

## An SRN-based Availability Modeling and Analysis for Edge Computing Systems with Correlated Failures

Damsub Lim\*, Tuan Anh Nguyen\*\*, Dugki Min\*\*\*, MinGi KYUNG\*

\*Researcher, Dept. of Computer Science and Engineering, Konkuk University, Seoul, Korea

\*\*Professor, Dept. of Software Engineering, Ho Chi Minh City University of Industry and Trade(HUIT),  
Ho Chi Minh City, Vietnam

\*\*\*Professor, Dept. of Computer Science and Engineering, Konkuk University, Seoul, Korea

### [Abstract]

Edge computing is a key technology for overcoming cloud latency and bandwidth constraints, but distributed node environments are vulnerable to both independent and correlated failures. Accordingly, this study 1) extends the existing SRN model to incorporate correlated failures, 2) defines a Capacity-Oriented Availability (COA) metric that reflects the overhead of HA and LM policies, and 3) performs sensitivity analysis through discrete event simulation. The analysis reveals that when failure intervals are long, the combined HA+LM policy performs better; however, a policy performance reversal occurs when the number of correlated failure nodes exceeds the threshold (5 nodes). This is because during large-scale failures, the node search overhead of LM delays recovery, making the HA-only policy provide higher COA. This study suggests the necessity of adaptive recovery strategies based on the scale of correlated failures.

▶ **Key words:** Availability, Correlated Failures, Edge Computing, Migration, Stochastic Reward Nets

### [요 약]

엣지 컴퓨팅은 클라우드의 지연 시간 및 대역폭 제약을 극복하는 핵심 기술이나, 분산된 노드 환경은 독립 장애와 상관된 장애에 취약하다. 이에 본 연구는 1) 선행 SRN 모델을 확장하여 상관된 장애를 통합하고, 2) HA 및 LM 정책의 오버헤드를 반영한 용량 중심 가용성(COA) 메트릭을 정의한 후, 3) 이산 사건 시뮬레이션으로 민감도 분석을 수행하였다. 분석 결과, 장애 발생 간격이 긴 경우 HA+LM 결합 정책이 우수했으나, 상관된 장애 노드 수가 임계값(5개)을 초과하면서 정책 성능 역전이 발생하였다. 대규모 장애 시 LM의 노드 탐색 오버헤드가 복구를 지연시켜 HA 단독 정책이 더 높은 COA를 제공하기 때문이다. 본 연구는 상관된 장애 규모에 따른 적응형 복구 전략의 필요성을 시사한다.

▶ **주제어:** 가용성, 상관된 장애, 엣지 컴퓨팅, 마이그레이션, 확률적 보상 망

• First Author: Damsub Lim, Corresponding Author: Dugki Min

\*Damsub Lim (miaow89@konkuk.ac.kr), Dept. of Computer Science and Engineering, Konkuk University

\*\*Tuan Anh Nguyen (anhngt@huit.edu.vn), Dept. of Software Engineering, Ho Chi Minh City University of Industry and Trade(HUIT)

\*\*\*Dugki Min (dkmin@konkuk.ac.kr), Dept. of Computer Science and Engineering, Konkuk University

\*MinGi KYUNG (MinGi.KYUNG@gmail.com), Dept. of Computer Science and Engineering, Konkuk University

• Received: 2025. 11. 06, Revised: 2025. 11. 21, Accepted: 2025. 11. 27.

## I. Introduction

최근 엣지 컴퓨팅은 네트워크 지연과 대역폭 제약을 극복하기 위한 핵심 기술로 부상했다. 엣지 노드를 사용자에게 근접 배치하여 서비스 처리를 중앙 클라우드에서 분산 환경으로 전환하고 있다[1]. 그러나 분산 환경에서는 노드 장애와 자원 불균형으로 인해 가용성이 저하될 수 있으므로 체계적인 평가와 모델링이 필수적이다.

중앙 클라우드는 물리적 거리로 인한 높은 왕복 시간(RTT)과 코어 네트워크의 혼잡으로 인해 이러한 엄격한 요구 사항을 충족하는 데 한계가 있다. Table I은 중앙 클라우드와 엣지 컴퓨팅의 주요 성능 지표를 비교한 것이다 [20, 21]. 데이터 처리 위치를 사용자 단말에 가까운 엣지로 이동시킴으로써, 초저지연(Ultra-Low Latency)과 대역폭 절감이 가능함을 알 수 있다.

엣지 환경은 장애 대응과 서비스 지속성 확보가 기존 클라우드보다 더 어렵다. 노드 고장, 네트워크 단절, 자원 고갈 등으로 인해 서비스 중단 위험이 높기 때문에[2], 고가용성(High Availability, 이하 HA) 구현이 매우 중요하다. HA의 대표적인 접근 방법으로는 마이크로서비스 복제와 장애 발생 시 장애 조치(Failover)가 있다. 특히 라이브 마이그레이션(Live Migration, 이하 LM)은 노드 부하 급증이나 유지보수 시에도 서비스를 끊김 없이 이동시켜 장애를 최소화한다[8, 9].

하지만 기존 HA 및 LM 연구의 대부분은 장애를 하나의 독립적인 장애, 고장으로 가정했다. 반면 실제 엣지 환경에서는 공유 전력, 공유 네트워크 또는 물리적 인접성으로 인해 여러 노드가 동시에 또는 연쇄적으로 장애가 발생하는 상관된 장애(Correlated Failures)가 빈번히 발생한다 [10, 11]. 이러한 상관된 장애는 전통적인 독립 장애 가정하의 HA 분석으로는 충분히 예측하기 어렵다.

Table 1. Comparison of Cloud and Edge Computing [20, 21]

Feature	Centralized Cloud	Mobile Edge Computing
Latency (RTT)	High (>100ms)	Ultra-Low (<10ms)
Bandwidth Cost	High (Core Network)	Low (Local Offloading)
Proximity	Far (Internet / WAN)	Near (Radio Access Network)
Location Awareness	Low	High

본 논문은 독립적인 장애(Independent Failures)를 가정한 이전 연구[7]를 확장한 것이다. 기존 연구는 모든 장애가 독립적으로 발생한다는 가정으로 인해 여러 엣지 노드에 영향을 미칠 수 있는 실제 시나리오를 반영하지 못하는 한계가 있었다. 본 연구의 핵심 기여는 이전 연구에 상관된 장애 개념을 추가하여 기존 모델보다 현실성을 높여 분석한 것이다. 본 논문에서는 엣지 컴퓨팅 환경에서 다음과 같은 주제에 대해 가용성 분석을 수행한다.

- **엣지 노드, 마이크로서비스 및 스토리지 장애:** 다중 장애 모드와 상관된 장애 등 복잡성을 반영
- **Microservice HA and Live Migration:** 불확실한 네트워크 환경에서 복제, 재시작, 마이그레이션 검토
- **Capacity-Oriented Availability (COA) via SRN:** 장애 시 마이크로서비스 COA 측정을 위한 SRN 보상 함수 제안 및 HA/LM 오버헤드를 고려함
- **민감도 분석 및 이산 사건 시뮬레이션:** SRN 결과를 시뮬레이션으로 검증하고, 파라미터 변화가 엣지 생태계에 미치는 영향을 분석함

## II. Related Works

MEC(Mobile Edge Computing) 환경의 가용성을 정량화하기 위해 Table II와 같이 다양한 확률론적 모델이 활용되어왔다. SPN(Stochastic Petri Nets)을 사용하여 MEC 시스템의 성능과 가용성을 함께 평가하는 모델을 제안하고, 이중화(redundancy) 정책의 영향을 분석했다[3]. 마이크로서비스 수준에서 클라우드 네이티브 엣지 환경에서 마이크로서비스의 가용성과 지연 시간을 함께 고려했으며, 마르코프 체인을 기반으로 배포 모델을 평가했다[6]. 이러한 연구들은 엣지 환경의 가용성 분석의 중요성을 보여주지만, 실시간 마이그레이션(LM)이나 상관된 장애와 같은 복잡한 동적 장애 시나리오는 상세히 다루지 않았다.

