

# S<sub>Co</sub>LE: Scalable Cooperative Latency Estimation

Michal Szymaniak

Guillaume Pierre

Maarten van Steen

Vrije Universiteit Amsterdam  
{michal,gpierre,steen}@cs.vu.nl

# Problem

- Context:
  - Large wide-area network
    - e.g.: Internet
  - Distributed system with  $M$  nodes
    - $M$  is **very large**, say  $O(\text{million})$
    - e.g., peer-to-peer file-sharing platform
- How to estimate latencies between arbitrary nodes?
  - Quite easy, as long as  $M$  is very small..
  - ..but much harder, once  $M$  becomes large

# Solution: Network Positioning

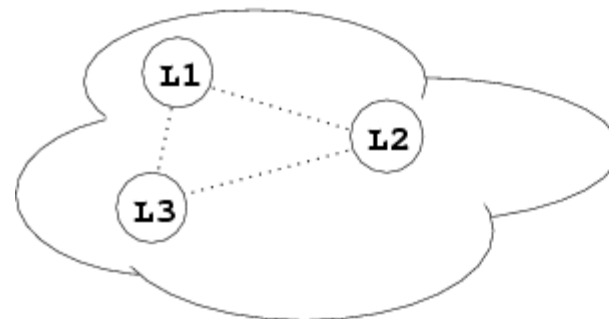
- GNP -- Global Network Positioning
  - by T.S. Eugene Ng and Hui Zhang (CMU)
  - Model the Internet as N-dimensional geometric space
  - For each node H, calculate its position  $P(H)$  in the space
  - **For any 2 nodes A and B:**
    - $\text{latency}(A,B) \sim \text{distance}(P(A),P(B))$
    - $\sim ==$  estimate with
- Main benefit:
  - In a system with M nodes, GNP reduces the number of necessary measurements:
    - all-to-all :  $O(M^2)$
    - GNP :  $O(M)$

# Talk Agenda

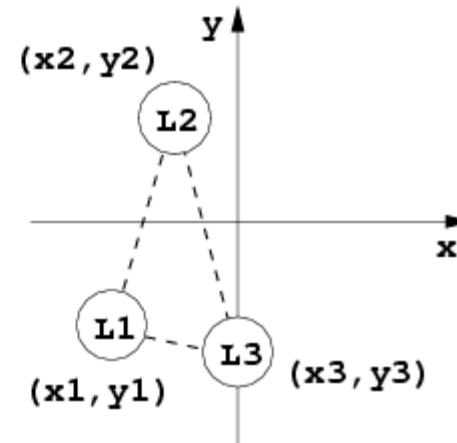
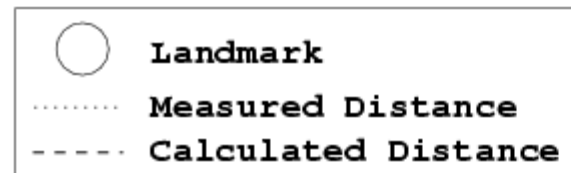
- GNP
  - Details
  - Performance
  - Limitations
- SCoLE
  - Personalized GNP
  - Architecture
  - Deployment
- Proposed Project
- Conclusion

# GNP: Space Construction

- N-dimensional space defined by N+1 reference nodes:
  - Select N+1 reference nodes, called landmarks:  $L_i$ ,  $1 \leq i \leq N+1$
  - Measure the latency between each pair of landmarks
  - Assign landmark positions  $P(L_i)$  such that:
    - For any  $i, j$ :  $\text{distance}(P(L_i), P(L_j)) \sim \text{latency}(L_i, L_j)$
    - In practice: minimize the global distance-vs-latency error



