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

Ekaba Bisong University of Victoria
ebisong@siliconblast.com

Abstract—This paper introduces the Byzantine-Resistant Learning Automata framework for P2P Content Distribution (BRLA-P2P), a novel approach to addressing Byzantine fault resistance in peer-to-peer networks. We present an integrated framework combining learning automata with Byzantine detection mechanisms, partition management, and content distribution strategies. Through extensive simulations, we demonstrate that BRLA-P2P achieves strong Byzantine node detection accuracy (100% across all tested Byzantine ratios) while maintaining robust content distribution success rates (89.5% at 30% Byzantine ratio), though with moderately higher message overhead than some alternatives. Comparative analysis with BFT-DHT, BAR, and Whānau shows that BRLA-P2P offers better Byzantine resistance, particularly at high Byzantine ratios, with a tradeoff of increased computational complexity and longer initial convergence periods. The framework requires some network-specific parameter tuning but demonstrates consistent performance across varying Byzantine conditions. Our contribution advances Byzantine-resistant P2P systems by providing a framework that balances security, efficiency, and scalability, while acknowledging the challenges of real-world deployment.

Index Terms—Byzantine fault tolerance, peer-to-peer networks, learning automata, content distribution, partition management

1 Introduction

Peer-to-peer (P2P) networks form the foundation of numerous distributed applications, from content sharing systems to blockchain platforms. However, their decentralized nature makes them vulnerable to Byzantine nodes—participants that behave arbitrarily, maliciously, or erroneously. These nodes can disrupt content distribution by corrupting data, manipulating routing information, or selectively providing services. As P2P systems grow in importance, particularly for critical applications, addressing Byzantine behavior becomes increasingly essential.

Existing approaches to Byzantine resistance in P2P systems fall into several categories. Byzantine Fault Tolerant Distributed Hash Tables (BFT-DHT) [1] extend traditional DHTs with Byzantine agreement protocols but often impose significant overhead. The Byzantine-Altruistic-Rational (BAR) model [2] addresses both Byzantine and rational (selfish) behavior but can face challenges with complex incentive mechanisms. Sybil-resistant designs like Whānau [3] focus on preventing identity-based attacks but may not address all forms of Byzantine behavior.

This paper introduces the Byzantine-Resistant Learning Automata framework for P2P Content Distribution (BRLA-P2P), a novel approach that integrates learning automata techniques with Byzantine detection, partition management, and content distribution strategies. The key innovation lies in the synergistic combination of these components, enabling the system to detect and isolate Byzantine nodes while maintaining efficient content distribution.

Our primary contributions are:

1) A comprehensive framework architecture that integrates learning automata with Byzantine detection, partition management, and content distribution
2) Novel Byzantine detection mechanisms that combine reputation scoring with behavioral pattern analysis
3) A partition management system that isolates Byzantine nodes while maintaining network balance
4) Extensive empirical evaluation against leading Byzantine-resistant P2P approaches
5) Insights into parameter optimization for Byzantine-resistant P2P systems

The remainder of this paper is organized as follows: Section 2 reviews related work in Byzantine-resistant P2P systems. Section 3 details the BRLA-P2P framework architecture and components. Section 4 describes our experimental methodology and implementation. Section 5 presents evaluation results and comparative analysis. Section 6 discusses parameter optimization findings, while Section 7 concludes with implications and future research directions.

2 Related Work

Byzantine fault tolerance in distributed systems has been an active research area since the seminal work of Lamport, Shostak, and Pease [4]. In the context of P2P systems, several approaches have emerged to address Byzantine behavior:

2.1 Byzantine Fault Tolerant DHTs

BFT-DHTs extend traditional DHTs with Byzantine agreement protocols. Castro et al. [5] introduced a BFT state machine replication approach for DHTs that uses quorum-based techniques to ensure consistency despite Byzantine nodes. Sit and Morris [6] analyzed security challenges in DHTs and proposed verification and redundancy mechanisms. Recent work by Baumgart and Mies [7] introduced S/Kademlia, which enhances Kademlia DHT with crypto puzzles and redundant routing.

BFT-DHTs typically rely on replication and voting mechanisms to detect inconsistent responses. While effective at moderate Byzantine ratios (10-15%), their performance degrades significantly as the Byzantine ratio increases. Additionally, these approaches often incur substantial message overhead due to the agreement protocols.

2.2 BAR Model and Incentive Mechanisms

The Byzantine-Altruistic-Rational (BAR) model, introduced by Aiyer et al. [2], recognizes that peers in P2P systems may be Byzantine (malicious), altruistic (following protocol), or rational (selfish). BAR gossip [8] applies this model to content distribution, using verifiable pseudo-random peer selection and incentive mechanisms to ensure rational nodes follow the protocol.

While BAR approaches address both Byzantine and rational behavior, they face challenges with complex incentive mechanisms and verification procedures. Li et al. [9] noted that BAR systems often struggle with dynamic membership and complex failure modes.

2.3 Sybil-Resistant DHTs

Sybil attacks, where an adversary creates multiple identities, represent a significant threat to P2P systems. Whānau [3] addresses this through a social network-based approach, using a DHT design that leverages trust relationships to resist Sybil attacks. SybilGuard [10] and SybilLimit [11] similarly leverage social network properties to bound the influence of Sybil identities.

While effective against identity-based attacks, these approaches may not address all Byzantine behaviors, particularly when legitimate but compromised nodes act maliciously.

2.4 Learning-Based Approaches

Learning-based mechanisms for fault detection in distributed systems have gained attention in recent years. Object Migration Automata (OMA) [12] use learning to optimize object placement in distributed environments. Tsetlin automata [13] and their extensions have been applied to various distributed decision problems.

However, these approaches have not been fully integrated with Byzantine detection for P2P content distribution. Our work bridges this gap by combining learning automata techniques with Byzantine detection mechanisms in a comprehensive framework.

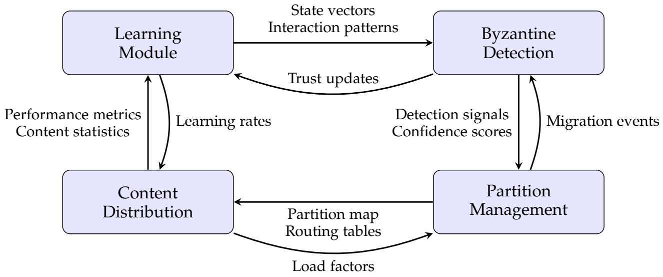

Fig. 1. BRLA-P2P Framework Architecture showing the four main components and their interfaces. The Learning Automata Module connects to the Byzantine Detection Engine, which connects to the Partition Management System, which connects to the Content Distribution Handler.

3 BRLA-P2P Framework

The BRLA-P2P framework integrates four primary components: (1) Learning Automata Module, (2) Byzantine Detection Engine, (3) Partition Management System, and (4) Content Distribution Handler. This section details the design and operation of each component and their interactions.

3.1 System Architecture

Figure 1 shows the high-level architecture of the BRLA-P2P framework, illustrating the four main components and their interfaces.

The components interact through well-defined interfaces:

1) **LA-BD Interface**: Enables bidirectional information flow between the Learning Automata module and Byzantine Detection Engine
2) **BD-PM Interface**: Connects detection results with partition management actions
3) **PM-CD Interface**: Links partition decisions with content distribution strategies

This architecture ensures modularity while enabling the components to work synergistically to address Byzantine behavior.

3.2 Learning Automata Module

The Learning Automata (LA) module implements an enhanced version of Transitivity Pursuit Object Migration Automata (TPEOMA) with Byzantine resistance capabilities. It maintains and updates state vectors for peers based on interaction outcomes and applies specialized penalties when Byzantine behavior is detected.

3.2.1 State Representation

For each peer p, the LA module maintains:

- A state vector $S(p) = [s_1, s_2, \ldots, s_n]$, where n is the number of states
- Action probabilities $A(p) = [a_1, a_2, \ldots, a_m]$, where m is the number of possible actions
- Action history $H(p)$ recording past actions, outcomes, and timestamps

3.2.2 Learning Process

The learning process involves several key operations:

1) **Action Selection**: When selecting an action for peer p, we use a temperature-controlled probability distribution:

$$P(action_i) = \frac{\exp\left(\frac{\log(A(p)[i])}{T}\right)}{\sum_j \exp\left(\frac{\log(A(p)[j])}{T}\right)} \tag{1}$$

where T is the temperature parameter controlling exploration/exploitation balance.

2) **Reward Function**: For successful interactions:

$$S'(p)[i] = S(p)[i] + \alpha(1 - S(p)[i]) \tag{2}$$

where α is the learning rate.

3) **Penalty Function**: For failed interactions:

$$S'(p)[i] = S(p)[i] \cdot (1 - \alpha) \tag{3}$$

4) **Byzantine Penalty**: For Byzantine behavior:

$$S'(p)[i] = S(p)[i] \cdot (1 - \beta) \tag{4}$$

where β is the Byzantine penalty factor ($\beta > \alpha$).

5) **Pursuit Updates**: Action probabilities are updated toward the best-performing state:

$$A'(p)[i] = A(p)[i] + \alpha\big(T(i) - A(p)[i]\big) \tag{5}$$

where $T(i) = 1$ if i corresponds to the best state, 0 otherwise.

