BRLA-P2P: A Byzantine-Resistant Learning Automata Framework for P2P Content Distribution

Research Update Report

Ekaba Bisong
University of Victoria
ebisong@siliconblast.com

March 24, 2025

This research update presents significant progress on our Byzantine-Resistant Learning Automata framework for P2P content distribution (BRLA-P2P). We detail the finalized architectural design with well-defined interfaces between the Learning Automata module, Byzantine detection engine, partition management system, and content distribution handler. A functional prototype has been implemented that integrates these components, which will enable comprehensive evaluation of the framework's performance. This report outlines our planned evaluation methodology for assessing Byzantine detection performance, content distribution efficiency, scalability characteristics, and comparative advantages over existing approaches. We also describe our parameter optimization strategy to identify optimal configurations for the framework's learning rate, Byzantine penalty, and partition count. The evaluation plan we present is designed to rigorously assess the BRLA-P2P framework's performance across various Byzantine ratios and network conditions, providing insights into the framework's robustness and applicability in real-world peer-to-peer environments.

Contents

1. Framework Design Refinement

The BRLA-P2P framework integrates Learning Automata (LA) techniques with Byzantine detection and dynamic partition management to provide robust content distribution in peer-to-peer networks. Through iterative design refinement, we have finalized a comprehensive architecture that addresses the key challenges of Byzantine resistance while maintaining high performance and scalability.

1.1. Architectural Design Overview

The refined BRLA-P2P framework consists of four primary components, each with distinct responsibilities but designed to work synergistically:

Learning Automata Module. This component implements enhanced Transitivity Pursuit Object Migration Automata (TPEOMA) with Byzantine resistance capabilities. It maintains and updates state vectors for each peer based on interaction outcomes and applies specialized penalties when Byzantine behavior is detected. The module continuously evolves action probabilities to favor reliable peers and paths while avoiding those exhibiting suspicious behavior.

Byzantine Detection Engine. This module combines reputation scoring with behavioral pattern analysis for robust malicious node detection. It maintains interaction histories, calculates weighted reputation scores, analyzes behavioral patterns for anomalies, and implements confidence-based classification to minimize false positives. The engine generates detailed detection reports that inform both learning and partition management processes.

Partition Management System. Implementing Partition Size Required Object Migration Automaton (PSR-OMA), this component manages dynamic partitioning with Byzantine awareness. It maintains partition size constraints, isolates detected Byzantine

nodes into a dedicated partition, manages peer migrations between partitions, validates partition constraints, and provides a reintegration mechanism for formerly Byzantine peers that demonstrate improved behavior.

Content Distribution Handler. This component manages content routing, replication, and availability with Byzantine-aware path selection. It implements graph-based routing algorithms that dynamically update based on Byzantine detection and partition structure, maintains strategic content replication for availability, implements popularity-based caching, and collects detailed distribution performance metrics.

Figure 1 illustrates the high-level architecture of the BRLA-P2P framework, showing the key components and their interfaces.

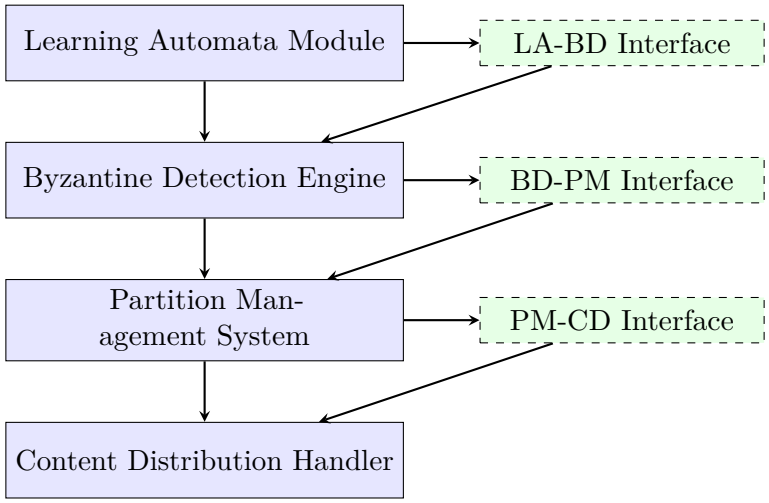

Figure 1: BRLA-P2P Framework Architecture

1.2. Component Interface Specifications

A key aspect of our design refinement has been the formalization of interfaces between components. These well-defined interfaces ensure clean separation of concerns, enable independent testing and maintenance, and facilitate system evolution.

