**Fall 2023 ISEC 885 : Project Overview Report: Research on Asynchronous Consensus Protocols**

**Introduction:**

The exploratory research for the Fall 2023 ISEC 885 course aims to develop a problem direction for a doctoral research idea concept paper leading to an idea paper, with the ultimate objective of improving certain aspects within the field of asynchronous consensus, yet to be determined. Asynchronous consensus holds immense significance in distributed systems and blockchain technology, as it forms the underlying architecture for coming to an agreement and coordination amongst peers that all operate independently, all without time synchronization. Previous literature has addressed the need for further exploration in the field of asynchronous consensus as being arguably the most appropriate solutions for building high-assurance and intrusion tolerant permissioned blockchain environments, as asynchronous protocols inherently perform more robustly against timing and denial-of-service (DoS) attacks. Especially over unprotected networks such as the internet (Duan et al., 2018). The Honey Badger Byzantine Fault Tolerance (HBFT) protocol claims to be “the first practical asynchronous Byzantine Fault Tolerant (BFT) protocol, which guarantees liveness without making any timing assumptions“ (Miller et al., 2016 "Abstract," p. 1). HBFT will be chosen as a bench mark for this research as it has been done before in the field of asynchronous consensus through newer protocols such as BEAT, DUMBO, and Asynchronous Byzantine Fault Tolerance (ABFT) that will later be discussed. (Knudsen et al., 2021,Duan et al., 2018, and Guo, Lu, Tang, Xu, & Zhang, 2020)

The initial goal of the research will be to redeploy an instance of HBFT and establish a baseline to compare previous and future work to. The deployment will mimic that of the HBFT deployment of 32, 40, 48, 56, 64, and 104 Amazon EC2 t2.medium instances uniformly distributed throughout its 8 regions spanning 5 continents. The batch sizes of these transactions will mimic that of the HBFT such that each node will propose 256, 512, 1024, 2048, 4096,8192, 16384, 32768, 65536, or 131072 transactions. (Miller et al., 2016 "Experiments on Amazon EC2," p. 9). The size of each transaction will be a constant of 250 Bytes each. (Miller et al., 2016 "Bandwidth Breakdown and Evaluation." p. 9) The results of the HBFT findings should mimic the original findings that state the upper bound limits of “throughput exceeding 20,000 transactions per second for medium size networks of up to 40 nodes. For a large 104 node network, we attain more than 1,500 transactions per second.” (Miller et al., 2016 "Experiments on EC2" p. 10) Noting that the HBFT fault tolerance parameter is set to Faulty Node (F) = Nodes (N) / 4, such that 32N/8F 40N/10F 48N/12F 56N/14F 64N/16F 104N/26F was used. The formula F=N/4 was chosen instead of the greater fault tolerance of F=N/3 for ease of division. (Miller et al., 2016 "Experiments on EC2" p. 9) The high level design of the HBFT protocol will be replicated and can be summed up in the abstract of DUMBO stating “ The core of (HBFT) is to achieve batching consensus using Asynchronous Common Subset protocol (ACS) of (Ben-Or, Kelmer, & Rabin, 1994) constituted with n Reliable Broad Cast protocol (RBC) to have each node propose its input, followed by (N) Asynchronous Binary Agreement protocol (ABA) to make a decision for each proposed value (N is the total number of nodes)” (Guo, Lu, Tang, Xu, & Zhang, 2020). It is important to establish a baseline deployment of HBFT that replicates the original findings before creating newer research to test against.

Simultaneously as the baseline deployment of the HBFT matures, further research will continue off the works of the BEAT, DUMBO, and ABFT asynchronous protocols. The three of which all use the HBFT as a baseline for their research. To begin BEAT takes the works of HBFT and creates “five asynchronous BFT protocols that are designed to meet different goals (e.g., different performance metrics, different application scenarios) ” ( Duan et al., 2018 “Abstract” p 2). The BEAT instantiations act as follows when compared to HBFT: BEAT0 uses a different threshold encryption, BEAT1 uses a different erasure-coded broadcast, BEAT2 changes the HBFT logic by opportunistically moving the encryption part of the threshold encryption to the client, BEAT3 changes the HBFT primitive and becomes a BFT storage system by replacing the RBC with Bandwidth-efficient Asynchronous Verifiable Information Dispersal (AVID-FP), and BEAT4 reduces read bandwidth making it more suitable for clients who read only a fraction of stored transactions. (Duan et al., 2018 “The BEAT protocols” p. 2) BEAT was chosen for this research because it builds upon the logic of HBFT and creates different instances with different design goals. Next, DUMBO was chosen for the research because its extends the research of BEAT and HBFT by creating two protocols DUMBO1 and DUMBO2 based off the HBFT codebase and the findings from BEAT. DUMBO1 runs a small K (independent of N) instances of ABA while DUMBO2 reduces it further down to a constant. The premise behind DUMBO being that “(1) reducing the number of ABA instances significantly improves efficiency; and (2) using multi-valued validated Byzantine agreement (MVBA) which was considered sub-optimal for ACS in HBFT in a more careful way could actually lead to a much more efficient ACS” (Duan et al., 2018 “Abstract” p. 1). DUMBO was chosen for this research because it extends the research of BEAT and HBFT. Lastly, the ABFT protocol was chosen for this research because it is an amalgamation of the works of HBFT, BEAT, and DUMBO (Knudsen, Li, Notland, Haro, & Ræder, 2021 "ABFT Design and Implementation on EC2" p. 9). The basis of the ABFT logic when compared to HBFT is that it “integrates threshold Elliptic Curve Digital Signature Algorithm (ECDSA) signatures and optimization of erasure coding parameters, as well as additional, implementation-level optimizations” (Knudsen et al., 2021, "Abstract," p. 1). Because ABFT combines the works of HBFT, BEAT, and DUMBO, it was also chosen for the research.