시스템의 동적 복원력을 모델링하는 연구도 활발히 진행되었다. SRN 모델을 활용하여 클라우드 및 엣지 환경에서 VM 배포 및 실시간 마이그레이션(LM) 전략을 심도 있게 분석했다. 이 연구는 SRN의 가드 함수(Guard Functions)를 사용해 LM 정책을 정교하게 모델링했다는 점에서 본 연구의 방법론과 밀접한 관련이 있다[5].

하지만, 앞서 언급된 연구들은 장애가 독립적으로 발생한다고 가정하는 경우가 많았다. 이러한 가정의 한계는 [4]의 연구에서 지적된다. 이들은 엣지-클라우드 협업 환경 신뢰성을 평가하며 상관된 장애가 중요한 요소임을 확인

Table 2. Comparison of Prevailing Literature and Our Proposed Research

Ref.	Architecture	Primary Focus	HA Methods	LM	Correlated Failure?	Stochastic Modeling
[3]	MEC	Performance and Availability Planning	Redundancy	No	Not addressed	Yes(SPN)
[4]	Edge-Cloud	Dependability & Correlated Failures	Redundancy	No	Yes	Yes(Markov chain)
[5]	Edge-Cloud	VM Deployment & Migration Strategies	Load Balancing	Yes	Not addressed	Yes(SRN)
[6]	Cloud Native Edge	Availability & Latency of Micro Services	Deployment Models	No	Not addressed	Yes(Markov Chain)
[7]	MEC based	HA+LM	Failover	Yes	Not addressed	SRN+DES
Our Work	MEC based	HA+LM	Yes(Failover, Replication)	Yes	Yes, like 37% of all	SRN+DES

하고, 이를 마르코프 체인으로 분석했다.

이러한 배경에서 MEC 기반 환경에서 HA와 LM 정책을 SRN과 이산 사건 시뮬레이션(DES)으로 결합하여 분석하는 포괄적인 프레임워크를 제안했다. 본 논문은 독립 장애를 가정한 선행 연구[7]의 HA 및 LM 정책 모델을 기반 프레임워크로 재사용한다. 그러나 [7]은 노드 간 장애의 독립성을 전제하여 실제 엣지 환경의 취약점을 온전히 반영하지 못했다. 이를 극복하기 위해, 본 연구는 [7]의 모델에 상관된 장애(Correlated Failures) 메커니즘을 새롭게 통합하고 확장한다. 구체적으로, 상관 장애 발생을 제어하는 별도의 Clock 모델을 설계하고 이를 시스템 모델과 연동함으로써, 기존의 독립 장애 모델로는 분석 불가능했던 동시 다발적 장애 시나리오에서 가용성 변화를 정량화한다.

### III. System Architecture

제안하는 시스템 아키텍처는 Fig. 1에 도식화 되어 있으며, 엣지 컴퓨팅 노드는 마이크로서비스가 배포 및 실행되는 엣지 계층(Edge Layer)의 물리적 또는 가상화된 자원을 의미한다. 엣지 컴퓨팅 서비스를 이용하는 클라이언트(Client1, Client2)가 생성하는 사용자 입력 데이터는 다수의 엣지 컴퓨팅(Edge Computing 1 to n)로 전달된다. 각 엣지 컴퓨팅 노드는 링(Ring) 구조의 상호 연결된 토폴로지를 형성하여 마이크로 서비스(Microservice)를 클라이언트에게 제공한다. 본 아키텍처에서 각 노드는 물리적으로 가장 인접한 노드( $k-1$  또는  $k+1$ )를 우선적인 장애 조치(Failover) 및 마이그레이션 대상으로 지정하여 오프로딩을 수행한다고 가정한다. 일반적인 엣지 컴퓨팅 노드는 1부터 n까지 임의의 개수로 확장하거나 축소가 가능하며, 본 논문에서는 모델링 복잡도와 시뮬레이션 상태폭과

같은 효율성을 고려하여  $n=6$ 으로 가정하였다. 정상적으로 운영되었을 때, Client 1은 네트워크 지연(Network Latency)을 최소화할 위해 지리적으로 가장 근접한 Edge Computing1을 Primary Node로써 사용하고 마이크로서비스를 제공받는다. 만약 Edge Computing 1의 고장이 발생하여 장애상황 발생 시, 장애 상태(Failure State)로 진입하게 되면, 시스템은 고가용성(HA)을 보장을 위해 다음으로 가까운 Edge Computing k를 백업 노드로 선택하여 마이크로서비스를 복제하여 이중화(Redundancy)를 유지한다. 또한, Edge Computing 1의 컴퓨팅 자원(CPU, 메모리, 스토리지 등)이 임계치를 초과하여 부족할 경우, 서비스의 연속성을 유지하기 위해서 인접한 엣지 노드로 실시간 마이그레이션 (LM)을 실행 한다. 이러한 Redundancy와 Migration 메커니즘은 엣지 환경에서 서비스 가용성을 보장할 수 있다.

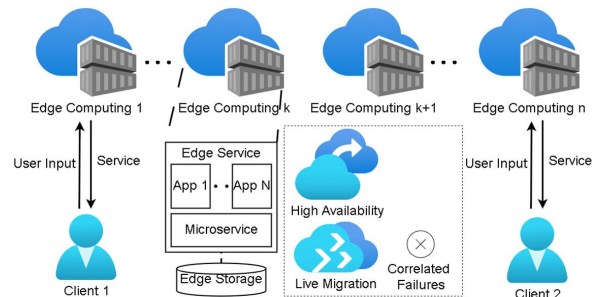


Fig. 1. System Architecture on Edge Computing

### IV. Stochastic Models

본 연구는 Fig. 1에서 제안한 시스템 구조를 기반으로, N개의 엣지 컴퓨팅 노드로 구성된 분산 환경의 장애-복구 동작(failure and recovery behavior) [12]을 확률론적으

로 모델링하기 위해 SRN(Stochastic Reward Net)기법을 적용하였으며, 그 전체 모델의 구조는 Fig. 2로 도식화되어 있다. Table III은 Fig. 1의 시스템 아키텍처 구성요소가 Fig. 2의 SRN 모델에서 어떻게 매핑 되는지를 요약한 것이다. 이를 통해 물리적/논리적 시스템 환경이 확률적 모델로 변환되는 과정을 정의한다. 제안하는 모델의 주요 구성 요소 및 동작 메커니즘은 다음과 같다.

- 엣지 노드: 각 노드는 이진 상태로 모델링된다. 'up'은 정상 동작, 'down'은 장애 상태를 의미한다. 장애는 마이크로서비스 실행 여부와 무관하게 발생하며, 하드웨어 고장, 네트워크 단절, 소프트웨어 크래시 등 다양한 시나리오를 포함한다.

- 마이크로서비스: 엣지 노드의 장애 감지 시 자동 복구 메커니즘이 활성화된다. 원본 노드의 failure 이벤트 탐지 시, 시스템은 Running State이고 충분한 자원을 보유한 대체 노드를 탐색하여 서비스를 재시작한다. Table IV의 가드 함수 조건 만족 시 LM을 통해 다른 노드로 이전된다.

- 고가용성: 시스템 복원력 평가를 위해 MTTR(Mean Time to Recovery)을 도입했다. 엣지 노드, 마이크로서비스, 스토리지 노드의 MTTF와 MTTR을 확률 분포로 표현하고, 단순 복구뿐 아니라 HA 전환과 LM을 SRN 모델에 통합하여 동적 장애 대응 전략을 모델링했다.

