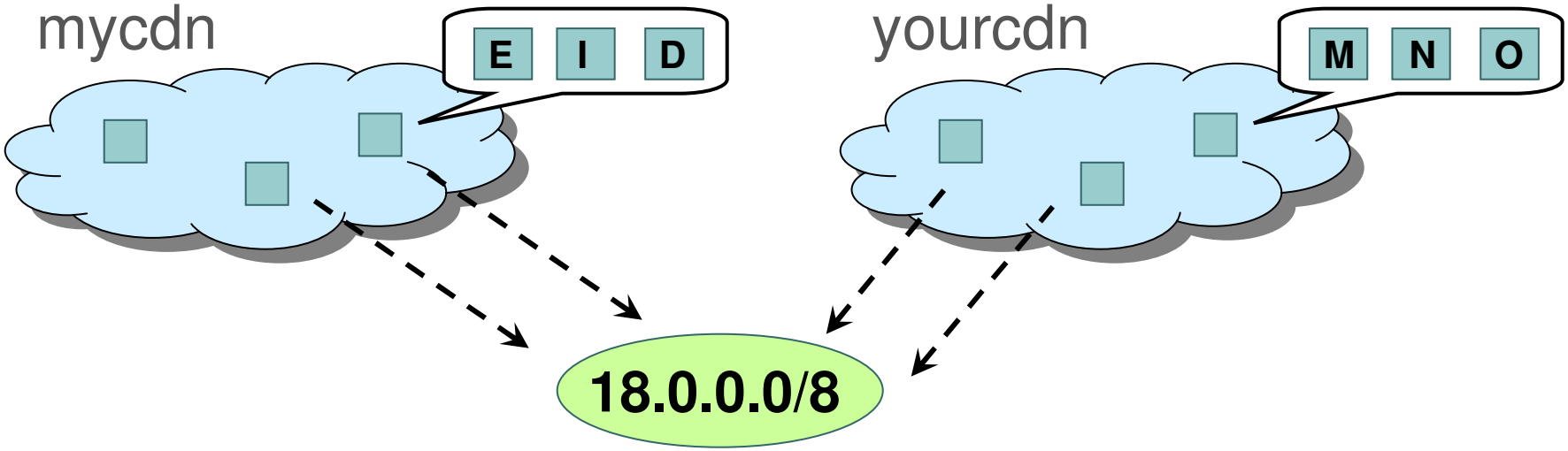
(99% of /24s by stub AS at the same location)



- ✓ Result highly accurate
- ✘ ~~Lots of probing~~
- ✘ ~~Slow to compute~~

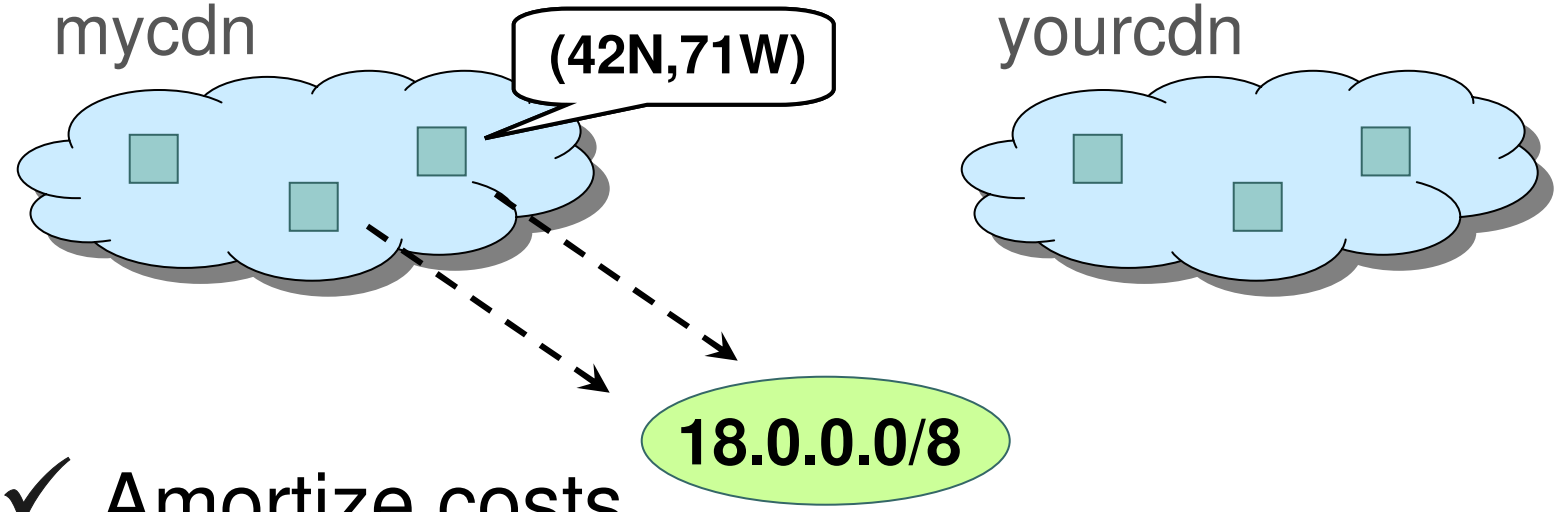
This is the problem Akamai must solve

What about *yourcdn*?



- ✓ Result highly accurate
- ✗ ~~Lots of probing~~
- ✗ ~~Slow to compute~~

Idea: Use geographic coordinates



- ✓ Amortize costs
- ✓ Stable across services, time, and failures
- ✓ Result highly accurate
- ✘ ~~Lots of probing~~
- ✘ ~~Slow to compute~~

Assume all replicas know geo-coords

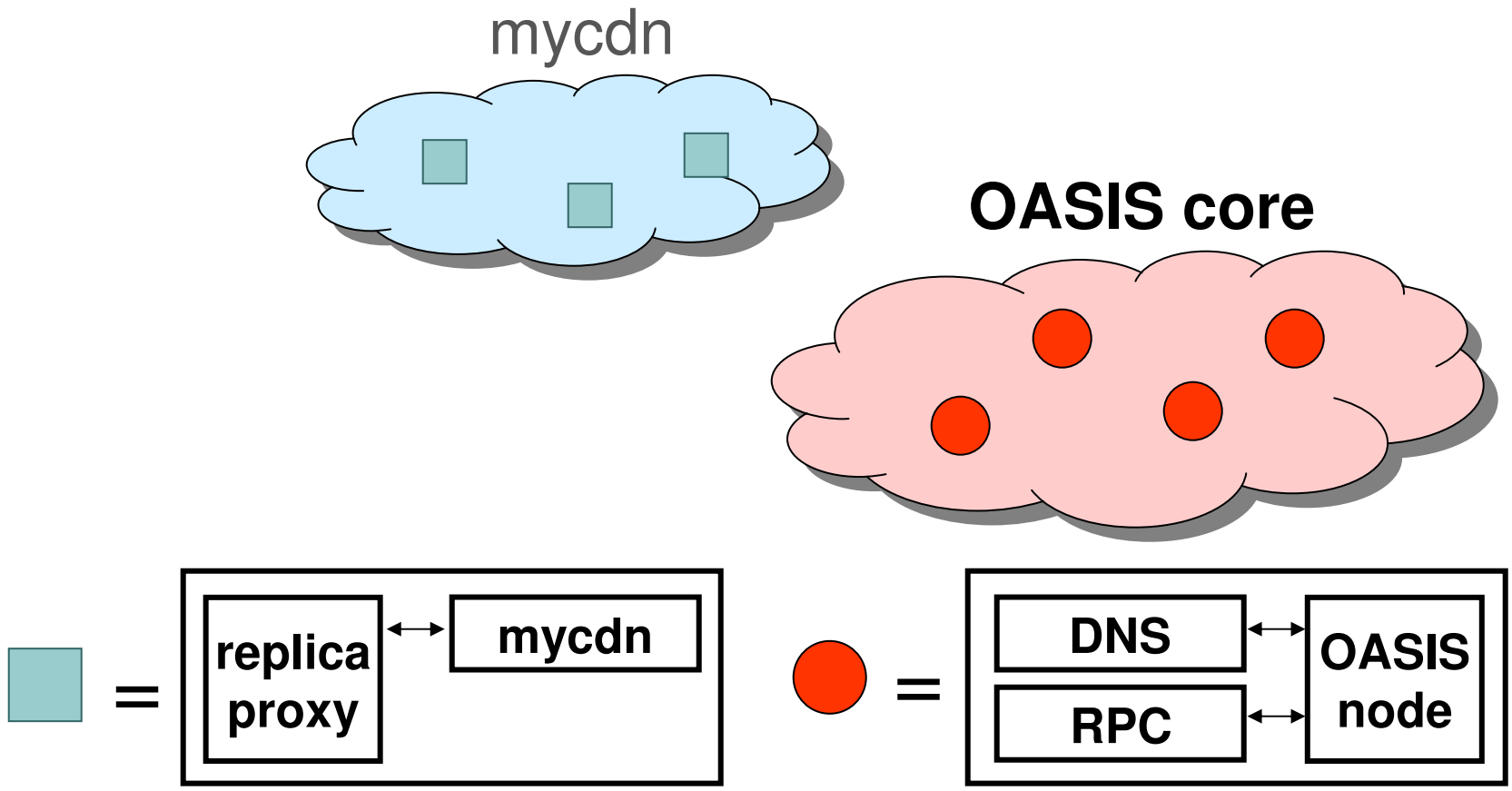
OASIS provides…

- ✓ Amortize costs
- ✓ Stable across time, services, and failures
- ✓ Result highly accurate
- ✓ Fast response time
- ✓ Supports flexible anycast policies
  - Balances tension between:
    - Performance: finding nearest replica
    - Cost: minimizing 95% bandwidth usage

Outline

- Architecture and design decisions
- Detailed design
- Evaluation
- Deployment and integration lessons
  - OASIS deployed since November 2005
  - Currently in use by 10 services

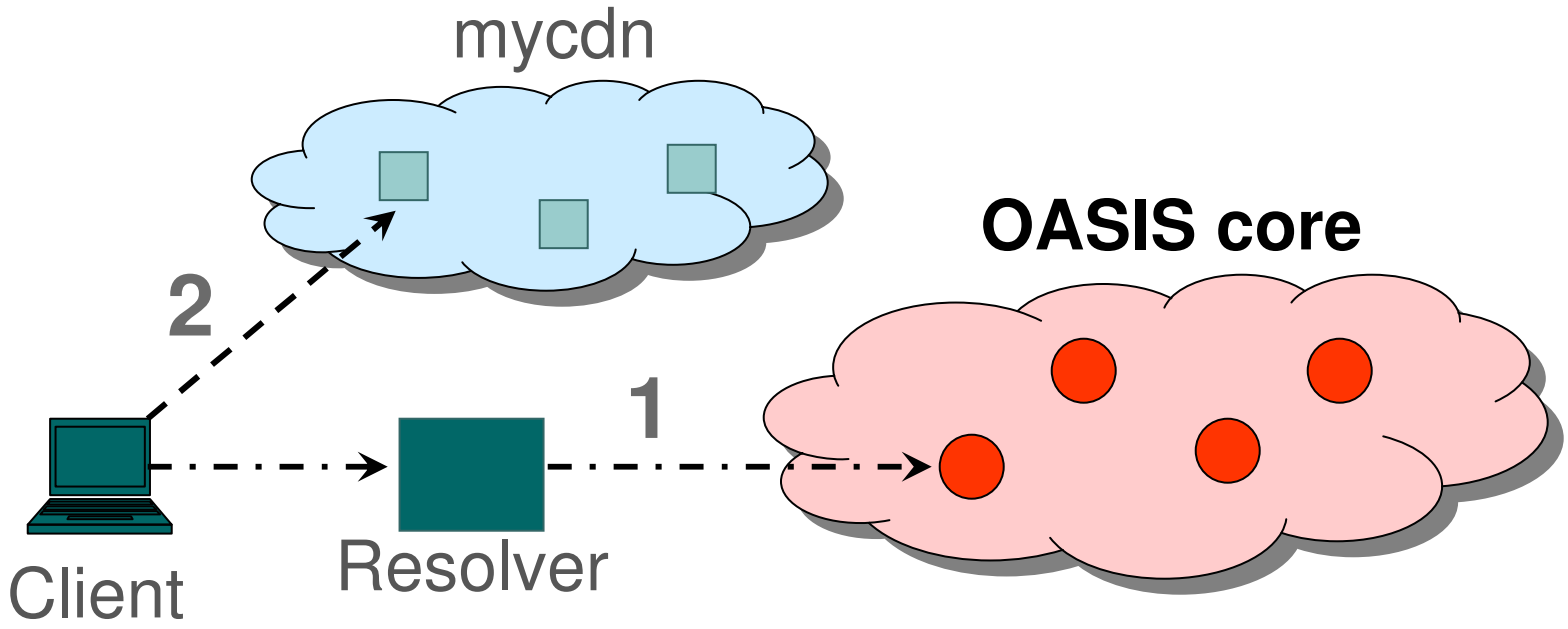
Two-tier architecture



Large set of *replicas* that assist in measurement

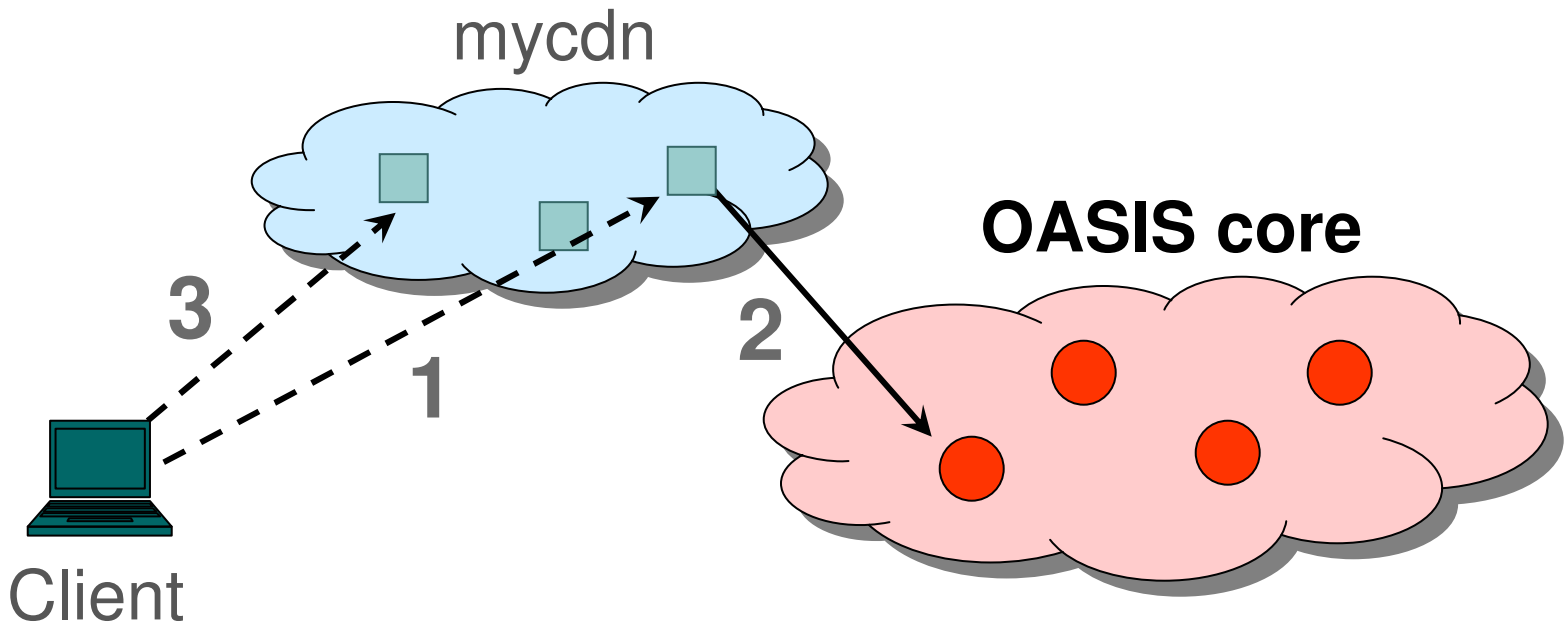
Reliable *core* of hosts that implement anycast

