Enhancing Byzantine Fault Tolerance in Blockchain Networks through Dynamic Clustering

Teppei Okada* Noriaki Kamiyama* Akihiro Fujihara†

*Ritsumeikan University

†Chiba Institute of Technology

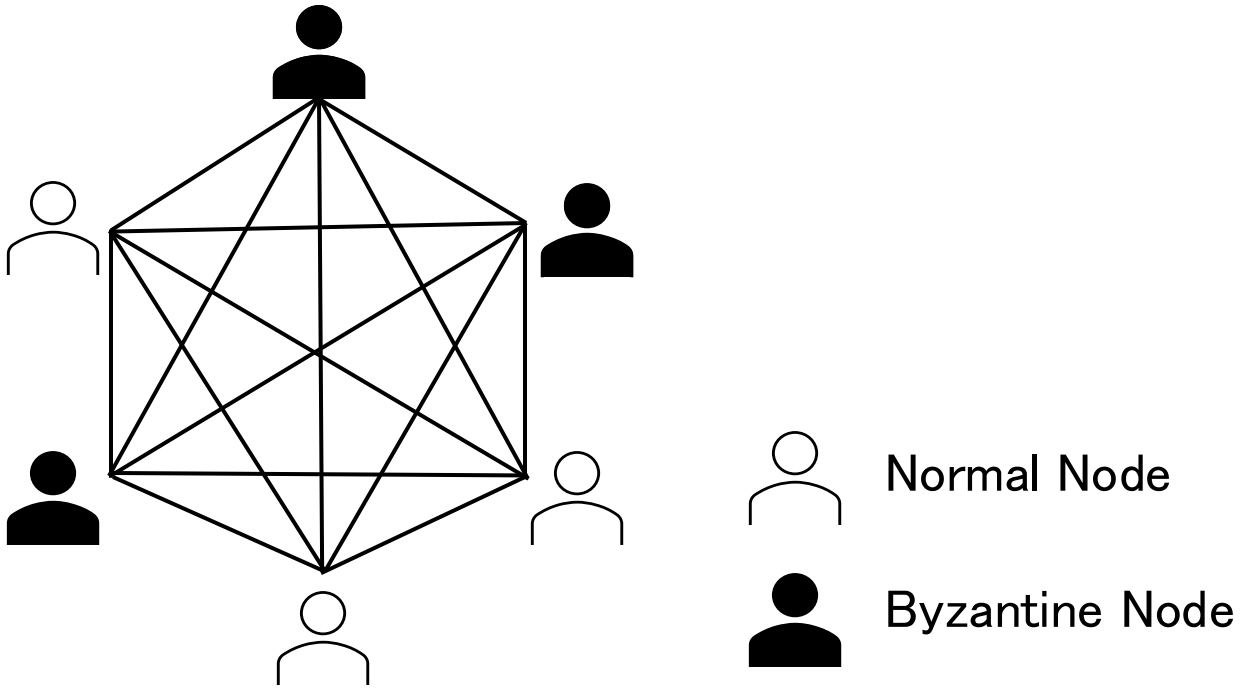
Background (1/2)

- Blockchain
  - Securely share and manage transactions across multiple computers on a ledger, resistant to tampering
  - Public type
    - No administrator
    - ex. Bitcoin, Ethereum
  - Consortium, Private type
    - With administrator
    - ex. Hyperledger
- Consensus Formation
  - Verification by all participants when new blocks or transactions are created
  - Detection and removal of potentially tampered transactions
  - PBFT (Practical Byzantine Fault Tolerance) is primarily used in consortium blockchains

Background (2/2)

- PBFT
  - Algorithm resistant to faulty or attacked nodes (Byzantine nodes)
  - Achieves correct consensus if Byzantine nodes are less than one-third of total

- Drawback
  - Achieving correct consensus becomes difficult if Byzantine nodes exceed one-third