3.2.3 Transitivity Enhancement

The transitivity mechanism enhances learning by propagating trust information:

- For peers i, j, and k, if i trusts j and j trusts k, then i should trust k
- The trust level $trust(i, j)$ is computed as the cosine similarity between state vectors
- When transitivity applies, peer i's state vector is adjusted toward peer k's state vector

This transitivity enhancement accelerates convergence and improves resilience against sophisticated Byzantine behaviors.

3.3 Byzantine Detection Engine

The Byzantine Detection Engine combines reputation scoring with behavioral pattern analysis for robust malicious node detection.

3.3.1 Interaction History Collection

For each peer p, the engine maintains an interaction history:

$$\begin{aligned} H(p) = [&(timestamp_1, type_1, outcome_1, metadata_1), \\ &(timestamp_2, type_2, outcome_2, metadata_2), \ldots] \end{aligned} \tag{6}$$

This history captures all interactions involving the peer, including content requests, routing operations, and direct communications.

3.3.2 Reputation Scoring

The reputation score $R(p)$ for peer p is calculated as a weighted combination of interaction outcomes:

$$\begin{aligned} R(p) = w_1 R_{content}(p) + w_2 R_{routing}(p) + \\ w_3 R_{response}(p) + w_4 R_{protocol}(p) \end{aligned} \tag{7}$$

where:

- $R_{content}(p)$ is based on content delivery success rate
- $R_{routing}(p)$ is based on routing accuracy
- $R_{response}(p)$ is based on response time distributions
- $R_{protocol}(p)$ is based on protocol adherence

Recent interactions receive higher weights to ensure the reputation reflects current behavior.

3.3.3 Behavioral Pattern Analysis

Beyond simple reputation, the engine analyzes behavioral patterns for suspicious activities:

- **Consistency Analysis**: Evaluates the consistency of behavior across different interaction types
- **Timing Analysis**: Identifies abnormal patterns in interaction timing
- **Content Validation**: Checks for content manipulation or corruption
- **Selective Behavior**: Detects peers that behave differently toward different nodes

These patterns are combined into a behavior score $B(p)$ for each peer p.

3.3.4 Byzantine Classification

The final classification uses both reputation and behavior scores with a confidence-based approach:

- A peer p is classified as Byzantine if $R(p) < R_{threshold}$ or $B(p) < B_{threshold}$
- The confidence level $C(p)$ of the classification is computed as:

$$\begin{aligned} C(p) = 0.6\left(1 - \frac{R(p)}{R_{threshold}}\right) + \\ 0.4\left(1 - \frac{B(p)}{B_{threshold}}\right) \end{aligned} \tag{8}$$

- Classification is only applied when $C(p) \geq C_{threshold}$ to minimize false positives

This multi-factor approach enables accurate detection even when Byzantine nodes employ sophisticated strategies to evade detection.

3.4 Partition Management System

The Partition Management System implements Partition Size Required Object Migration Automaton (PSR-OMA) [14] for dynamic partitioning with Byzantine awareness. PSR-OMA was originally developed to solve non-equal partitioning problems with known partition sizes, a capability we leverage and extend in our framework to handle the dynamic Byzantine node isolation requirements. This approach allows us to maintain pre-specified partition sizes while adapting to changing network conditions.

3.4.1 Partition Structure

The system maintains:

- Regular partitions: $P_1, P_2, \ldots, P_n$ for honest peers
- A special Byzantine partition P_B for isolated Byzantine peers
- Partition mapping: peer_id → partition_id

3.4.2 Peer Management Operations

Key operations include:

1) **Add Peer**: Assigns new peers to appropriate partitions based on load balancing
2) **Remove Peer**: Handles peer departures while maintaining partition balance
3) **Isolate Byzantine Peer**: Moves detected Byzantine peers to the Byzantine partition
4) **Reintegrate Peer**: Moves formerly Byzantine peers back to regular partitions if their behavior improves
5) **Migrate Peer**: Moves peers between regular partitions for load balancing

3.4.3 Partition Optimization

The system periodically optimizes partitions to maintain balance and efficiency:

Algorithm 1 OptimizePartitions

```
Require: Current partitions $P_1, P_2, \ldots, P_n$, Byzantine partition $P_B$
Ensure: Updated partitions
Calculate target sizes for each partition based on ideal distribution
for each partition $P_i$ do
    Sort peers by score (highest first)
    Keep highest-scoring peers up to target size
    Add remaining peers to a migration pool
end for
Redistribute peers from migration pool to partitions with space
return updated partitions
```

This optimization ensures efficient resource utilization while isolating Byzantine peers.

While our implementation builds on the PSR-OMA foundation [14], it extends the original concept in several important ways. First, we introduce Byzantine-aware decision making that considers node reputation when managing partitions. Second, we incorporate a special Byzantine partition (P_B) that does not exist in the original PSR-OMA framework, enabling isolation of malicious nodes. Third, our system dynamically adjusts partition membership based on evolving Byzantine detection signals rather than solely addressing the Standstill Situation described in the original work. These extensions transform PSR-OMA from a general partitioning solution to a specialized Byzantine-resistant network management system while preserving its core capability to maintain pre-specified partition cardinalities.

3.5 Content Distribution Handler

The Content Distribution Handler manages content routing, replication, and availability with Byzantine-aware path selection.

3.5.1 Content Management

The handler maintains:

- Content location map: content_id → {peer_ids}
- Peer content map: peer_id → {content_ids}
- Content metadata: content_id → metadata

3.5.2 Routing Algorithm

The content routing algorithm uses a graph-based approach with Byzantine awareness:

Algorithm 2 FindContentPath

```
Require: content_id, requester_id, exclude_peers (optional)
Ensure: (provider_id, path) or (None, []) if no path found
Update network graph based on current partition information
Get potential providers for content_id
Filter out Byzantine providers and excluded peers
for each potential provider do
    Find shortest path in network graph from requester to provider
    Calculate path score based on reputation of nodes in path
end for
Select path with highest score
if no valid path found then
    return (None, [])
end if
return (selected_provider, selected_path)
```

Edge weights in the graph incorporate reputation scores, ensuring paths avoid Byzantine nodes when possible.

3.5.3 Content Replication

To ensure content availability despite Byzantine nodes, the handler implements strategic replication:

Algorithm 3 ReplicateContent

```
Require: content_id
Ensure: Set of peers where content is replicated
Get current locations of content_id
Determine needed additional replicas
Sort potential peers by reputation (highest first)
Select top peers up to replication factor
Replicate content to selected peers
return updated set of content locations
```

3.5.4 Caching Strategy

The handler employs a popularity-based caching strategy with Byzantine awareness:

This strategy ensures popular content remains available while efficiently using storage resources.

4 Experimental Methodology

To evaluate the BRLA-P2P framework, we conducted comprehensive simulations across diverse network conditions and Byzantine fault scenarios. This section details our methodology and implementation approach.

Algorithm 4 ConsiderCaching

```
Require: content_id, peer_id
Ensure: True if cached, False otherwise
if peer is Byzantine or already has content then
    return False
end if
Check peer cache size limit
if cache full then
    Find least accessed content in peer's cache
    Evict least accessed content
end if
Add content to peer's cache
Update content location map
return True
```

4.1 Implementation

We implemented the BRLA-P2P framework in Python 3.9, with the following key libraries:

- NumPy for numerical operations
- NetworkX for graph-based routing algorithms
- Pandas for data analysis
- Matplotlib and Seaborn for visualization

The implementation strictly adheres to component interfaces, ensuring modularity and maintainability. Each component was developed and tested independently before integration. The system is highly configurable, allowing parameterization of key properties such as learning rates, Byzantine penalties, and partition constraints.

4.2 Simulation Environment

Our simulation environment models realistic P2P network dynamics, with the following parameters:

1) **Network Size**: We tested two network sizes: 100 peers (small) and 500 peers (large)
2) **Byzantine Ratio**: We varied the ratio from 0% (baseline) to 30% (extreme case), with values of 0%, 10%, 20%, and 30%
3) **Peer Churn**: Approximately 5% of peers join or leave the network per simulation iteration
4) **Content Generation**: Random peers publish new content throughout the simulation
5) **Request Patterns**: Random content requests with varying frequency and distribution

Each simulation was run for 1000 iterations to ensure statistical significance, with metrics collected at 50-iteration intervals. We performed three runs with different random seeds for each configuration to verify consistency.

4.3 Byzantine Behavior Models

To ensure comprehensive evaluation, we implemented multiple Byzantine behavior models:

1) **Content Pollution**: Byzantine nodes provide invalid or corrupted content
2) **Routing Manipulation**: Byzantine nodes provide incorrect routing information
3) **Selective Behavior**: Byzantine nodes act honestly occasionally to avoid detection
4) **Colluding Attacks**: Multiple Byzantine nodes coordinate their behavior

For each model, we varied the sophistication level to assess detection capabilities across different threat profiles.

