

Review of "Meridian: A Lightweight Network Location Service without Virtual Coordinates"

Nicolas Bonvin

December 3, 2007

Summary The authors introduce in this paper a lightweight, precise and scalable framework that tries to select nodes based on their network location using direct measurements instead of a network embedding.

Top 3 contributions Here are some important contributions described in this paper:

- *Presentation of a lightweight, accurate and scalable framework* which is able to:
 - discover the closest node to a targeted reference point,
 - find a node that offers minimal latencies to a given set of nodes,
 - find a set of nodes in a region whose boundaries are defined by latency constraints;
- *Theoretical analysis*: they argue analytically that their system scales well:
 - they present a rigorous definition capturing the quality of the ring sets and show that small ring cardinalities are enough to ensure good quality,
 - they show that the nearest-neighbor queries return exact or near-exact neighbors in a logarithmic number of steps,

- they argue that the system is load-balanced if the ring sets of different nodes are stochastically independent;

- *Empirical results*: the authors evaluated the framework on a large scale simulator and on PlanetLab. The results provided in this paper confirm the theoretical analysis.

Most glaring problems Some issues need to be mentioned:

- They use a certain amount of *magic numbers*, which is not a desirable property especially in dynamic environments,
- They only test the framework in a simulator and in PlanetLab which does not reflect the real behavior of Internet (latencies, NAT, firewall, asymmetric bandwidth, ...).

Implications for distributed systems

Meridian is a very useful framework for a lot of different applications like P2P overlays or content distribution networks. One interesting aspect of Meridian is that the target node doesn't need to be part of the Meridian overlay, which is not the case in traditional structured P2P networks.