Using OASIS via DNS



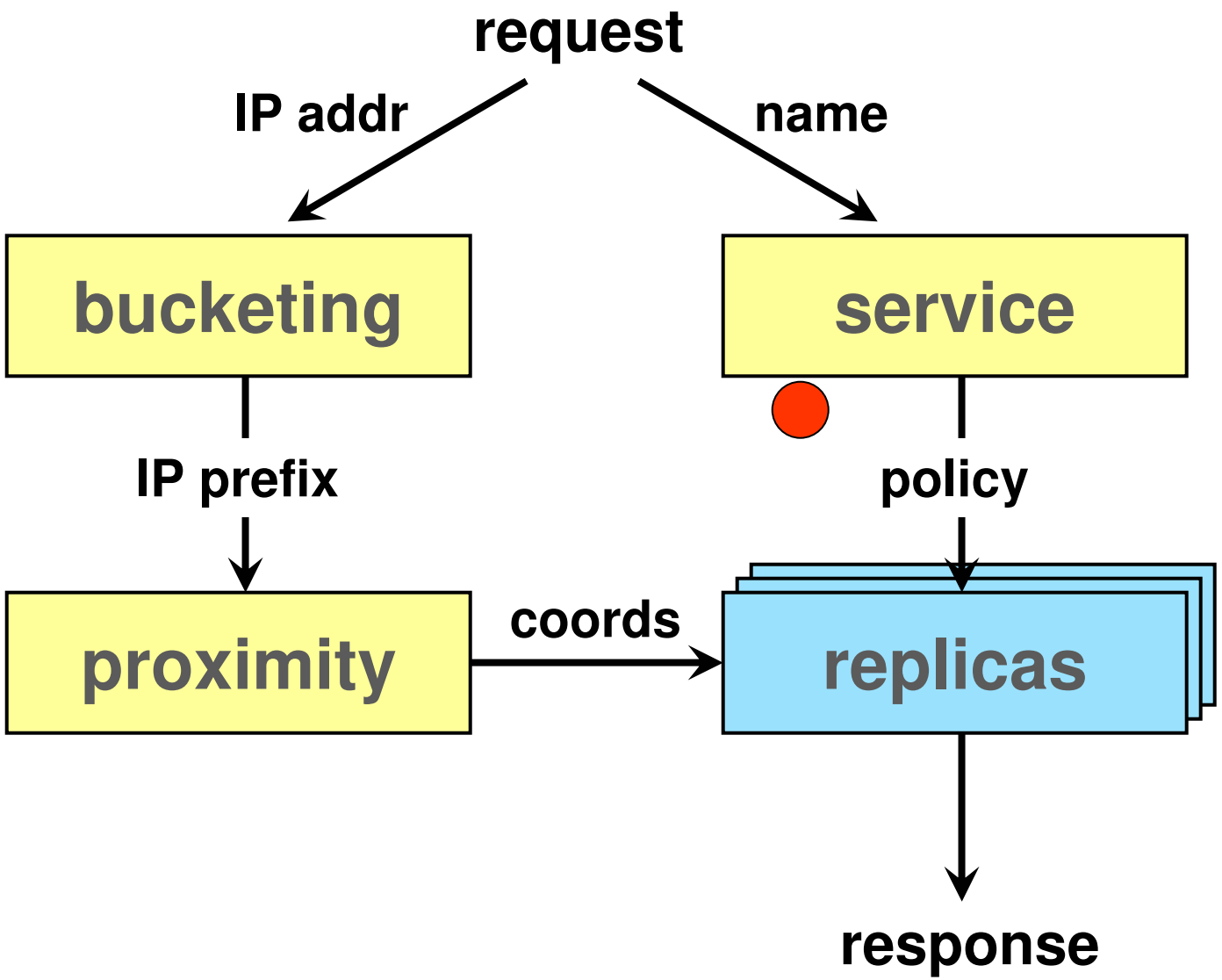
1. Client issues DNS request for ***mycdn**.nyuld.net*
2. OASIS redirects client to nearby application replica

Using OASIS via HTTP

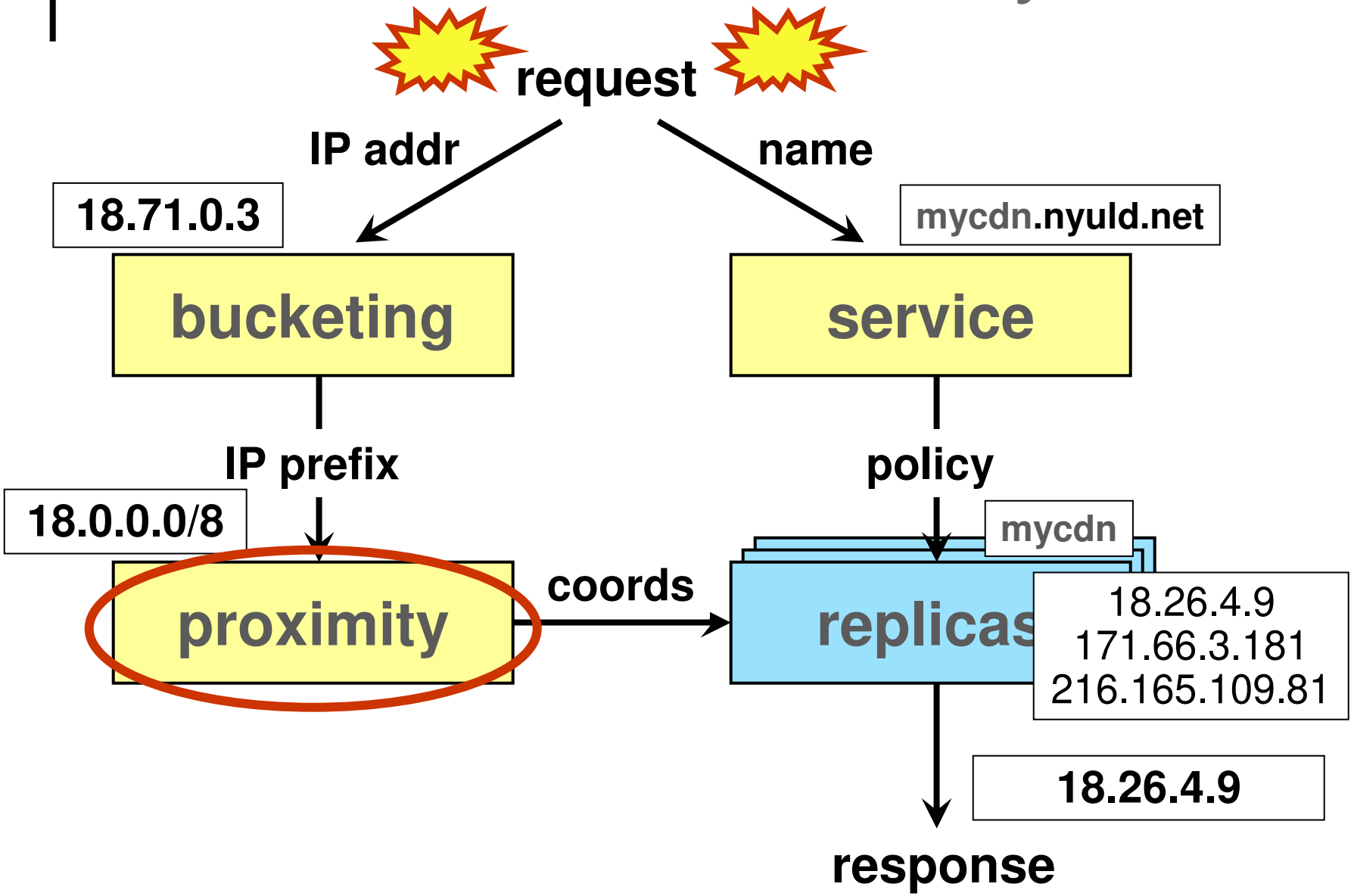


1. Client issues HTTP request
2. Web cgi-bin issues RPC to OASIS core
3. Client redirected to nearby application replica

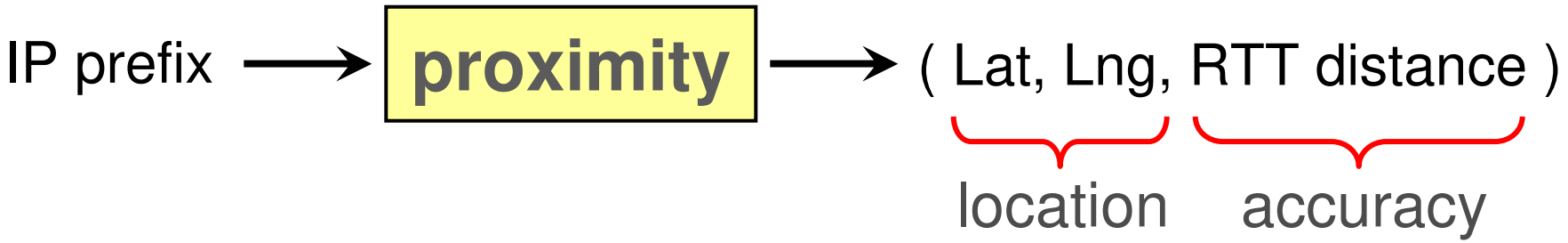
How does core answer anycast?



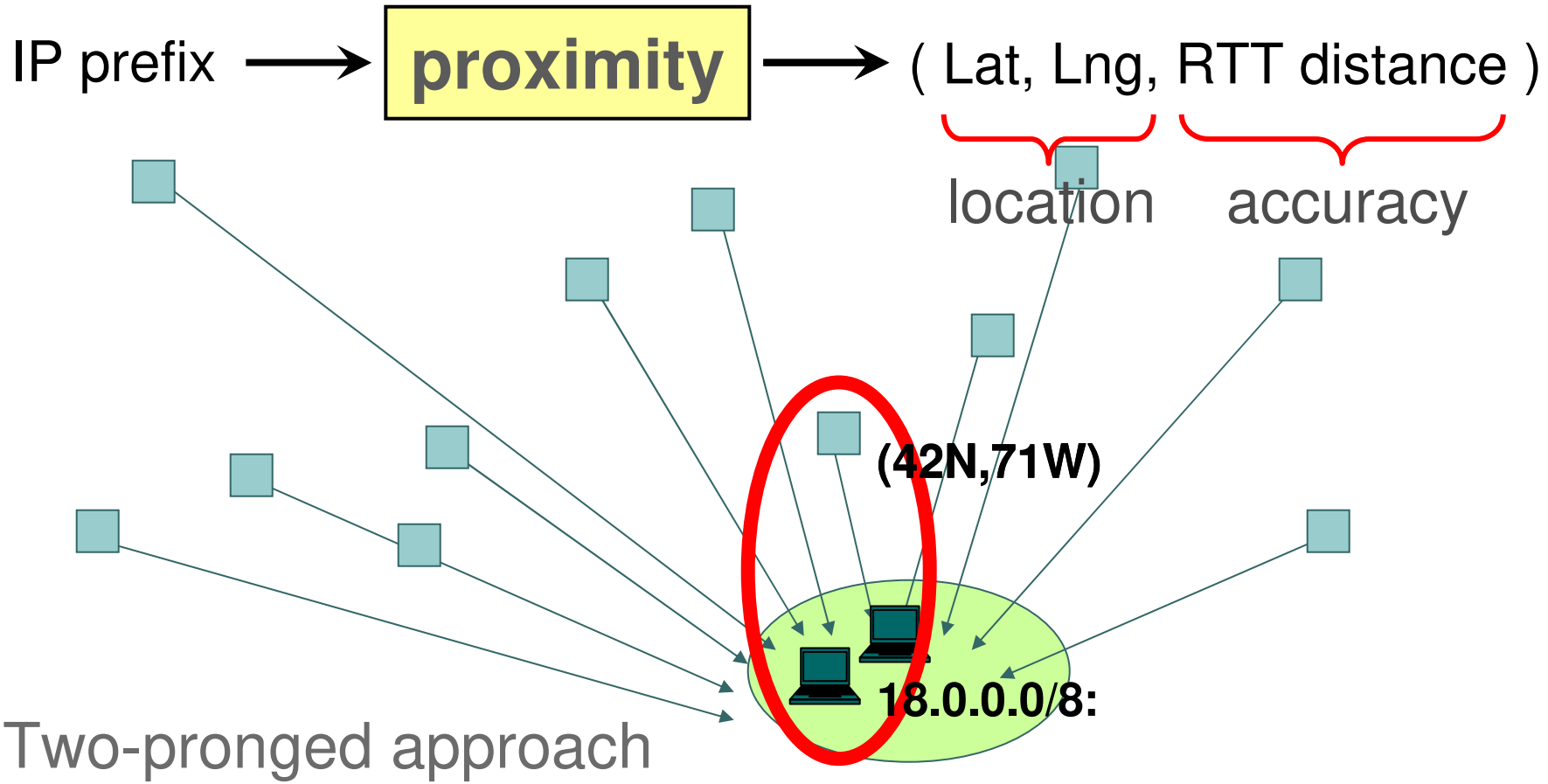
How does core answer anycast?



