

# Simulating Disaster Scenarios and Geographically-Correlated Resilient Overlay Networks

## Heuristics for Location-based Routing

Kyle E. Benson and Zhipeng Huang

Department of Computer Science  
University of California, Irvine  
Irvine, California 92697

[kebenson@uci.edu](mailto:kebenson@uci.edu) and [zhipengh@uci.edu](mailto:zhipengh@uci.edu)

June 14, 2013

# Challenges

- ▶ **Goal:** Improve data delivery ratio/speed during geographically-correlated network failures (earthquakes)
- ▶ No control over and limited knowledge about routing infrastructure
- ▶ But can gather location information
- ▶ **Claim:** Resilient Overlay Networks can help establish more reliable connections between end-hosts with limited topology knowledge

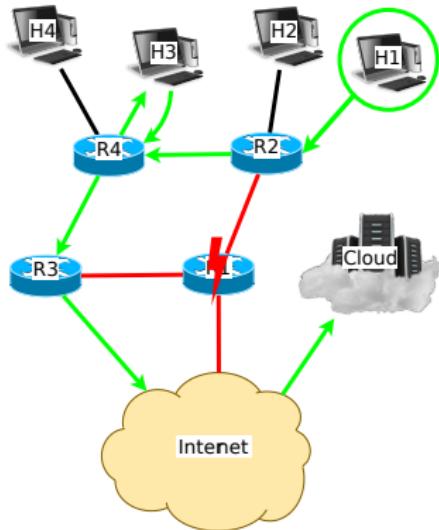
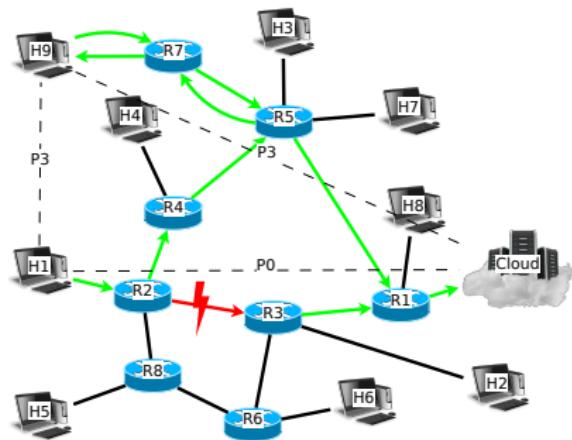


Figure: H1 routes around failed R1 using H3 as an overlay.

# Exploiting Location Information

- ▶ Contact overlay nodes outside of local region to avoid failures
- ▶ Choose overlay nodes from diverse regions not previously attempted
- ▶ Failure likely along path to server or in local area
- ▶ Choose node outside local region to avoid overlapping paths
- ▶ Choose path avoiding as much of the direct path as possible
- ▶ Overlay node may use similar route to sensor

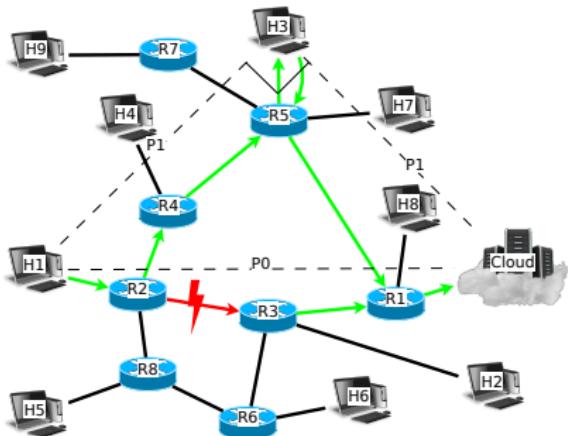


**Figure:** Sensors try to contact nodes outside of the local area and not along the straight-line path to the server.

# Orthogonal Distant Path Heuristic

## Algorithm 1:

```
begin
    /* Sensor, overlay and server
    node are a, c, b respectively */
    idealDist = |0.5 · dist(a, b)|
    perpDist = |sin(angA) · dist(a, c)|
    if angA >  $\frac{\pi}{2}$  or angA + angC <  $\frac{\pi}{2}$  or
    perpDist > dist(a, b) then
        likelihood = 0
    else
        likelihood = 0.5 · ((1.0 - (||perpDist| -
        |idealDist||/idealDist)2) + (1.0 -
        (| $\frac{\pi}{2}$  - angC|/ $\frac{\pi}{2}$ )2))
    end
```



**Figure:** The intuition of this heuristic is to avoid the straight path, without diverging from it too much. It strikes a middle ground by choosing a path at an ideal angle of  $45^\circ$ , which makes the angle at the top orthogonal, hence the name.