Table 3. Mapping of System Architecture components to SRN Model Elements

System Architecture (Fig. 1)	SRN Model Elements (Fig. 2) $k \in (1, \dots, 6)$
Edge Computing k Node	Place $P_{eu}^k$ (Up State), $P_{ef}^k$ (Failure State)
Microservice	Place $P_{mu}^k$ (Running Service)
Edge Storage	Place $P_{stru}$ (Up State), $P_{strf}$ (Failure State)
High Availability (Failover)	Transition $T_{ha}^{k,k+1}$ (Recovery)
Live Migration	Transition $T_{lm}^{k,k+1}$ (Migration)
Correlated Failure Environment	Transition ( $T_{eff}$ & Place $P_{efp}$ )

## 1. Proposed SRN Models

본 연구의 엣지 컴퓨팅 시스템에 대한 전체 SRN (Stochastic Reward Net) 모델 구조는 Fig. 2에 도식화되어 있다. 이 모델은 [7]에서 제시된 SRN 모델을 기반으로 재설계 및 확장하여, 새로운 시뮬레이션 실험과 가용성

분석을 수행하였다. Fig. 2(a)은 개별 엣지 노드의 장애-복구 사이클(failure-recovery cycle)을 표현하며, Fig. 2(c)은 N개의 엣지 노드로 구성된 확장 가능한 아키텍처에 고가용성(HA) 및 실시간 마이그레이션(LM) 메커니즘을 통합한 SRN 모델이다.

SRN 모델의 표기법에서 위첨자(superscript)는 엣지 노드의 인덱스를 나타내고, 아래첨자(subscript)는 해당 Place 또는 Transition의 의미론적 역할을 말한다. Fig. 2(a)의 구체적 동작을 살펴보면, Place  $P_{eu}^1$ 는 초기 토큰 1개를 보유하고, 이는 엣지 노드가 정상 운영 상태임을 모델링한다. Transition  $T_{ef}^1$ 의 발화(firing) 시 토큰은  $P_{ef}^1$ 로 전이되어 엣지 노드 1의 장애 발생을 나타낸다. 이후  $T_{ed}^1$ 가 발화되면 토큰은  $P_{ef}^1$ 에서  $P_{ed}^1$ 로 이동하여 장애 탐지(Failure Detection) 완료를 표현한다. 최종적으로  $T_{er}^1$ 의 발화는 엣지 노드 복구(Repair) 과정의 완료를 의미한다. Edge Node k( $E_k$ ), Edge Node k+1( $E_{k+1}$ ), Edge Node n( $E_n$ )도 동일한 상태 전이 패턴을 따른다.

스토리지 컴포넌트 모델의  $P_{stru}$ 는 초기 토큰 1개를 가지며 엣지 스토리지의 정상 상태를 나타낸다.  $T_{strf}$ 의 발화로 토큰이  $P_{strf}$ 로 이동하면 스토리지 장애 상태를 표현하고,  $T_{strf}$ 의 발화는 복구 완료 후  $P_{stru}$ 로의 상태 복귀를 모델링 한다. Fig. 2(c) 상단 중앙의 “HA and LM Policy” 파란색 점선 블록은 본 연구에서 비교 평가하는 세가지 시나리오를 모델 수준에서 동적으로 전환할 수 있는 정책 제어 메커니즘을 나타낸다. 이는 Table IV의 Guard Function에서 사용되어 (i) without HA and LM, (ii) with only HA, (iii) with HA and LM의 세가지 시나리오를 설정하여 모델의 가용성 분석을 진행할 수 있다.

Fig. 2(c)은 파란색 점선으로 구획된 Edge Node k submodel을 기본 단위로 하여 n개의 노드 모델이 상호 연결된 구조를 보여준다. 특히, Fig. 2(c)의 트랜지션 중  $T_{ha}^{k,k+1}$ ,  $T_{lm}^{k,k+1}$ 는 앞서 정의한 링 토폴로지 가정을 반영하여, 임의의 노드가 아닌 인접한 노드 ( $k+1$  또는  $k-1$ )로만 토큰이 이동할 수 있도록 모델링 되었다. Place  $P_{mu}^k$ 는 초기에  $m$ 개의 토큰을 보유하며, 여기서  $m$ 은 Table V의  $res\_max$  파라미터와 연관된다.  $res\_max$ 는 엣지 노드가 수용 가능한 최대 자원 용량을 정의하며, 초기 자원 할당량  $m$ 은  $res\_max \geq m$  제약조건을 만족해야 한다.  $m$  값은 해당 노드에서 동시 실행 가능한 마이크로서비스 인스턴스의 최대 개수를 나타낸다.

마이크로서비스 중단 시나리오는 두 가지로 분류된다:

(i) 호스트 Edge Node k의 장애, (ii) Ek에서 실행 중인 마이크로서비스 자체의 장애 시나리오 (i)의 경우,  $T_m^k$ 이 발화하여 토큰이  $P_{mdn}^k$ 으로 전이되며,  $1/\mu_m$ 의 지연 시간 이후  $T_{mr}^k$ 이 발화되어 서비스가 복구된다. 시나리오 (ii)에서는  $T_{mf}^k$ 가 발화하여 토큰이  $P_{mf}^k$ 로 이동하고, 동일한 복구 지연  $1/\mu_m$  후  $T_{mrp}^k$ 의 발화로 마이크로서비스가 재시작된다.

#### a) High Availability

$P_{eu}^k$ 에서 장애가 발생하면 마이크로서비스의 고가용성 메커니즘이 실행된다. 이는 엣지 노드 k( $P_{mu}^k$ )에서 실행 중이던 서비스 인스턴스를 대체 노드로 재배포하는 정책으로, 타겟 노드의 가용 용량에 기반하여 수행된다. 가드 함수  $gT_m^{kk+1}$ 의 조건이 충족되면  $T_m^{kk+1}$ 이 발화하고,  $P_{mu}^k$ 의 토큰이 중간 상태  $P_{mha}^{kk+1}$ 로 전이된다. Table V의  $1/\gamma_{ha}$ 로 정의된 장애복구 지연 이후, 최종적으로 토큰은  $P_{mu}^{k+1}$ 로 이동하여 서비스 재개한다. 이는  $P_{eu}^k$ 의 마이크로서비스가  $P_{mu}^{k+1}$ 로 마이그레이션되어 재시작되는 과정을 모델링한다. 역방향 장애복구의 경우,  $P_{mu}^{k+1}$ 의 장애 시  $T_m^{k+1k}$ 이 발화하여 토큰이  $P_{mha}^{k+1k}$ 로 전이되고,  $1/\gamma_{ha}$ 의 중복성(redundancy) 확보 시간 경과 후  $T_{ha}^{k+1k}$ 가 발화하여  $P_{mu}^k$ 로 서비스를 이전함으로써 장애 상황에서도 서비스 연속성을 보장한다.

#### b) Live Migration

Live Migration 정책은 엣지 노드 간 자원 활용의 균형을 동적으로 최적화하는 핵심 전략이다. 이 메커니즘은 자원 포화 상태에 근접한 노드로부터 마이크로서비스를 여유 자원을 보유한 인접 노드로 우선순위를 할당하여 이전한다. 노드의 실시간 자원 상태 모니터링을 통해 마이크로서비스 배치를 적응적으로 재조정함으로써 성능 병목을 예방하고 일관된 서비스 가용성을 유지한다. 구체적 예로,  $P_{mu}^k$ 의 마이크로서비스 실행 개수가 임계치  $res_{max}$ 에 도달하기 전에 인접 노드의 잔여 용량을 평가한다.  $P_{mu}^{k+1}$ 에 충분한 여유 자원이 확인되면  $T_{lm}^{kk+1}$ 이 발화하여 토큰을 이전시킨다. 반대 상황일 경우  $T_{lm}^{k+1k}$ 의 발화로  $P_{mu}^k$ 로 토큰이 이동하여 마이그레이션을 실행한다.

