Blockchain-based Multi-Robot Path Planning

AMR MOKHTAR
NOEL MURPHY
JENNIFER BRUTON
Agenda

- Background
- Problem Statement
- Objective
- Solution & Results
- Discussion & Conclusion
Background
Path Planning for Robots

The field of research that is concerned by finding a global optimum navigational path from a starting to a destination point.

Tasks of Path Planning:

- Environment Modelling
- Motion Planning
- Trajectory Planning
- Obstacle Avoidance
- Motion Coordination
- Communication (Mobile Ad-hoc Networks)
Blockchain Technology & Smart Contracts

Blockchain is the digital asset of Bitcoin, a peer-to-peer cryptocurrency that was invented by Satoshi Nakamoto to eliminate the need for an intermediary financial institution and to protect from double spending.

- **Blockchain as a distributed decision making algorithm:**
  - Users transact via private/public key pairs in a public peer-to-peer network
  - Transactions are cryptographically secured and digitally signed
  - Users reach a consensus on executed transactions through a distributed process (bitcoin: mining)
  - Miners need to provide Proof-of-Work “PoW” in order for a block to get accepted to the distributed ledger
  - PoW allows the goodwill to overpower attackers and intruders

- **Smart contracts are executable scripts that are stored on the blockchain and accordingly executed by all the peers on the network in the endeavor of fulfilling the terms of an agreed contract**
Problem Statement

- IoRT security, privacy and trust issues
- Time-sensitivity of cloud robotics
- IoRT distributed computing challenges
- Robotics' limited compute & power resources
Objective

- Secure and trust-worthy distributed coordination
- Time-sensitivity fulfillment
- Cloud robotics architecture with edge computing in mind
- Enterprise-grade blockchain as a distributed control system
- Smart contracts for distributed decision making
Solution & Results
Hyperledger Fabric Network

- **Multi-Robot test application**: high-level program supervising overall robot client applications
- **Robot client applications**: implemented using Hyperledger Fabric Java SDK
- **One Fabric peer node**: hosting the multi-robot chaincode & own ledger replica
- **SOLO ordering service**: achieves the consensus between peers
- **Certificate Authority (CA)**: issuer of digital certificates of all network nodes
1. Query chaincode for workspace configuration – invoking `getWorkspace()` method
2. Query other robots’ path plans – invoking `getAllPaths()` method invocation.
3. Initiate PRM planning – roadmap buildup & path planning
5. Chaincode executes and a response is generated
6. Transaction goes through endorsement policy
7. Ordering service creates a block and appends to the ledger
8. The new block is broadcasted and `path-committed` event is fired
8 robots are instructed to plan their own paths from a start to a goal point
- Instantiation of robot client applications is in *prioritized order*
- **Consensus** is achieved when all robots reach the state of all paths are collision-free
- Capture path planning, ledger commit and consensus latencies
Consensus vs Path Planning Times

Path planning vs Ledger Commit Latencies

- Total path planning time consumes \( \approx 81.46\% \) of the total consensus time
- Average Ledger Commit Latency \( \approx 112.46 \) ms
- Enterprise-grade blockchain latencies are in the order of 100s of milliseconds as compared to 10 minutes of Bitcoin and 7 seconds of Ethereum
Prioritized vs Interactive Path Planning

For an interactive 8-robot scenario with a path of 18 segments:

- 144 ledger commits
- Consensus time ≈ **19002.99** ms, as compared to ~**27507** ms of the prioritized approach

**How will the interactive approach scale with larger path plans and increased number of robots?**

**Prioritized**

- Blocking
- Less interactions on the ledger
- Must rebuild roadmap with every new path committed

**Interactive**

- Non-blocking
- More interactions on the ledger – impacted by *ledger commit latency*
- Roadmap is computed in parallel and for once
Conclusion

- Secure & trust-worthy distributed control for multi-robotic systems is attainable using enterprise-grade blockchain frameworks

- Smart contracts enable distributed coordination and control for multi-robots

- Latency is in the order of milliseconds for enterprise-grade blockchain-based robotic systems

- A blockchain-based robotic system is suitable for deployment in cloud and edge computing architecture
Thanks!