Research Objective

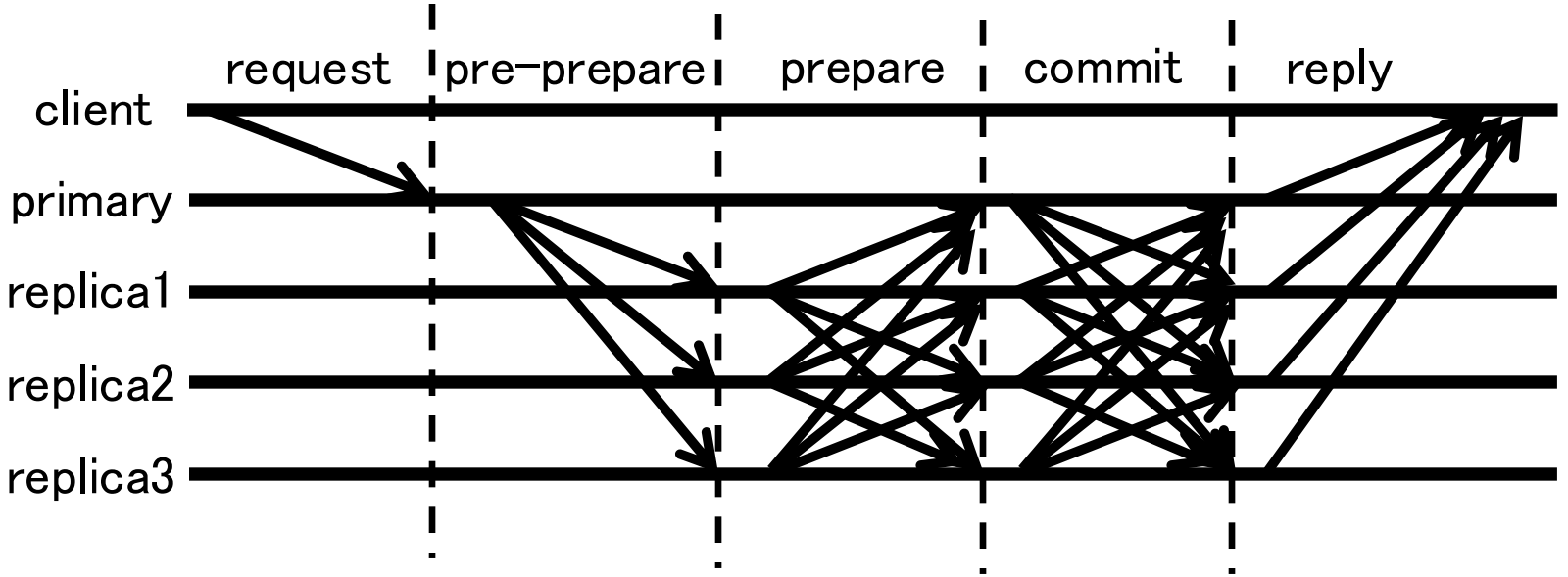
- PBFT
  - Agreement depends on number of Byzantine nodes.

- We propose a method to form clusters with resistance to Byzantine nodes by estimating the attacker's location through clustering

- Approach
  - Divides topology using k-means, forms consensus within and between clusters using PBFT
    - Merge clusters sending minority opinions

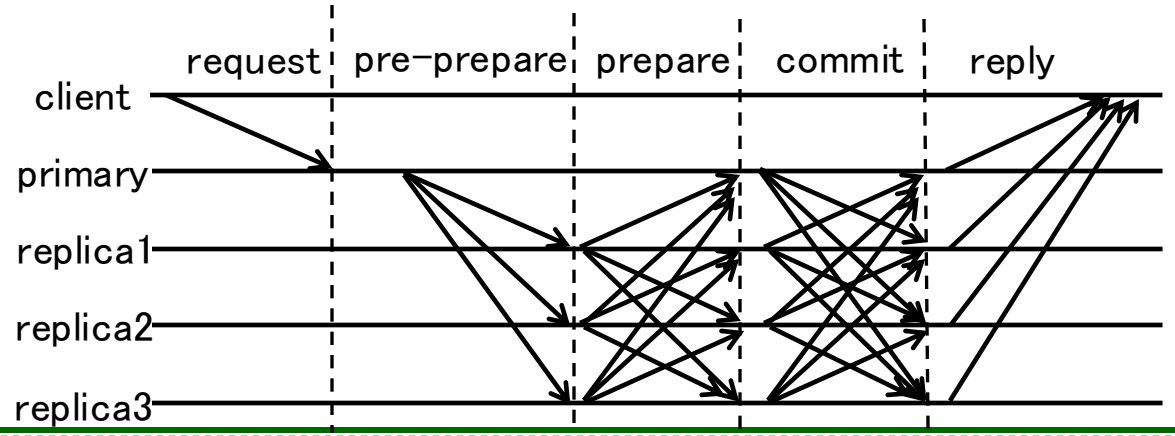
PBFT (1/2)