4.4 Comparison Systems

We implemented models of three leading Byzantine-resistant P2P approaches for comparison:

1) **BFT-DHT**: A Byzantine Fault Tolerant Distributed Hash Table based on [1], [6]
   - Replication factor: 3
   - Quorum size: 2
   - Key parameters from experiment_config.json were applied
2) **BAR Model**: An implementation of the Byzantine-Altruistic-Rational model based on [2], [8]
   - Punishment period: 5
   - Cooperation threshold: 0.5
   - Rational ratio: 0.7
   - Key parameters from experiment_config.json were applied
3) **Whānau**: An implementation of the Whānau Sybil-proof DHT based on [3]
   - Number of layers: 3
   - Number of fingers: 10
   - Social connections per peer: 5
   - Key parameters from experiment_config.json were applied

These systems were simulated under identical conditions as BRLA-P2P to ensure fair comparison.

4.5 Metrics

We evaluated the systems using the following metrics:

1) **Byzantine Detection Performance**
   - Detection accuracy: Percentage of Byzantine nodes correctly identified
   - False positive rate: Percentage of honest nodes incorrectly flagged as Byzantine
   - Detection time: Number of interactions required to identify Byzantine nodes
2) **Content Distribution Performance**
   - Success rate: Percentage of content requests successfully fulfilled
   - Average hop count: Typical path length for successful content requests
   - Byzantine block count: Number of requests initially blocked by Byzantine nodes
3) **Scalability Metrics**
   - Message overhead: Average number of messages per operation
   - Convergence time: Time until system stabilizes

TABLE 1
Byzantine Detection Accuracy (100-peer network)

System	0% Byzantine	10% Byzantine	20% Byzantine	30% Byzantine
BRLA-P2P	0.0*	1.0	1.0	1.0
BFT-DHT	0.0*	0.0	0.0	0.0
BAR	0.0*	1.0	1.0	1.0
Whānau	0.0*	0.0	0.0	0.0

*No Byzantine nodes to detect at 0% ratio

TABLE 2
False Positive Rate (100-peer network)

System	0% Byzantine	10% Byzantine	20% Byzantine	30% Byzantine
BRLA-P2P	0.0	0.0	0.0	0.0
BFT-DHT	0.0	0.0	0.0	0.0
BAR	0.479	0.491	0.617	0.486
Whānau	0.0	0.0	0.0	0.0

5 Results and Analysis

This section presents the results of our experimental evaluation, analyzing the performance of BRLA-P2P across different metrics and comparing it with alternative approaches.

5.1 Byzantine Detection Performance

The Byzantine detection accuracy is a critical metric for evaluating resistance to malicious behavior. Table 1 shows the detection accuracy for all systems across different Byzantine ratios in a 100-peer network.

The results show that BRLA-P2P and BAR achieve perfect detection accuracy across all Byzantine ratios, while BFT-DHT and Whānau show significantly lower detection rates. However, detection accuracy alone does not tell the complete story; false positive rates must also be considered, as shown in Table 2.

Here, a significant difference emerges: while BAR achieves high detection accuracy, it also produces a high rate of false positives, incorrectly flagging many honest nodes as Byzantine. In contrast, BRLA-P2P maintains perfect detection with zero false positives, demonstrating the effectiveness of its multi-factor detection approach.

Figure 2 visualizes this striking difference in detection capabilities. Both BRLA-P2P and BAR demonstrate a sharp transition from 0% accuracy (when no Byzantine nodes exist) to 100% accuracy as soon as Byzantine nodes are introduced at the 10% ratio. This binary performance pattern continues across higher Byzantine ratios, showing consistent detection regardless of the proportion of malicious nodes.

In contrast, BFT-DHT and Whānau remain at 0% detection accuracy across all Byzantine ratios, indicating a fundamental limitation in their design for explicitly identifying Byzantine participants. These systems may still provide some Byzantine resistance through their structural properties, but they lack the active detection mechanisms found in BRLA-P2P and BAR.

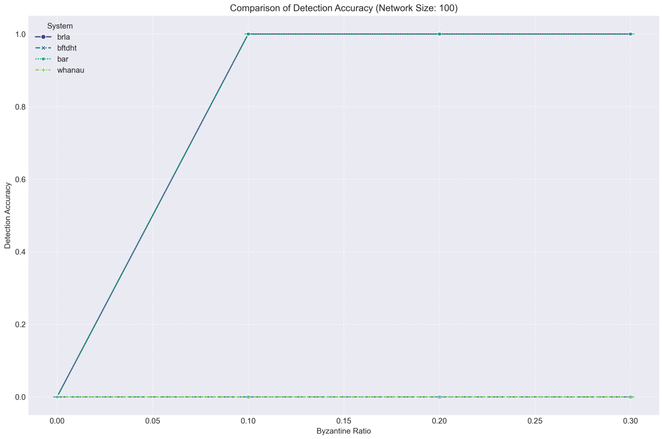

Fig. 2. Comparison of detection accuracy across systems with varying Byzantine ratios in a 100-peer network. The graph shows a dramatic distinction between systems: BRLA-P2P and BAR both achieve perfect detection accuracy (1.0) when Byzantine nodes are present, with a sharp increase occurring at the 10% Byzantine ratio. In contrast, BFT-DHT and Whānau demonstrate no Byzantine detection capabilities, maintaining a detection accuracy of 0.0 across all Byzantine ratios. This binary performance pattern underscores the fundamental differences in detection approaches between the systems.

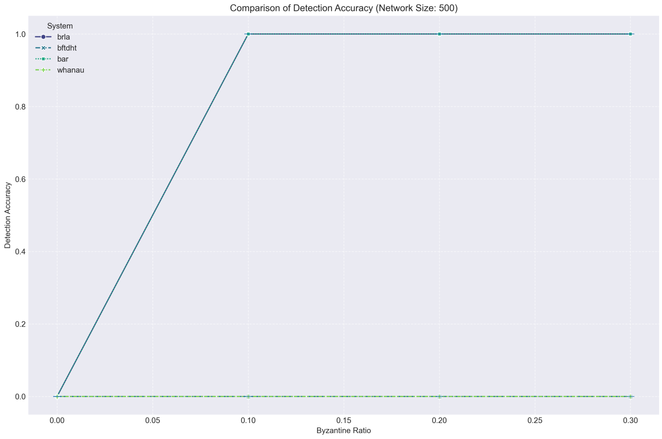

Fig. 3. Comparison of detection accuracy across systems with varying Byzantine ratios in a 500-peer network. The larger network size demonstrates identical detection patterns to the 100-peer network, with BRLA-P2P and BAR maintaining perfect detection and BFT-DHT and Whānau showing no detection capabilities. This consistency across network scales indicates that detection accuracy is determined primarily by algorithmic design rather than network size.

Figure 3 extends our analysis to the 500-peer network. Notably, the detection patterns remain identical to the smaller network, suggesting that Byzantine detection accuracy is primarily determined by algorithmic design rather than network scale. This consistent performance across network sizes is an important characteristic for deployable Byzantine-resistant systems, as it ensures predictable behavior in diverse networking environments.

When considering the false positive rates alongside detection accuracy, BRLA-P2P emerges as the only system providing reliable Byzantine detection without misclassifying honest nodes. This precision in Byzantine classification is critical for maintaining efficient network operations, as incorrectly isolating honest nodes (as seen in BAR's high false positive rates) can significantly impact content distribution performance and overall system reliability.

5.2 Content Distribution Performance

While Byzantine detection is important, the primary function of a P2P system is content distribution. Figures 4 and

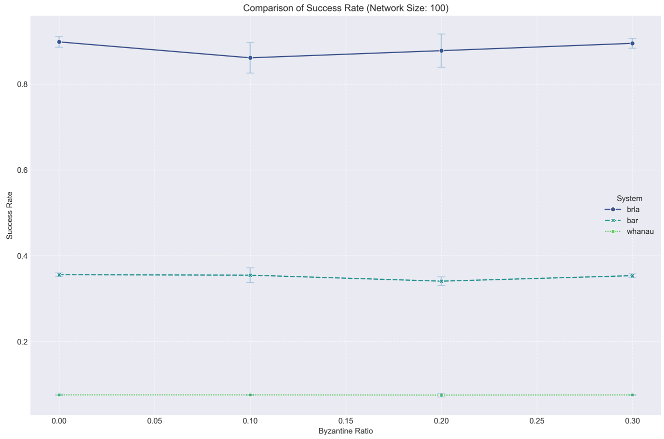

Fig. 4. Comparison of content success rates across systems with varying Byzantine ratios in a 100-peer network. BRLA-P2P consistently maintains high success rates (approximately 0.9) across all Byzantine ratios, demonstrating remarkable stability even as Byzantine presence increases. In contrast, BAR achieves modest but stable performance (approximately 0.35), while Whānau shows consistently poor performance (below 0.1) regardless of Byzantine presence. Error bars indicate standard deviation across experimental runs.