#### c) 확장 가능한 n-노드 아키텍처

Fig. 2(c)은 시스템 확장성(scalability)을 중심 설계 원

칙으로 채택하여, 엣지 노드를 1개부터 시작해 점진적으로 n개까지 확장 가능하도록 모델링하였다. 본 SRN 모델의 노드 구조는 Fig. 2(c)의 Edge Node k submodel을 기본 모델 단위로 활용하여, 각 노드를 독립적 컴포넌트처럼 구성하고 통합하는 모듈형 설계를 채택한다. 이러한 접근법은 시스템을  $(P_\mu^1, P_\mu^2, P_\mu^3, \dots, P_\mu^k, P_\mu^{k+1}, \dots, P_\mu^n)$ 의 가변적 노드 집합으로 표현함으로써 n개까지의 엣지 노드 확장을 용이하게 한다. 예를 들어, 6개 노드 구성에서는  $k=4, k+1=5, n=6$ 으로 설정하여  $P_\mu^4, P_\mu^5, P_\mu^6$ 로 구체화할 수 있다. 9개 노드로 확장하는 경우, 기존 6개 노드 모델에  $P_\mu^7, P_\mu^8, P_\mu^9$  서브모델을 추가 연결하여 구현한다.

#### d) 상관된 장애 모델 확장

본 연구의 상관된 장애 모델은 [7]에서 제시된 독립 장애 가정인 SRN 모델을 기반으로 활용한다. 기반 모델은 N개의 엣지 노드로 구성되며, 각 노드의 장애-복구 주기를 분석한다. 고가용성 및 실시간 마이그레이션 정책 하에서 용량 중심 가용성(COA)을 평가하였다. 독립 장애 가정의 현실적 한계를 극복하기 위해, 본 연구는 Fig. 2(b)에 제시된 상관된 장애 clock 모델과 Fig. 2(a)의  $T_{cff}^1, \dots, T_{cff}^k, T_{cff}^{k+1}, \dots, T_{cff}^n$ 을 추가하여 기존 모델에 통합하였다. Fig. 2(b)의  $P_{cf}$ 는 1개의 토큰을 보유하고 있고, 상관된 장애가 발생할 시간이 흐른 후에  $T_{cfint}$  (correlated failures interval time)이 발화하여 토큰이  $P_{cfp}$ 로 이동한다. 만약 상관된 장애로 인한 엣지 노드가 동시에 2개 장애가 발생하면  $P_{cfp}$  상태에 토큰이 존재할 때 Fig. 2(a)의  $T_{cff}^1, \dots, T_{cff}^k, T_{cff}^{k+1}, \dots, T_{cff}^n$  중 2개가 발화되어 엣지 노드 2개는 고장 상태로 강제로 변화하게 된다. 그 후  $T_{cfd}$ 이 발화되어 다시 MTTCF(Mean Time to Correlated Failure) 시간까지 대기하게 된다.

## V. Experiment and Analysis

제한한 SRN 모델의 정량적 분석은 Mercury Tool [13]을 이용한 이산 사건 시뮬레이션으로 실험을 진행하였다. DES의 확률적 특성으로 인한 결과 변동성을 제어하고 통계적 신뢰도를 확보하기 위해서, 독립적인 시뮬레이션을 다수 반복 실행하여 그 결과 종합 분석하였다. 모든 실험은 CPU: Intel(R) Core(TM) i5-13400, RAM: 16 GB of DDR4 환경에서 수행하였다.

## 1. Default Input Parameters and Model Configuration

Fig. 2에 제시된 SRN 모델은 엷지 컴퓨팅 시스템의 전체 구조를 포괄적으로 표현하며, Table IV에 정의된 가드 함수와 Table V에 명시된 기본 입력 파라미터를 기반으로 구성 한다[14-16]. Table IV는 시스템의 즉시 전이(Immediate Transition)와 시간 지연 전이(Timed Transition)에 대한 설명을 제공하며, 각 전이의 파라미터 표기법, 연관된 전이 식별자, 가드 함수의 논리적 조건문, 기본값(default value), 그리고 측정 단위를 기술하였다. 특히, 복구 정책별 소요 시간은 각 메커니즘의 기술적 특성을 반영하여 설정하였다. HA( $1/\gamma_{ha}$ )는 장애 발생 후 새로운 인스턴스를 프로비저닝하고 초기화하는 Cold Start 방식을 가정하여 5분으로 설정하였다. 반면, LM( $1/\gamma_{lm}$ )은 서비스 실행 상태를 유지하며 메모리 페이지만 전송하는 Pre Copy 방식을 가정한다[8, 9]. 엷지 노드 간 고속 네트워크 환경에서는 이러한 라이브 마이그레이션이 전체 재시작보다 효율적이므로, LM의 소요 시간을 HA보다 빠른 1분으로 설정하였다.

## 2. Existing Models and Our Contributions

본 연구의 실험은 선행 연구 [7]에서 제안된 파라미터와 모델 구조를 재구현하여 검증하는 것에서 시작한다. 하지만 [7]의 분석 범위가 독립 장애에 국한된 반면, 본 연구는 [17-19]에 기반하여 상관된 장애를 새로운 변수로 추가하였다. 이를 통해 기존 연구에서 다루지 못한 대규모 장애 상황에서의 정책적 한계와 역전 현상을 새롭게 규명한다. 이러한 상관된 장애 모델링의 필요성은 실제 대규모 시스템 데이터로도 뒷받침된다. 분산 시스템 연구에 따르면, 전체 장애의 약 37%가 두 개 이상의 노드가 동시에 고장나는 상관된 장애이며[17], Google 데이터에서도 다수의 동시 재부팅이 관측되었다[18]. 또한 Alibaba의 데이터 센터 연구 역시 최대 36.9%에 달하는 Intra Node 및 Rack 상관된 장애를 보고하였다[19].

따라서 엷지 노드에서의 상관된 장애율을 37%로 정의하였고, 다음과 같은 수식을 통해 Table V의  $T_{cfint}$ 의 MTTCF를 설정하였다.

상관된 장애를 반영한 엷지 노드 장애율을 계산한다. 엷지 노드의 평균 무고장 시간(MTTF)은 Table V의  $1/\lambda_e$ 에 해당하는 31,536,000초(약 1년)이다. 클러스터( $N = 6$ 개 노드)의 전체 독립 고장율은 다음과 같으며, 다음 수식 Eq. (1)과 같다.

$$\begin{aligned}\lambda_{total\_e} &= N \times \lambda_e \\ &\approx 6 \times \frac{1}{31,536,000} \approx 1.90 \times 10^{-7} s^{-1} \quad (1)\end{aligned}$$

상관된 장애는 여러 노드가 동시에 고장나는 사건을 의미한다. 본 모델에서는 상관된 장애가 전체 고장의 37%( $f_{cf}$ )를 차지하고 2개의 노드가 동시에 상관된 장애( $k = 2$ )를 가진다고 가정하였으며 다음 수식 Eq. (2)과 같다.

$$\begin{aligned}\lambda_{ecf} &= \frac{f_{cf} \times \lambda_{total\_e}}{k} \\ &\approx \frac{0.37 \times 1.90 \times 10^{-7}}{2} \approx 3.52 \times 10^{-8} s^{-1} \quad (2)\end{aligned}$$

이는 약 329일마다 한 번씩 상관된 장애가 발생하는 빈도에 해당하며, Table V의  $1/\lambda_{ecf}$ 의 28,445,000s, 즉 7,901h로 기본 파라미터를 설정하여 모델을 분석한다.