How to map IP prefix to coords?

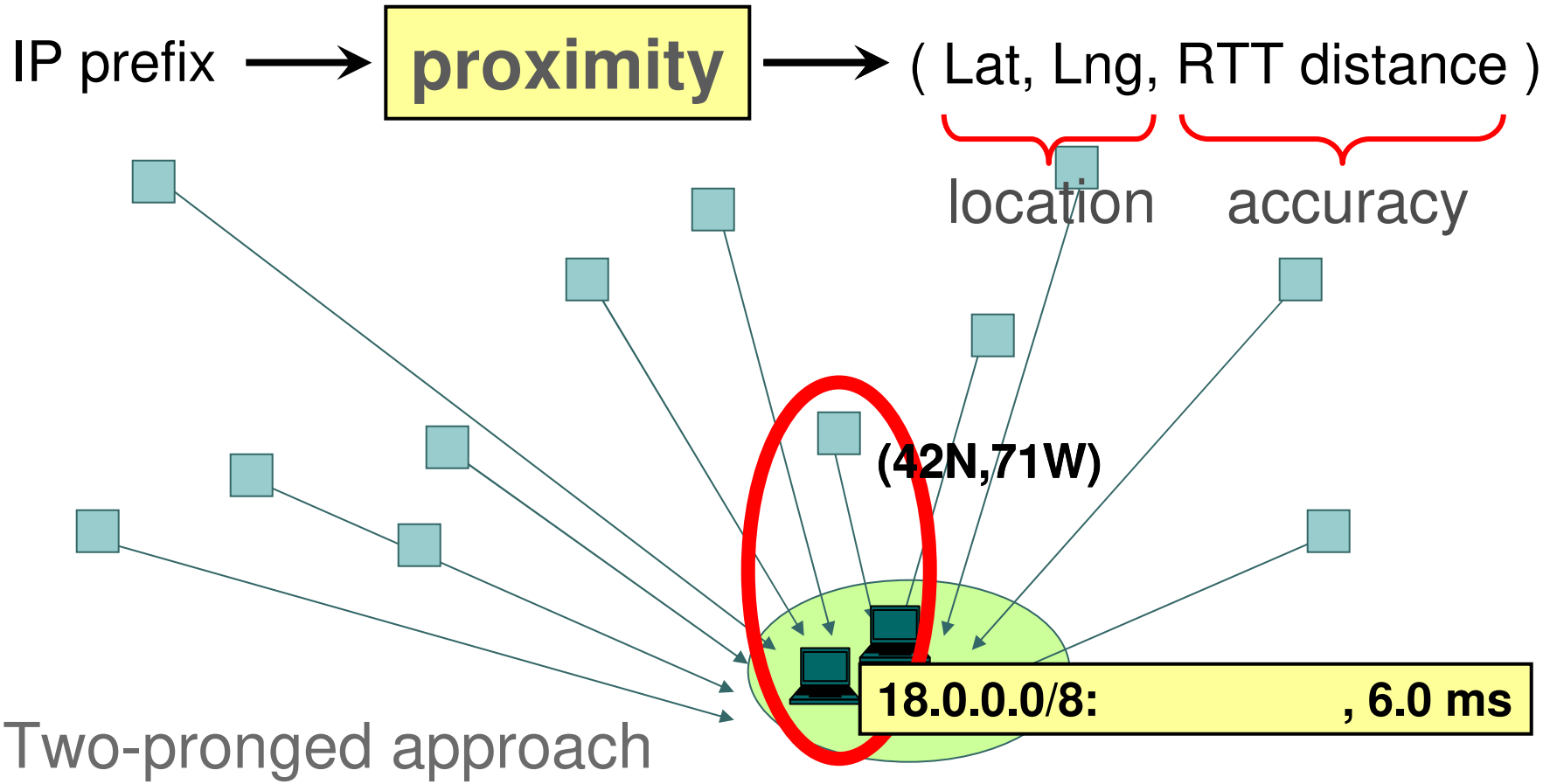


How to map IP prefix to coords?



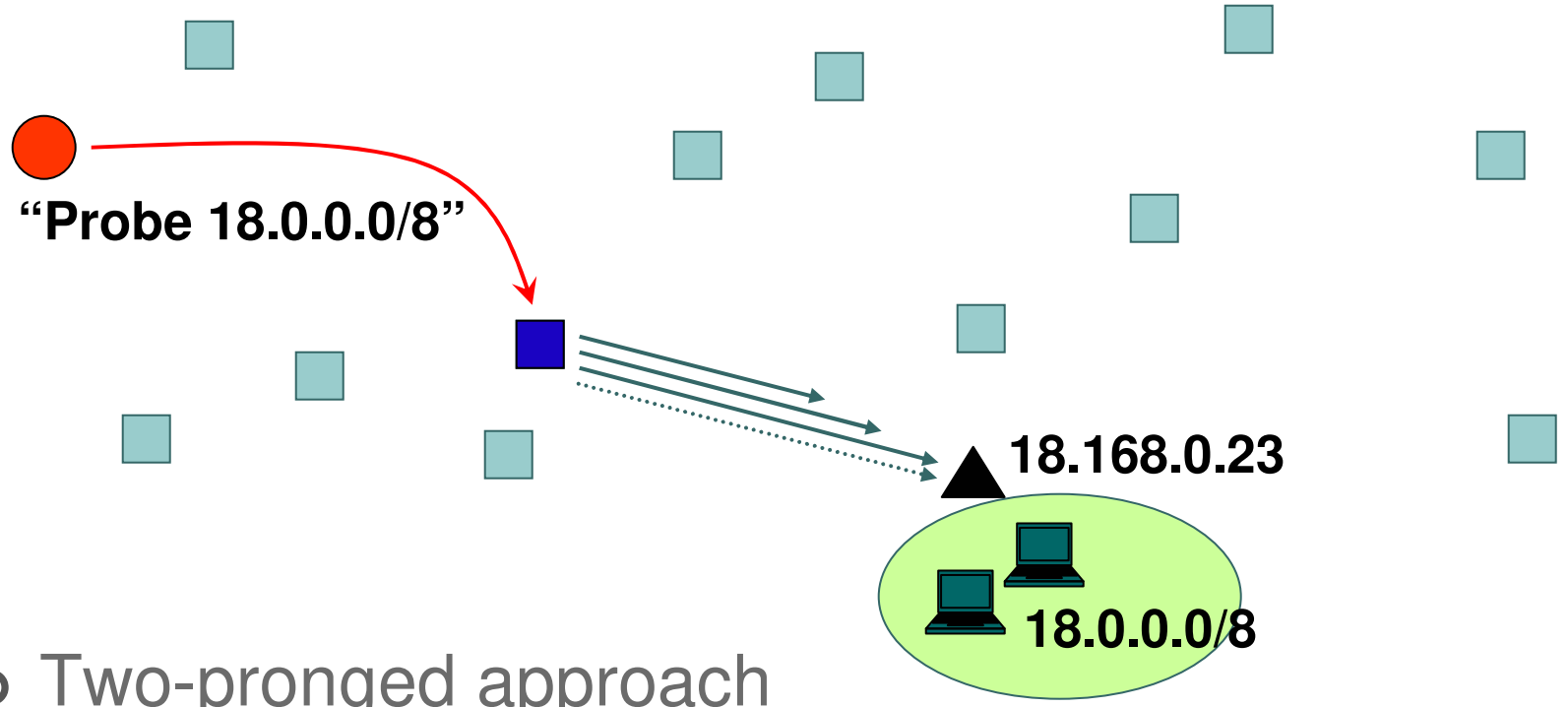
- Two-pronged approach
  - Find *closest* replica proxy

How to map IP prefix to coords?



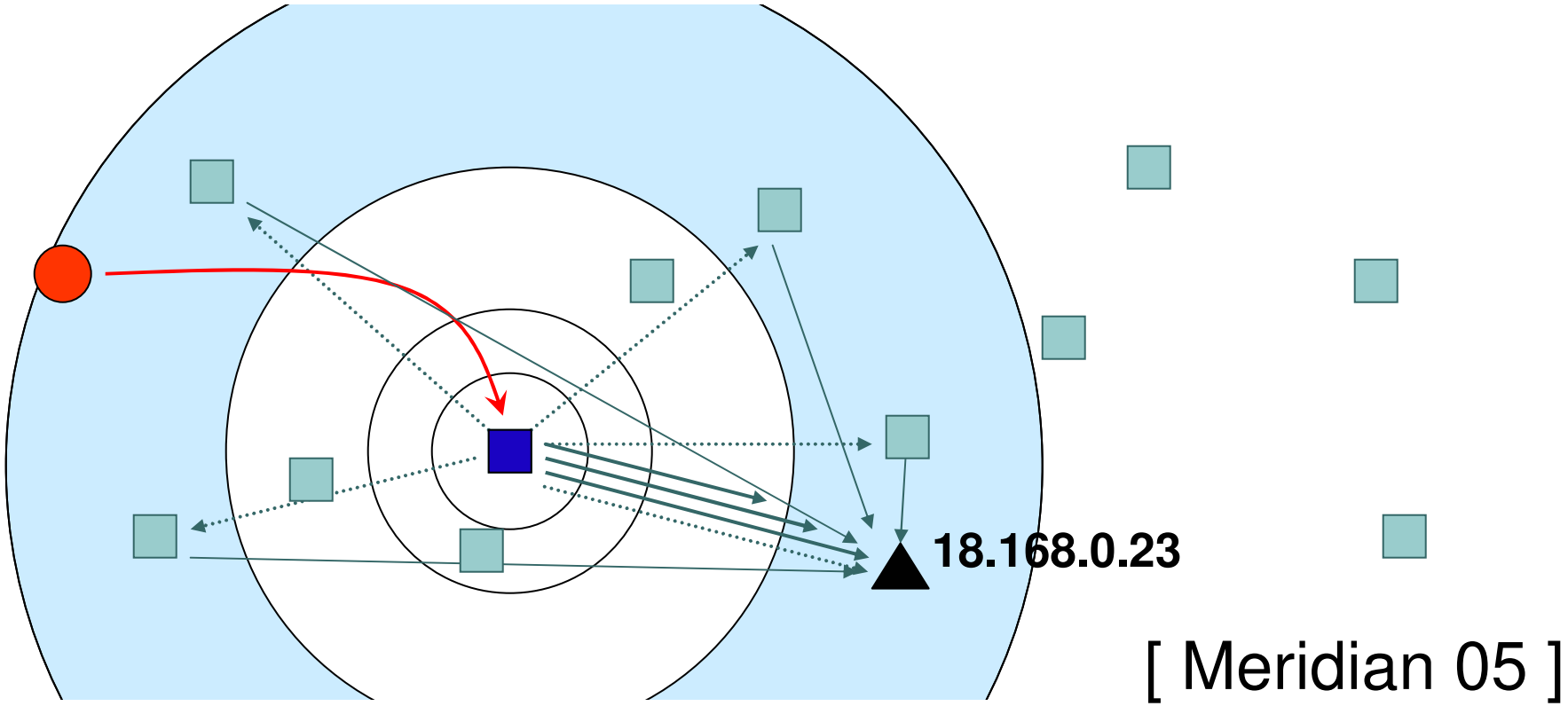
- Two-pronged approach
  - Find *closest* replica proxy
  - Use closest replica's geo-coords + error RTT as location

Find replica nearest prefix efficiently



- Two-pronged approach
  - Find *closest* replica proxy **with less probing**
  - Use closest replica's geo-coords + error RTT as location

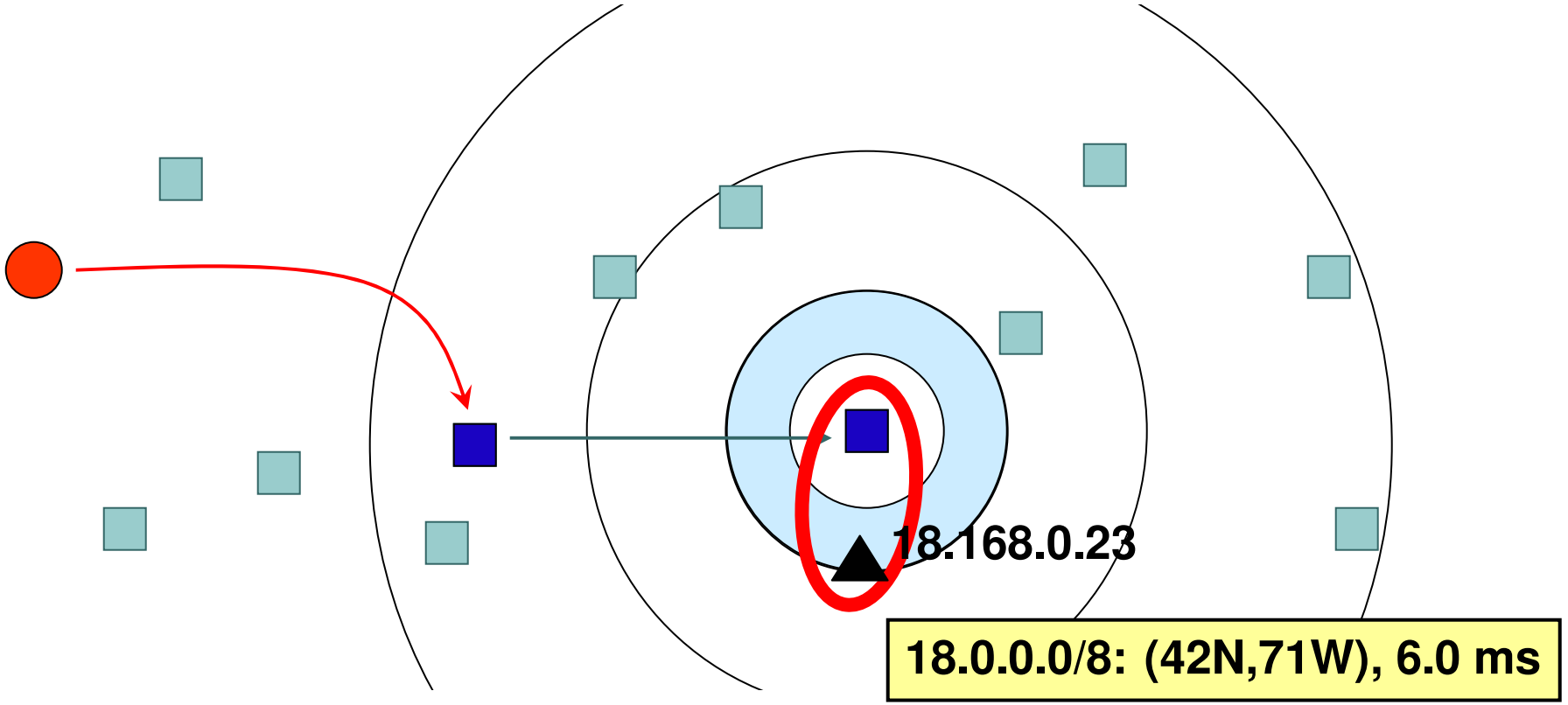
Find replica nearest prefix efficiently



[Meridian 05]