Learning Automata to Byzantine Detection (LA-BD) Interface. The LA-BD interface enables bidirectional information flow between the Learning Automata module and the Byzantine Detection Engine. It allows the Learning Automata to provide peer state vectors and interaction histories to the Byzantine Detection Engine, which uses this information for analysis. Conversely, it enables the Byzantine Detection Engine to signal detected Byzantine behavior back to the Learning Automata module, which can then apply appropriate penalties. The interface also supports trust coefficient updates that modify Learning Automata parameters based on detection insights, creating a feedback loop that enhances detection accuracy over time.

Byzantine Detection to Partition Management (BD-PM) Interface. This interface connects detection results with partition management actions. When the Byzantine Detection Engine identifies a malicious node, it uses this interface to request isolation of the peer in the Byzantine partition. It also communicates changes in peer classification status, which may trigger partition reorganization. In the reverse direction, the Partition Management System provides statistics about partition composition and behavior, which can inform detection processes. The interface also includes migration notifications that alert the Byzantine Detection Engine about peer movement between partitions, allowing it to update its behavioral models accordingly.

Partition Management to Content Distribution (PM-CD) Interface. The PM-CD interface links partition decisions with content distribution strategies. When partition structures change, this interface enables the synchronization of routing tables with the new configuration. It supports path optimization by determining optimal content routes across partitions while avoiding Byzantine nodes. The Content Distribution Handler reports distribution performance metrics back through this interface, providing valuable feedback about the effectiveness of current partition arrangements. Additionally, the interface includes load balancing signals that communicate partition imbalances affecting distribution efficiency, enabling dynamic partition adjustments.

1.3. Data Flow and System Dynamics

The BRLA-P2P framework implements a cyclical data flow pattern that enables continuous learning and adaptation. When peers interact, the system records detailed interaction information, including participants, interaction type, outcome, and contextual metadata. The Learning Automata module then updates state vectors based on these interaction outcomes, adjusting trust levels and action probabilities. Periodically, the Byzantine Detection Engine analyzes accumulated patterns to identify potentially malicious nodes, using both reputation metrics and behavioral anomalies.

When Byzantine nodes are detected, the Partition Management System isolates them into a dedicated partition and optimizes the remaining partitions to maintain balance and efficiency. Content requests are then routed through paths that avoid identified Byzantine nodes, maximizing delivery success rates. The performance metrics from these distribution operations feed back into the Learning Automata module, completing the cycle and enabling continuous improvement in Byzantine resistance.

Figure 2 illustrates this cyclical data flow, showing how information moves between components.

This cycle creates a self-reinforcing system that continuously improves Byzantine resistance while maintaining efficient content distribution.

2. Prototype Implementation

We have developed a functional prototype of the BRLA-P2P framework to validate our approach. The implementation encompasses all core components with their interfaces

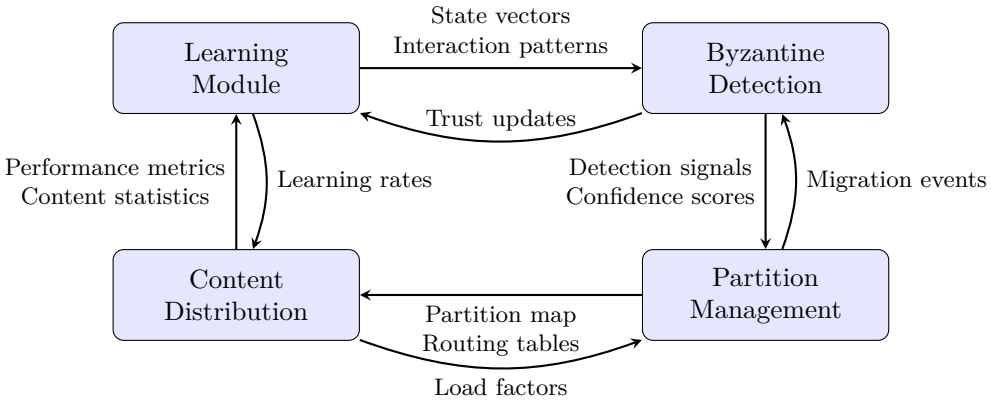

Figure 2: Data Flow in BRLA-P2P Framework

and provides comprehensive simulation capabilities. This prototype will serve as the foundation for our planned evaluation and parameter optimization studies.