- Composed of 3 types of entities
  - Client
    - Generates transactions and sends them to the network
  - Replica Node
    - Participates in consensus
  - Primary Node
    - Specific replica node
    - Receives transactions from clients and forwards them to other replica nodes



PBFT (2/2)

- Pre-prepare phase
  - Primary node forwards the transaction to other replica nodes
  - Each node verifies the transaction's validity and broadcasts the result to other nodes
- Prepare phase
  - Replica nodes confirm that the verification result matches the primary node's verification result, and broadcasts to other nodes
- Commit phase
  - If agreeing with the message received in the prepare phase, a commit message is sent to other nodes
- Receiving commit messages from two-thirds of all nodes
  - Send a reply message to the client
    - Confirming that consensus has been successfully reached



Challenges of PBFT

- Consensus formation becomes difficult in high-attacker environments

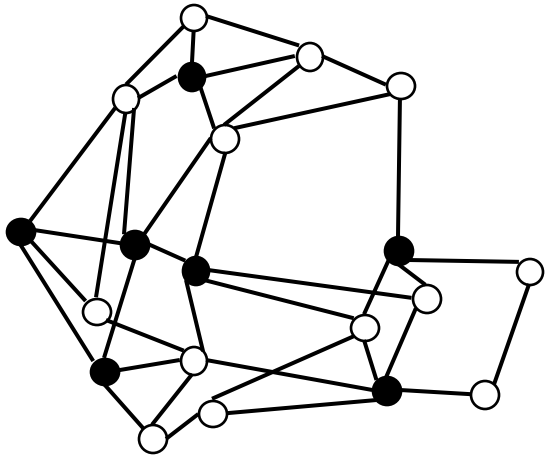
- By utilizing clustering, clusters with a high number of attackers can be identified and merged, reducing the overall proportion of attacker-dominated clusters.
  - Attackers are concentrated into specific clusters
  - As a result, the consensus rate increases

Proposed Method

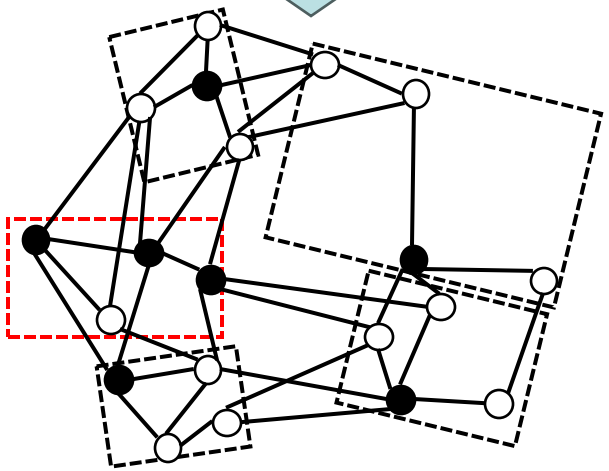
1. Execute PBFT across the entire network
2. If consensus cannot be formed, apply k-means clustering to the network
3. Run PBFT within each cluster
4. Run PBFT between clusters
5. If consensus is not reached, determine that the cluster sending the minority opinion has many attackers
   - Merge minority clusters
6. If consensus formation is successful, share the result with the entire network and terminate

Proposed Method (e.g.)

- Above figure (Before applying proposed method)
  - Out of 20 nodes, 7 nodes are Byzantine nodes
  - Number of Byzantine nodes exceeding one-third of the total, consensus cannot be formed even with conventional PBFT



- Below figure (After applying proposed method)
  - Clustering results in 5 clusters, and 4 clusters have fewer than one-third of Byzantine nodes
  - When PBFT is executed between clusters, consensus is successfully reached

