Paper Review: Autobahn:Seamless high speed BFT

1 Introduction

The paper presents Autobahn, a Byzantine Fault Tolerant state machine replication protocol, which has been designed to meet the performance and robustness challenges in a partially synchronous environment. Indeed, by decoupling data dissemination from consensus, Autobahn enables continuous transaction broadcasting when the network splits. The protocol is organized in a two-layer architecture: a data dissemination layer, comprising multiple parallel lanes where each replica independently proposes transactions, and a consensus layer which agrees on snapshots of the data layer(cuts). This decoupling allows Autobahn to maintain high throughput during stable periods while minimizing performance degradation (hangovers) during and after network disruptions (blips).

2 Strong Points

- **Seamless recovery from network disruptions**: Autobahn is designed to minimize performance degradation after experiencing network issues. The data dissemination layer continues working and, hence the whole data backlog can instantly be committed as soon as the network recovers.
- **High throughput**: The parallel lane structure allows each replica to propose transactions independently, thus allowing horizontal scalability.Compared to the state-of-the-art DAG-based BFT protocols, Autobahn achieves equal throughput with much lower latency.
- **Low latency during stable periods**: The consensus layer of Autobahn comes with a fast path mechanism that could bring down the latency during good network conditions to that of a single round-trip, thus improving the latency of traditional BFT protocols.

3 Weaknesses

- Autobahn makes use of uncertified DAGs that lack robustness, as they omit certification, which can lead to unwanted synchronization delays and

susceptibility to timeout violations.

- While the design of Autobahn tries to mitigate most of the negative consequences of timeouts, it still depends on them for liveness guarantees. Setting timeouts appropriately is extremely hard, especially under dynamic network conditions.

- Scalability constraints may emerge under network load, as uncertified DAGs face increased synchronization and fetch demands when scaling in highly unreliable network conditions.

4 Detailed Feedback

- **Limitation of Uncertified DAGs in Ensuring Robustness**: Uncertified DAGs assumes that all nodes would have the required data to begin with. This assumption of theirs fails when a network is unstable or when there are Byzantine faults, in which case nodes have to fetch missing data. This may take substantial delay, roughly two message delays per round, and may weaken latent benefits to be gained by employing this approach. On the other hand, protocols like Shoal++[1] emphasize that, with proper engineering, certification does not have to introduce additional latency and that robustness can be ensured by making data available at all nodes.

- **Reliance on timeouts**: Autobahn's reliance on timeouts for view changes can be problematic in highly dynamic environments where network conditions fluctuate frequently. Setting timeouts too low may lead to unnecessary view changes, while setting them too high might delay recovery from actual leader failures. This could potentially be improved by using adaptive timeouts similar to those used in Spotless[2].

- **Scalability in Real-World Scenarios**: The nodes in non-certified DAGs need to constantly fetch and reconcile non-existent data to validate proposals by removing all certifications. This usually originates with increased network load and higher communication overhead in case of the scale-up of a network or volume of transactions. This reduces the scalability advantage that this should have provided, particularly in the case of a network with high transactions or an unreliable one, as constant fetching and retransmission of data is required, which could again result in potential performance bottlenecks.

5 Conclusion

In general, this is a very promising approach for improving BFT consensus protocols on partially synchronous environments. On the other hand, given the complexity of the protocol and reliance on timeouts, due care must be considered with regard to the system.

References

[1] Balaji Arun, Zekun Li, Florian Suri-Payer, Sourav Das, and Alexander Spiegelman. 2024. Shoal++: High Throughput DAG BFT Can Be Fast! arXiv preprint arXiv:2405.20488 (2024).

[2] D. Kang, S. Rahnama, J. Hellings and M. Sadoghi, "SpotLess: Concurrent Rotational Consensus Made Practical Through Rapid View Synchronization," 2024 IEEE 40th International Conference on Data Engineering (ICDE), Utrecht, Netherlands, 2024, pp. 1916-1929, doi: 10.1109/ICDE60146.2024.00157. keywords: Costs;Prototypes;Throughput;Fabrics;Consensus protocol;Complexity theory;Synchronization,