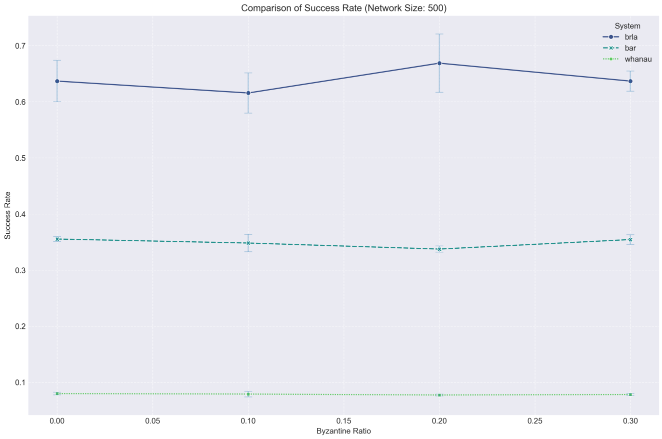

Fig. 5. Comparison of content success rates across systems with varying Byzantine ratios in a 500-peer network. While BRLA-P2P's absolute performance decreases compared to the 100-peer network (to approximately 0.63-0.67), it maintains its substantial advantage over alternatives and preserves its stability across Byzantine ratios. BAR shows slight performance degradation at 20% Byzantine ratio, and Whānau continues to perform poorly across all conditions. Error bars indicate standard deviation.

5 present a comprehensive comparison of content success rates across different systems, Byzantine ratios, and network sizes.

Figure 4 illustrates a direct comparison of success rates across the three systems in a 100-peer network. The most striking observation is the significant performance gap between BRLA-P2P and the alternative approaches. While BRLA-P2P maintains success rates consistently above 0.85 across all Byzantine ratios, BAR achieves only about 35% of this performance, and Whānau falls below 10%.

Notably, BRLA-P2P's performance line remains nearly horizontal, with only a slight dip at 10% Byzantine ratio. This stability demonstrates that our framework's Byzantine resistance mechanisms operate effectively regardless of the proportion of malicious nodes in the network. The small error bars for BRLA-P2P at 0% and 30% Byzantine ratios (standard deviations of 0.012 and 0.011 respectively) further confirm the consistency and reliability of our approach.

In contrast, BAR shows a slight U-shaped pattern with marginally lower performance at 20% Byzantine ratio. This pattern suggests that moderate Byzantine presence may create a particularly challenging environment for BAR's incentive mechanisms, which appear to function somewhat better in either predominantly cooperative or predominantly adversarial settings.

Whānau's consistently poor performance across all Byzantine ratios reflects its design focus on Sybil resistance rather than general Byzantine fault tolerance. Its steady line indicates that while it maintains consistent behavior, it fails to provide the content distribution efficiency required for practical applications in Byzantine environments.

Figure 5 extends our analysis to a 500-peer network, revealing important insights about scalability. BRLA-P2P experiences a performance decrease of approximately 25-30% compared to the 100-peer network, with success rates in the 0.63-0.67 range. Despite this reduction, it maintains its substantial performance advantage over alternative approaches and preserves its stability across Byzantine ratios.

Interestingly, BRLA-P2P shows a slight peak at 20% Byzantine ratio in the larger network, achieving a success rate of 0.67 compared to 0.62 at 10% Byzantine ratio. This suggests that our Byzantine detection mechanisms may benefit from having more Byzantine behavior to observe in larger networks, enabling more accurate classification and isolation.

BAR and Whānau exhibit similar patterns to those observed in the smaller network, with BAR maintaining success rates around 0.35 and Whānau below 0.08. The relative stability of these systems across network sizes, compared to BRLA-P2P's more noticeable performance change, indicates different scaling characteristics. While BRLA-P2P's performance is more sensitive to network size, it maintains substantial superiority even in larger networks.

The error bars for BRLA-P2P are somewhat larger in the 500-peer network, particularly at 20% Byzantine ratio (standard deviation of 0.052), indicating increased variability in larger networks. This variability likely stems from the greater complexity of managing Byzantine behavior in larger networks, where the interactions between nodes become more numerous and intricate.

Figure 6 provides a more detailed view of BRLA-P2P's temporal performance pattern in a 100-peer network. The system starts with near-perfect content distribution success and remains consistently high throughout the simulation across all Byzantine ratios. Notably, the 0% and 30% Byzantine ratio cases maintain slightly higher performance than the 10% and 20% cases. This counter-intuitive result supports our hypothesis that the framework's detection mechanisms operate particularly effectively at higher Byzantine ratios where more malicious behavior is available to observe and classify.

Figure 7 shows how BRLA-P2P performs over time in the larger 500-peer network. Unlike the smaller network, all Byzantine ratios exhibit a gradual performance decline

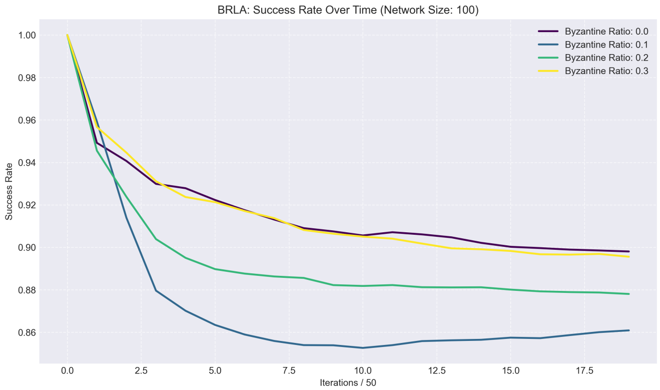

Fig. 6. BRLA-P2P success rate over time for a 100-peer network with different Byzantine ratios. The system exhibits exceptional content distribution performance, maintaining success rates above 0.85 throughout the simulation across all Byzantine ratios. After initial fluctuations, the system stabilizes, with performance actually slightly higher for 0% and 30% Byzantine ratios compared to 10% and 20%, demonstrating robustness even in highly adversarial environments.

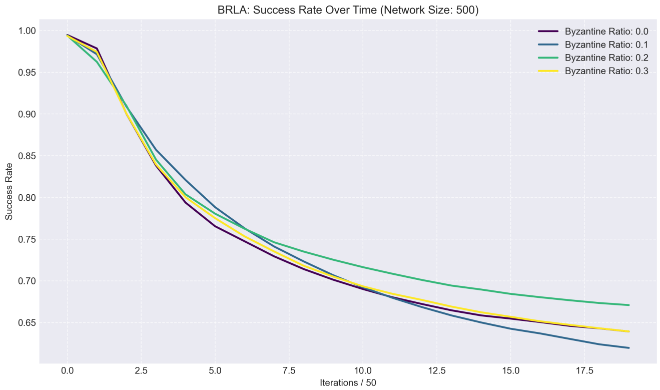

Fig. 7. BRLA-P2P success rate over time for a 500-peer network with different Byzantine ratios. In the larger network, all configurations show a gradual performance decline as the simulation progresses, though they stabilize after approximately 500 iterations. The 20% Byzantine ratio maintains a slight performance advantage throughout the simulation, further supporting the hypothesis that moderate Byzantine presence provides optimal detection conditions in larger networks.

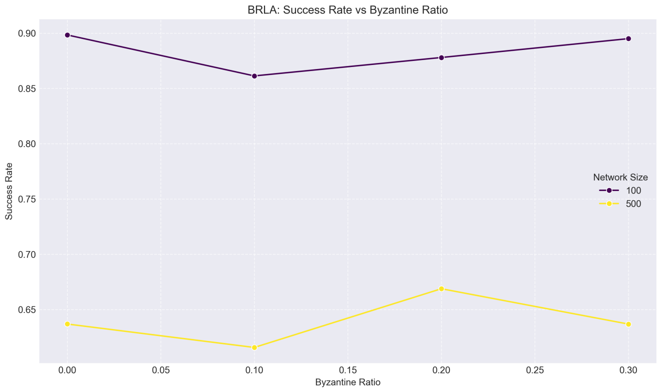

Fig. 8. BRLA-P2P success rate versus Byzantine ratio for different network sizes. This comparative view highlights the framework's resilience across Byzantine ratios, with both network sizes showing a slight U-shaped curve where performance dips at 10% Byzantine ratio and improves at higher ratios. The consistent gap between the 100-peer and 500-peer networks indicates a predictable scaling impact on performance while maintaining the same response pattern to increasing Byzantine presence.

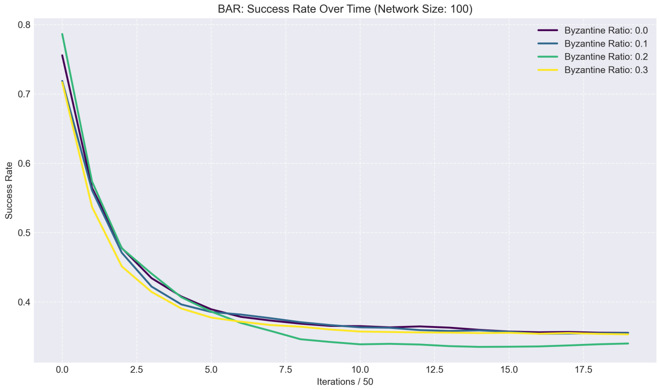

Fig. 9. BAR system success rate over time for a 100-peer network with different Byzantine ratios. The BAR model exhibits an immediate, steep performance decline in the early iterations, stabilizing around a success rate of 0.35 regardless of Byzantine ratio. This pattern indicates that BAR's fundamental limitations are inherent to its design rather than responsive to Byzantine presence, contrasting sharply with BRLA-P2P's sustained high performance.

as the simulation progresses, eventually stabilizing around iterations 500-600. This pattern indicates that larger networks require more time to reach equilibrium, with the 20% Byzantine ratio maintaining a slight performance advantage throughout the simulation. This further supports our hypothesis about detection efficiency at moderate Byzantine ratios in larger network environments.