## 3. Experiment Metrics: Capacity-Oriented Availability

용량 중심 가용성(COA)은 특정 시점에서 사용자에게 제공 가능한 자원 용량의 총량을 측정함으로써 시스템의 가용성을 평가하는 메트릭이다. 단순히 시스템의 정상 가동 상태만 측정하는 것이 아닌 COA는 부분적인 시스템 장애가 서비스 제공 능력을 정량적으로 반영가능하다. Eq. (3)은 엷지 노드가 정상 작동하는 조건 하에서 모든 엷지 노드에 걸쳐 실행 중인 마이크로서비스의 총 개수를 평가함으로써 COA를 산출하는 보상 함수 (Reward Function)로 정의한다.

$$COA_n = \sum_{i=1}^n (\#P_{mu}^n) \times P\{\#P_{stru} \equiv 1\} \quad (3)$$

Eq.(3)에 기반하여, 본 논문은 6개의 엷지 노드로 구성된 시스템에 대해 SRN 모델링하고 분석을 수행한다. 따라서 6개의 엷지 노드에 해당하는 보상함수는  $((\#P_{mu}^1 + \#P_{mu}^2 + \#P_{mu}^3 + \#P_{mu}^4 + \#P_{mu}^5 + \#P_{mu}^6) \times P\{\#P_{stru} = 1\})$ 로 정의한다.

## 4. Stationary Analysis

Fig. 2의 SRN 모델에 Table V에 명시된 기본 입력 파라미터를 적용하고, Eq. (3)의 보상 함수를 활용하여 정상 상태 (steady-state)에서의 예상 동작을 검증한다. 이후 민감도 분석을 하기 위해 3가지 정책 시나리오를 구성하고 COA를 분석한다: (i) HA/LM 미적용, (ii) HA만 적용, (iii) HA 및 LM 동시 적용. 3가지 정책 시나리오는 Fig. 2(c)의 중앙 상단의 파란색 점선으로 표시되어있는 HA and LM Policy으로 매 시뮬레이션마다 선택하여 분석한다. HA and LM Policy의  $P_{Policy\_HA}$ 와  $P_{Policy\_LM}$ 은 Table IV

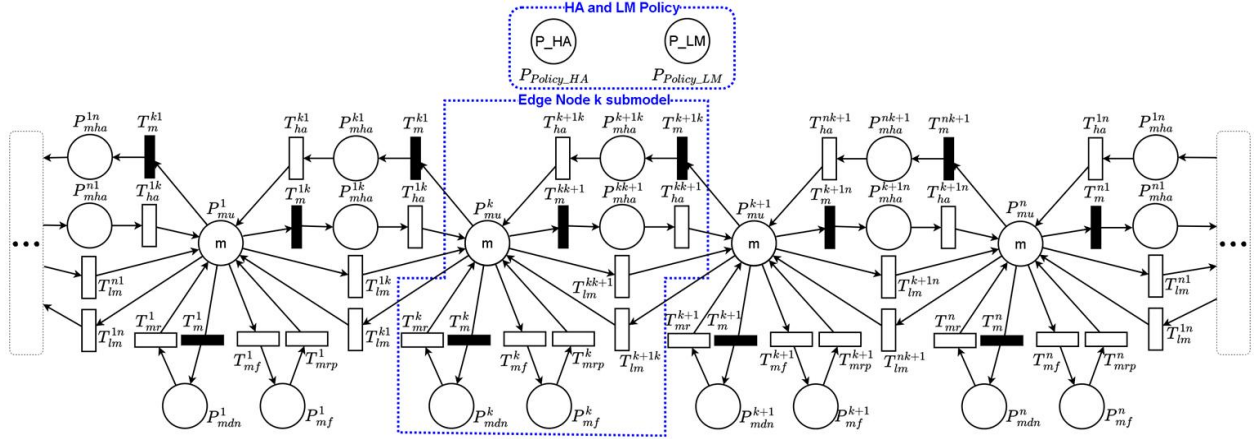
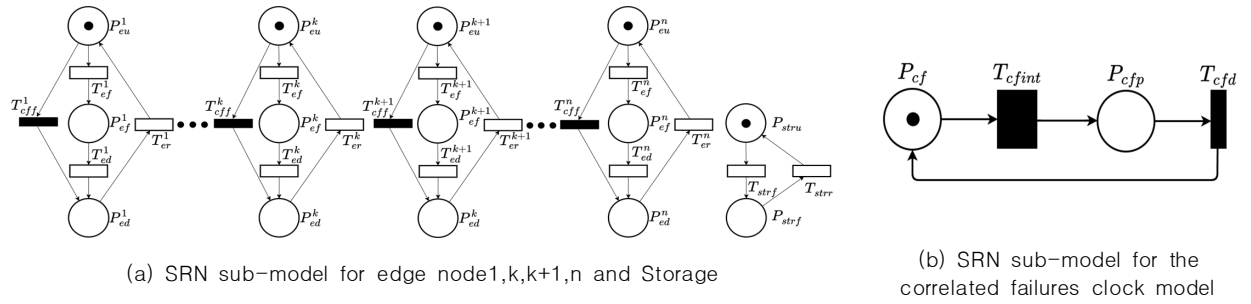


Fig. 2. SRN model for N Edge Nodes and Storages

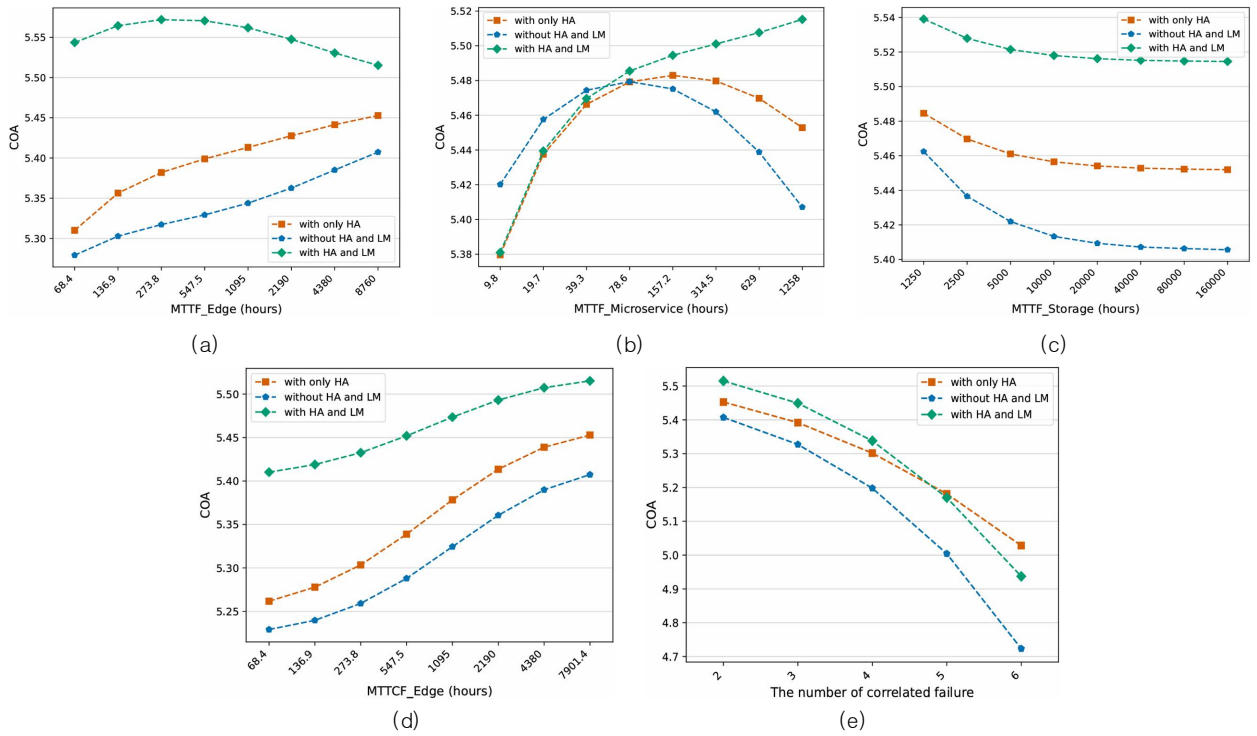


Fig. 3. Capacity-Oriented Availability(COA) with respect to Sensitivity Parameter.

