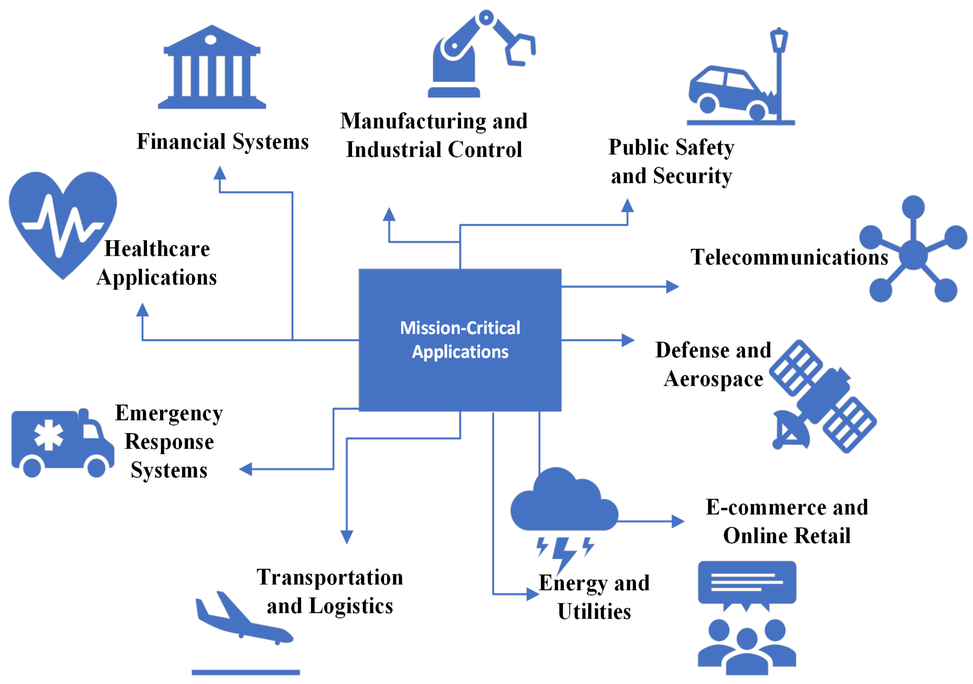
Fault Tolerance for Mission-Critical Edge Applications

Tanaz (Nayereh) Rasouli, Cristian Klein, and Erik Elmroth

Dept of Computing Science, Umeå University
nrasouli@cs.umu.se

Mission-critical applications rely heavily on fault tolerance in IT systems. Natural disasters like earthquakes and hurricanes can damage components or disrupt networks, impacting essential applications and complicating disaster relief efforts. Although prior research has addressed disaster management in Mobile Edge Computing (MEC) systems, integrating robust fault tolerance remains understudied. Our study utilizes contemporary technologies like Kubernetes to manage node failures effectively. We introduce an infrastructure with RabbitMQ as a resilient message broker, ensuring reliable communication even during severe disruptions. Our tailored fault tolerance solution for MEC systems includes a holistic disaster recovery strategy, validated by a case study with weather stations in urban-forest adjacent areas, demonstrating the system's capability to sustain dual node failures and maintain 99.966% availability for mission-critical applications.

What are Mission-Critical Applications?