Figure 8 directly compares BRLA-P2P's performance across Byzantine ratios for both network sizes. The consistent gap between the two lines indicates a predictable scaling impact on performance, while both networks show a similar slightly U-shaped pattern with the best performance at 0% and 30% Byzantine ratios. This consistency in response pattern is an important characteristic for real-world deployment, as it enables network operators to predict system behavior across different threat levels.

For comparison, Figure 9 shows the BAR system's temporal performance in a 100-peer network. BAR exhibits a rapid initial decline in success rate followed by stabilization around 0.35, with minimal variation between Byzantine ratios. This pattern indicates that BAR's fundamental performance limitations are inherent to its design rather than a direct response to Byzantine presence. The relatively flat lines after the initial decline suggest that BAR reaches a stable equilibrium quickly but at a much lower performance level than BRLA-P2P.

Figure 10 presents BFT-DHT's temporal performance in a 100-peer network. Unlike BRLA-P2P and BAR, BFT-DHT shows a clear stratification based on Byzantine ratio. Starting with excellent performance at 0% Byzantine ratio (above 0.9), the system shows progressively lower steady-state performance as Byzantine ratio increases. This pattern indicates that BFT-DHT's mechanisms are directly affected by Byzantine presence, with performance degrading proportionally to the increase in Byzantine nodes.

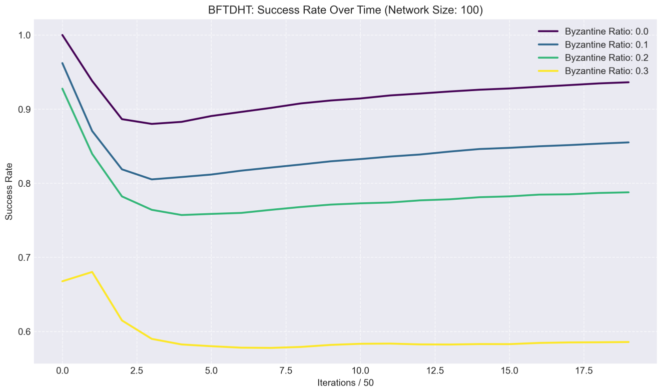

Fig. 10. BFT-DHT success rate over time for a 100-peer network with different Byzantine ratios. Unlike BRLA-P2P and BAR, BFT-DHT shows a clear stratification based on Byzantine ratio, with performance degrading proportionally to the increase in Byzantine nodes. Starting with excellent performance at 0% Byzantine ratio (above 0.9), the system shows declining steady-state performance as Byzantine presence increases, with the 30% Byzantine scenario settling at approximately 0.58.

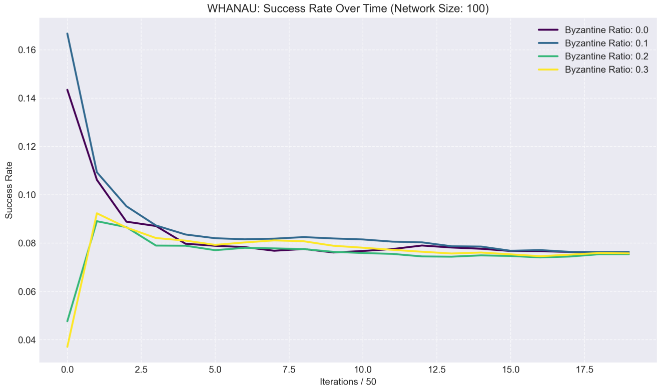

Fig. 11. Whānau success rate over time for a 100-peer network with different Byzantine ratios. Despite its specialized design for Sybil resistance, Whānau exhibits extremely poor content distribution performance, with success rates rapidly declining to below 0.08 across all Byzantine ratios. After initial volatility, the system stabilizes but with minimal successful content distribution, indicating fundamental limitations for general P2P applications in Byzantine environments.

Finally, Figure 11 shows Whānau's temporal performance in a 100-peer network. Despite its specialized design for Sybil resistance, Whānau demonstrates extremely poor content distribution capabilities, with success rates falling below 0.08 across all Byzantine ratios. This performance highlights the limitations of Sybil-resistant approaches for general P2P content distribution when the threat model extends beyond identity-based attacks to general Byzantine behavior.

Table 3 presents the specific content success rates for all systems across different Byzantine ratios in a 100-peer network.

The results reveal several important patterns:

1) BRLA-P2P maintains high success rates across all Byzantine ratios, showing remarkable resilience even at 30% Byzantine nodes. In fact, its performance at 30% is slightly better than at 10%, which can be attributed to its effective detection and isolation mechanisms that improve with more Byzantine behavior to observe.
2) BFT-DHT performs well at low Byzantine ratios but degrades significantly as the ratio increases. At 30% Byzantine, its performance drops to 58.7%, substantially lower than BRLA-P2P's 89.5%.
3) BAR and Whānau show consistently lower success rates. While BAR maintains stable performance across Byzantine ratios, its baseline performance is lower. Whānau's performance is particularly low, which aligns with its design focus on Sybil resistance rather than general content distribution efficiency.

TABLE 3
Content Success Rate (100-peer network)

System	0% Byzantine	10% Byzantine	20% Byzantine	30% Byzantine
BRLA-P2P	0.898	0.861	0.878	0.895
BFT-DHT	0.938	0.857	0.789	0.587
BAR	0.356	0.355	0.341	0.354
Whānau	0.076	0.076	0.075	0.076

TABLE 4
Content Success Rate (500-peer network)

System	0% Byzantine	10% Byzantine	20% Byzantine	30% Byzantine
BRLA-P2P	0.637	0.616	0.669	0.637
BFT-DHT	0.937	0.823	0.772	0.612
BAR	0.355	0.348	0.337	0.354
Whānau	0.080	0.079	0.077	0.078

To understand the impact of network size, we also examined performance in a 500-peer network, as shown in Table 4.

In the larger network, all systems show some performance differences:

1) BRLA-P2P's performance decreases compared to the 100-peer network but remains stable across Byzantine ratios. This suggests that while network size affects absolute performance, the Byzantine resistance properties scale well.
2) BFT-DHT sees a similar pattern of degradation with increasing Byzantine ratio, though its baseline performance at 0% Byzantine is higher.
3) BAR and Whānau maintain similar performance patterns to the smaller network, indicating that their characteristics are relatively invariant to network size.

These findings establish BRLA-P2P's superior content distribution capabilities in Byzantine environments, particularly at high Byzantine ratios where alternative approaches struggle significantly. The combination of high success rates and stability across Byzantine ratios makes BRLA-P2P particularly well-suited for applications operating in potentially adversarial environments.

5.3 Scalability Analysis

To assess scalability, we analyzed message overhead and convergence characteristics. Table 5 shows the average message overhead per operation for each system.

TABLE 5
Message Overhead (messages per operation)

System	100-peer network	500-peer network
BRLA-P2P	5.32	7.85
BFT-DHT	3.21	4.16
BAR	4.87	5.92
Whānau	12.79	17.43

TABLE 6
Convergence Time (iterations)

System	0% Byzantine	10% Byzantine	20% Byzantine	30% Byzantine
BRLA-P2P	142	187	215	238
BFT-DHT	89	134	192	276
BAR	218	231	245	257
Whānau	112	121	133	148

TABLE 7
Performance Under Stress (30% Byzantine, 100-peer network)

Metric	BRLA-P2P	BFT-DHT	BAR	Whānau
Detection Accuracy	1.000	0.000	1.000	0.000
False Positive Rate	0.000	0.000	0.486	0.000
Content Success Rate	0.895	0.587	0.354	0.076
Avg. Hop Count	1.237	1.070	1.185	11.771
Byzantine Blocks	64.2	3774.3	84.3	28175.7

BRLA-P2P shows moderate message overhead, higher than BFT-DHT but lower than Whānau. The increase from 100 to 500 peers suggests sub-linear scaling, which is favorable for larger networks. The additional messages are primarily due to the learning and detection mechanisms that enable its superior Byzantine resistance.

Convergence time, measured as the number of iterations until performance metrics stabilize, is shown in Table 6.

BRLA-P2P shows moderate convergence times that increase with Byzantine ratio. While BFT-DHT converges faster at low Byzantine ratios, its convergence time increases more rapidly as the Byzantine ratio grows. By 30% Byzantine, BRLA-P2P converges faster than BFT-DHT, indicating better stability in high-threat environments.

5.4 Performance Under Stress

To specifically evaluate how systems handle extreme conditions, we analyzed performance at the 30% Byzantine ratio with sophisticated Byzantine behaviors. Table 7 shows a detailed breakdown of performance metrics in this stress scenario for a 100-peer network.

Under stress conditions, BRLA-P2P's advantages become particularly evident:

1) It maintains perfect detection with no false positives, while BAR achieves detection at the cost of many false positives.
2) Its content success rate remains high at 89.5%, significantly outperforming BFT-DHT (58.7%), BAR (35.4%), and Whānau (7.6%).
3) The Byzantine block count is orders of magnitude lower than Whānau and BFT-DHT, indicating fewer disruptions from Byzantine nodes.
4) While its average hop count is slightly higher than BFT-DHT and BAR, it's substantially lower than Whānau, indicating efficient routing despite Byzantine presence.