2.1. Implementation Approach

Our implementation approach prioritized component separation, ensuring strict adherence to interface boundaries between the Learning Automata module, Byzantine Detection Engine, Partition Management System, and Content Distribution Handler. Each component was developed and tested independently before integration, ensuring modular functionality and facilitating isolated debugging. The system has been designed with high configurability, allowing parameterization of key properties such as learning rates, Byzantine penalties, detection thresholds, and partition constraints. We have incorporated comprehensive instrumentation throughout the code to enable detailed metric collection during simulations, capturing everything from detection accuracy to message overhead. Additionally, the implementation includes robust simulation capabilities that model realistic P2P network dynamics, including peer churn, content publication, request patterns, and Byzantine behaviors.

3. Planned Simulation and Evaluation

Our evaluation plan for the BRLA-P2P framework involves comprehensive simulation studies across diverse network conditions and Byzantine fault scenarios. This section outlines our planned simulation methodology and the metrics we will use to assess the framework's performance.

3.1. Simulation Methodology

Our simulation environment will model realistic P2P network dynamics to ensure meaningful evaluation of the BRLA-P2P framework. The network size will vary from small (100 peers) to large (1000+ peers) to assess scalability characteristics. We will test

with different Byzantine ratios, ranging from 0% (baseline) to 30% (extreme case), with particular focus on the 10-20% range that represents realistic threat levels in many P2P networks. The simulations will incorporate peer churn with approximately 5% of peers joining or leaving the network per iteration, reflecting the dynamic nature of real-world P2P systems. We will implement dynamic content generation where random peers publish new content items throughout the simulation, and random content request patterns that vary in frequency and distribution to model different usage scenarios.

Each simulation configuration will run for at least 1000 iterations to ensure statistical significance, with multiple runs using different random seeds to verify consistency of results. We will collect metrics at regular intervals throughout each simulation to observe convergence patterns and adaptation to changing conditions.

3.2. Byzantine Behavior Modeling

To ensure comprehensive evaluation of Byzantine resistance, we will implement multiple Byzantine behavior models with increasing sophistication. The most basic model will involve content pollution, where Byzantine nodes provide invalid or corrupted content in response to requests. A more advanced model will implement routing manipulation, where Byzantine nodes provide incorrect routing information to disrupt content distribution. We will also model selective behavior, where Byzantine nodes act honestly occasionally to avoid detection, making them more difficult to identify. The most sophisticated model will incorporate colluding attacks, where multiple Byzantine nodes coordinate their behavior to maximize disruption while minimizing detection probability.

For each behavior model, we will test with varying levels of Byzantine sophistication to assess the framework's detection capabilities across different threat profiles. These models will be parameterized to allow controlled experimentation with different attack strategies and intensities.

3.3. Evaluation Metrics

Our evaluation will focus on several key performance aspects of the BRLA-P2P framework:

Byzantine Detection Performance. We will evaluate the system's ability to identify Byzantine nodes accurately using several metrics. Detection accuracy will be measured as the percentage of Byzantine nodes correctly identified, a critical metric for assessing the framework's resistance to malicious behavior. False positive rate will be calculated as the percentage of honest nodes incorrectly flagged as Byzantine, an important consideration for usability and fairness. We will also measure detection time, defined as the number of interactions required before a Byzantine node is identified, which indicates the system's responsiveness to threats. Additionally, we will assess the framework's resilience to different Byzantine behavior models, examining how detection performance varies across different attack strategies and sophistication levels.