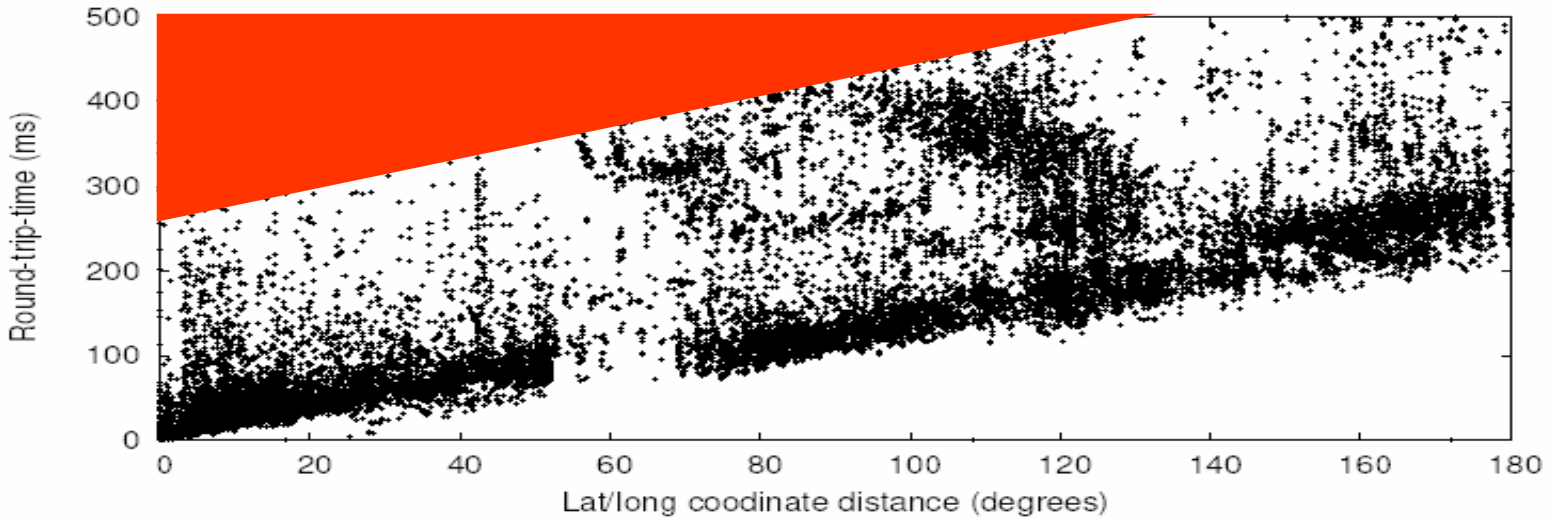
- Two-pronged approach
  - Find *closest* replica proxy **with less probing**
  - Use closest replica's geo-coords + error RTT as location

Find replica nearest prefix efficiently



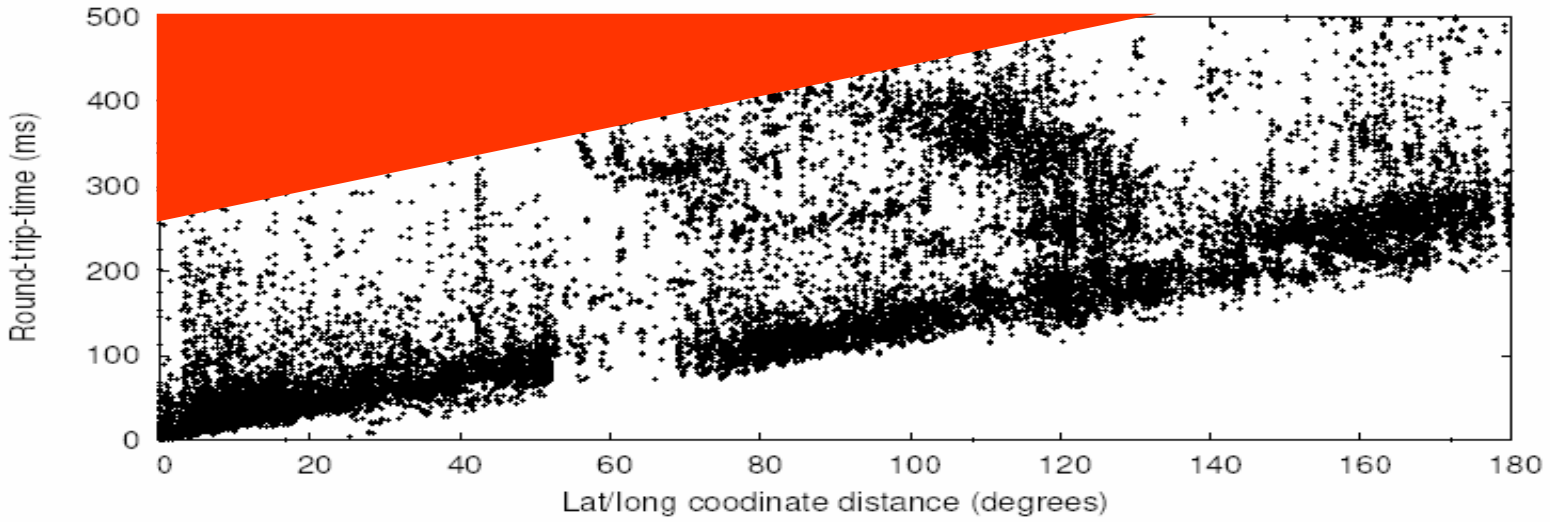
- Two-pronged approach
  - Find *closest* replica proxy **with less probing**
  - Use closest replica's geo-coords + error RTT as location

Geographic distance vs. RTT



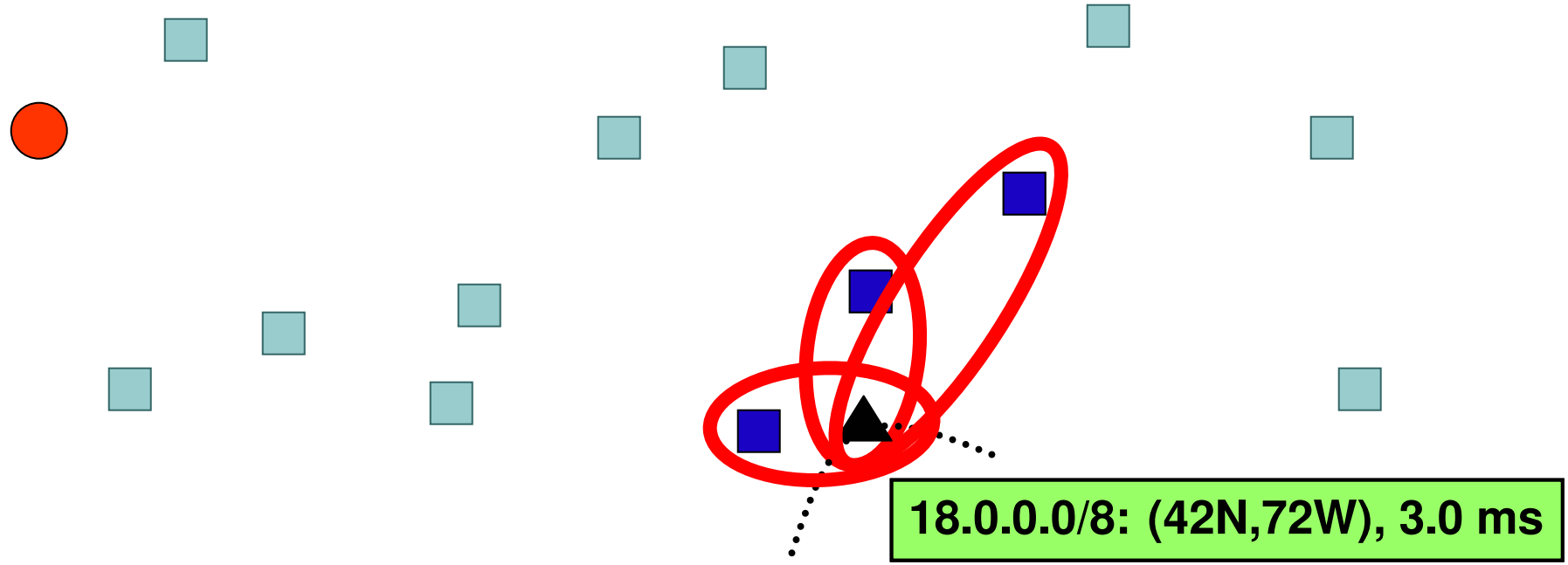
- Strong correlation b/w geographical distance and RTT

Geographic distance vs. RTT



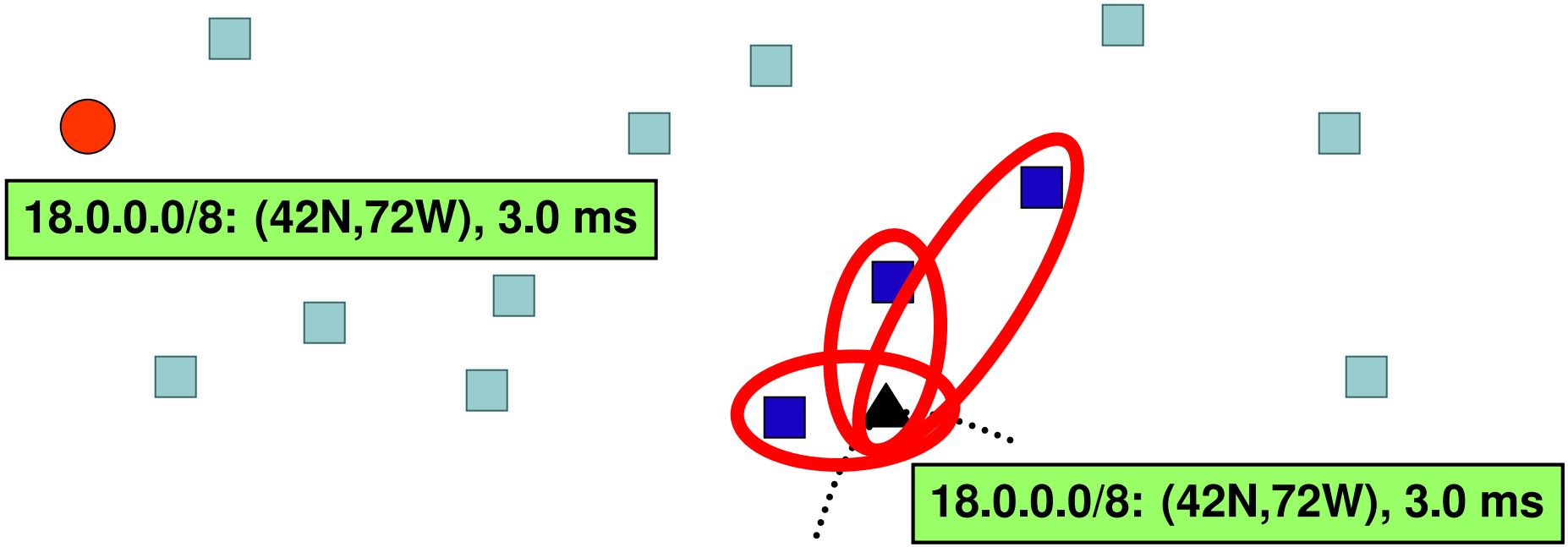
- Strong correlation b/w geographical distance and RTT
- RTT accuracy has real-world meaning
  - Check if new coordinates improve accuracy vs. old coords

Geographic distance vs. RTT



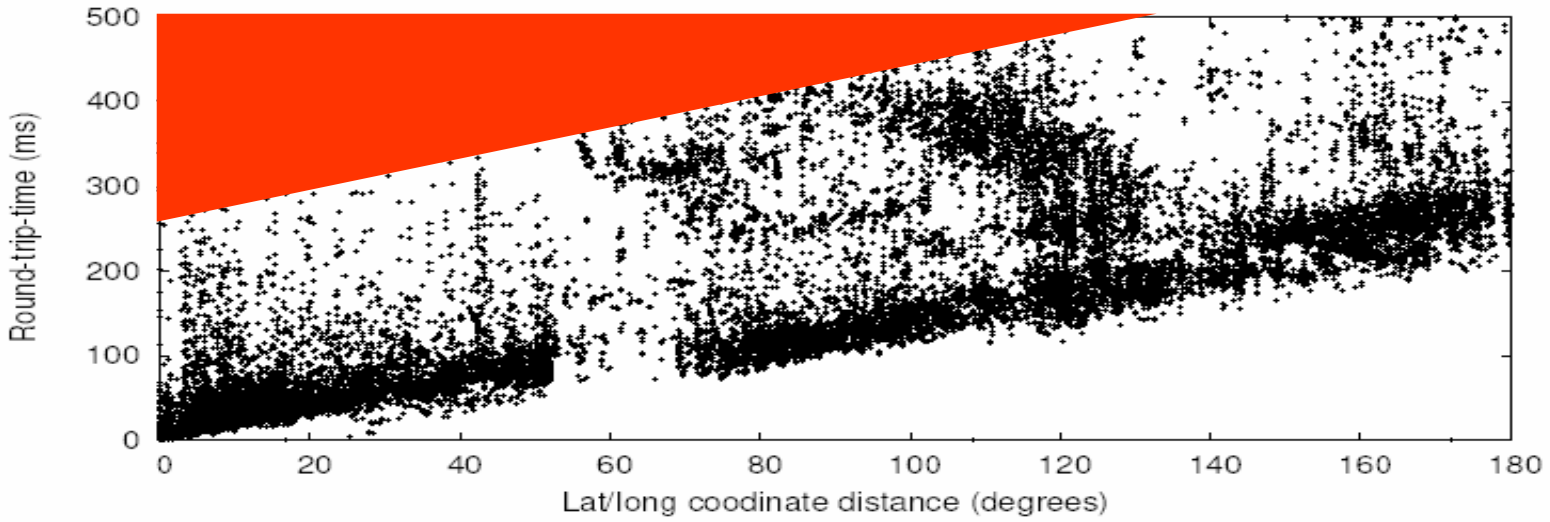
- RTT accuracy has real-world meaning
  - Check if new coordinates improve accuracy vs. old coords