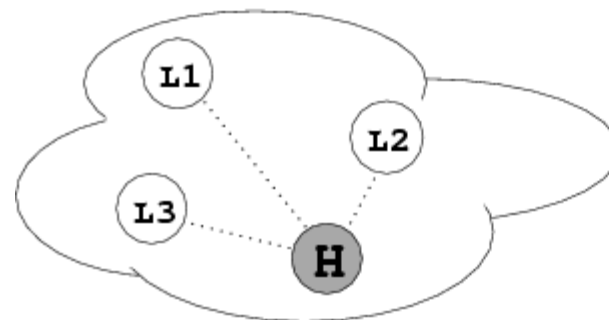
The Internet



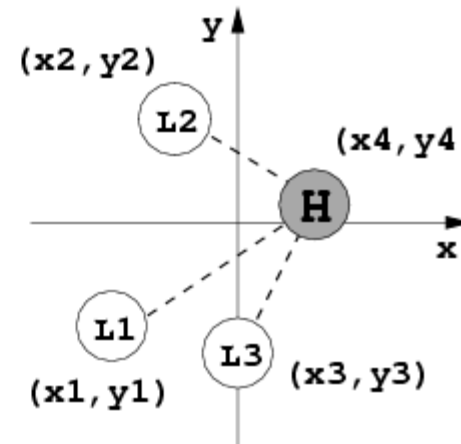
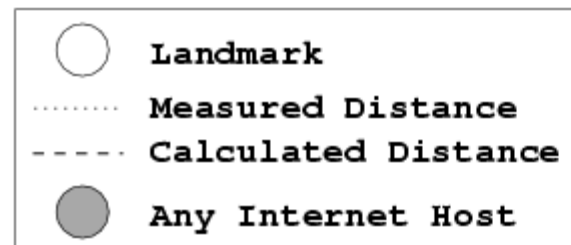
2-Dimensional Euclidean Space

# GNP: Node Positioning

- Node H positioning:
  - Measure the latencies between H and each landmark  $L_i$
  - Assign  $P(H)$  such that:
    - For any  $i$ :  $\text{distance}(P(H), P(L_i)) \sim \text{latency}(H, L_i)$
    - Again, apply global error minimization



The Internet

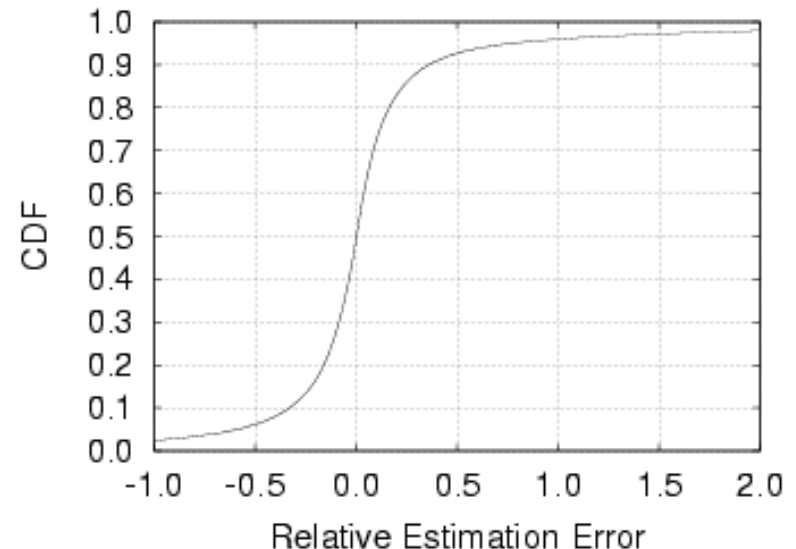


2-Dimensional Euclidean Space

# GNP: Cost/Performance

- Cost:
  - N is a small **constant**, we use N=6
  - In terms of measurements performed:
    - Space construction : **O(1)** (21 for N=6, clique of 7 landmarks)
    - Single node positioning : **O(1)** (7 for N=6, 1 per landmark)
    - Total for M nodes : **O(M)** (21 + 7 \* M)
    - Single latency estimation : **0** (once the positioning is done)

- Performance:
  - For **90%** of latency estimations:  
 **$\frac{2}{3}$  real < estimated <  $\frac{3}{2}$  real**

