A Quicker Way to Discover Nearby Peers

Gwendal SIMON
Institut TELECOM / TELECOM Bretagne, France
gwendal.simon@telecom-bretagne.eu

Yiping CHEN, Ali BOUDANI, Gilles STRAUB
Thomson R&D France
firstname.lastname@thomson.net

***Abstract*— The match between a peer-to-peer overlay and the physical Internet infrastructure is a constant issue. Time-constrained peer-to-peer applications such as live streaming systems are even more challenging because participating peers have to discover their closest neighbors as quickly as possible. We propose in this paper an approach based on landmarks and a management server in order to discover, as quickly as possible, its closest neighbors among a large population of peers.**

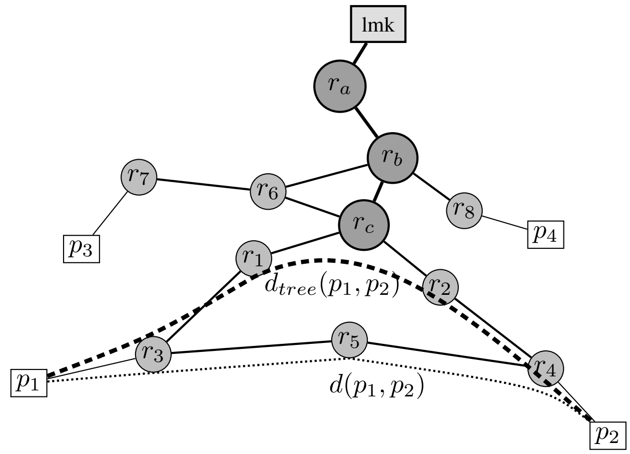
I. Context

Designers of peer-to-peer systems are usually concerned by the match between logical overlay and physical infrastructure: a peer should preferentially be connected with the peers that are the closest in Internet [10]. However determining its closest neighbors in a wide population spread on a large-scale network as Internet is still a challenge. Among the most promising ideas, coordinate-based schemes aim to fix the location of any host on Internet in a Euclidean space, so to allow two peers to estimate their latency by a basic distance computation [7], [3]. Previous works show that these virtual coordinates can be obtained by active probing, *i.e.* by collecting round-trip-time (RTT) measurements between peers and a small set of *landmarks* [5].

Unfortunately, network coordinate systems require a substantial amount of time before to deliver accurate information. This time is actually an issue. Consider live streaming peer-to-peer systems [11]: when a newcomer joins, it first experiences a *setup delay* before the video becomes actually visible. During this time, the peer has to set up many parameters including the playback delay. In typical mesh-based applications [9], the playback delay of a peer should ideally be the same than the ones of its neighbors because chunk exchanges are easier to manage when neighbors focus simultaneously on the same set of chunks. As the setup delay should be as short as possible and as the proximity between peer is crucial in such real-time applications, we are interested with a system that allows a newcomer to discover *as quickly as possible* its closest peers.

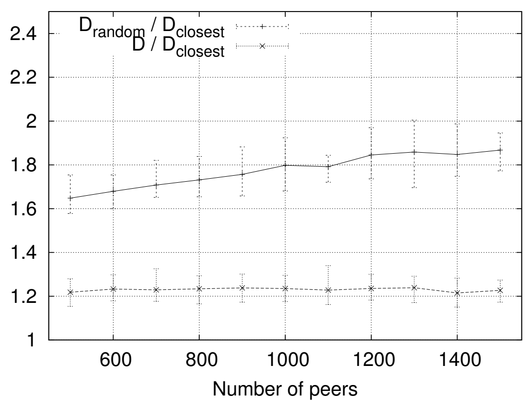
II. Proposal

We envision a mechanism based on a `traceroute`-like tool [2]. We make the following assumption: if a server can store the exact routes between each peer and a landmark, it would be able to estimate accurately the closest peers of any newcomer. The reasoning refers to the statistical regularities observed in the large-scale structure of Internet [8]. The apparent heavy-tailed degree distribution in the graph of routers makes most of the routes from a same point passe through



the network core. This characteristic refers to the notion of centrality [1]. We imply that (i) the shortest path between most pairs of network edges use the network core and (ii) the path between a peer and the landmark reveals the quickest way to reach the network core. We describe a typical situation in a drawing. The routers r_a, r_b and r_c are within the network core and have a large number of connections with other routers (not represented in the figure). On the contrary, the routers r_i with $i \in \{1 \dots 9\}$ are small routers with a small degree. If the landmark lmk knows both routes from p_1 and p_2 to itself, it would be able to deduce the path $d_{tree}(p_1, p_2)$ by noting that r_c is the first common router in the network core. In this example, this inferred path is not the shortest path between p_1 and p_2. But we expect that most cases verify $d(p_1, p_2) = d_{tree}(p_1, p_2)$.

