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 and zhipengh@uci.edu

June 13, 2013

Challenges

- Geographically-correlated network failures
- Improve data delivery ratio/speed
- Observations
  - No control over routing infrastructure, only end hosts
  - May not have reliable knowledge of underlying network topology
    - Traceroutes may follow different paths
    - Network administrator may obfuscate links/routers
  - But can gather location information from host-sensors
    - GPS, IP address, user-specified
- Claim: Resilient Overlay Networks can help establish more reliable connections between end-hosts with limited topology knowledge

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

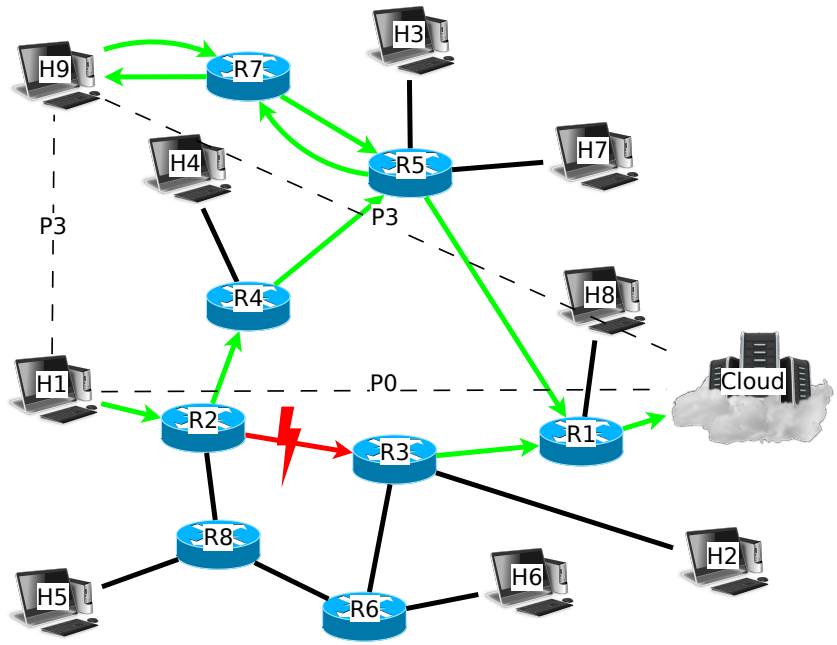

Figure: Sensors try to contact nodes outside of the local area and not along the straight-line path to the server.

Orthogonal Distant Path Heuristic

Algorithm 1:

begin

/* Monitor both the distance and the angle of the overlay node. */

$idealDist = |0.5 \cdot dist(a, b)|$

$perpDist = |\sin(angA) \cdot dist(a, c)|$

if $angA > \frac{\pi}{2}$ **or** $angA + angC < \frac{\pi}{2}$ **or** $perpDist > dist(a, b)$ **then**

$likelihood = 0$

else

$likelihood = 0.5 \cdot ((1.0 - (||perpDist| - |idealDist||/idealDist)^2) + (1.0 - (|\frac{\pi}{2} - angC|/\frac{\pi}{2})^2))$