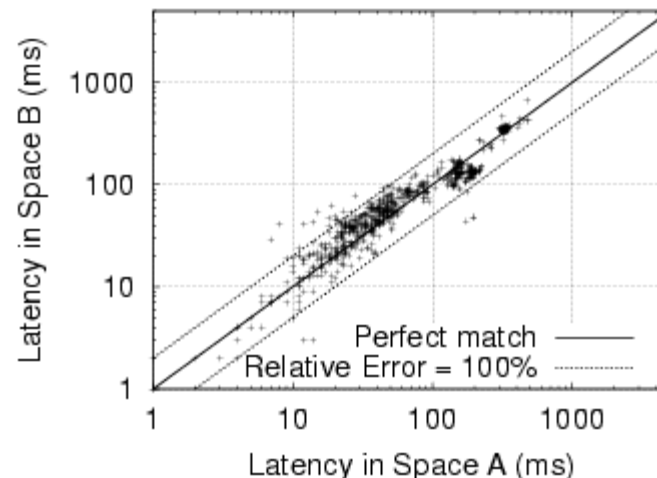


# GNP: Limitations

- GNP uses global landmarks
  - All the nodes must agree on which landmarks they use
    - Global negotiation + global knowledge = limited scalability
  - The same landmarks seldom suit all the nodes
    - Lack of flexibility
- Both problems can be removed..
  - ..if only we let nodes choose their landmarks.
  - But how can we calculate global positions then?
  - We can't. But we do not need them, either.
  - Hint: we only care about latencies.

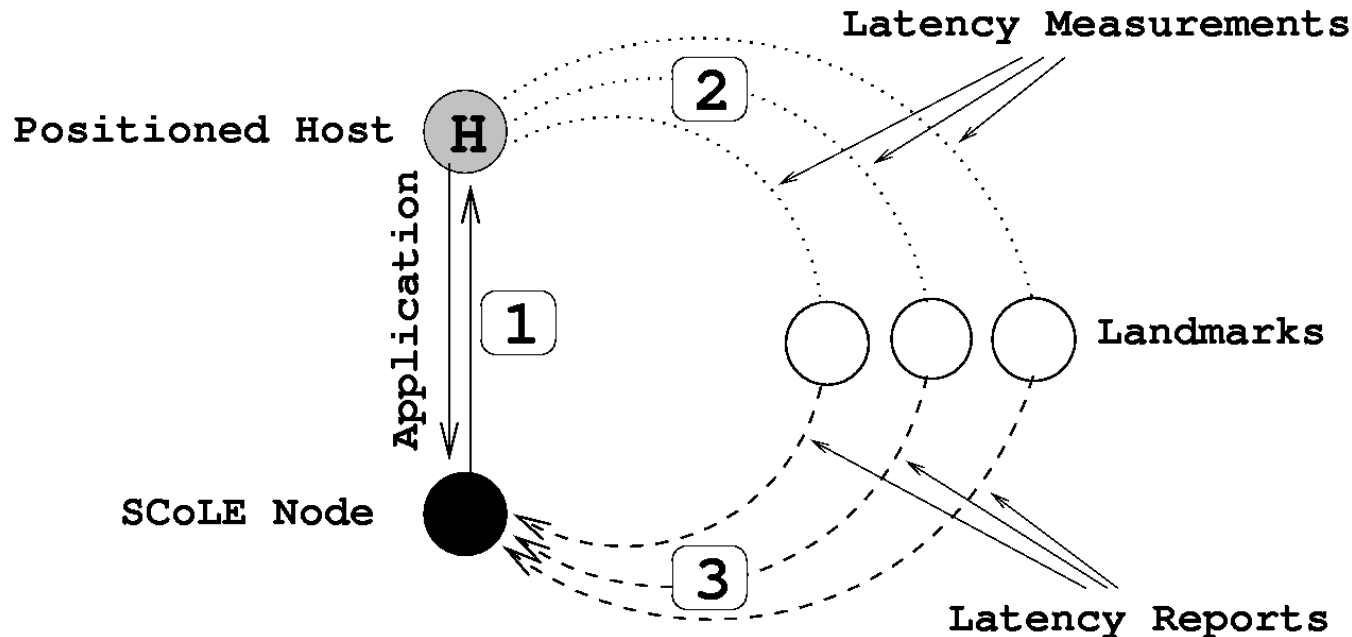
# SCoLE: Personalized GNP

- SCoLE – Every node runs its own GNP
  - select your landmarks, position any nodes you want
- Properties:
  - no global negotiation nor knowledge
  - estimation adjustable on a per-node basis
  - positions calculated by different nodes may be different
  - but: latency estimates globally correlated