Content Distribution Performance. To evaluate the primary function of the P2P system, we will measure content distribution efficiency under various Byzantine conditions. Content success rate will be calculated as the percentage of content requests successfully fulfilled, the most direct measure of the system's ability to maintain functionality despite Byzantine presence. Average hop count will indicate the typical path length for successful content requests, with increases potentially reflecting the need to route around Byzantine nodes. We will also measure Byzantine block count, representing the number of requests initially blocked by Byzantine nodes but eventually fulfilled through alternative paths. Content availability will be assessed based on the percentage of content items successfully retrievable from at least one non-Byzantine node. Additionally, we will measure distribution latency under different Byzantine ratios to understand the performance impact of malicious nodes on content delivery times.

Scalability Analysis. The scalability of BRLA-P2P is a critical consideration for real-world applicability, so we will analyze performance characteristics across different network sizes. Message overhead will be measured as the average number of messages required per operation (e.g., content request, Byzantine detection), with particular attention to how this scales with network size. Memory usage will be tracked across different network configurations to understand resource requirements. Computational complexity will be assessed based on CPU time for key operations. We will also measure convergence time across different network sizes to understand how quickly the system stabilizes as the network grows.

Comparison with Existing Approaches. A comprehensive evaluation requires comparison with existing Byzantine-resistant P2P systems. We will implement models of leading alternatives, including Byzantine Fault Tolerant Distributed Hash Table (BFT-DHT), the Byzantine-Altruistic-Rational (BAR) model, and the Whānau Sybil-proof DHT. These systems will be simulated under identical conditions as BRLA-P2P to ensure fair comparison. We will compare detection accuracy and false positive rates across systems, particularly focusing on performance under high Byzantine ratios. Content distribution metrics, including success rates and message overhead, will be directly compared to assess relative efficiency. We will also examine convergence characteristics, comparing how quickly each system stabilizes under changing network conditions. Finally, we will evaluate the robustness of each approach to sophisticated Byzantine behaviors, particularly collusion attacks, which are challenging for many existing systems.

Each of these metrics will be collected across all simulation configurations, enabling comprehensive analysis of the framework's performance characteristics. The data will be aggregated to produce statistical measures (mean, median, standard deviation) and visualized through appropriate graphs to highlight trends and relationships.

3.4. Analysis Approach and Performance Metrics

Our analysis will focus purely on empirical measurements collected during the simulation studies. For Byzantine detection performance, we will measure accuracy (percentage of

correctly identified Byzantine nodes) and false positive rates (percentage of honest nodes incorrectly flagged) across different Byzantine ratios. We will examine when performance begins to decline as Byzantine ratios increase, and how different Byzantine behavior models affect detection efficacy.

For content distribution, we will quantify success rates (percentage of content requests successfully fulfilled) under varying Byzantine conditions. We will measure hop count increases to understand the network's adaptation to Byzantine presence, and identify any threshold points where performance significantly changes. The relationship between Byzantine ratio and performance metrics will be derived entirely from experimental data.

Regarding scalability, we will empirically measure message overhead at different network sizes to determine the actual scaling behavior of the framework. Similarly, memory usage will be measured directly during simulations rather than relying on theoretical predictions. These measurements will provide concrete data about the framework's resource requirements at scale.

When comparing with existing approaches, we will implement BFT-DHT, BAR model, and Whānau under identical experimental conditions and directly measure the performance difference. This head-to-head comparison will involve identical workloads, Byzantine behaviors, and network conditions to ensure fair assessment.

All analysis will be driven by experimental data, using statistical methods to identify correlations between metrics and simulation parameters. We will apply regression analysis to characterize relationships, conduct ANOVA tests to determine significant factors affecting performance, and employ time-series analysis to understand how metrics evolve throughout simulation runs. This empirical approach ensures that all conclusions are firmly grounded in experimental evidence rather than theoretical expectations.

4. Parameter Optimization Methodology

To maximize the performance of the BRLA-P2P framework, we plan to conduct systematic parameter optimization studies. This process will identify optimal configurations for different network conditions and use cases.

