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

Ekaba Bisong
University of Victoria
ebisong@siliconblast.com

This research update focuses on two critical components of our work: (1) the refined System Model and Problem Formulation that captures the dynamic behavior of P2P networks under Byzantine threats, and (2) the enhanced BRLA-P2P Framework, which integrates advanced Learning Automata with robust Byzantine detection and dynamic partition management. Together, these updates aim to improve both the efficiency and security of content distribution in large-scale peer-to-peer environments.

Contents

1. System Model and Problem Formulation

In our updated model, the peer-to-peer (P2P) network is represented as a dynamic graph

$$G(t) = (V(t), E(t)),$$

where $V(t)$ denotes the set of peers and $E(t)$ the set of connections between them at time t. Each peer $p_i \in V(t)$ is characterized by the tuple:

$$p_i = \{\tau_i, \kappa_i, \rho_i, \sigma_i\},$$

with:

- τ_i: Storage capacity,
- κ_i: Bandwidth capacity,
- ρ_i: Processing capability,
- σ_i: Current state in the Learning Automata.

Network dynamics are influenced by:

- λ_c: Peer churn rate (rate of joining/leaving),
- λ_r: Content request rate.

Peers are organized into K partitions, $\Pi = \{\pi_1, \pi_2, \ldots, \pi_K\}$, each adhering to a size constraint:

$$|\pi_k| = \rho_k \quad \text{with} \quad \sum_{k=1}^{K} \rho_k = |V(t)|.$$

1.1. Threat Model

The model considers Byzantine nodes—malicious or colluding peers—denoted by $B \subset V(t)$ with $|B| \leq f|V(t)|$, where f is the maximum fraction tolerated. Byzantine nodes may engage in:

1. Content pollution,
2. Routing table manipulation,
3. Collusion and Sybil attacks.

They are, however, limited by:

1. Inability to break cryptographic primitives,
2. Inability to forge valid signatures,
3. Control of no more than a fraction f of the network.

1.2. Problem Statement

Our goal is to maximize the efficiency of content distribution while ensuring Byzantine resilience. This is formalized as:

$$\max_{P \in \Pi} \sum_{k=1}^{K} \eta_k(P)$$

subject to:

$$\begin{aligned}
&|\pi_k| = \rho_k, \quad \forall k, \\
&\sum_{k=1}^{K} \rho_k = |V(t)|, \\
&P(B_{detect}) \geq \theta_{detect}, \\
&P(F_{positive}) \leq \theta_{false}, \\
&L_{overhead} \leq \theta_{overhead}.
\end{aligned}$$

Here, $\eta_k(P)$ is the efficiency metric for partition k (a function of content availability, request satisfaction, and stability), while the constraints ensure appropriate partition sizes, high detection accuracy, and limited communication overhead.

2. BRLA-P2P Framework

The BRLA-P2P framework builds upon our system model by integrating an enhanced Learning Automata (LA) module with mechanisms for robust Byzantine detection and dynamic partition management.

2.1. Learning Automata Structure

At the core, the LA component is defined as:

$$LA = \{S, \alpha, \beta, p, T\},$$

where:

- $S = \{s_1, s_2, \ldots, s_m\}$ is the set of states,
- $\alpha = \{a_1, a_2, \ldots, a_n\}$ is the action set,
- β is the feedback function,
- $p : S \times \alpha \to [0, 1]$ defines state transition probabilities,
- T is a temperature parameter regulating exploration versus exploitation.

The state transition function is modified to incorporate Byzantine penalties:

$$T(s_i, a_j, o) = \begin{cases} s_i + \alpha(1 - s_i), & \text{if } o = \text{SUCCESS}, \\ s_i(1 - \beta), & \text{if } o = \text{FAILURE}, \\ s_i - \delta, & \text{if Byzantine}(p_i), \end{cases}$$

with δ serving as the penalty factor when Byzantine behavior is detected.

2.2. Byzantine Detection Mechanism

The framework deploys a multi-faceted Byzantine detection method:

- **Reputation Scoring:** Each peer i is assigned a reputation score:

  $$Rs(i) = \sum_{k=1}^{n} w_k I_k,$$

  where w_k are weights for different interaction types and I_k are the corresponding outcomes.

- **Behavioral Analysis:** A behavior profile $BA(p_i)$ is maintained:

  $$BA(p_i) = \{\lambda_i, \pi_i, \nu_i\},$$

  where λ_i is the interaction frequency, π_i the pattern similarity score, and ν_i the content validation rate.

A detection decision is then made as:

$$D(p_i) = \begin{cases} 1, & \text{if } Rs(i) < \theta_r \text{ or } BA(p_i) < \theta_b, \\ 0, & \text{otherwise.} \end{cases}$$

2.3. Dynamic Partition Management

To maintain balanced partitions, the framework employs a Partition Size Required (PSR) approach. Peer adjustments are governed by:

$$Adjust(\pi_i, \pi_j) = \begin{cases} \text{swap}(p_i, p_j), & \text{if } Valid(\pi_i, \pi_j), \\ \text{isolate}(p_i), & \text{if Byzantine}(p_i), \\ \text{maintain}(\pi_i), & \text{otherwise.} \end{cases}$$

This mechanism ensures that Byzantine nodes are promptly isolated and that partition size constraints remain satisfied.

2.4. Enhanced Learning Features

Additional features of the framework include:

- **Pursuit Learning:** The action probability vector is updated as:

  $$p(t+1) = p(t) + \lambda \left[e(t) - p(t)\right],$$

  where λ is the learning rate and $e(t)$ is the target distribution.

- **Transitivity Enhancement:** For any peers p_i, p_j, and p_k, trust is propagated via:

  $$\text{if } \; Trust(p_i, p_j) \wedge Trust(p_j, p_k) \text{ then } Trust(p_i, p_k) = \min\{Trust(p_i, p_j),\, Trust(p_j, p_k)\}.$$

Algorithm 1 BRLA-P2P Learning Update

```
procedure UpdateLearning($peer_i, peer_j, outcome$)
    $R_{ij} \leftarrow$ current reputation score of $peer_i$
    if $outcome$ = SUCCESS then
        $R_{ij} \leftarrow R_{ij} + \alpha(1 - R_{ij})$
    else
        $R_{ij} \leftarrow R_{ij}(1 - \beta)$
    end if
    if DetectByzantine($peer_i$) then
        TriggerIsolation($peer_i$)
    end if
    UpdateTransitivity($peer_i, peer_j$)
end procedure
```

An outline of the learning update process is provided in Algorithm 1.

This integrated framework facilitates rapid convergence, maintains partition integrity, and provides robust defense against Byzantine attacks, thereby ensuring efficient and secure content distribution across dynamic P2P networks.

3. Recent Progress and Future Work

In our research, we have made significant strides in developing the BRLA-P2P framework. Recent progress includes the formulation of a comprehensive system model and problem formulation, as well as the integration of advanced learning automata techniques with robust Byzantine detection and dynamic partition management.

Future work will consist of:

- **Implementation:** Developing a full-scale implementation of the BRLA-P2P framework, including all modules such as the learning engine, Byzantine detection module, partition manager, and content distribution handler.
- **Experimental Evaluation:** Conducting extensive experiments to evaluate system performance, scalability, and resilience under various network conditions and Byzantine attack scenarios.
- **Discussion:** Analyzing experimental results in detail, discussing practical implications, and identifying potential bottlenecks or areas for further improvement.
- **Conclusion:** Summarizing findings, highlighting the contributions of the work, and outlining future research directions.