Atomic Multicast in Asymmetric Byzantine Quorum Systems

Cosmin-Adrian Moldoveanu
Delft University of Technology

1. Background

In distributed systems, a **Byzantine failure** occurs when a node behaves maliciously, sending conflicting data to trick other nodes or to stall the network.

Byzantine Fault Tolerant (BFT) algorithms ensure agreement despite these traitors, but traditionally rely on symmetric trust: a global assumption that all nodes trust the same peers and share a common failure threshold (n > 3f).

Asymmetric Byzantine Quorum Systems (ABQS) is an asymmetric trust model where each node defines their own assumptions for which subsets of nodes can fail and turn against the network.

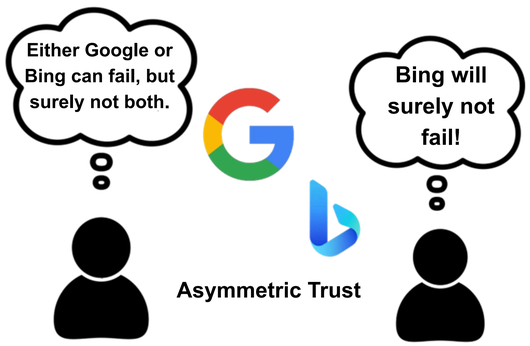
Atomic Multicast is a fundamental primitive of distributed systems that ensures a consistent partial ordering of messages across groups of processes.

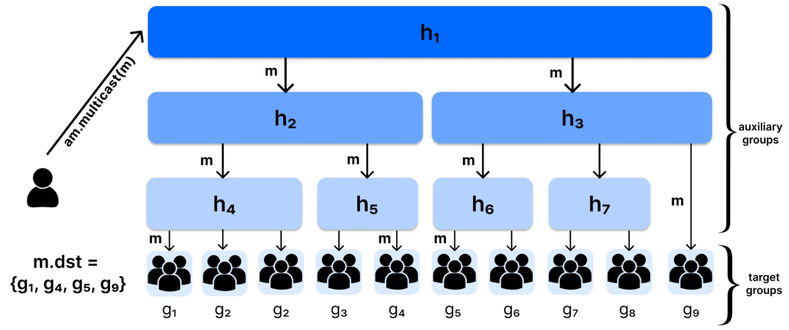
2. Research Question

How can Atomic Multicast be implemented in the ABQS model?

3. Related Work

1. ByzCast, P. Coelho et al. (2018) - first BFT atomic multicast
2. O. Alpos et al. (2024) - formalized ABQS, designed algorithms for strong consensus and reliable broadcast
3. O. Alpos et al. (2019) - designed rules for composing 2 quorum systems into a joint system, without introducing a global trust model
4. Amores-Sesar et al. (2025) - designed a DAG-based consensus and atomic broadcast algorithm for ABQS





4. Our Work

ABQS Atomic Broadcast

We define and implement Atomic Broadcast in the ABQS model by:

- building on ABQS Reliable Broadcast and Strong Consensus abstractions [2]
- reducing Atomic Broadcast to these abstractions using the Chandra-Toueg approach
- proving:
  - consistency for **wise** processes
  - liveness with probability 1 for processes in the **maximal guild**

ABQS Atomic Multicast

We then construct Atomic Multicast from Atomic Broadcast, following the structure of the ByzCast [1] algorithm.

Message propagation

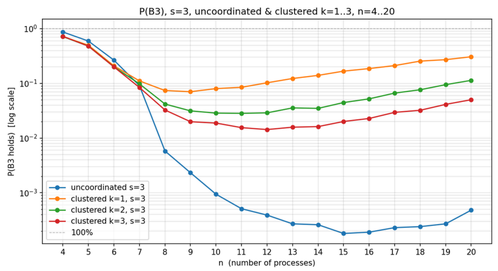
- processes are organized into:
  - disjoint target groups
  - auxiliary groups forming an overlay tree, created by composing [3] children groups' trust assumptions
- a message is first broadcast to the LCA of the destination groups
- nodes in that group broadcast an echo message to child groups
- once process p receives matching echoes from enough processes, it forwards the message further down the tree
- propagation continues until the message reaches all target groups
- target groups can then safely deliver the message

Result: Atomic Broadcast and Multicast can be achieved under asymmetric trust assumptions in an asynchronous network. Chandra-Toueg construction achieves **stronger safety guarantees** for atomic broadcast than existing alternatives [4].

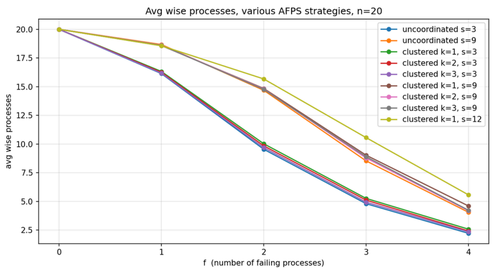
5. Robustness Experiments

We evaluated how randomly generated localized trust choices affect overall network safety (number of secure nodes) and liveness (number of nodes guaranteed to make progress). Nodes either choose trust assumptions completely independently (Uncoordinated) or aligned around a few shared profiles with minor local noise (Clustered).

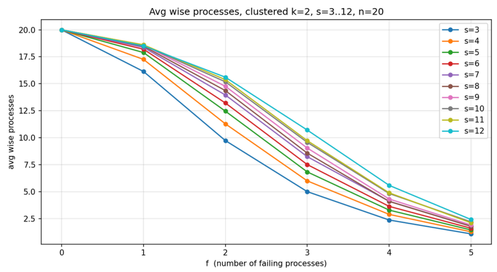
- **Clustering Restores Network Viability**: When nodes choose trust independently, the probability that the network can run the protocol drops exponentially toward zero as network size grows. Clustering fixes this scalability trap; compatibility hits a floor and rebounds because shared alignment neutralizes individual noise (see Figure a).
- **Local Capacity Drives Safety**: As Byzantine faults increase, the number of secure nodes drops roughly linearly (see Figure b). This safety resilience is entirely independent of network layout; instead, sweeping local capacity from 3 to 12 proves that higher local capacity uniformly raises the safety baseline (see Figure c).
- **Coordination Prevents Total Collapse**: Without coordination, the system is much more susceptible to liveness issues. In contrast, clustered trust flattens this failure curve, keeping the system alive even under multiple concurrent faults (see Figure d).



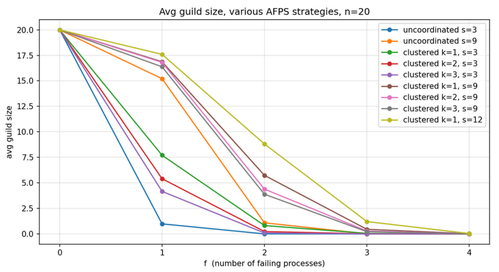
(a)



(b)



(c)



(d)

c.a.moldoveanu@student.tudelft.nl