(a) Mean time for Edge Node Failure (hours), (b) Mean time for Microservice Failure (hours), (c) Mean time for Meta-Edge Storage Failure (hours), (d) Mean time to Correlated Failure (hours), (e) The number of correlated failure.

Table 4. Guard Functions for the Six Edge Storage Stochastic Model

Name	Transition	Guard function Description ( $k \in \{1, \dots, 6\}$ )	
gTmn	$T_m^1, T_m^2$ $T_m^3, T_m^4$ $T_m^5, T_m^6$	If $\#P_{eu}^k = 0$	:1 otherwise:0
gTmkn	$T_m^{12}, T_m^{23}$ $T_m^{34}, T_m^{45}$ $T_m^{56}, T_m^{61}$	If $(\#P_{eu}^k = 0) \text{ AND } (\#P_{eu}^{k+1} = 1) \text{ AND } (\#P_{nu}^{k+1} < res\_max + 1) \text{ AND } (\#Policy\_HA > 0)$	:1 otherwise:0
	$T_m^{21}, T_m^{32}$ $T_m^{43}, T_m^{54}$ $T_m^{65}, T_m^{16}$	If $(\#P_{eu}^{k+1} = 0) \text{ AND } (\#P_{eu}^k = 1) \text{ AND } (\#P_{nu}^k < res\_max + 1) \text{ AND } (\#Policy\_HA > 0)$	:1 otherwise:0
gTlmkn	$T_{lm}^{12}, T_{lm}^{23}$ $T_{lm}^{34}, T_{lm}^{45}$ $T_{lm}^{56}, T_{lm}^{61}$	If $(\#P_{eu}^{k+1} = 1) \text{ AND } (\#P_{nu}^k > m) \text{ AND } (\#P_{nu}^{k+1} < m) \text{ AND } (\#P_{nu}^{k+1} < res\_max + 1) \text{ AND } (\#Policy\_LM > 0)$	:1 otherwise:0
	$T_{lm}^{21}, T_{lm}^{32}$ $T_{lm}^{43}, T_{lm}^{54}$ $T_{lm}^{65}, T_{lm}^{16}$	If $(\#P_{eu}^{k-1} = 1) \text{ AND } (\#P_{nu}^k > m) \text{ AND } (\#P_{nu}^{k-1} < m) \text{ AND } (\#P_{nu}^{k-1} < res\_max + 1) \text{ AND } (\#Policy\_LM > 0)$	:1 otherwise:0
gTlmnr	$T_{nr}^1, T_{nr}^2$ $T_{nr}^3, T_{nr}^4$ $T_{nr}^5, T_{nr}^6$	If $\#P_{eu}^k = 1$	:1 otherwise:0
gTcff	$T_{cff}^1, T_{cff}^2$ $T_{cff}^3, T_{cff}^4$ $T_{cff}^5, T_{cff}^6$	If $\#P_{cfp} = 1$	:1 otherwise:0
gTcfd	$T_{cfd}$	If $(\sum_{k=1}^6 (\#P_{ed}^k) \geq num\_cf)$ with Priority = 2	:1 otherwise:0

Table 5. Input Parameters for the Six Edge Storage Stochastic Model

Parameters	Related Transitions	Description	Default Value
$1/\lambda_e$	$T_{ef}^1, T_{ef}^2, T_{ef}^3, T_{ef}^4, T_{ef}^5, T_{ef}^6$	mean time for edge nodes failure	8760 hours
$1/\lambda_m$	$T_{mf}^1, T_{mf}^2, T_{mf}^3, T_{mf}^4, T_{mf}^5, T_{mf}^6$	mean time for microservice failure	1258 hours
$1/\lambda_s$	$T_{strf}$	mean time for edge storage node failure	40000 hours
$1/\delta_{ed}$	$T_{ed}^1, T_{ed}^2, T_{ed}^3, T_{ed}^4, T_{ed}^5, T_{ed}^6$	mean time for detection of edge nodes failure	10 seconds
$1/\mu_e$	$T_{er}^1, T_{er}^2, T_{er}^3, T_{er}^4, T_{er}^5, T_{er}^6$	mean time to repair edge nodes	1 hours
$1/\mu_m$	$T_{mr}^1, T_{mr}^2, T_{mr}^3, T_{mr}^4, T_{mr}^5, T_{mr}^6$ $T_{nmp}^1, T_{nmp}^2, T_{nmp}^3, T_{nmp}^4, T_{nmp}^5, T_{nmp}^6$	mean time to repair microservice	0.238 hours
$1/\mu_s$	$T_{str}$	mean time to repair a edge storage node	5 hours
$1/\mu_{ha}$	$T_{ha}^{12}, T_{ha}^{21}, T_{ha}^{23}, T_{ha}^{32}, T_{ha}^{34}, T_{ha}^{43}$ $T_{ha}^{45}, T_{ha}^{54}, T_{ha}^{56}, T_{ha}^{65}, T_{ha}^{16}, T_{ha}^{61}$	mean time to migrate a edge node for HA	5 minutes
$1/\mu_{lm}$	$T_{lm}^{12}, T_{lm}^{21}, T_{lm}^{23}, T_{lm}^{32}, T_{lm}^{34}, T_{lm}^{43}$ $T_{lm}^{45}, T_{lm}^{54}, T_{lm}^{56}, T_{lm}^{65}, T_{lm}^{16}, T_{lm}^{61}$	mean time to migrate a edge node for LM	1 minutes
$m$		Initialize resource capacity of the Meta-Edge Stroage nodes	1
$res\_max$	$res\_max \geq m$	Maximum resource capacity of the Meta-Edge Stroage nodes	2
$num\_cf$	$T_{cfd}$	The number of Correlated Failures	2
$1/\lambda_{ecf}$	$T_{cftnt}$	mean time to correlated edge nodes failure	7901 hours

의 가드 함수를 통해 제어되며, 각 정책별 시나리오 설정은 다음과 같다. 예를 들어, 시나리오 (i)에서는  $P_{Policy\_HA} = 0, P_{Policy\_LM} = 0$ 으로 설정하여 복구 정책을 비활성화하고, 시나리오 (ii)에서는  $P_{Policy\_HA} = 1, P_{Policy\_LM} = 0$ 으로 HA 메커니즘만 활성화하며, 시나리오 (iii)에서는 두 파라미터 모두 1로 설정 하여 HA와 LM 메커니즘으로 복원력 정책을 구현한다.

정상 상태에서의 COA를 정량화하기 위해 각 엣지 노드에  $m = 1, 2, 3, 4$ 개의 마이크로서비스를 할당하되,

$res\_max \geq m$  제약 조건을 만족하도록 설정하였다. Eq. (3)을 통해 산출된 Table VI의 결과는  $m$ 값의 증가에 따라 COA가 상승함을 보여준다. HA 메커니즘을 활성화 한 경우에도 정책 시나리오 미적용 때와 달리 성능 향상을 관찰할 수 있으며, HA와 LM을 동시에 구성하였을 때 가장 높은 COA로 평가되었다. 이 수치는 모델에 정의된 HA 및 LM이 반영된 상태에서 시스템이 장기적으로 도달하는 정상 상태 (Steady State)의 기대 성능을 나타낸다. 즉, Table VI는 일반적인 운영 환경에서의 각 정책이 제공할

수 있는 가용성의 기준을 제시하고 있으며, HA와 LM의 결합이 오버헤드를 상쇄하고도 성능을 입증한다. 다만, 오버헤드가 급증하는 특정 장애 시나리오에서의 동적 변화는 다음 4. Sensitivity Analysis의 민감도 분석에서 상세히 다룬다.