end

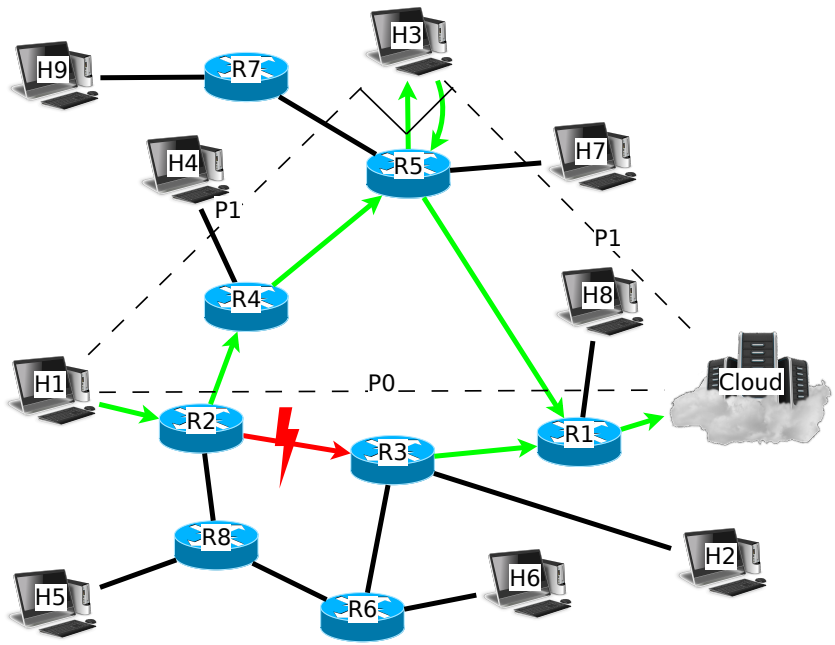

Figure: The intuition of this heuristic is to avoid the straight path, without diverging from it too much. It strikes a middle ground by choosing a path at an ideal angle of 45°, which makes the angle at the top orthogonal, hence the name.

New Region Heuristic

Algorithm 2:

begin

```
    /* Discover any node that has
    not been discoverd in this
    region                                */
```

if *RegionsAttempCount*(*region*$_c$*ount*)
then *likelihood* $= 0$
else *likelihood* $= 1.0$

end

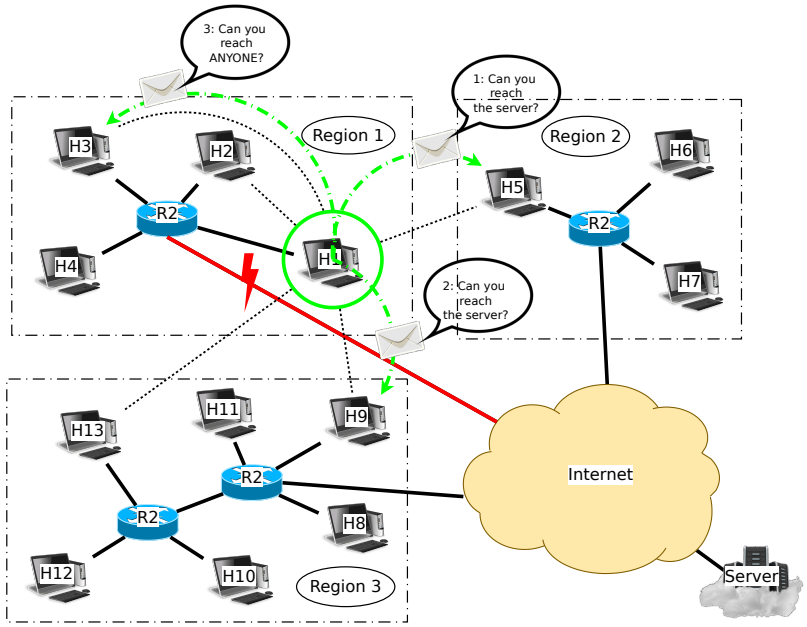

Figure: The intuition of this heuristic is to avoid regions that have been previously attempted unsuccessfully. We assume that no further attempts to contact such a region will succeed.

New Angle Path Heuristic

Algorithm 3:

begin

/* The angle could be either acute or obtuse. */

if $angle = \pi$ **or** $angle = 0$ **then**

$likelihood = 0$

else if $angle < \pi$ **then**

$likelihood = \cos(angle - \frac{\pi}{4})$

else if $angle > \pi$ **then** $likelihood = \cos(2 \cdot ((2\pi - angle) - \frac{\pi}{4})/3)$

likelihood *= pastLikelihood;