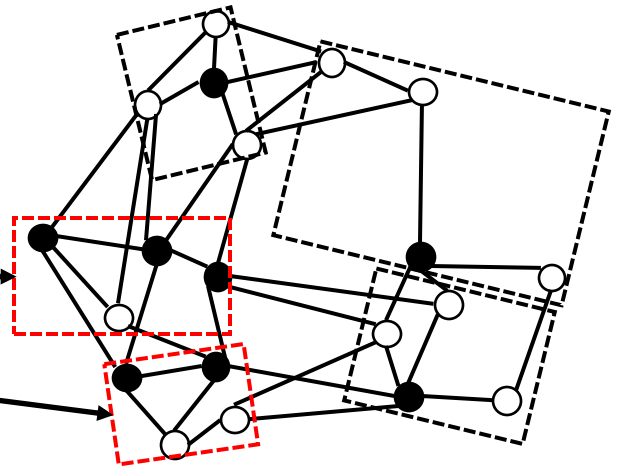


Normal Node

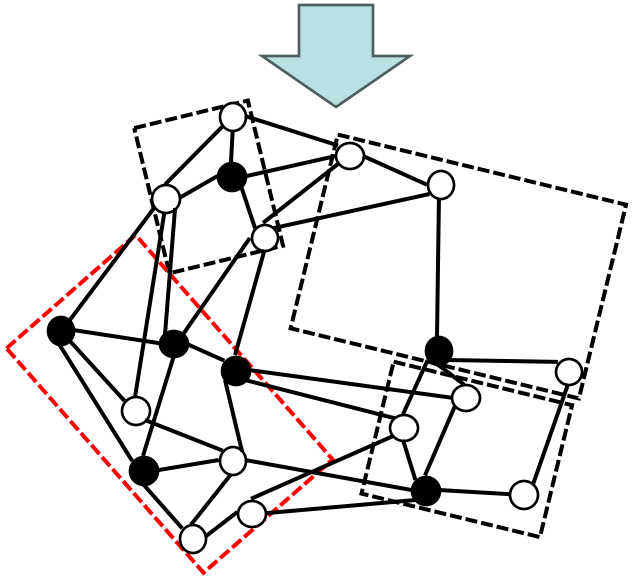
Byzantine Node

Merging Clusters

- Above figure
  - 3 clusters with fewer than one-third Byzantine nodes
    - Consensus cannot be formed
  - Merge the minority clusters



- Below figure
  - Out of 4 clusters, 3 have fewer than one-third Byzantine nodes
  - →Consensus can be successfully formed



○ Normal Node

● Byzantine Node

Performance Evaluation

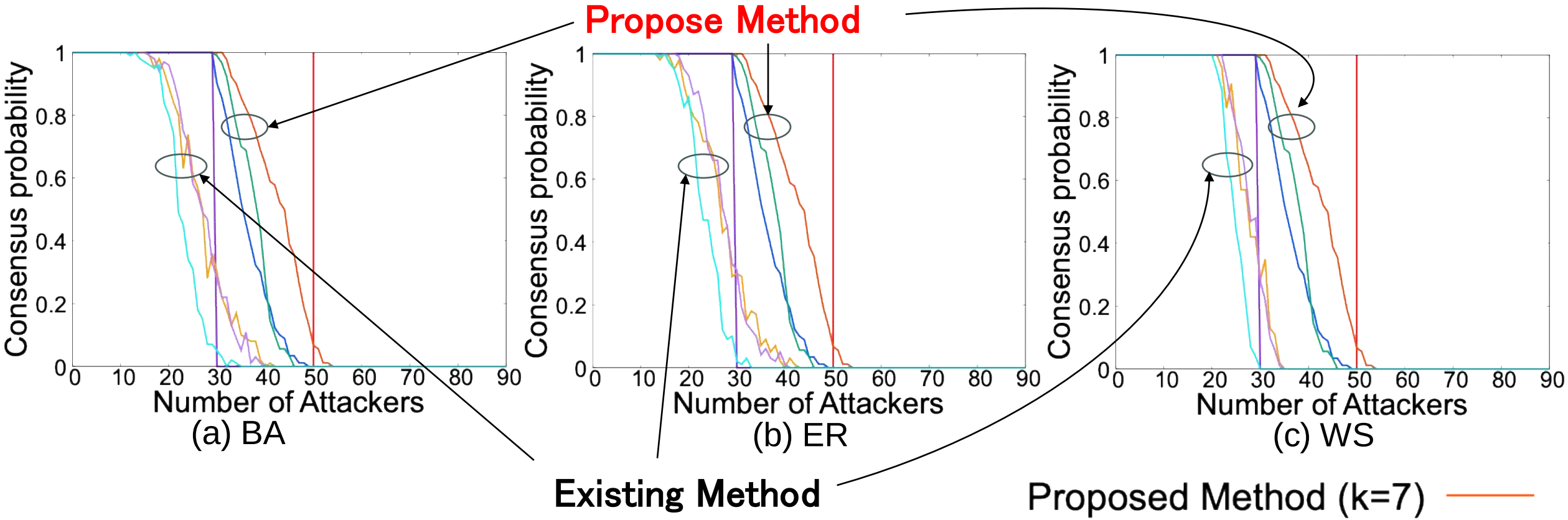
- Evaluated through computer simulation
  - Consensus rate
    - Count the number of times consensus is reached out of 100 trials
  - Traffic volume