Table 6. COA Stationary Analysis for the Three Scenario

m	Case(i) without HA and LM	Case(ii) with onlyHA	Case(iii) with HA and LM
1	5.4070956	5.4528552	5.5151915
2	11.0713639	11.1815374	11.2781010
3	16.6805417	16.8331186	16.9538582
4	22.2337129	22.4309034	22.5707045

## 5. Sensitivity Analysis

COA의 변화를 체계적으로 분석하기 위해 Fig. 2의 SRN 모델에서  $k \in \{1, \dots, 6\}$ ,  $n = 6$ ,  $res\_max = 2$ ,  $m=1$ 를 고정하고, Table V의 파라미터  $\lambda_e, \lambda_m, \lambda_s$ 를 독립 변수로 변화시키며 분석을 수행하였으며, 각 파라미터의 변화율은 [7]에서 제안하고 있는 변화량만큼 그대로 입력 파라미터로 사용하여 시뮬레이션을 실행하고 이를 분석하였다. Fig. 3a, 3b, 3c의 전체적인 COA값의 경향은 베이스 모델[7]의 결과 값과 비슷하나 실제 수치는 다르므로 본 논문은 기반모델을 올바르게 재구현하였고 새롭게 이산 사건 시뮬레이션을 수행했다는 것을 의미한다.

### a) 엣지 노드 MTTF (Fig. 3a)

시나리오 (iii)은 MTTF 변화와 무관하게 지속적으로 높은 COA 수준(약 5.54 이상)을 나타냈다. 반면, 시나리오 (i)과 (ii)는 MTTF의 증가에 따라 상대적으로 낮은 초기값(약 5.30)으로부터 점증적 향상 추세를 보였으며, 이는 HA 및 LM 정책의 동시 적용이 노드 장애 빈도에 대한 시스템의 강건성(robustness)을 크게 향상시킴을 시사한다.

### b) 마이크로서비스 MTTF (Fig. 3b)

마이크로서비스 MTTF가 짧은 초기 구간(9.8시간)에서 시나리오 (i)은 5.42 의 최고 COA를 기록하였으나, 약 78.6시간을 경과하며 극대값을 형성한 후 하향 곡선을 그렸다. 시나리오 (ii)는 약 157.2시간 부근에서 피크를 보인 뒤 감소 패턴으로 전환되었다. 시나리오 (iii)은 전 구간에서 증가 경향을 나타내어 1258 시간에서 최고값(약 5.51)에 도달하였으며, 이는 HA와 LM의 시너지 효과가 서비스 수준 장애 대응에서 특히 효과적임을 입증한다.

### c) 스토리지 MTTF (Fig. 3c)

세 시나리오 모두 스토리지 MTTF의 최소값(1250시간)에서 COA 극대값을 달성하였으며, 이후 MTTF 증가 구간에서 완만한 감소세를 보이다가 10000시간 이상의 영역에서 수렴한다. 이러한 경향은 스토리지 신뢰성과 시스템 가용성 간의 비선형 상관관계를 반영한다.

### d) 상관된 장애 MTTF (Fig.3d)

상관된 장애의 발생 주기가 길어질수록(MTTCF 증가), 전체 시나리오에서 COA의 일관된 증가가 관찰되었다. 시나리오 (iii)은 MTTCF의 전 범위에 걸쳐 높은 COA(약 5.41)을 안정적으로 유지하여, HA 와 LM의 결합이 상관된 장애 환경에서도 강력한 복원력을 제공함을 실증한다. 시나리오 (i)과 (ii) 또한 MTTCF의 확대에 따른 지속적 가용성 증가 추세를 나타냈으나, 시나리오 (i) 은 최하위, 시나리오 (iii)은 고가용성을 보여주었다. 이는 상관된 장애 빈도가 감소하더라도 복구 정책의 유무가 시스템 가용성의 결정적 차별 요인으로 작용함을 시사한다.

### e) 상관된 장애 규모 (Fig.3e)

상관된 장애 규모의 증가는 모든 시나리오에서 COA의 선형 하락하는 경향을 보인다. 시나리오 (i)은 장애 규모 확대에 가장 취약한 반응을 보여 COA가 4.72까지 급격히 저하되었으며, 이는 복구 매커니즘 부재 시 대규모 장애에 대한 시스템의 한계를 명확히 드러낸다. 주목할 점은 상관된 장애 규모가 4개 이상 구간에서는 시나리오 (iii)이 (ii)보다 우위를 점했으나, 5개 노드(전체  $N=6$  중  $N-1$ 개 노드) 부근에서 두 곡선이 교차하였다. 이는 시스템 내 가용할 수 있는 백업 자원이 거의 고갈된 대규모 장애 시나리오( $N-1$  Failure)를 의미한다. 이러한 현상은 본 모델의 인접 노드 기반 토폴로지 제약과 밀접하게 연관 된다.  $N-1$ 개 노드가 동시에 고장 나는 임계 상황에서는 LM을 수용할 수 있는 인접 노드가 부재하여, LM 정책은 실행 가능한 노드를 탐색하는 과정에서 불필요한 오버헤드만 소모하게 된다. 그 결과, 단순 복구 정책인 HA보다 성능이 저하되는 역전이 발생한다. 본 연구에서는 SRN 모델의 상태 공간 폭발(State Space Explosion) 문제로 인해 노드 수를  $N=6$ 으로 제한하였으나, 이러한 경향성은  $N$ 이 증가하더라도  $N-1$ 개 노드가 동시에 장애를 겪는 임계 상황에서 동일하게 나타날 것으로 예측된다. 결과적으로 LM 정책의 복잡성이 오히려 복원력을 저해할 수 있음을 시사하며, 상관된 장애 규모에 따른 정책 선택의 필요성을 보여준다.

## VI. Conclusions

본 연구는 엣지 컴퓨팅 시스템의 가용성을 보다 더 현실적으로 평가하기 위해, 기존 독립 장애 가정 모델[7]을 확장하여 상관된 장애(Correlated Failures)를 통합한 SRN 기반 분석 프레임워크를 제안한다. HA와 LM 정책을 포함한 상세한 모델링을 통해, 상관된 장애가 시스템의 COA에 미치는 영향을 정량적으로 규명하였다. 주요 분석 결과, 상관된 장애 발생 빈도(CMTTF)가 낮을수록 HA과 LM을 적용한 시나리오(iii)가 장애 빈도 변화에 가장 안정적인 가용성을 보장하였다. 그러나 가장 주목할 만한 실험결과는 상관된 장애 규모에 따른 가용성 역전 현상이다. 4개 이하 노드의 동시 장애 시에는 HA와 LM 시너지가 우수하였으나, 5개 이상의 대규모 장애 시에는 LM의 노드 탐색 오버헤드가 장애 복구 시간을 지연시켜 시나리오 (ii)인 HA 단독 전략 보다 낮은 COA를 보여준다. 이는 HA와 LM의 무조건적 결합이 최적이지 않음을 보여주며, 상관된 장애 규모에 따라 LM 활성화를 동적으로 결정하는 적응형 장애 복구 메커니즘의 필요성을 시사한다. 본 연구는 이러한 상황 인지형 복구 정책 설계의 이론적 근거와 정량적 평가 방법론을 제시한다. 향후 본 연구의 결과는 스마트 도심 항공 교통(UAM), 디지털 트윈 시스템, 자율 주행 인프라 등 일시적인 서비스 중단이 치명적인 결과를 초래할 수 있는 엣지 컴퓨팅 분야의 가용성 설계 및 운영 가이드라인으로 폭넓게 활용될 수 있을 것으로 기대된다.