/* aggregate likelihoods to make punishments on wrong choices */

end

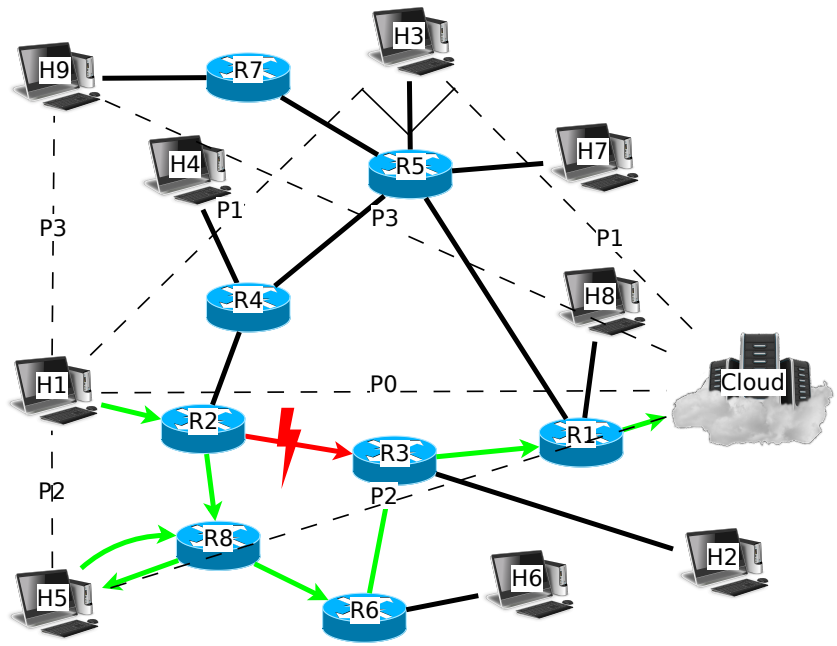

Figure: This heuristic attempts paths along new angles different from the ones previously attempted.

Distance-Dependent Path Heuristic

Algorithm 4:

begin

```
    /* Define idealDist as half of
    the distance from the source to
    the destination, dist as the
    distance from the source to the
    candidate node, minDist as a
    lower boundary to avoid
    candidate node being too close
    to the source.                  */
```

if $dist <= minDiste$ **then**

$likelihood = 0$

else if $minDist < dist < idealDist$

then $likelihood = dist^2 / idealDist^2$

else if $dist >= idealDist$ **then**

$likelihood = idealDist / dist$

end

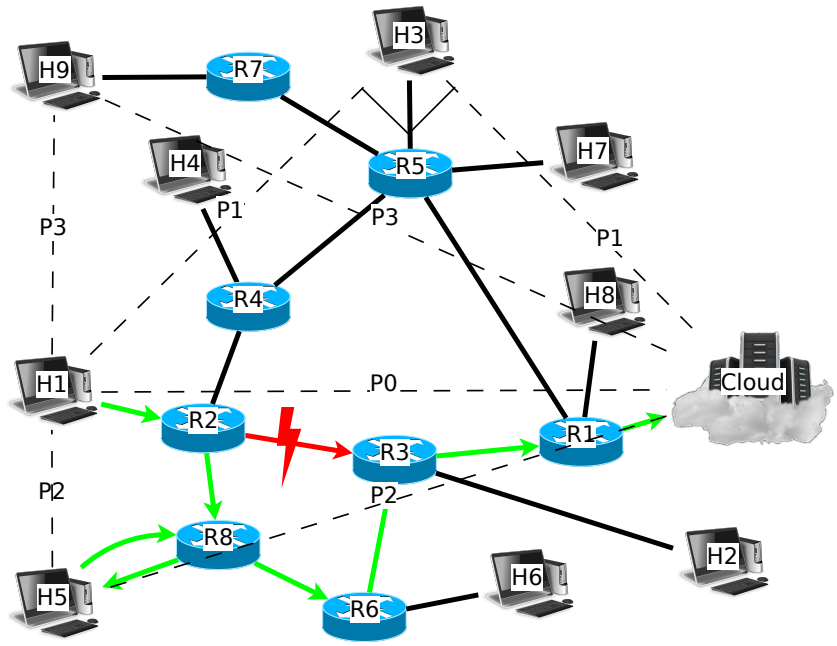

Figure: This heuristic attempts paths that use overlay nodes at an ideal distance from the sensor. This ideal distance is chosen as a radius that reaches half-way to the server.