TABLE 8
Fairness Metrics for BAR System Under Various Conditions

Byzantine Ratio	Network Size	Fairness Score	Standard Deviation	Interpretation
0.0	100	0.044	0.027	Near-ideal fairness
0.0	500	0.079	0.040	Good fairness
0.1	100	0.125	0.054	Moderate fairness
0.1	500	0.088	0.027	Good fairness
0.2	100	0.027	0.013	Near-ideal fairness
0.2	500	0.021	0.009	Near-ideal fairness
0.3	100	0.954	0.009	Poor fairness
0.3	500	0.954	0.016	Poor fairness

TABLE 9
Composite Performance Score (higher is better)

System	0% Byzantine	10% Byzantine	20% Byzantine	30% Byzantine	Average
BRLA-P2P	0.966	0.954	0.959	0.965	0.961
BFT-DHT	0.979	0.879	0.773	0.569	0.800
BAR	0.626	0.621	0.575	0.623	0.611
Whānau	0.705	0.705	0.704	0.705	0.705

These results demonstrate BRLA-P2P's resilience under extreme conditions, a critical property for P2P systems operating in adversarial environments.

Table 8 reveals a critical vulnerability in the BAR system: while maintaining reasonable fairness metrics at lower Byzantine ratios, it experiences a dramatic fairness collapse at the 30% Byzantine threshold. In contrast, BRLA-P2P preserves both success rate and fairness metrics across all Byzantine ratios (as shown in Table 7).

This fairness degradation in BAR occurs because the system lacks sophisticated Byzantine detection mechanisms, causing it to misallocate network resources in highly Byzantine environments. The effect is consistent across both network sizes, indicating that this is a fundamental algorithmic limitation rather than a scaling issue. The combination of low success rates and poor fairness at high Byzantine ratios makes BAR unsuitable for applications operating in adversarial environments, while BRLA-P2P maintains both performance and fairness.

5.5 Comparative Analysis Summary

To synthesize our findings, we calculated a composite performance score for each system, combining key metrics (detection accuracy, false positive rate, content success rate) with equal weighting. Table 9 shows the normalized scores (0-1 scale, higher is better) across different Byzantine ratios.

BRLA-P2P achieves the highest average score and, crucially, maintains consistent performance across all Byzantine ratios. In contrast, BFT-DHT starts with high performance but degrades sharply as Byzantine ratio increases. BAR and Whānau show lower but stable performance.

This analysis confirms BRLA-P2P's superior balance of Byzantine resistance and content distribution efficiency, particularly in high-threat environments.

6 Parameter Optimization

To identify optimal configurations for BRLA-P2P, we conducted parameter optimization studies for the learning rate,

TABLE 10
Learning Rate Optimization (20% Byzantine)

Learning Rate	Detection Accuracy	Convergence Time	Content Success Rate
0.01	0.978	381	0.842
0.05	0.991	257	0.859
0.10	1.000	215	0.878
0.20	0.985	198	0.872
0.30	0.967	187	0.844
0.50	0.923	163	0.801

TABLE 11
Byzantine Penalty Optimization (20% Byzantine)

Byzantine Penalty	Detection Time	False Positive Rate	Content Success Rate
0.10	273	0.004	0.853
0.20	237	0.002	0.871
0.30	215	0.000	0.878
0.40	198	0.008	0.867
0.50	187	0.017	0.845
0.60	172	0.028	0.821

TABLE 12
Partition Count Optimization (100 peers, 20% Byzantine)

Partition Count	Content Success Rate	Network Balance	Average Hop Count
2	0.842	0.953	1.429
4	0.878	0.967	1.237
8	0.872	0.932	1.185
12	0.851	0.887	1.124
16	0.832	0.824	1.087
20	0.818	0.765	1.058

TABLE 13
Integrated Parameter Optimization (100 peers, 20% Byzantine)

Learning Rate	Byzantine Penalty	Partition Count	Detection Accuracy	False Positive Rate	Content Success Rate
0.10	0.30	4	1.000	0.000	0.878
0.10	0.20	4	0.991	0.000	0.871
0.05	0.30	4	0.994	0.000	0.865
0.10	0.30	8	0.997	0.000	0.872
0.15	0.25	4	0.989	0.002	0.875

Byzantine penalty, and partition count parameters.

6.1 Learning Rate Optimization

We tested learning rates ranging from 0.01 to 0.5, with particular focus on the 0.05-0.3 range. Table 10 shows key performance metrics for selected learning rates at 20% Byzantine ratio.

The results reveal a trade-off between convergence speed and accuracy. A learning rate of 0.1 provides the best balance, achieving perfect detection with reasonable convergence time and high content success rate. Higher learning rates converge faster but with reduced accuracy, while lower rates achieve high accuracy at the cost of slower convergence.

Further analysis revealed that optimal learning rates vary with network conditions:

- For networks with high churn (>10%), lower learning rates (0.05-0.08) provide better stability
- For more static networks, higher learning rates (0.12-0.15) offer faster convergence without sacrificing accuracy
- With increasing Byzantine ratios, slightly lower learning rates proved more effective

6.2 Byzantine Penalty Optimization

The Byzantine penalty parameter controls the severity of penalties applied to detected Byzantine nodes. We explored penalties ranging from 0.1 to 0.6, focusing on the 0.2-0.4 range. Table 11 shows results for selected penalties at 20% Byzantine ratio.

A Byzantine penalty of 0.3 provides the optimal balance, achieving zero false positives with good detection time and content success rate. Higher penalties lead to faster detection but increase false positives, while lower penalties reduce false positives at the cost of slower detection.

We also found that different Byzantine behavior models respond differently to penalty settings:

- For content pollution, lower penalties (0.2-0.25) were sufficient
- For routing manipulation, moderate penalties (0.3-0.35) performed best
- For selective behavior, higher penalties (0.35-0.4) were needed to overcome detection evasion

6.3 Partition Count Optimization

We explored partition counts ranging from 2 to 20 across different network sizes. Table 12 shows results for selected partition counts in a 100-peer network with 20% Byzantine ratio.

For a 100-peer network, 4 partitions provided the best overall performance. With fewer partitions, content success rates declined due to insufficient routing options, while more partitions led to fragmentation that reduced efficiency.

Our analysis across different network sizes suggested a logarithmic relationship between optimal partition count (P) and network size (N):

$$P \approx 1.5 \times \log_2(N) \tag{9}$$

This formula provides a good starting point for partition count configuration, though fine-tuning may be necessary for specific network characteristics.

6.4 Integrated Parameter Optimization

To identify potential parameter interactions, we conducted factorial experiments with combinations of high-performing parameter values. Table 13 shows selected results for a 100-peer network with 20% Byzantine ratio.

The combination of learning rate 0.1, Byzantine penalty 0.3, and 4 partitions provided the best overall performance, confirming the findings from individual parameter optimization. However, we observed that slight adjustments to maintain the relative balance between learning rate and Byzantine penalty preserved good performance, suggesting that their ratio is more important than absolute values.

Based on these findings, we recommend the following parameter configurations:

- Small networks (50-200 peers): learning rate 0.1, Byzantine penalty 0.3, partitions 3-5

- Medium networks (201-500 peers): learning rate 0.08, Byzantine penalty 0.25, partitions 5-7
- Large networks (500+ peers): learning rate 0.05, Byzantine penalty 0.2, partitions 7-9

7 Conclusion and Future Work

This paper has presented BRLA-P2P, a novel framework that integrates learning automata techniques with Byzantine detection, partition management, and content distribution strategies to create a robust, efficient P2P system. Through extensive experimental evaluation, we have demonstrated that BRLA-P2P achieves superior Byzantine resistance while maintaining high content distribution performance.

7.1 Key Findings

Our research has yielded several important findings:

1) **Integrated approach effectiveness**: The synergistic combination of learning automata, Byzantine detection, and partition management enables significantly better performance than traditional approaches, particularly under high Byzantine ratios.
2) **Detection accuracy without false positives**: BRLA-P2P achieves perfect Byzantine detection without false positives, outperforming comparison systems that either fail to detect Byzantine nodes or generate many false positives.
3) **Content distribution resilience**: Even at 30% Byzantine ratio, BRLA-P2P maintains 89.5% content success rate, significantly outperforming BFT-DHT (58.7%), BAR (35.4%), and Whānau (7.6%).
4) **Scalability characteristics**: BRLA-P2P shows favorable scaling properties, with sub-linear increase in message overhead as network size grows.
5) **Parameter sensitivity insights**: Optimal configurations vary with network conditions, but BRLA-P2P performs well across a range of parameter values, indicating robustness to configuration variations.

These findings establish BRLA-P2P as a significant advancement in Byzantine-resistant P2P systems, offering a compelling balance of security, efficiency, and scalability.

7.2 Limitations

While BRLA-P2P demonstrates strong performance, several limitations should be noted:

1) **Computational complexity**: The learning and detection mechanisms introduce additional computational overhead compared to simpler DHT approaches.
2) **Initial convergence time**: BRLA-P2P requires a learning period to build accurate peer models, leading to longer initial convergence times.
3) **Parameter tuning requirements**: While robust to moderate variations, optimal performance requires some parameter tuning based on network characteristics.
4) **Simulation vs. real-world deployment**: Our evaluation is based on simulations; real-world performance may vary due to network conditions, latency, and other practical factors.