Learning Rate Optimization. The learning rate parameter significantly impacts both convergence speed and stability in the Learning Automata module. Our optimization study will test learning rates ranging from very low (0.01) to relatively high (0.5), with particular focus on the 0.05-0.3 range that preliminary testing suggests is most promising. For each learning rate, we will run complete simulations across multiple Byzantine ratios and network sizes. The key metrics for evaluation will include convergence time (number of iterations until stable partitioning), detection accuracy (percentage of Byzantine nodes correctly identified), and a stability score that measures the consistency of partitioning over time. We will analyze trade-offs between these metrics to identify optimal learning rates for different scenarios. For networks with high churn rates, we hypothesize that lower learning rates may provide better stability, while networks with more static membership might benefit from higher learning rates for faster convergence.

Byzantine Penalty Optimization. The Byzantine penalty parameter controls the severity of penalties applied to detected Byzantine nodes in the Learning Automata module. Our optimization study will explore penalties ranging from 0.1 (mild) to 0.6 (severe), with detailed focus on the 0.2-0.4 range. For each penalty value, we will measure detection time (iterations required to identify Byzantine nodes), isolation speed (iterations from detection to complete isolation), and false positive rate (honest nodes incorrectly penalized). The optimization goal is to identify penalty values that enable rapid isolation of Byzantine nodes while minimizing false positives. We anticipate that different Byzantine behavior models may require different penalty settings for optimal performance, with sophisticated selective behaviors potentially requiring higher penalties to overcome.

Partition Count Optimization. The optimal number of partitions depends on network size and content distribution patterns. Our optimization approach will test partition counts ranging from very few (2-4) to many (20+) across different network sizes. For each configuration, we will measure content success rate, average hop count, and network balance (distribution of peers across partitions). Based on preliminary understanding of partitioning dynamics, we hypothesize that the optimal partition count may follow a logarithmic relationship with network size. We will quantify this relationship through regression analysis of the optimization results, potentially yielding a formula that can predict optimal partition counts for any given network size.

Integrated Parameter Optimization. While individual parameter optimization provides valuable insights, interactions between parameters may affect overall system performance. Therefore, we will conduct integrated optimization using a factorial experimental design that tests combinations of high-performing parameter values identified in the individual studies. This approach will allow us to identify interaction effects and potential dependencies between parameters. For instance, higher learning rates might work better with lower Byzantine penalties, or certain partition counts might require adjusted learning rates. The results will be analyzed using statistical methods to quantify both main effects and interaction effects, leading to a set of recommended parameter configurations for different use cases and network conditions.

The optimization process will be iterative, with initial broad-range studies followed by more focused exploration of promising regions in the parameter space. The end result will be a set of parameter configuration guidelines that system administrators can use to tune BRLA-P2P for specific deployment scenarios.

5. Conclusion and Future Work

5.1. Summary of Progress

This research update has presented significant progress on the BRLA-P2P framework. We have finalized the detailed architectural design with well-defined interfaces between components, ensuring modularity and maintainability. We have implemented a functional

prototype that encompasses all core components and provides comprehensive simulation capabilities. We have developed a detailed evaluation plan that will assess Byzantine detection performance, content distribution efficiency, scalability characteristics, and comparative advantages over existing approaches. Additionally, we have outlined a systematic parameter optimization methodology that will identify optimal configurations for different network conditions.

5.2. Planned Next Steps

Our immediate next steps involve executing the simulation and evaluation plan described in this report. We will run comprehensive simulations across multiple Byzantine ratios, network sizes, and behavior models to collect the metrics outlined in our evaluation methodology. We will analyze the resulting data to quantify the performance characteristics of BRLA-P2P and compare it with existing approaches. We will also conduct the parameter optimization studies to identify optimal configurations and document configuration guidelines.

Beyond these immediate steps, we plan to explore several extensions to the BRLA-P2P framework. We will investigate enhanced detection mechanisms for coordinated Byzantine attacks, which remain challenging for many existing systems. We will explore dynamic parameter adaptation that automatically adjusts framework parameters based on observed network conditions, reducing the need for manual configuration. We also plan to investigate the applicability of BRLA-P2P to specific domains such as content delivery networks and blockchain systems, potentially developing domain-specific extensions to the framework.