A Global Name Service for a Highly Mobile Internetwork

Abhigyan Sharma, Xiaozheng Tie, Hardeep Uppal, Arun Venkataramani, David Westbrook, Aditya Yadav

School of Computer Science
University of Massachusetts Amherst
Mobile arrived, but Internet unmoved

- Unidirectional communication initiation
- Redundant app-specific mobility support

Cleaner separation of location and identity commonly advocated wisdom
But DNS does separate identity / location

- Domain names ↔ IP addresses
- connection migration techniques

- **Challenge:** scaling to handle update cost of frequent mobility while returning up-to-date values
  - Example: 10B devices, 100 addresses/day ≈ 1M updates/sec
  - DNS update propagation can take hours or days today!

**How to force browsers/ISPs to look for my new DNS?**

1. I have changed the DNS for my domain. What code (or header) should I use in my old server to tell the visitor's browser or ISP that it should check for my new DNS and the current content is old?

2. Is the temporary redirecting to a subdomain should help?

3. Or do you know a better way?
Goal: A massively scalable, logically centralized GNS to enable secure, name-based communication with flexible endpoint principals with arbitrary (fixed) names despite high mobility.
Outline

- Poor intrinsic support for mobility today
- **Case for a next-generation GNS**
- Auspice GNS design
- Implementation and evaluation
- Related work, open issues, summary
GNS critical to handle mobility

- Pre-lookup mobility
- Connect-time mobility
- Individual mobility
- Simultaneous mobility

Global name service

GNS critical or can significantly benefit mobility handling in any network architecture
DNS limitations

Passive caching
Static placement
Hierarchical names

Authoritative name-server ns.xyz.net

Single root of trust

“JohnSmith2178@Amherst”
“Living room chandelier”
“Taxis near Times Square”
Outline

- Poor intrinsic support for mobility today
- Case for a next-generation GNS
- Auspice GNS design
- Implementation and evaluation
- Related work, open issues, summary
GNS: Decoupling certification and resolution

Domain name system

- Root name service (ICANN, US. Dept. of Commerce)
- TLD name services
- Auth. name services
- Managed DNS services
- Local name services

Global name system

- Name certification services
- Certificate search services
- Auspice-like global name services
- Managed DNS services
- Local name services

Name: “Alice’s phone”

GUID=X, GNS=Auspice

getAddress(X) \[ [P_1, P_2, ...]\]
### Active replication cost-benefit tradeoff

Update cost for name $i \propto (#\text{replicas}_i) \times (\text{update\_rate}_i)$

Lookup latency for name $i$?

<table>
<thead>
<tr>
<th>Resource cost</th>
<th>Name lookup latency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Auspice</strong></td>
<td></td>
</tr>
<tr>
<td>Replicate-at-all-locations</td>
<td></td>
</tr>
<tr>
<td>Resource limit</td>
<td>Consistent hashing with (static) $k$-replication</td>
</tr>
</tbody>
</table>

**Equation:**

$$\text{Update cost} = i \propto (#\text{replicas}_i) \times (\text{update\_rate}_i)$$
Demand-aware active replication

- \#replicas of name \( i \) \( \propto \) \( \frac{\text{read_rate}_i}{\text{update_rate}_i} \)

Geolocality-aware  \( i \)  \( j \)  Load-aware
Placement reconfiguration engine

Consistent hashing based placement control plane

Planned demand-aware placement for data plane
Outline

- Poor intrinsic support for mobility today
- Case for a next-generation GNS
- Auspice GNS design
- Implementation and evaluation
- Related work, open issues, summary
Implementation

- Geo-distributed key-value store

  ```
  GUID: {
    {IPs: [123.45.67.89, 98.76.54.321]},
    {geoloc:[lat, long]},
    {TE_prefs: [“prefer WiFi”,...]},
    {ACL: {whitelist: [...]}}},
  ...
  }
  ```

- Name certification service

  Human-readable name: abhigyan@cs.umass.edu:phone
  GUID: 21EC2020-3AEA-4069-A2DD-08002B30309D

- **msocket** user-level socket library with Auspice integration

  ```
  MSocket socket = new MSocket(abhigyan@cs.umass.edu:phone);
  MServerSocket socket = new MServerSocket(8080);
  ```
 Placement schemes comparison

**Testbed**: 16 server cluster emulating with 80 NS an 80 local NS

**Workload**: 90% mobile names (geolocality 0.75), 10% service names

Auspice gives close to best throughput and up to 9x lower latency than DHT+Popularity.
Managed DNS comparison

Ultra DNS (16 replicas) vs. Auspice 5/10/15 replicas out of 80 locations

One-third replication cost, similar latency

Auspice reduces cost/latency over today’s managed DNS

60% less latency, similar cost
Related work

- Classical name services [Grapevine/ClearingHouse] used *static* replication
  - Context-based names like Lampson’s “descriptive names”

“the XEROX system [Grapevine] was then … the most sophisticated name service in existence, but it was not clear that its heavy use of replication, light use of caching … were appropriate”

```python
msocket.bind([lat, long, radius])
msocket.send(msg)
```
Auspice GNS summary

Enables secure, name-based communication

- arbitrary name/location representation
- flexible endpoint principals
- handles all types of mobility

- Key differences from DNS for today’s Internet
  - federation decoupling certification and resolution
  - active replication
  - demand-aware placement

Get your GUID at: http://gns.name