# New Region Heuristic

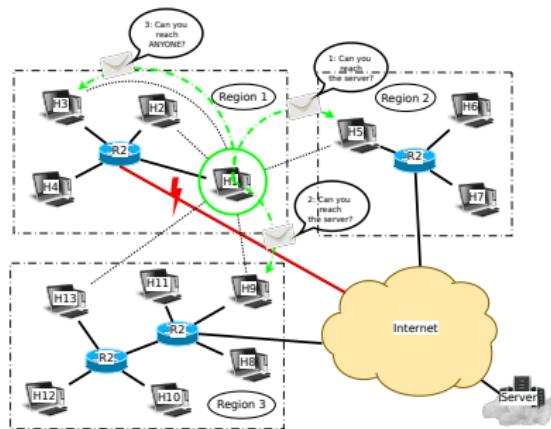
---

## Algorithm 2:

---

```
begin
    if peer.region ∈ regionsAttempted
    then likelihood = 0
    else likelihood = 1.0
end
```

---



**Figure:** The intuition of this heuristic is to avoid regions that have been previously attempted unsuccessfully. We assume that no further attempts to contact such a region will succeed.

# New Angle Path Heuristic

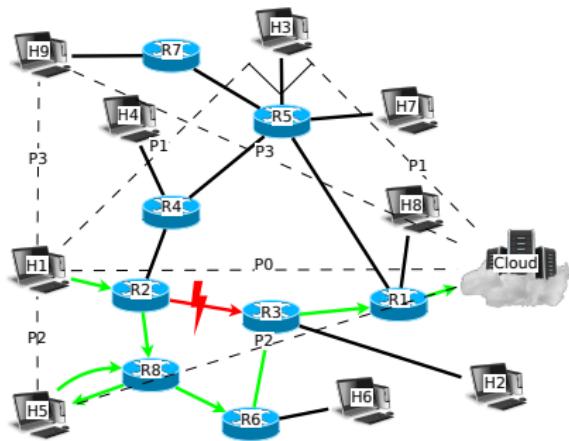
---

## Algorithm 3:

---

```
begin
    angle = Angle(overlay, server)
    if angle =  $\pi$  or angle = 0 then
        initLikelihood = 0
    else if angle <  $\pi$  then
        initLikelihood =  $\cos(\angle - \frac{\pi}{4})$ 
    else if angle >  $\pi$  then
        initLikelihood =
             $\cos(2 \cdot ((2\pi - \angle) - \frac{\pi}{4})/3)$ 
    foreach path  $\in$  pathsAttempted do
        thisLikelihood =  $|\frac{\sin(\angle)}{2}|$ 
        newLikelihood* = thisLikelihood
    end
```