# S-CoLE: Architecture

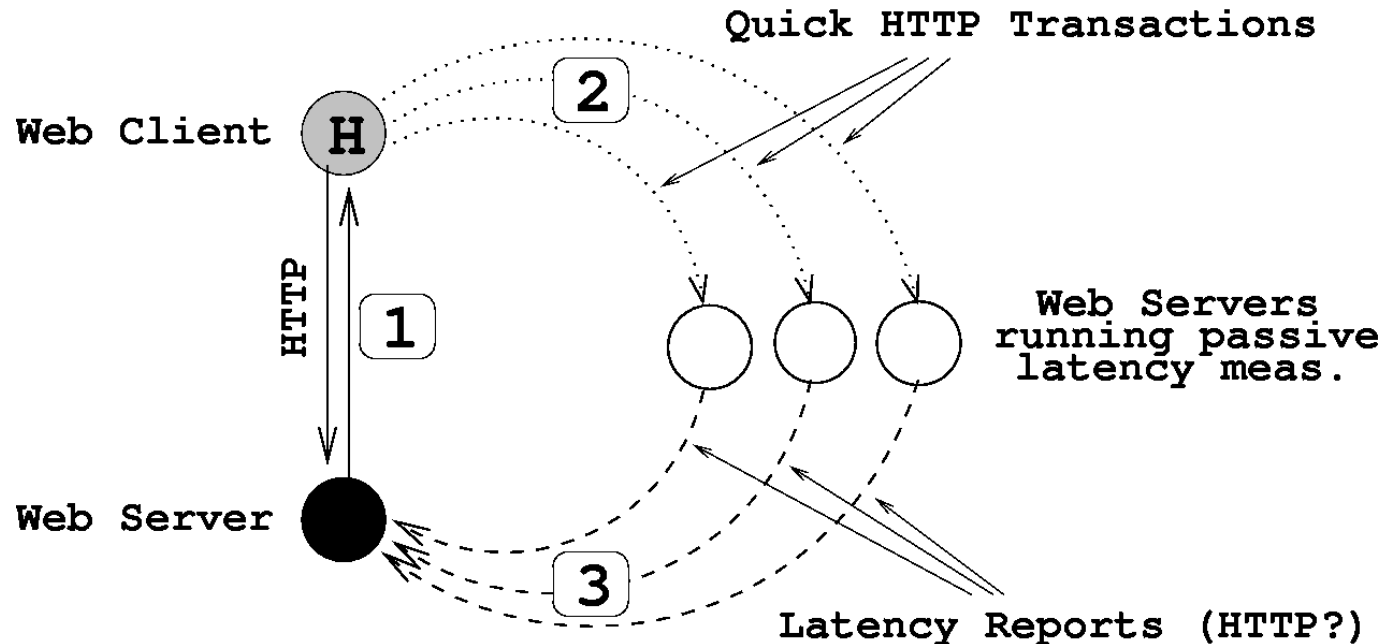
- S-CoLE Instance:



- Watch out:
  - landmarks must be distributed (important for estimation accuracy)
  - landmarks measure latencies to each other (space construction)