Geographic distance vs. RTT



- RTT accuracy has real-world meaning
  - Check if new coordinates improve accuracy vs. old coords

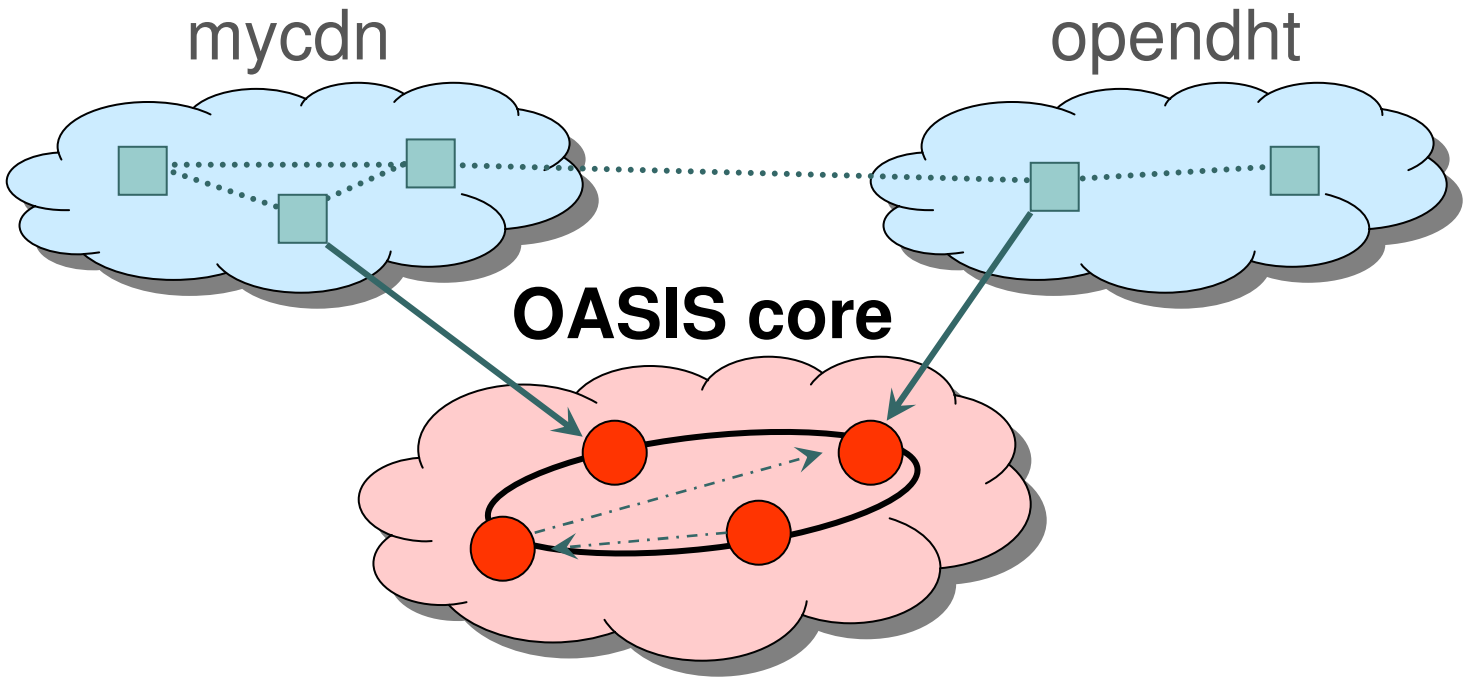
Geographic distance vs. RTT



- Strong correlation b/w geographical distance and RTT
- RTT accuracy has real-world meaning
  - Check if new coordinates improve accuracy vs. old coords
- Useful for sanity check for network peculiarities
  - Do multiple results satisfy constraints (e.g., speed of light) ?

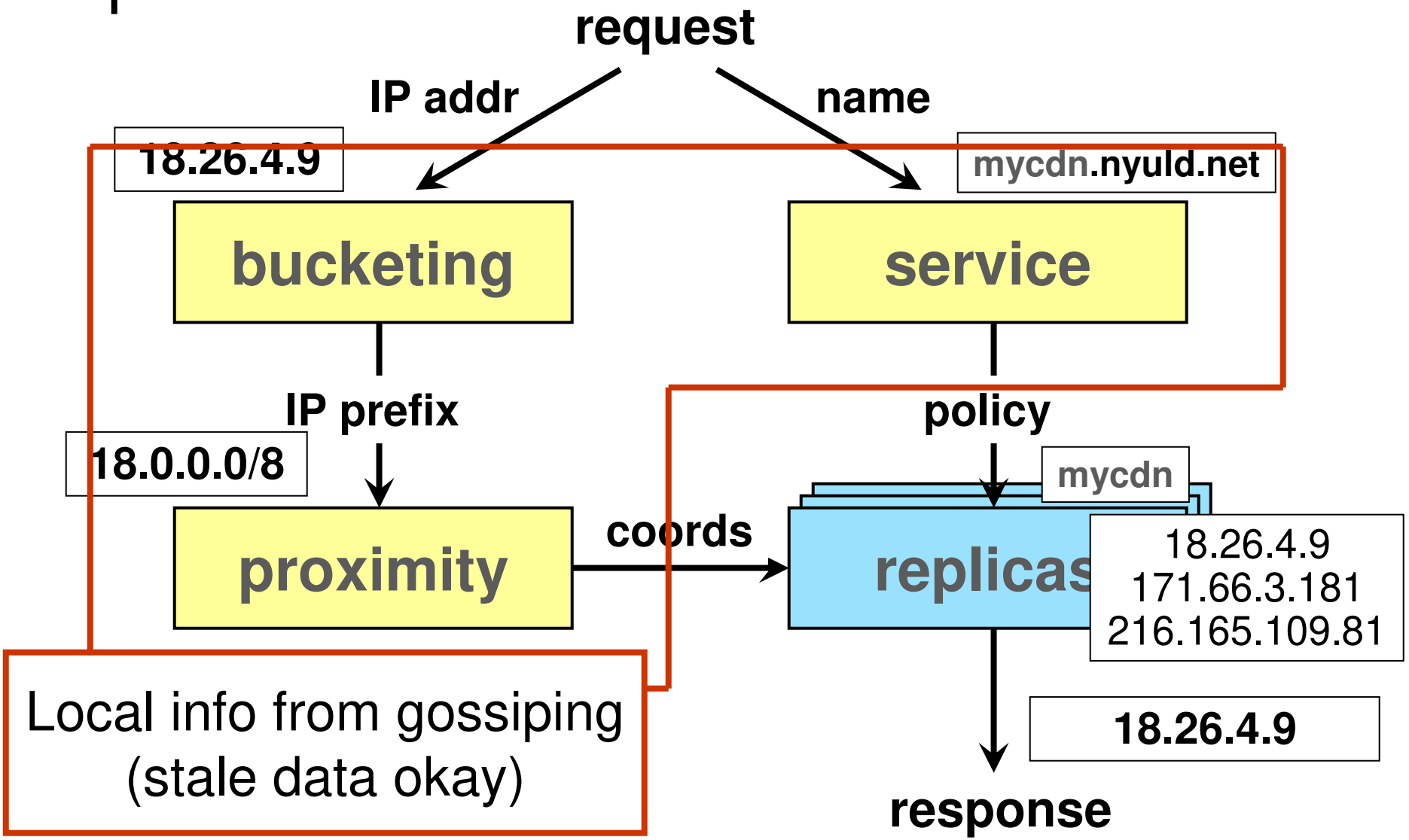
Outline

- Architecture and design decisions
- Detailed design
- Evaluation
- Deployment and integration lessons
  - OASIS deployed since November 2005
  - Currently in use by 10 services

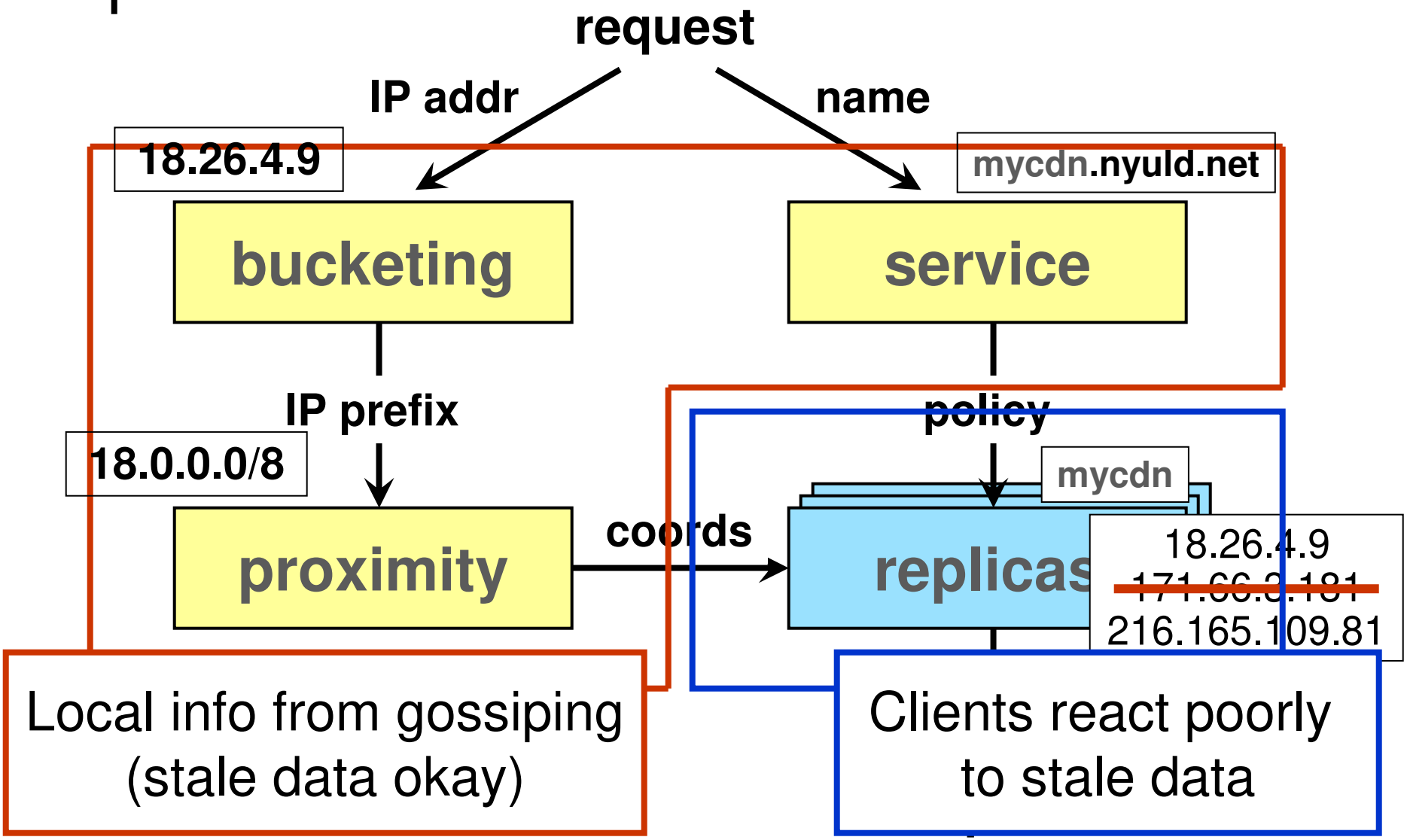


- OASIS core
  - Global membership view
  - Epidemic gossiping
    - Scalable failure detection
    - Spread policies, prefix→coords
  - Consistent hashing
    - Divide up responsibility for prefixes
- Service replicas
  - Heartbeats to OASIS node
  - Form global Meridian overlay for probing

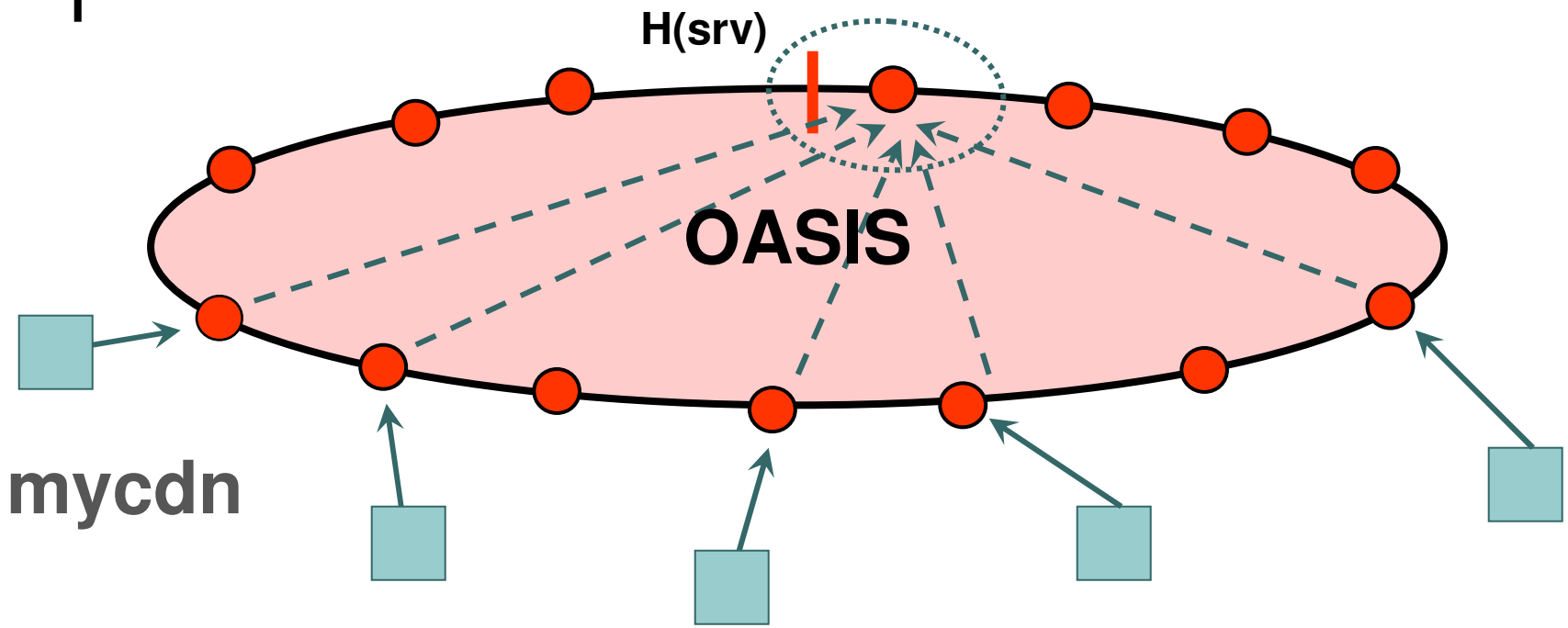
How to find “nearby” nodes?



