Point-by-point response to reviewers and editors

The full diff can be found here (it may reveal the authors):

`https://github.com/fsantanna-no/p2p-tml-paper/compare/33080a93a5ae42144ca08eb5ed3371b9a293c061..fa21b9477620cdfe6dffc52430ad50d613a01d52`

Comment 1

> Please enhance the description and comparison with other current schemes so that to show your advantages. In especially, try to do the data/performance analysis and comparison, which is important to prove the value of your proposal.

We added a half-page discussion at the end of the Related Work section comparing the two centralized schemes with our proposal. We also split the section in two to improve its structure. Even though the centralized schemes have better performance, we show competitive numbers for similar scenarios. Follows the added text:

Many aspects may affect the performance of symmetric distributed applications, such as the network latency, the number of nodes in the network, the rate of user events, and the application frame rate. It is expected that centralized solutions exhibit a better performance in comparison to peer-to-peer applications, since nodes are only 2 hops aways from each other. The performance of Croquet depends exclusively on the network latency, which restricts the application to a maximum FPS. For instance, since all frame ticks need to be transmitted over the network, a latency of 50ms supports at most 20 FPS. Also, the higher is the FPS, the higher is the network traffic and server load, which may also degrade performance. GALS detaches latency from FPS, since each node has its own timeline. However, the protocol requires two roundtrips to the server, which degrades performance considerably. The authors claim to achieve a responsiveness of 350ms in applications with 25 nodes at 25 FPS, where 350ms represents the time it takes for all nodes to apply an event since its initial occurrence. As we discuss in Section 4, the performance of our midlleware is measured in terms of rollbacks, i.e., the more the peers need to resynchronize and travel back in time, the lower is the performance of the application. In a similar scenario with 21 peers at 50 FPS, and average of 5 hops with 50ms latency, we show 3.1% of rollbacks with responsiveness of 500ms, which we classify as moderate performance (Figure 7, framed rectangle in the center). As expected, the peer-to-peer performance is lower in comparison to centralized solutions, but still viable in many scenarios, as detalied in Section 4. Regarding CRDTs, recall that they only provide eventual consistency with no notion of a shared timeline in which peers go through identical steps. Therefore, measuring the performance of event responsiveness, maximum FPS, or clock synchronization is meaningless, given that events can be applied locally and broadcasted using a best-effort policy.