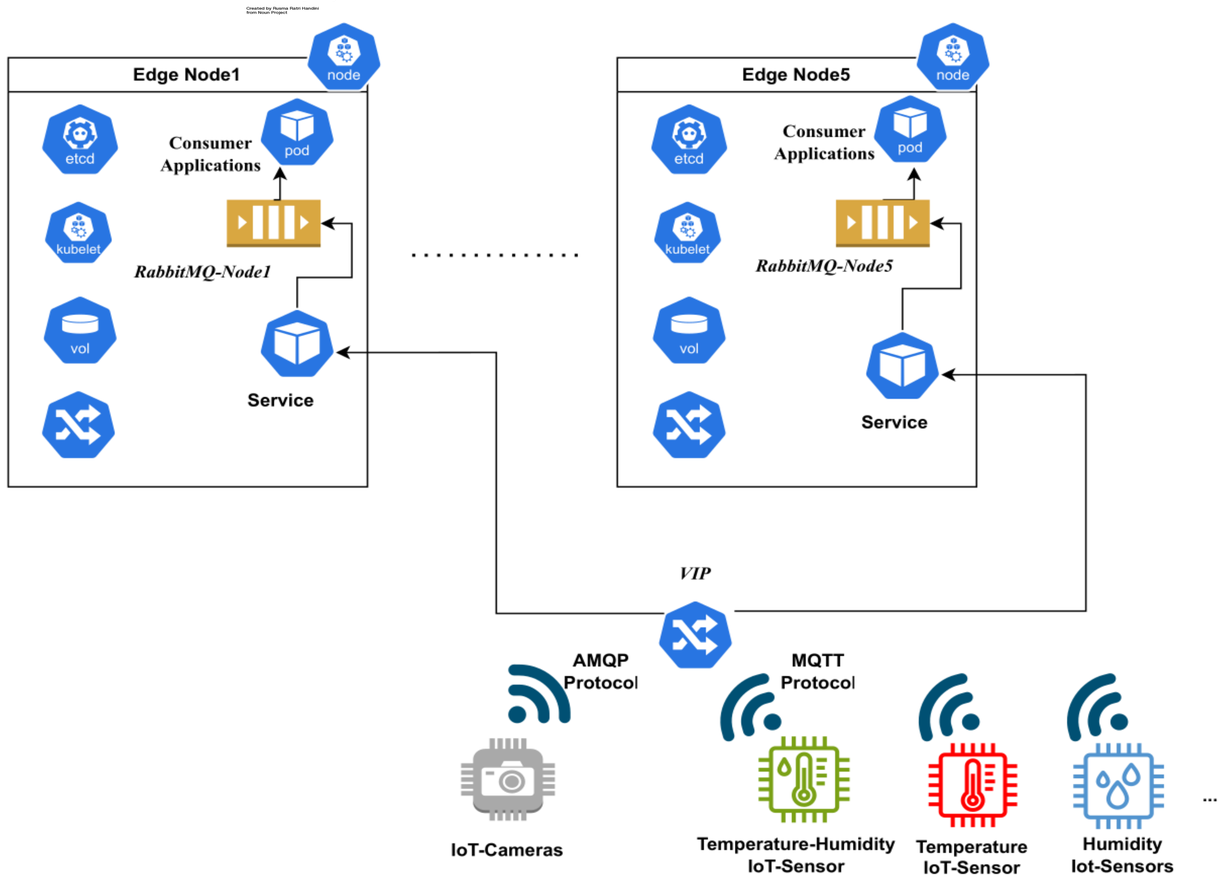


Research Questions

Q1. How can off-the-shelf cloud-native technology be used in edge computing to handle node failures without compromising system integrity?

Q2. What are the impacts of disaster-induced failures on the performance and reliability of mission-critical applications?