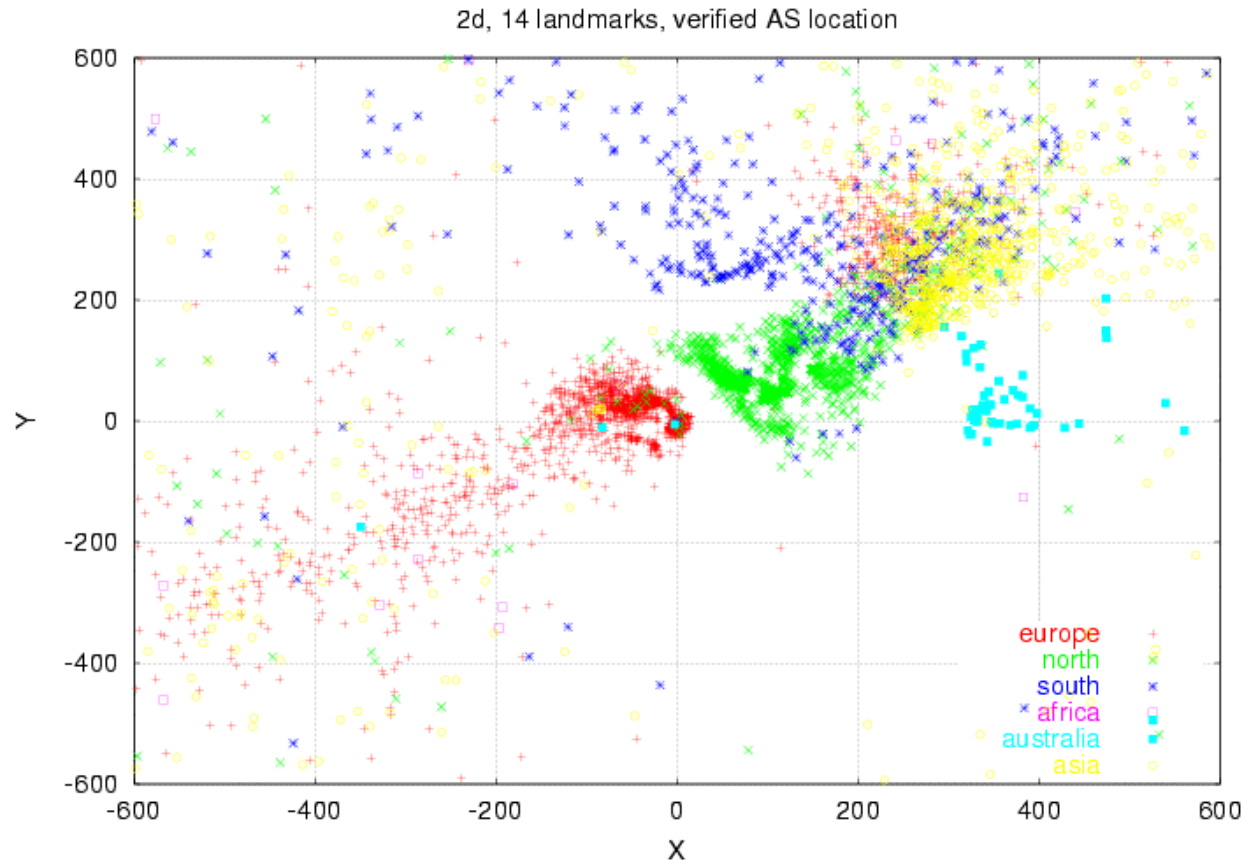
# S CoLE: Deployment

- Example system:
  - CDN supporting latency-based redirection of Web clients



# S CoLE: Prototype

- Deployed on the VU Website / PlanetLab nodes
- Clients positioned in 2D space:



# Proposed Project

- Turn TTM Test-Boxes into landmarks
  - They measure latencies among each other anyway
  - They can also measure and report other latencies upon request
  - They are physically distributed
- Why?
  - So RIPE clients could run their own SCoLE instances (using any subset of Test-Boxes as landmarks)
- Benefits?
  - Latency estimation between any pair of Internet hosts
  - Useful for: client redirection, replica placement, etc.

# Conclusion

- Network positioning:
  - Allows for scalable latency estimation
  - Is cheap in terms of number of measurements
  - Offers reasonable accuracy
- Can be personalized:
  - Each node runs its own GNP instance
  - Each instance can be adjusted to the owner's needs
  - Latency estimates are consistent across instances
- Should RIPE support such a service?

Questions?