## ACKNOWLEDGEMENT

This research was partially supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. RS-2020-NR049595). This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. RS-2021-NR059684). This paper was supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (P0020536, HRD Program for Industrial Innovation)

## REFERENCES

- [1] D. Penglin and Md Likhan, A Survey of Emerging Trends in Edge Computing, 2024. DOI:10.13140/RG.2.2.22183.76962
- [2] S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing: Concepts, Applications and Issues," in Proceedings of the 2015 Workshop on Mobile Big Data, Hangzhou China: ACM, Jun.21,2015, pp.37-42,DOI:10.1145/2757384.2757397
- [3] C. Brito et al., "Stochastic Model Driven Performance and Availability Planning for a Mobile Edge Computing System," Applied Sciences, vol. 11, no. 9, p. 4088, Apr. 2021, DOI: 10.3390/app11094088.
- [4] J. Liang, B. Ma, S. Ali, and J. Huang, "Model-Based Evaluation and Optimization of Dependability for Edge Computing Systems," in Collaborative Computing: Networking, Applications and Worksharing, H. Gao and X. Wang, Eds., vol.406, Cham: Springer International Publishing, 2021, pp.728-747, DOI: 10.1007/978-3-030-92635-942
- [5] Y. Liu, K. Wang, L. Ge, L. Ye, and J. Cheng, "Adaptive Evaluation of Virtual Machine Placement and Migration Scheduling Algorithms Using Stochastic Petri Nets," IEEE Access, vol. 7, pp. 79 810-79 824, 2019, DOI:10.1109/ACCESS.2019.2923592
- [6] S. Arora, A. Ksentini, and C. Bonnet, "Availability and Latency Aware Deployment of Cloud Native Edge Slices," in GLOBECOM 2022 - 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil: IEEE, Dec. 2022, pp. 3441-3446, DOI:10.1109/GLOBECOM48099.2022.10001341
- [7] D. Lim et al., "Metaverse Distributed Storages: High Availability Quantification using Stochastic Reward Nets," in 2025 International Conference on Metaverse Computing, Networking and Applications (MetaCom), Seoul, Korea, Republic of: IEEE, Aug. 2025, pp. 181- 188, DOI:10.1109/MetaCom65502.2025.00039
- [8] P. G. Jeba Leelipushpam and J. Sharmila, "Live VM migration techniques in cloud environment - A survey," in 2013 IEEE CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES, Thuckalay, Tamil Nadu, India: IEEE, Apr. 2013, pp. 408-413, DOI: 10.1109/CICT.2013.6558130
- [9] A. Rezazadeh and H. Lutfiyya, "Multi-Microservice Migration Modeling, Comparison, and Potential in 5G/6G Mobile Edge Computing: A Non-Average Parameter Values Approach," IEEE Access, vol. 11, pp. 140 464- 140 480, 2023, DOI:10.1109/ACCESS.2023.3342190
- [10] E. Zhai, R. Chen, D. I. Wolinsky, and B. Ford, "Heading off correlated failures through independence-as-a-service," in Proceedings of the 11th USENIX Conference on Operating Systems Design and Implementation, ser. OSDI'14, Broomfield, CO: USENIX Association, 2014, 317-334
- [11] A. Aral and I. Brandic, "Dependency Mining for Service

- Resilience at the Edge,” in 2018 IEEE/ACM Symposium on Edge Computing (SEC), Seattle, WA, USA: IEEE, Oct. 2018, pp. 228–242, DOI: 10.1109/SEC.2018.00024
- [12] D. S. Kim, J. B. Hong, T. A. Nguyen, F. Machida, J. S. Park, and K. S. Trivedi, “Availability modeling and analysis of a virtualized system using stochastic reward nets,” in 2016 IEEE International Conference on Computer and Information Technology (CIT), IEEE, 2016, pp. 210–218
- [13] P. Maciel et al., “Mercury: Performance and dependability evaluation of systems with exponential, expoly-nomial, and general distributions,” in 2017 IEEE 22nd Pacific Rim international symposium on dependable computing (PRDC), IEEE, 2017, pp. 50–57
- [14] M. F. F. da Silva Lisboa, G. L. Santos, T. Lynn, D. Sadok, J. Kelner, P. T. Endo, et al., “Modeling the availability of an e-health system integrated with edge, fog and cloud infrastructures,” in 2018 IEEE symposium on computers and communications (ISCC), IEEE, 2018, pp.00416–00 421
- [15] C. Brito et al., “Stochastic model driven performance and availability planning for a mobile edge computing system,” *Applied Sciences*, vol. 11, no. 9, p. 4088, 2021
- [16] S. Sebastio, R. Ghosh, and T. Mukherjee, “An availability analysis approach for deployment configurations of containers,” *IEEE Transactions on Services Computing*, vol. 14, no. 1, pp. 16–29, 2018
- [17] D. Ford et al., “Availability in globally distributed storage systems,” in 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 10), Vancouver, BC: USENIX Association, Oct. 2010
- [18] R. Alagappan, A. Ganesan, Y. Patel, T. S. Pillai, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, “Correlated Crash Vulnerabilities”, OSDI, Savannah, USENIX, Nov, 2016
- [19] S. Han, P. P. C. Lee, F. Xu, Y. Liu, C. He, and J. Liu, “An In-Depth Study of Correlated Failures in Production SSD-Based Data Centers,” *FAST 21*, pp 417-429, Feb, 2021
- [20] Y. Yu, “Mobile edge computing towards 5G: Vision, recent progress, and open challenges”, *China Commun*, vol 13, Supplement2, pp 89–99, 2016, doi: 10.1109/CC.2016.7833463.
- [21] E. Bastug, M. Bennis, M. Medard and M. Debbah, “Toward Interconnected Virtual Reality: Opportunities, Challenges, and Enablers”, *IEEE Commun. Mag.*, vol 55, 6, pp 110–117, 2017, doi: 10.1109/MCOM.2017.1601089.

## Authors



Damsub Lim Received the M.S. and Ph.D. degrees in Computer science and Engineering from Konkuk University, South Korea, in 2015 and 2025. He is a Postdoctoral researcher in the Department of Computer

science and Engineering at Konkuk University. His research interests include Distributed System, Cloud Computing, IoT and SDN, System Analysis and Modeling, Performance, Availability and Dependability Evaluation.



Tuan Anh Nguyen (Ph.D.'15, M.Sc.'10, B.Eng.'08) received his Ph.D. in Computer Science and Systems Engineering from Korea Aerospace University and his MSc and BEng in Mechatronics from Hanoi University of

Science and Technology. He is a member of IEEE, IEEE Computer, Robotics and Automation (IEEE RAS), Aerospace and Electronic Systems (IEEE AESS), and Reliability Societies. His research focuses on Deep Learning and Deep Reinforcement Learning, AI-based Digital Twin Systems, Autonomous Intelligent Systems, and Depenability and Fault-Tolerance of Computer Systems and Networks.



Dugki Min received a B.S. degree in industrial engineering from Korea University in 1986, an M.S. degree in 1991 and a Ph.D. degree in 1995, both in computer science from Michigan State University.

He is currently a Professor in Department of Computer Science and Engineering and Department of Artificial Intelligent (Graduate School) at Konkuk University. His research interests include Cloud and Fog computing, Distributed and Parallel Computing, Deep Learning and Deep Reinforcement Learning, Intelligent Multi-Agent Systems, Software Architecture



MinGi KYUNG received the M.S. and Ph.D. degrees in Computer science and Engineering from Konkuk University, South Korea, in 2010 and 2019. He is a Postdoctoral researcher in the Department of Computer

science and Engineering at Konkuk University. His research interests include embedded Linux systems, UAV communication optimization based on ROS2, multi-sensor integration for autonomous navigation, and performance evaluation of real-time systems.