Proposed Infrastructure

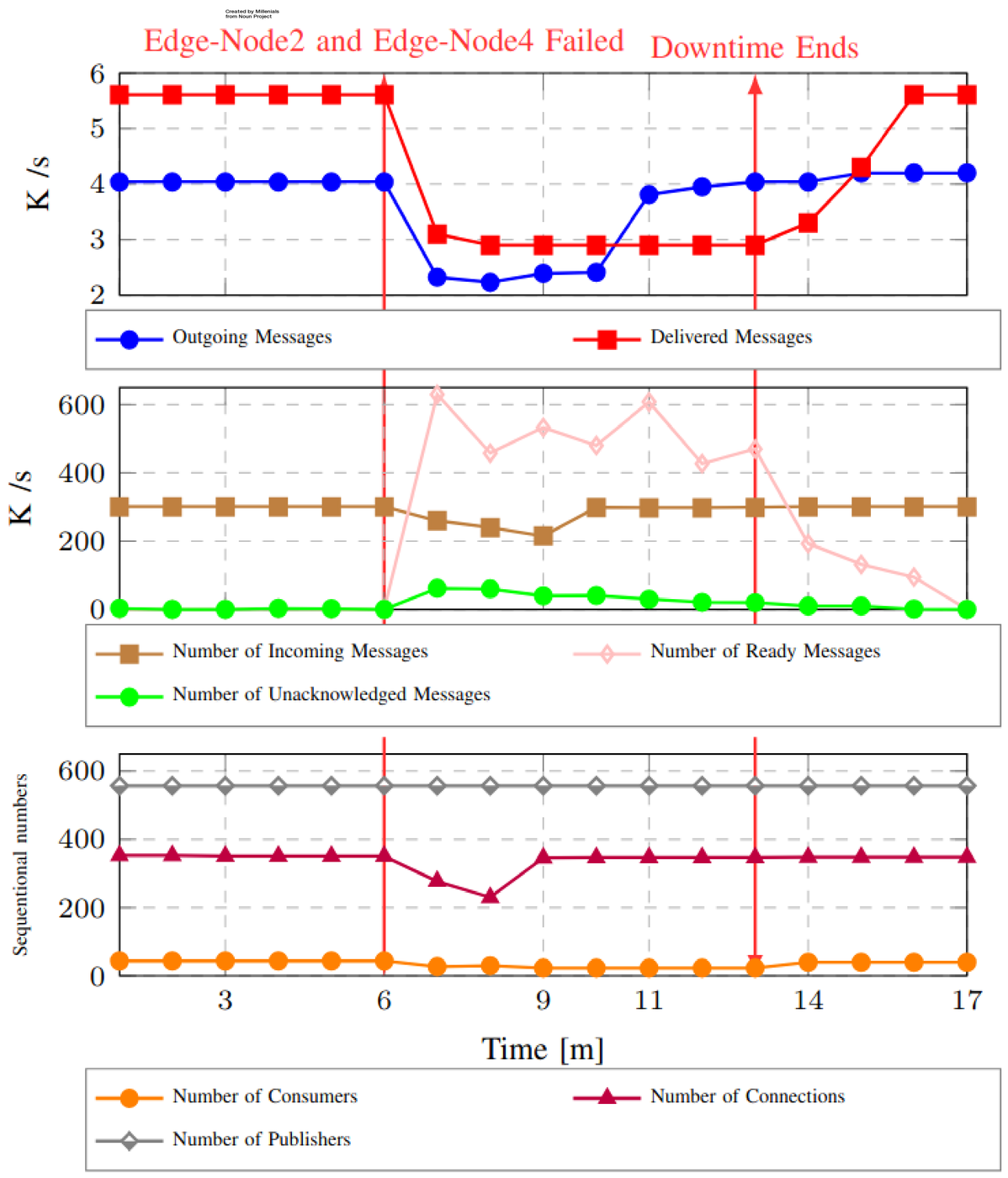


- This infrastructure uses a unified Kubernetes cluster with five dual-role nodes for high availability and fault tolerance. Each node acts as both master and worker, with a unique local-only API server setup to enhance resilience against single-node failures.
- It supports up to two control-plane node failures without affecting performance or data integrity, ensuring continuous operation for mission-critical applications. Any number of worker nodes can be added to the cluster.

Key Features of the Proposed Infrastructure

- **High Availability**
- **Efficient Resource Utilization**
- **Robust Data Handling**
- **Dynamic Resource Allocation**
- **Enhanced Disaster Recovery**

Experiment



Despite node outages, our system maintains operational latency within acceptable ranges and provides 99.966% availability.

Future Work

We will address **fault tolerance techniques** considering conflicting optimization targets such as **latency** and **availability**.