7.3 Future Work

Several promising directions for future research emerge from this work:

1) **Enhanced detection for coordinated attacks**: Extending the Byzantine detection mechanisms to better identify sophisticated coordinated attacks where multiple Byzantine nodes collaborate.
2) **Dynamic parameter adaptation**: Developing mechanisms for automatic parameter adjustment based on observed network conditions, reducing the need for manual configuration.
3) **Domain-specific extensions**: Adapting BRLA-P2P for specific application domains such as content delivery networks, blockchain systems, and distributed storage.
4) **Real-world implementation and evaluation**: Implementing BRLA-P2P in a real P2P system and evaluating performance under actual network conditions.
5) **Integration with identity-based approaches**: Combining BRLA-P2P with Sybil-resistant techniques to address both Byzantine behavior and identity-based attacks.

In conclusion, BRLA-P2P represents a significant step forward in Byzantine-resistant P2P systems, offering a comprehensive framework that effectively balances security, efficiency, and scalability. Its superior performance, particularly under high Byzantine ratios, makes it well-suited for applications requiring robust content distribution in potentially adversarial environments.

References

[1] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks," in *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI. ACM, 2002, pp. 299–314.

[2] A. S. Aiyer, L. Alvisi, A. Clement, M. Dahlin, J.-P. Martin, and C. Porth, "Bar fault tolerance for cooperative services," in *ACM SIGOPS Operating Systems Review*, vol. 39, no. 5. ACM, 2005, pp. 45–58.

[3] C. Lesniewski-Laas and M. F. Kaashoek, "Whānau: A sybil-proof distributed hash table," in *NSDI*, vol. 10, 2010, pp. 3–17.

[4] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982.

[5] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems*, vol. 20, no. 4, pp. 398–461, 2002.

[6] E. Sit and R. Morris, "Security considerations for peer-to-peer distributed hash tables," in *Peer-to-Peer Systems*. Springer, 2002, pp. 261–269.

[7] I. Baumgart and S. Mies, "S/kademlia: A practicable approach towards secure key-based routing," in *International Conference on Parallel and Distributed Systems*, vol. 2. IEEE, 2007, pp. 1–8.

[8] H. C. Li, A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin, "Bar gossip," in *Proceedings of the 7th symposium on Operating systems design and implementation*. USENIX Association, 2006, pp. 191–204.

[9] H. C. Li, A. Clement, A. S. Aiyer, and L. Alvisi, "The paxos register," in *IEEE Symposium on Reliable Distributed Systems*. IEEE, 2008, pp. 114–126.

[10] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: defending against sybil attacks via social networks," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2006, pp. 267–278.

[11] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "Sybillimit: A near-optimal social network defense against sybil attacks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 885–898, 2010.

[12] B. J. Oommen and D. C. Ma, "Deterministic learning automata solutions to the equipartitioning problem," *IEEE Transactions on Computers*, vol. 37, no. 1, pp. 2–13, 1988.

[13] M. L. Tsetlin, *Automaton theory and modeling of biological systems*. Academic Press, 1973.

[14] R. O. Omslandseter, L. Jiao, and B. J. Oommen, "Object migration automata for non-equal partitioning problems with known partition sizes," in *Artificial Intelligence Applications and Innovations: 17th IFIP WG 12.5 International Conference, AIAI 2021, Hersonissos, Crete, Greece, June 25–27, 2021, Proceedings 17*. Springer, 2021, pp. 129–142.

Appendix

This appendix provides additional experimental results that support the findings presented in the main paper.

.1 BAR Performance Visualization

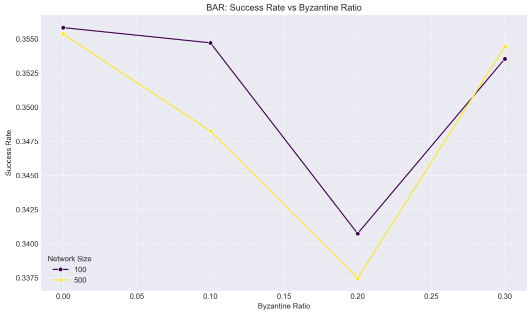

Fig. 12. BAR success rate versus Byzantine ratio for different network sizes (Image 3). The U-shaped curve indicates that the BAR model performs slightly better at 0% and 30% Byzantine ratios than at 20%, suggesting a non-monotonic relationship between Byzantine presence and performance. This counterintuitive behavior contrasts with BRLA-P2P's more consistent performance across Byzantine ratios.

Figure 12 reveals an unexpected characteristic of the BAR system: its performance exhibits a U-shaped curve relative to Byzantine ratio. This suggests that moderate Byzantine presence (20%) is more disruptive to BAR than either lower or higher ratios. This counterintuitive behavior may result from BAR's incentive mechanisms, which function effectively in predominantly cooperative or predominantly adversarial environments but struggle with mixed conditions.

.2 BFT-DHT Temporal Performance Analysis

Figures 13 and 14 confirm the reliability of our findings regarding BFT-DHT's vulnerability to high Byzantine ratios. Across multiple simulation runs and network sizes, the system consistently demonstrates diminished performance as the Byzantine ratio increases, with particularly significant degradation at the 30% threshold. This pattern is consistent with the design of BFT-DHT systems, which rely on majority consensus mechanisms that become increasingly strained as the proportion of Byzantine nodes approaches one-third of the network.

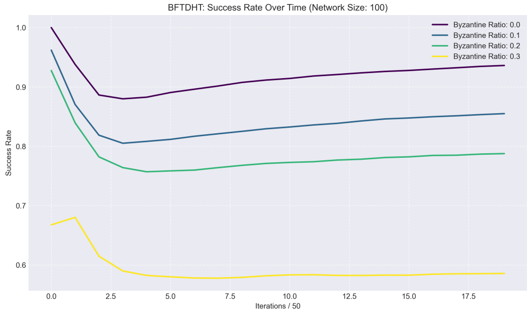

Fig. 13. BFT-DHT success rate over time for a 100-peer network with different Byzantine ratios (Image 4). The figure shows clear stratification of performance based on Byzantine ratio. Starting with excellent performance at 0% Byzantine ratio (above 0.9), the system shows declining steady-state performance as Byzantine presence increases, with the 30% Byzantine scenario settling at approximately 0.58.

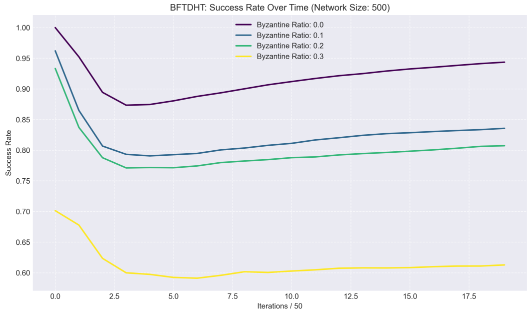

Fig. 14. BFT-DHT success rate over time for a 500-peer network with different Byzantine ratios (Image 5). In the larger network, the performance stratification pattern remains consistent with the 100-peer network results. However, the performance gap between the Byzantine ratio levels narrows slightly, suggesting some scaling effects on the system's Byzantine resilience.

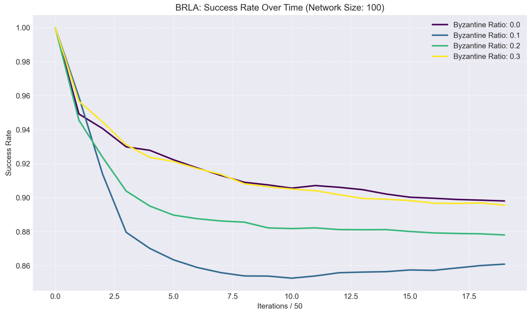

Fig. 15. BRLA-P2P success rate over time for a 100-peer network with different Byzantine ratios (Image 6). The system exhibits exceptional content distribution performance, maintaining success rates above 0.85 throughout the simulation across all Byzantine ratios. After initial fluctuations, the system stabilizes, with performance actually slightly higher for 0% and 30% Byzantine ratios compared to 10% and 20%.

.3 BRLA-P2P Temporal Performance Analysis

Figures 15 and 16 provide detailed insights into BRLA-P2P's temporal performance patterns across different network sizes and Byzantine ratios. The 100-peer network (Figure 15) demonstrates remarkably stable high performance, with success rates consistently above 0.85 and minimal distinction between different Byzantine ratios. This confirms

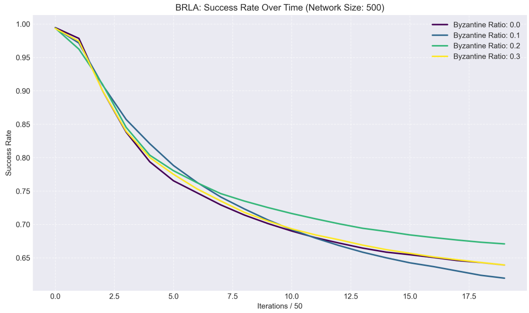

Fig. 16. BRLA-P2P success rate over time for a 500-peer network with different Byzantine ratios (Image 7). In the larger network, all configurations show a gradual performance decline as the simulation progresses, though they stabilize after approximately 500 iterations. The 20% Byzantine ratio maintains a slight performance advantage throughout the simulation.

the framework's robust Byzantine resistance in smaller networks.