Evaluation Conditions

- Number of nodes: 90
  - number of attackers varies from 0 to 90
- Network Topology
  - Barabasi-Albert (BA)
  - Erdos-Renyi (ER)
  - Watts-Strogatz (WS)
- Compare the proposed method with Existing PBFT and PBFT with k-means (k = 7, 10, 15) applied

Consensus Probability

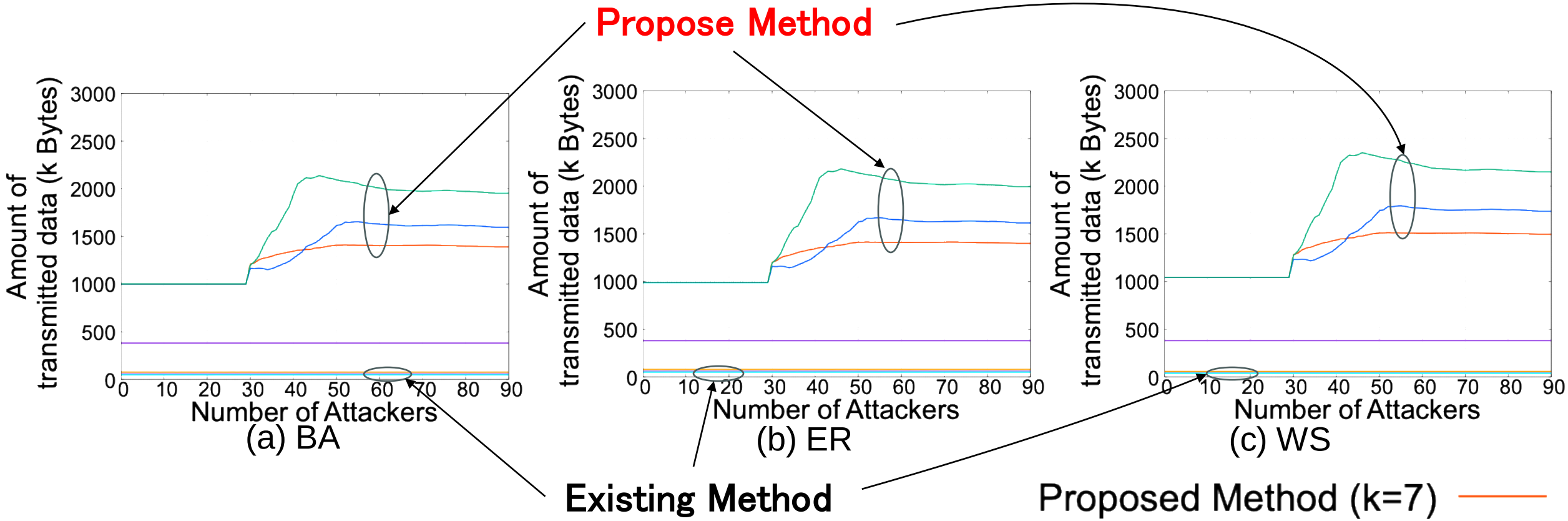
Consensus Probability



- Proposed method achieves a higher consensus rate compared to the existing method
  - Clusters with many attackers are aggregated, increasing the proportion of clusters that produce correct results

Amount of Traffic

Amount of Traffic



- Traffic volume increases compared to the existing method
  - Proposed method: PBFT execution after clustering and the need to broadcast data among nodes to perform the k-means algorithm

Conclusion

- We proposed a method to enhance resistance to Byzantine attacks by utilizing clustering and merging

- In the proposed method, the consensus rate and communication traffic volume were compared, and simulation evaluations confirmed the following:
  - Correct consensus formation even when attackers constitute more than one-third of the total
  - Traffic volume increased significantly compared to the conventional method due to an increase in communication frequency and data size

- Future works
  - Devising methods to reduce traffic volume while maintaining the consensus rate