We suggest a two-round approach using a *management server* and few *landmarks*: (i) a newcomer infers the path between itself and its closest landmark (in terms of latency) and (ii) the management server estimates the closest peers to this newcomer from the analysis of this path. The first round begins once a newcomer p joins the peer-to-peer system. The peer p makes use of a `traceroute`-like tool to discover the set of routers in the path between itself and one landmark. This path is then transmitted to the management server. This latter is responsible of the second round. It knows all peers and we assume moreover that it also knows the path between each peer and its respective landmark. It should answer to p a short list of peers that are the closest to p. We are currently working on an algorithm such that the complexity of a newcomer insertion is $\mathcal{O}(\log n)$ – the cost of inserting a new element in an ordered



list of elements – while the result is obtained with a complexity $\mathcal{O}(1)$ – accessing a data in a hash table.

III. Ongoing Works

We face a usual challenge of simulating a worldwide application for end-users: the simulator should conform to a model of the whole Internet, although this mapping is known to be a real issue [4]. Moreover, we are interested here in latency estimation that can hardly be done on a Autonomous System (AS) layer, but rather on a Internet Router (IR) layer. So we rely on a IR map obtained from a specific Internet mapper [6] integrated into the `PeerSim` simulator.

First we initialize an overlay by attaching n peers to routers with degree equals to one in the simulated network and few landmarks to routers with medium-size degree. Each peer receives from the management server a set of neighbors. We compute the sum D of the hop-distances between the peer and these neighbors, then we compare it to the best set of neighbors obtained by a brute-force algorithm. We display here one first result: when the number of peers increases, the quality of the algorithm is stable in comparison to the optimal choice and unsurprisingly outperforms the basic approach with a newcomer randomly choosing its neighbors.

Among the future works, we are interested with testing this approach on existing simulation platforms, even if they do not exactly emulate typical end-users. The mobility will require specific algorithms, managing both faulty peers and handover. Extensive simulations will also let us study various policies for the management of landmarks, including the number and their placement in the network. In the same manner, we are investigating the opportunity to use some super-peers. We also have to specify the accurate `traceroute`-like tool we require. This tool could be a decreased version of the original one because we are only interested with some routers along the path. Another major wish is related to a formal proof based on a graph-oriented analysis.

References

[1] M. Barthlemy. Betweenness centrality in large complex networks. *European Physical Journal B*, 38:163–168, 2004.

[2] S. Branigan, H. Burch, B. Cheswick, and F. Wojcik. What can you do with traceroute? *IEEE Internet Computing*, 5(5):96, 2001.

[3] F. Dabek, R. Cox, M. F. Kaashoek, and R. Morris. Vivaldi: a decentralized network coordinate system. In *SIGCOMM*, 2004.

[4] L. Dall'Asta, J. I. Alvarez-Hamelin, A. Barrat, A. Vázquez, and A. Vespignani. Exploring networks with traceroute-like probes. *Theoretical Computer Sciences*, 355(1):6–24, 2006.

[5] J. M. Kleinberg, A. Slivkins, and T. Wexler. Triangulation and embedding using small sets of beacons. In *FOCS*, 2004.

[6] D. Magoni and M. Hoerdt. Internet core topology mapping and analysis. *Computer Communications*, 28(5):494–506, 2005.

[7] T. S. E. Ng and H. Zhang. Predicting internet network distance with coordinates-based approaches. In *INFOCOM*, 2002.

[8] R. Pastor-Satorras and A. Vespignani. *Evolution and Structure of the Internet*. Cambridge University Press, 2004.

[9] F. Pianese, J. Keller, and E. W. Biersack. PULSE, a flexible P2P live streaming system. In *Global Internet Symposium*, 2006.

[10] S. Ratnasamy, M. Handley, R. M. Karp, and S. Shenker. Topologically-aware overlay construction and server selection. In *Infocom* 2002.

[11] A. Sentinelli, G. Marfia, M. Gerla, L. Kleinrock, and S. Tewari. Will IPTV ride the peer-to-peer stream. *IEEE Communications Mag.*, June 2007.