How to find “nearby” nodes?

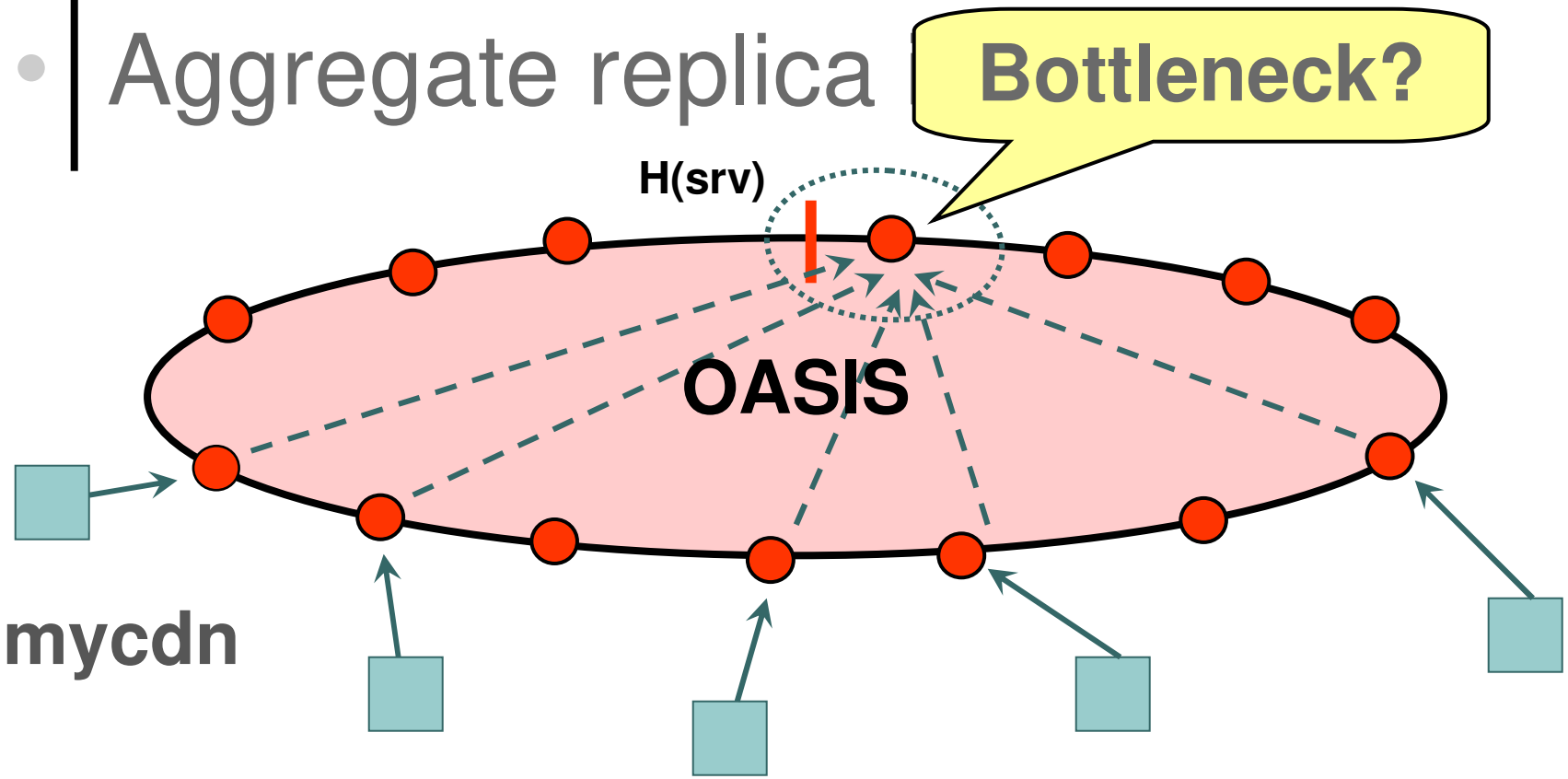


Aggregate replica information



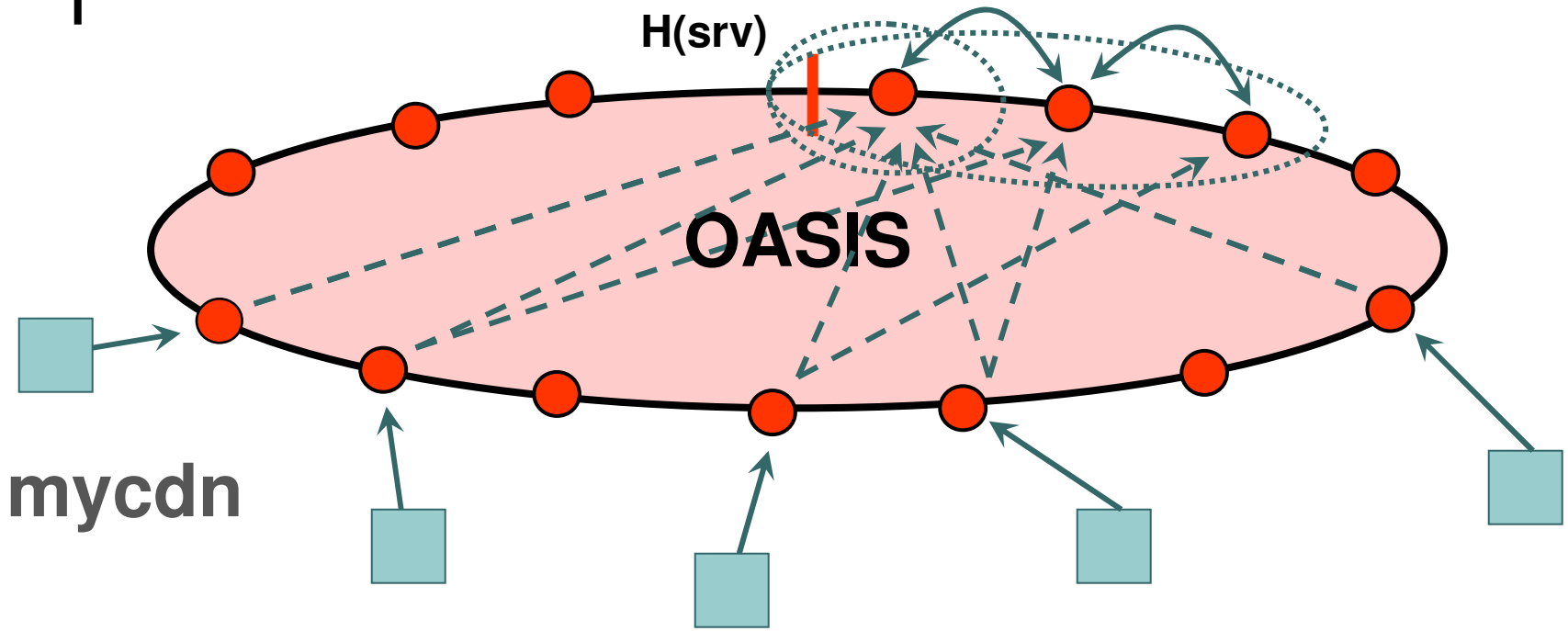
- Define service's rendezvous node via consistent hashing
- Service replicas send keepalives to nearby OASIS nodes
- Update rendezvous when replicas join, leave, large load change

Aggregate replica



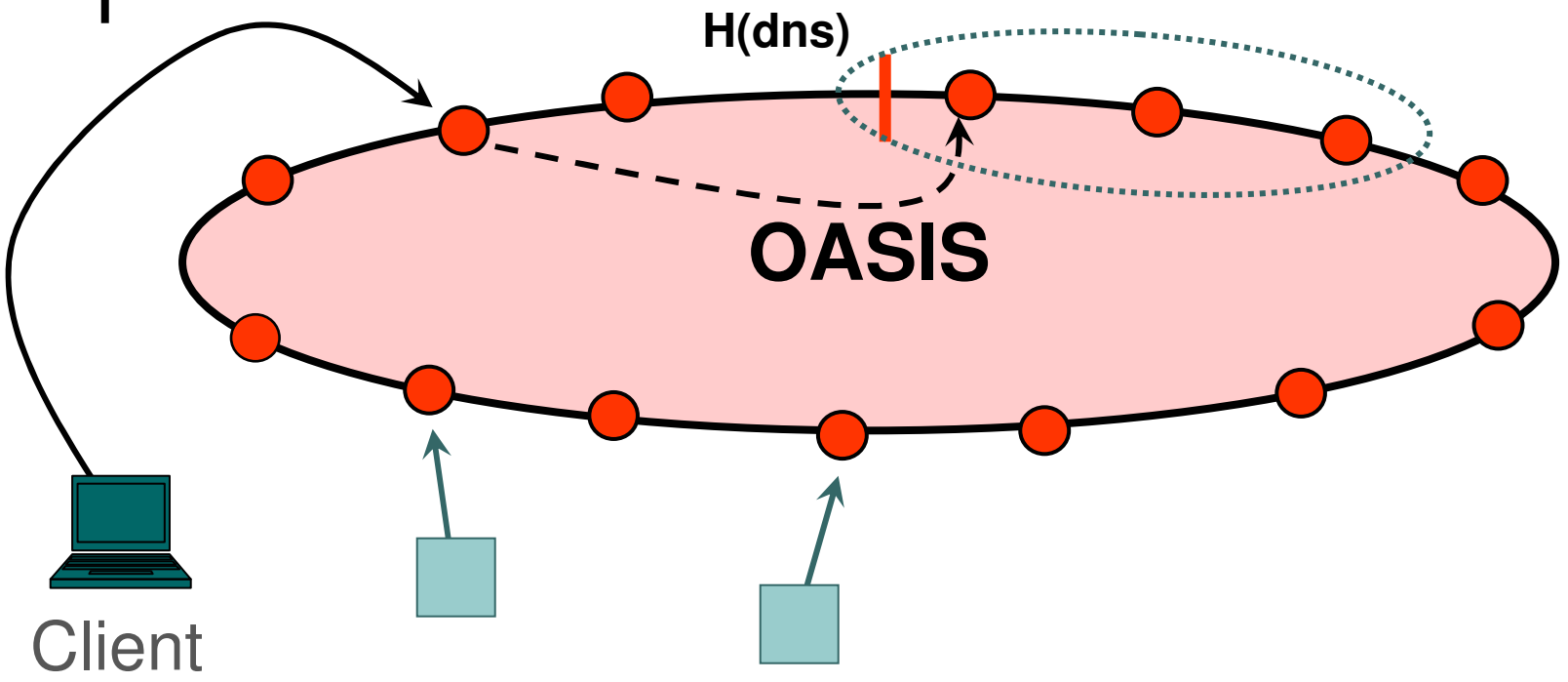
- Define service's rendezvous node via consistent hashing
- Service replicas send keepalives to nearby OASIS nodes
- Update rendezvous when replicas join, leave, large load change

Aggregate replica information



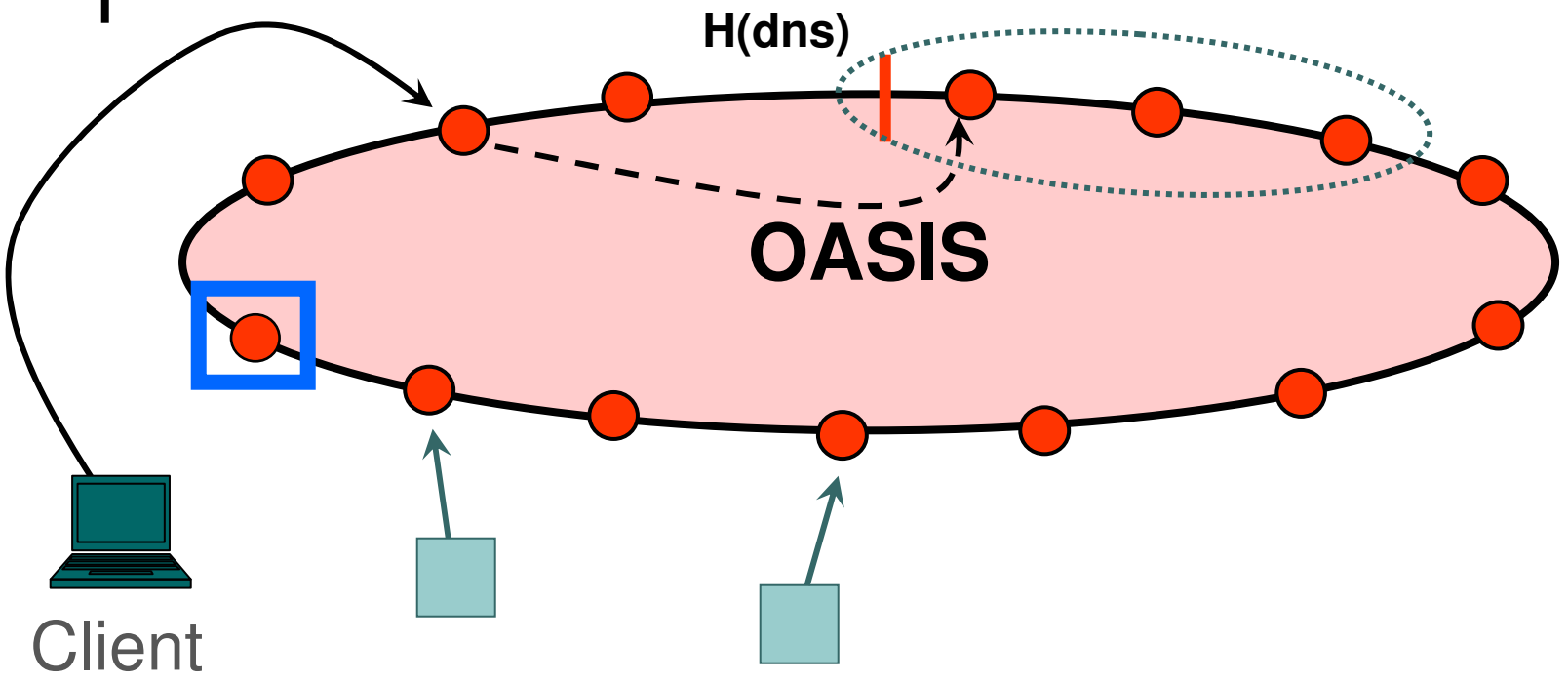
- Aggregate over *k* nodes for scalability
- Rendezvous gossip liveness state for loose consistency
- *k* can be dynamic for better scalability