Objectives:

*Problem Identification:* Identify and define specific problems within the field of asynchronous consensus through ABFT based on the existing literature.

*Concrete Problem Direction:* Establish a concrete problem direction and established a need that will serve as the foundation for a doctoral research idea concept paper.

*Goal Establishment:* Establish a clear and concise goal for further research, with a focus on improving some aspect of asynchronous consensus through ABFT.

*Literature Review:* Conduct a review of literature, including Knudsen et al. (2021), and the work cited to understand the current state of ABFT and its existing challenges and limitations.

*Experimental Validation:* Explore different experimental setups and network environments to validate the protocol's performance under various conditions and deployments, addressing concerns about external validity.

*Performance Analysis:* Investigate performance-related issues within ABFT, including the Threshold ECDSA Scheme, experimental setup, scalability, and handling network degradation.

*Quantitative Assessment:* Quantify the performance benefits of potential solutions, such as threshold ECDSA signatures, in comparison to previous threshold BLS signatures for efficiency.

*Resource Optimization:* Research methods to reduce computational overhead and explore efficient methods for delegating precomputing to other machines in the ABFT protocol.

*Scalability Improvement:* Develop strategies to make ABFT more scalable, particularly for very large networks or those with a high churn rate.

*Network Degradation Mitigation:* Investigate adaptive strategies to mitigate performance degradation when the number of affected nodes exceeds the fault tolerance threshold.

Current Level of Completion: At this stage, the project is in its early exploratory phase. Literature review and problem identification have been initiated, but specific goals, milestones, and deliverable are yet to be fully defined. A basic implementation of the HoneyBadgerBFT protocol still needs to be deployed.

Milestones and Deliverables: The project will progress through the following milestones and deliverables:

* Problem identification and definition
* Concrete problem direction, need, and goal
* Literature review completion
* Experimental setup exploration and validation
* Quantitative assessment and performance analysis
* Threshold ECDSA versus BLS Signatures trade off strategies
* Scalability improvement strategies
* Network degradation mitigation strategies

Current Accomplishments:

* Initiated literature review and identified key problems within ABFT.
* Began the process of defining performance-related challenges and researching potential solutions.
* Cloned the HoneyBadger Rust code base. <https://github.com/poanetwork/hbbft>
* Started communication between some of the HBFT contributors.

Scheduled Completions:

* Literature review completion by [Q4 2023]
* Finalization of concrete problem direction, need, and goal by [Q4 2023]
* Experimental setup exploration and validation by [Q2 2024]
* Quantitative assessment and performance analysis by [Q4 2024]
* Threshold ECDSA versus BLS Signatures trade off strategies by [Q2 2025]
* Scalability improvement strategies by [Q4 2025]
* Network degradation mitigation strategies by [Q4 2025]

Missed Targets: There are currently no missed targets as the project is still in its early stages.

Issues and Changes:

Still need to deploy the HoneybadgerBFT as a baseline.

Open Issues:

Still need to deploy the HoneybadgerBFT as a baseline.

Open Change Requests: None.

Next Phase Schedule (Start and Completion Targets): The next phase the “Literature review completion” will involve the detailed investigation into the literature of ABFT and related work. Extra time could be sent setting up the HBFT baseline network. The schedule for the literature review completion phase is as follows:

* Start Date: [September 2023]
* Completion Target: [December 2023]

Summary: The goal of the exploratory research, conducted as part of the Fall 2023 ISEC 885 course, is to establish a problem direction for a doctoral research idea concept paper, which will lead to a research idea paper. The goal is centered around the improvement of the field of asynchronous byzantine fault tolerance through the ABFT protocol. The final outcome will be a concrete problem, need, and goal. The technical goals of the research is to implement the HBFT protocol to create a baseline. This baseline will serve as a reference point for comparing both the original work of HBFT and any future research endeavors. After creating a base line the next technical goal will be replicating and contributing to ABFT such that previous results can be compared to and new research proposed.

## **References**

Abraham, I., Malkhi, D., & Spiegelman, A. (2019). Asymptotically optimal validated asynchronous byzantine agreement. In Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (pp. 337–346). ACM. <https://doi.org/10.1145/3293611.3331612>

Ben-Or, M., Kelmer, B., & Rabin, T. (1994). "Asynchronous secure computations with optimal resilience." In Proceedings of the thirteenth annual ACM symposium on Principles of distributed computing (pp. 183–192). ACM.

Duan, S., Reiter, M., & Zhang, H. (2018). BEAT: Asynchronous BFT Made Practical. *In ACM SIGSAC Conference on Computer and Communications Security,* pp. 2028–2041. <https://doi.org/10.1145/3243734.3243812>

Guo, B., Lu, Z., Tang, Q., Xu, J., & Zhang, Z. (2020). Dumbo: Faster Asynchronous BFT Protocols. Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20), 803–818. <https://doi.org/10.1145/3372297.3417262>

Knudsen, H., Li, J., Notland, J., Haro, P., & Ræder, T. (2021). High-Performance Asynchronous Byzantine Fault Tolerance Consensus Protocol. *IEEE International Conference on Blockchain*, pp. 476-483. <https://doi.org/10.1109/Blockchain53845.2021.00073>

Miller, A., Xia, Y., Croman, K., Shi, E., & Song, D. (2016). The Honey Badger of BFT Protocols. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 31–42. <https://doi.org/10.1145/2976749.2978399>