---



**Figure:** This heuristic attempts paths along new angles different from the ones previously attempted.

# Distance-Dependent Path Heuristic

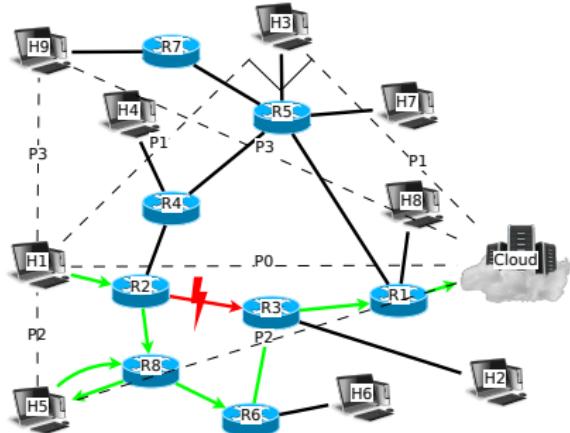
---

## Algorithm 4:

---

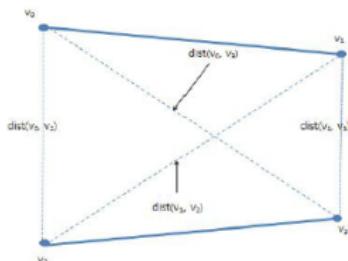
```
begin
    /* minDist = some reasonably
    small number */
    idealDist = 0.5 · dist(sensor, server)
    dist = dist(sensor, overlay) if
    dist <= minDist then likelihood = 0
    else if minDist < dist < idealDist
    then likelihood = dist2 / idealDist2
    else if dist >= idealDist then
    likelihood = idealDist / dist
end
```

---



**Figure:** This heuristic attempts paths that use overlay nodes at an ideal distance from the sensor. This ideal distance is chosen as a radius that reaches half-way to the server.

# Furthest First Path Heuristic

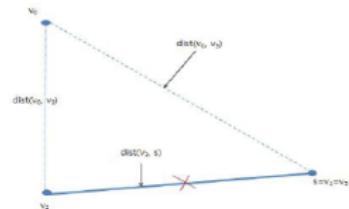


**Figure:** Definition of path similarity presented in W. Sun, “A Method for Overlay Network Latency Estimation from Previous Observation”, ICN-2013.

$$\text{geo\_similarity}(v_0, v_1, v_2, v_3) = \min[\text{dist}(v_0, v_2) + \text{dist}(v_1, v_3), \text{dist}(v_0, v_3) + \text{dist}(v_1, v_2)]$$

When considering one server,

$$\text{geo\_similarity}(v_0, v_1, v_2, v_3) = \text{dist}(v_0, v_2) = \text{geo}(v_0, v_2)$$



**Figure:** Thus, this heuristic considers overlay peers located further away to be more likely candidates.

---

**begin**

```
    /* minDistance = some constant  
       small number */  
    if  $dist(a, b) < minDistance$  then  
        likelihood = 0  
    else likelihood =  
         $(dist(a, b) - minDistance)/dist(a, b)$   
end
```

---

# Closest First Path Heuristic

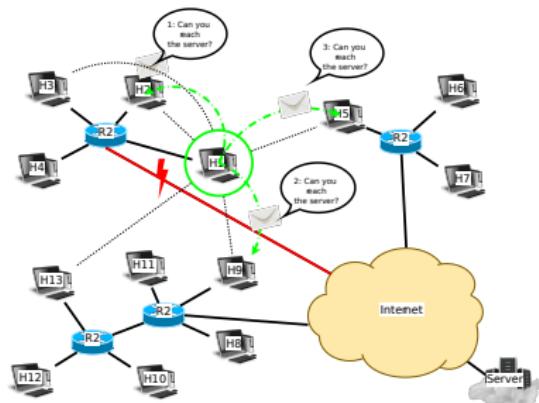
---

## Algorithm 5:

---

```
begin
    /* maxDistance = some constant
       large number */
    if dist(a, b) > maxDistance then
        likelihood = 0
    else likelihood = (maxDistance -
        dist(a, b))/maxDistance
end
```