A client's view: Finding a nameserver



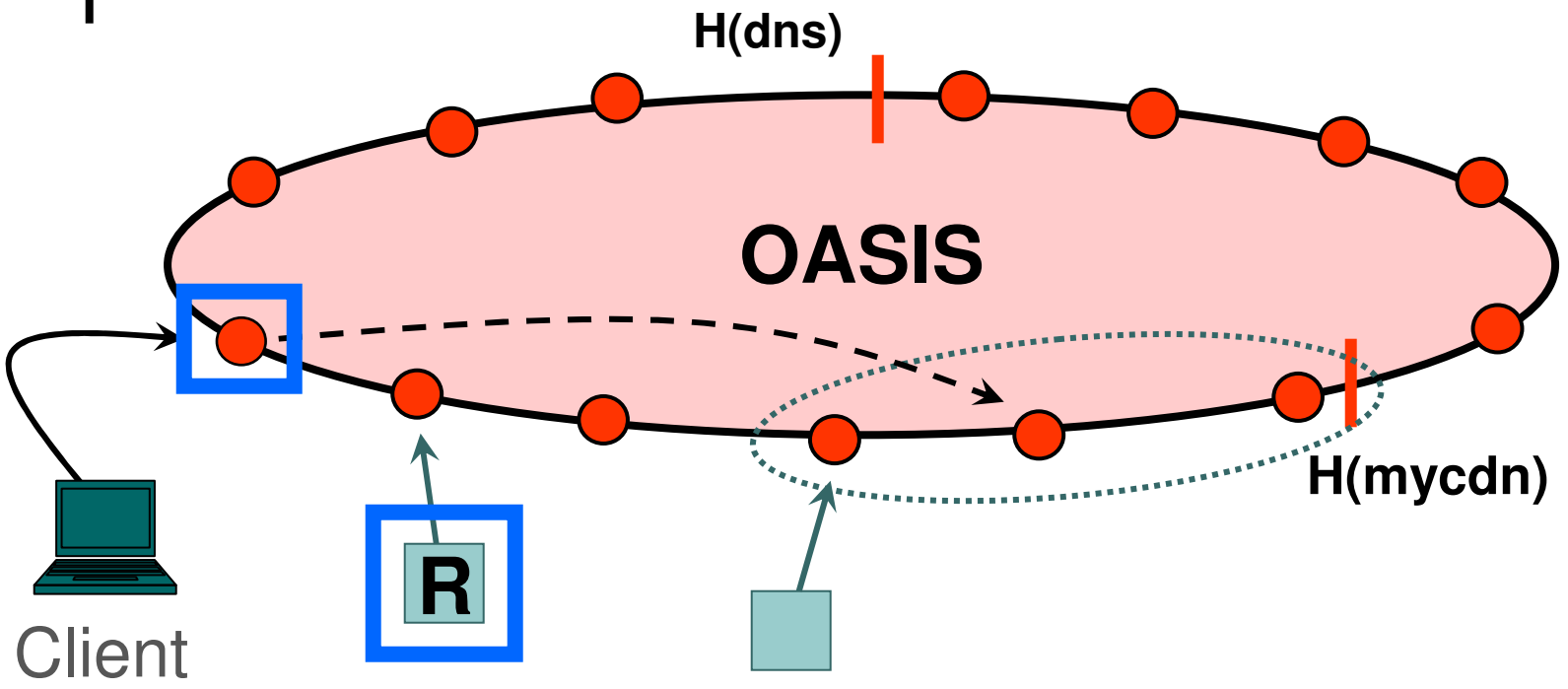
- Core lookup: Contacts 1 of 13 nameservers for *.nyuld.net*
  - OASIS "uses itself" to discover replica for service ***dns***

A client's view: Finding a nameserver



- Core lookup: Contacts 1 of 13 nameservers for *.nyuld.net*
  - OASIS "uses itself" to discover replica for service ***dns***
  - Returns nearby nameservers for subsequent requests

A client's view: Finding a replica



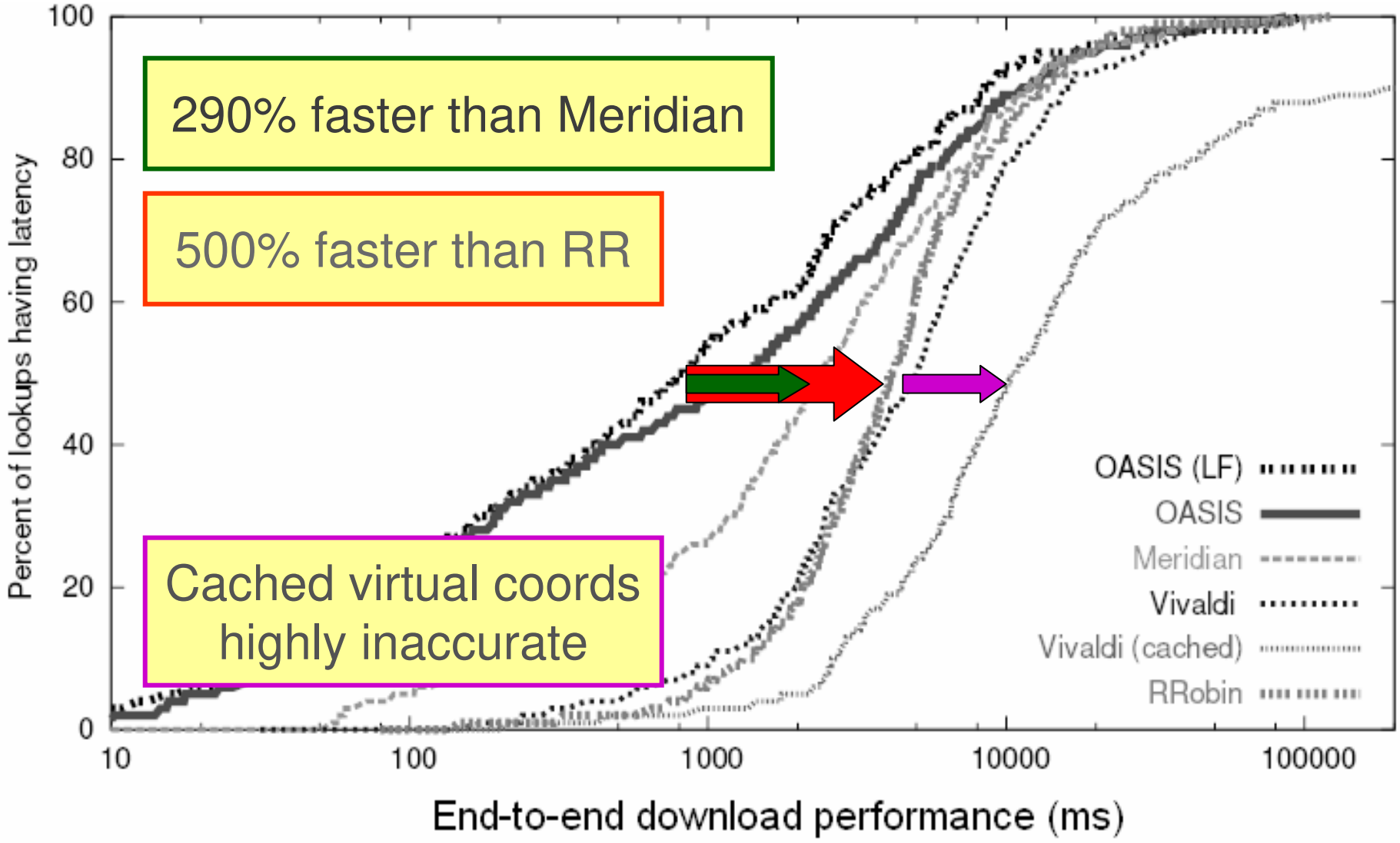
- Replica lookup: Client contacts nearby nameserver
  - OASIS discover replica for service ***mycdn***
  - Returns nearby replicas for application

Evaluation

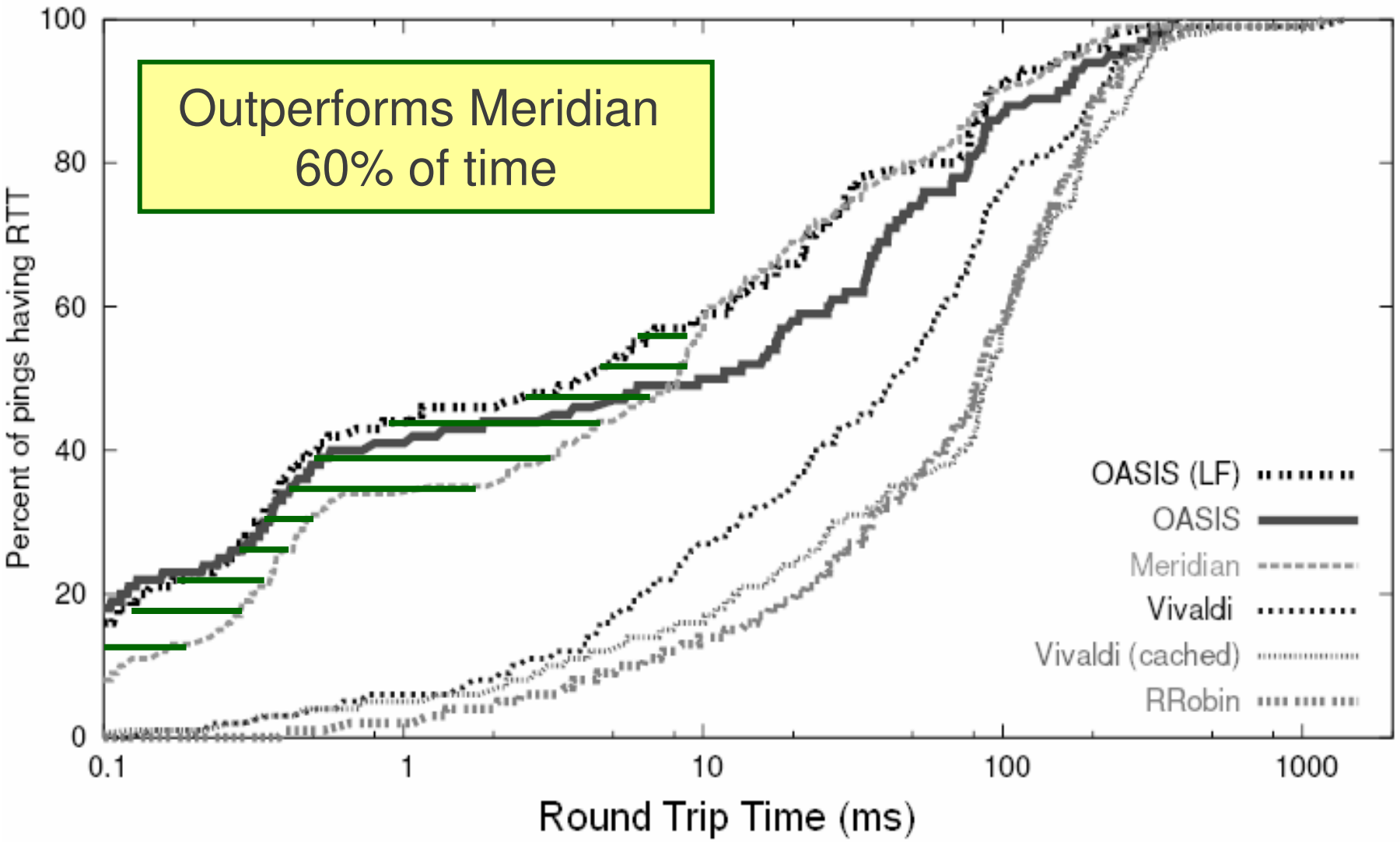
- Deployed on PlanetLab since November 2005

- How much end-to-end benefit from OASIS?
- How accurate is OASIS?
- Effective for load balancing?
- What are OASIS's bandwidth costs?

E2E download of web page



Client RTT to chosen replica



OASIS minimizes bandwidth spikes

95% bandwidth usage per replica (MB)

metric \ loc	CA	TX	NY	Germany
Latency Only	**23.3**	**0.0**	**0.0**	**0.0**
Load + Latency				

- 8 clients in CA repeatedly request 1 MB file
- Replicas report load as *log* (95% bandwidth per 1-min slot)

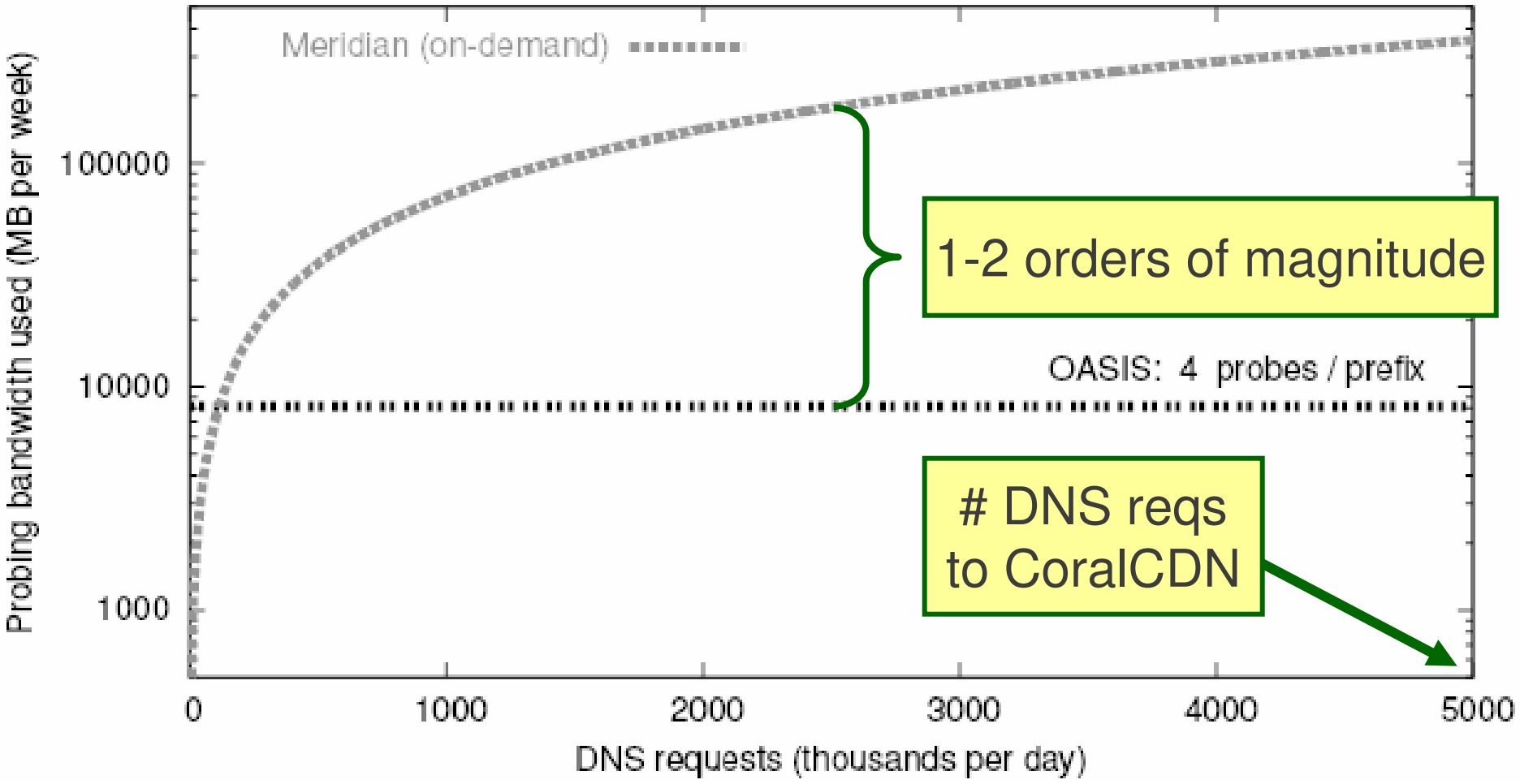
OASIS minimizes bandwidth spikes

95% bandwidth usage per replica (MB)

metric \ loc	CA	TX	NY	Germany
Latency Only	**23.3**	**0.0**	**0.0**	**0.0**
Load + Latency	**9.0**	**11.3**	**9.6**	**9.2**