Closest First Path Heuristic

Algorithm 5:

begin

```
    /* Set maxDistance to a
    considerably large number      */
```

if $dist(a, b) > maxDistance$ **then**
$likelihood = 0$
else $likelihood = (maxDistance - dist(a, b))/maxDistance$

end

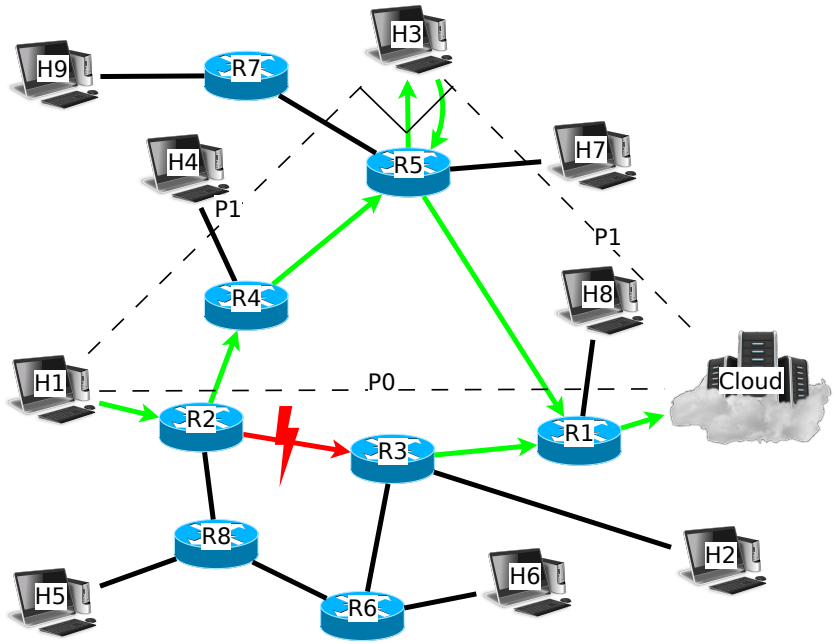

Figure: This heuristic select overlay node that has the closest distance to the source.

Furthest First Path Heuristic

Algorithm 6:

begin

```
    /* Set minDistance to a
    considerably small number      */
```

if $dist(a, b) < minDistance$ **then**

$likelihood = 0$

else $likelihood =$

$(dist(a, b) - minDistance)/dist(a, b)$

end

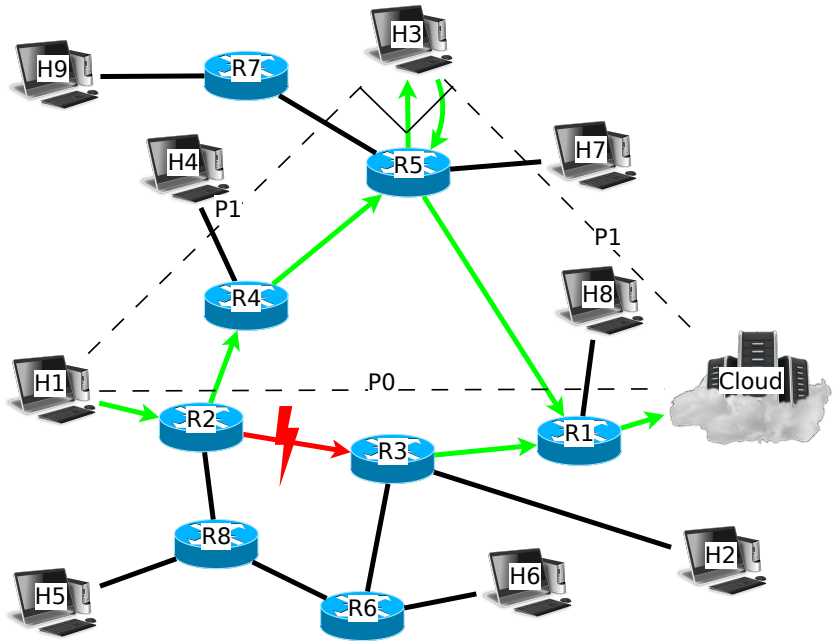

Figure: This heuristic select overlay node that has the furthest distance to the source.

Regions/AS'es Evaluated

Table:

Disaster Region							
Total Nodes							
Total Overlay Nodes							
Nodes in Region							
Overlay Nodes in Region							