---

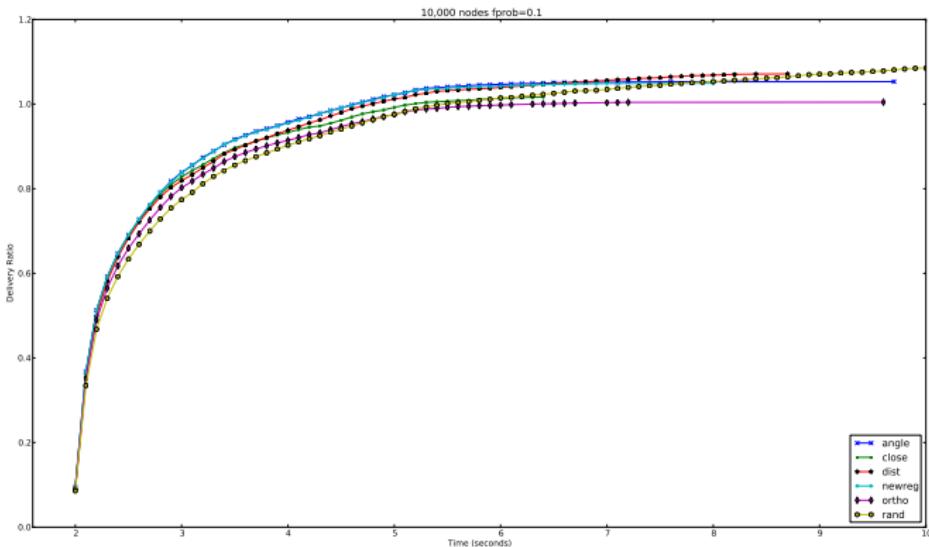


**Figure:** The intuition of this heuristic is to contact nearby overlay nodes that have found a path out of the local region. We found, however, that this approach does not always work well in practice, likely due to the path similarity inherent with nearby nodes.

## Simulation Environment

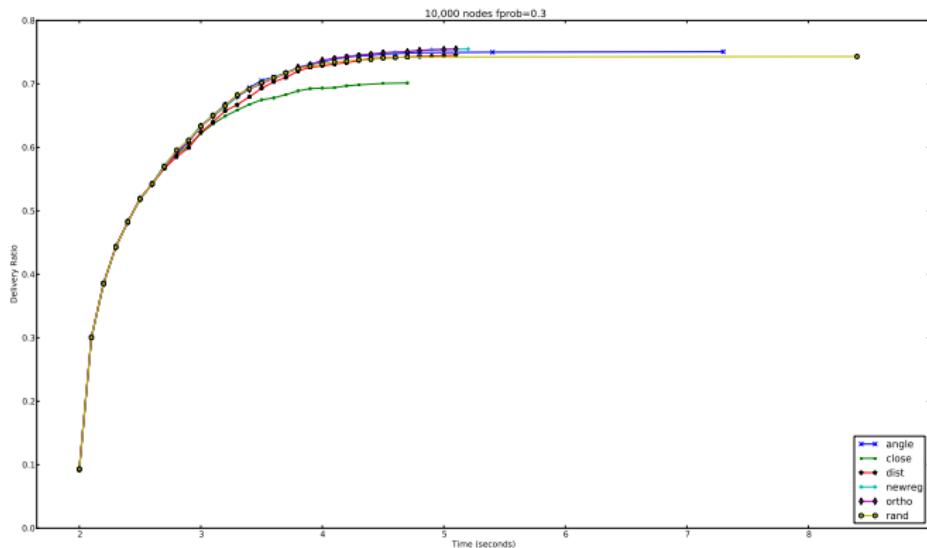
- ▶ Compared heuristics with baseline Random
- ▶ ns-3 simulation environment
- ▶ 10 seconds simulation time
  - ▶ >10s → data not as useful for early warning
  - ▶ Most damage done (earthquake over)
  - ▶ Routes start to recover
- ▶ Nodes retry every *timeout* seconds (up to 20 times)
- ▶  $\text{timeout} = 500\text{ms}$ : East → West RTT  $\approx 200\text{-}300\text{ms}$
- ▶ Nodes/links in disaster region fail uniformly with probability  $p = 0.1, 0.2, 0.3, 0.4, 0.5$
- ▶ All overlay nodes in region (one incident link) report to server during disaster
- ▶ Server chosen randomly from 10 highest-degree nodes outside disaster
- ▶ Each combination of parameters repeated 100 times with different RNG seeds
- ▶ BRITE topology generator (10,000 nodes, 50 AS'es, top-down Barabasi/Waxman model, 25 regions)

## Results



**Figure:** Delivery ratio of the various heuristics for a failure probability of 0.1. One can notice the way the heuristics sometimes overcome another's delivery ratio as time evolves and different paths are considered. NOTE: due to some necessarily aggressive data cleaning, the delivery ratio is exaggerated and skewed.

## Results



**Figure:** Delivery ratio for a failure probability of 0.3. Note that the more restrictive heuristics stop finding alternative connections before those that pick from more diverse areas, such as Random and New Angle. This pattern, as well as the relatively poor performance of Nearest Neighbor, grows proportionally to the failure probability.