- 8 clients in CA repeatedly request 1 MB file
- Replicas report load as *log* (95% bandwidth per 1-min slot)

Bandwidth costs: OASIS v. on-demand

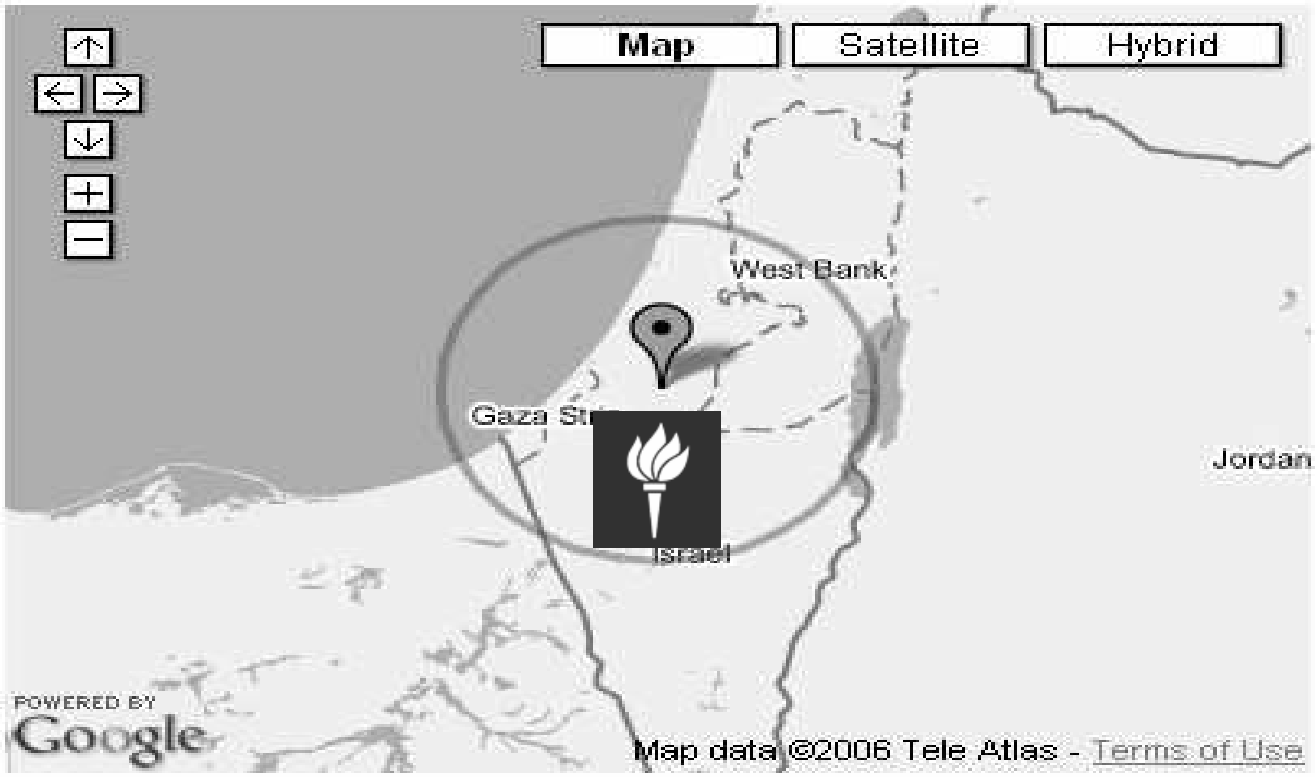


Outline

- Architecture and design decisions
- Detailed design
- Evaluation
- Deployment and integration lessons
  - OASIS deployed since November 2005
  - Currently in use by 10 services

Sanity check for network peculiarities

- Employ measurement redundancy
- Easy visualization significantly helped debugging



Netops have low tolerance for probing

- Probing generates abuse complaints
- Your service can get blacklisted!

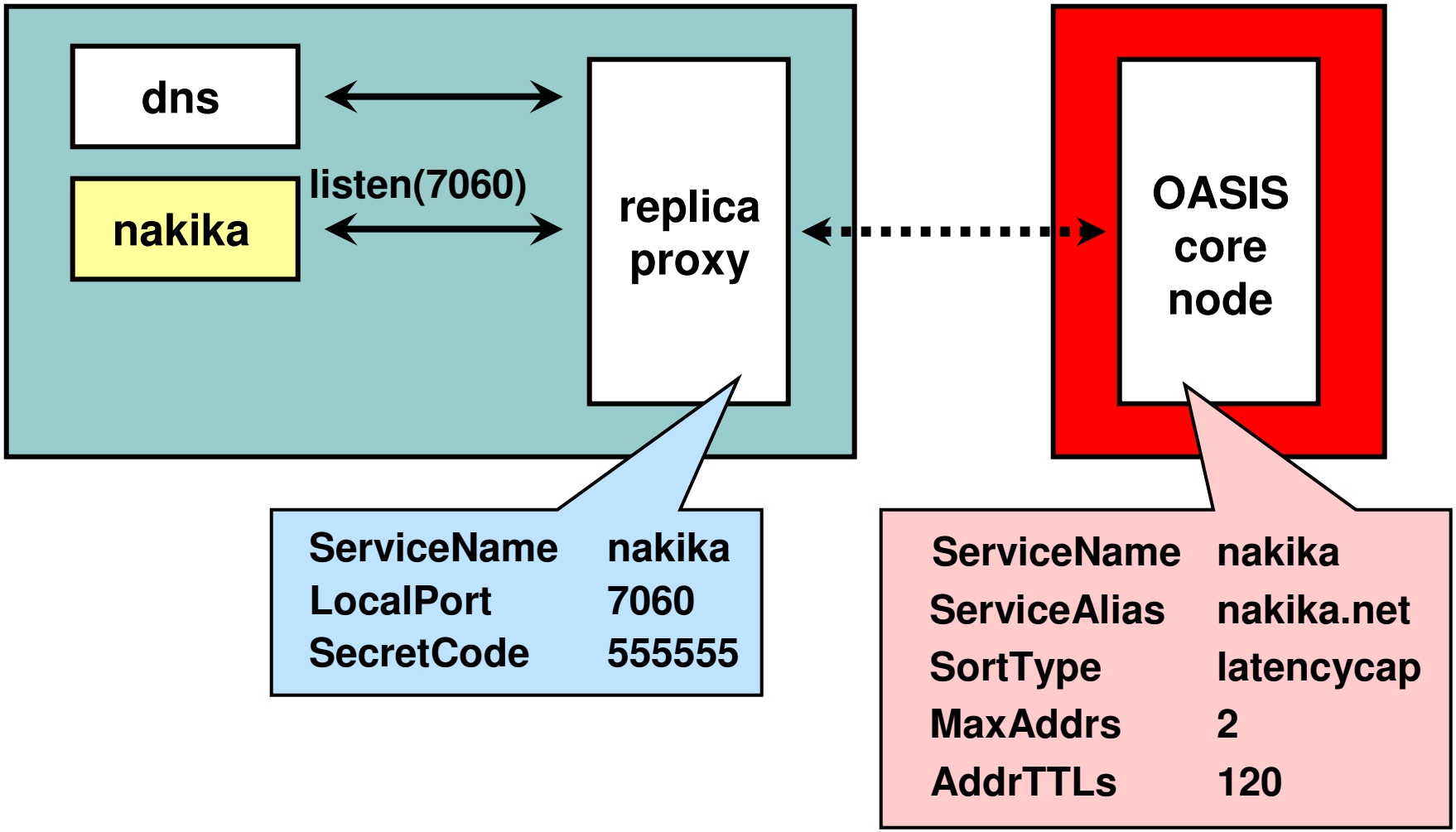
Keyword	Threads	Msgs	Keyword	Threads	Msgs
abuse	198	888	ICMP	64	308
attack	98	462	IDS	60	222
blacklist	32	158	intrusion	14	104
block	168	898	scan	118	474
complaint	216	984	trojan	10	56
flood	4	30	virus	24	82

Keyword frequency on PlanetLab support lists
9 months, 1820 threads, 4682 msgs

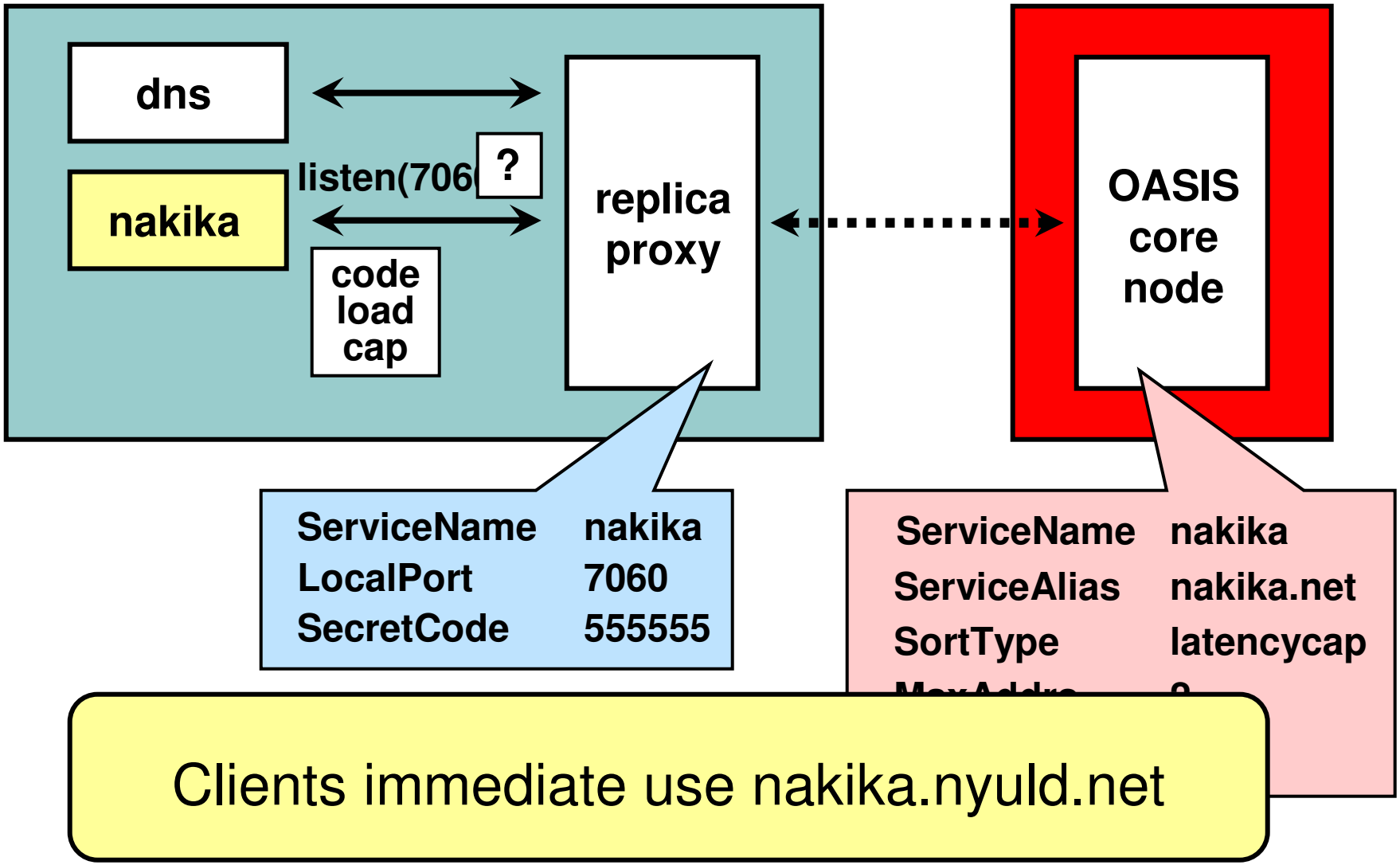
Netops have low tolerance for probing

- Be careful *what* you probe
  - Probe slowly and rarely
  - No random ports or obvious attack vectors (TCP port 22/23)
- Be careful *whom* you probe
  - Check blacklist for netblock *and* target IP (after traceroute)

Make it easy to integrate



Make it easy to integrate



Clients immediate use nakika.nyuld.net

Current services using OASIS…

- **Chunkcast** block anycast (Berkeley)
- **CoralCDN** (NYU)
- **Na Kika** content distribution (NYU)
- **OASIS**
  - RPC, DNS, HTTP interfaces
- **OCALA** overlay convergence (Berkeley)
  - Separate services for client and server IPs gateways
- **OpenDHT** public DHT service (Berkeley)
- **OverCite** distributed library (MIT): Deployed on RON

Summary

- OASIS is a general, open anycast service
  - Supports multiple services: more are better
  - Performs accurate server selection
  - Removes all on-demand probing
  - Provides easy integration

- Use OASIS for your distributed system!

http://oasis.coralcdn.org/