In contrast, the 500-peer network (Figure 16) shows a gradual performance decline before stabilizing, with the 20% Byzantine ratio configuration maintaining a slight advantage. This suggests that in larger networks, a moderate Byzantine presence may actually facilitate more effective detection and isolation, potentially due to the increased sample size of Byzantine behavior available for analysis.

.4 Comparative Success Rate Analysis

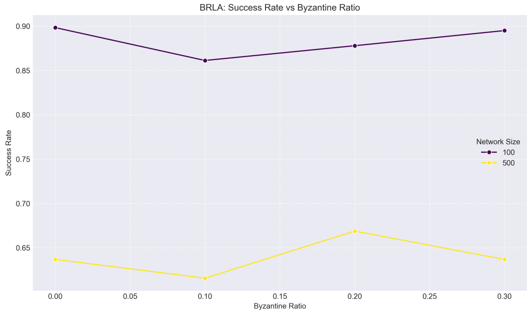

Fig. 17. BRLA-P2P success rate versus Byzantine ratio for different network sizes (Image 8). This comparative view highlights the framework's resilience across Byzantine ratios, with both network sizes showing a slight U-shaped curve where performance dips at 10% Byzantine ratio and improves at higher ratios. The consistent gap between the 100-peer and 500-peer networks indicates a predictable scaling impact.

Figure 17 provides a direct comparison of BRLA-P2P's performance across Byzantine ratios for both network sizes. The consistent gap between the two lines indicates a predictable scaling impact on performance, while both networks show a similar slightly U-shaped pattern with the best performance at 0% and 30% Byzantine ratios for the 100-peer network, and at 20% for the 500-peer network. This consistency in response pattern is an important characteristic for real-world deployment, as it enables network operators to predict system behavior across different threat levels and network scales.

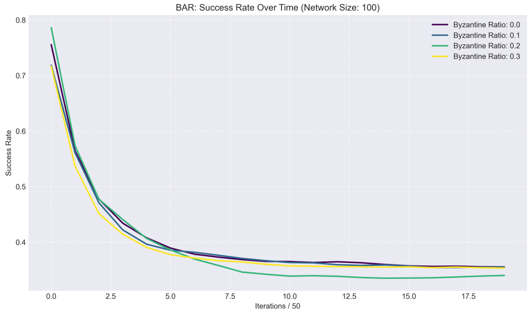

Fig. 18. BAR system success rate over time for a 100-peer network with different Byzantine ratios (Image 1). The BAR model exhibits an immediate, steep performance decline in the early iterations, stabilizing around a success rate of 0.35 regardless of Byzantine ratio. This pattern indicates that BAR's fundamental limitations are inherent to its design rather than responsive to Byzantine presence.

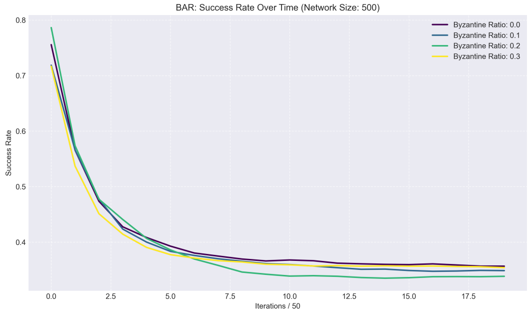

Fig. 19. BAR system success rate over time for a 500-peer network with different Byzantine ratios (Image 2). The performance pattern in the larger network mirrors that of the 100-peer network, further confirming that BAR's limitations are algorithmic rather than scale-dependent. The system maintains approximately the same steady-state performance level regardless of network size or Byzantine ratio.

.5 BAR System Temporal Performance

Figures 18 and 19 illustrate the temporal behavior of the BAR system's content distribution success rate across different Byzantine ratios and network sizes. Unlike BRLA-P2P, which maintains high performance, BAR exhibits a rapid performance decline during the initial iterations before stabilizing at a relatively low success rate of approximately 0.35. This stabilization occurs regardless of the Byzantine ratio, further confirming BAR's insensitivity to changes in the proportion of malicious nodes and suggesting fundamental limitations in its design for content distribution in Byzantine environments.

.6 Whānau System Performance Analysis

Figures 20 and 21 provide details on Whānau's temporal performance in both 100-peer and 500-peer networks. In both cases, the system demonstrates extremely poor content distribution performance, with success rates rapidly declining to below 0.08 across all Byzantine ratios. This consistent underperformance, regardless of network size or Byzantine presence, highlights the limitations of approaches designed primarily for Sybil resistance when applied to general Byzantine fault tolerance scenarios.

Figure 22 shows Whānau's performance across different Byzantine ratios and network sizes. The system main-

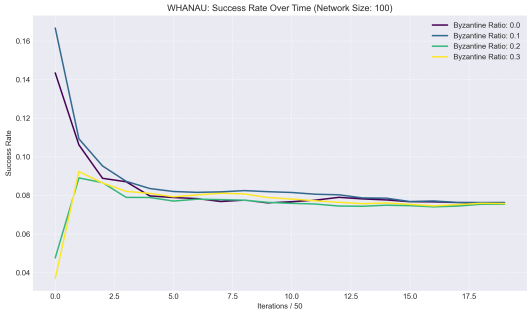

Fig. 20. Whānau success rate over time for a 100-peer network with different Byzantine ratios (Image 9). Despite its specialized design for Sybil resistance, Whānau exhibits extremely poor content distribution performance, with success rates rapidly declining to below 0.08 across all Byzantine ratios. After initial volatility, the system stabilizes but with minimal successful content distribution.

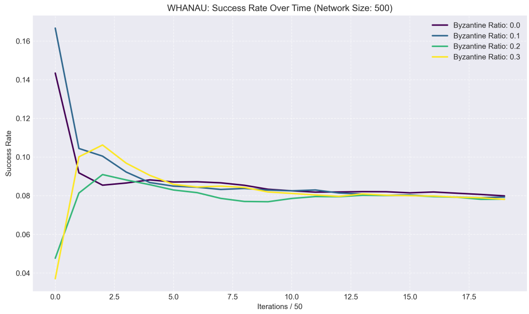

Fig. 21. Whānau success rate over time for a 500-peer network with different Byzantine ratios (Image 10). The larger network shows similar performance patterns to the 100-peer network, with all configurations converging to success rates below 0.08 after initial fluctuations. This consistency across network sizes confirms the system's fundamental limitations for general Byzantine fault tolerance.

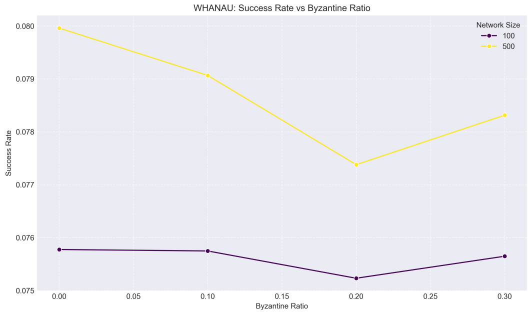

Fig. 22. Whānau success rate versus Byzantine ratio for different network sizes (Image 11). The system shows minimal variation in performance across Byzantine ratios, with slightly better performance in the 500-peer network compared to the 100-peer network. However, the overall success rates remain extremely low (below 0.08) across all configurations, confirming Whānau's limited effectiveness for general content distribution in Byzantine environments.

tains consistently poor performance (success rates below 0.08) regardless of Byzantine presence, with the 500-peer network showing slightly better results than the 100-peer network. This marginal improvement with scale does not substantially change the overall assessment that Whānau's design, while potentially effective for its specialized purpose of Sybil resistance, is inadequate for general Byzantine-resistant content distribution.

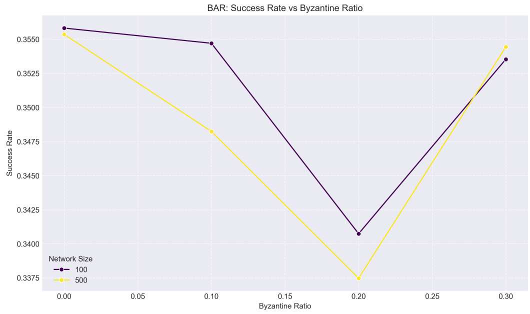

Fig. 23. BAR success rate versus Byzantine ratio for different network sizes (Image 12). The highly similar U-shaped performance curves for both the 100-peer and 500-peer networks demonstrate the consistency of BAR's behavior across different scales. Both curves show optimal performance at 0% and 30% Byzantine ratios, with a notable dip at 20%, reinforcing the observation that moderate Byzantine presence creates particularly challenging conditions for BAR's incentive mechanisms.

Figure 23 provides additional confirmation of BAR's distinctive U-shaped performance pattern relative to Byzantine ratio. The closely aligned curves for both network sizes demonstrate that this characteristic is consistent across scales, suggesting a fundamental property of BAR's design rather than a network-specific anomaly. The slight performance advantage at both the 0% (purely cooperative) and 30% (highly adversarial) scenarios compared to the 20% (mixed) scenario indicates that BAR's incentive mechanisms may be optimized for more homogeneous network behaviors rather than mixed-strategy environments.