# Edge Computing for the Internet of Things: A Case Study

## Table of Contents

* **Abstract**
* **I. Introduction**
* **II. Edge Computing : Classes and Architectures**
* **III. Enabling Technologies**
* **IV. Edge Computing for IoT Applications**
* **V. Use Case : Mobile Gaming**
* **VI. Discussion**
* **VII. Conclusion**
* **Contributions**

## Abstract

### 배경

센서·액추에이터 등 **IoT 단말의 데이터가 폭증**했고, 현재 대부분이 **원거리 클라우드 데이터센터**에서 처리됩니다. 이로 인해 **네트워크 대역폭 부담**과 **통신 지연**이 병목으로 부각됩니다.

### 제안

본 논문은 **상호작용형 애플리케이션**(예: AR/VR, 실시간 3D)에서의 **센서 스트림 활용**을 위해 **엣지 컴퓨팅**을 제안합니다.

### 기여

1. **엣지 컴퓨팅 아키텍처·플랫폼 분류 및 서베이**를 제시하고, **엣지에 적합한 IoT 시나리오**를 정리합니다.
2. **모바일 게임(고연산 3D)**을 대표 사례로, **여러 배치 시나리오에서의 지연**을 실험적으로 비교하여 **엣지 및 핵심 기술**의 효과를 평가합니다.

### 방법

* **자원 집약적 3D 애플리케이션**을 기준으로, **클라우드 단독 vs 엣지 포함** 등 **다양한 배치**에서의 **응답 지연**을 측정·비교합니다.

### 결과

* 엣지 컴퓨팅은 **가상/증강 현실** 등 **엄격한 지연 요구**를 가진 애플리케이션에서 **요구 성능 충족에 필수**임을 보입니다.

### 결론

* **현재 엣지 플랫폼으로 가능한 범위**와 **향후 기술 진화가 IoT 배치에 미칠 영향**을 논의합니다.

### Keywords

**edge computing**, **fog computing**, **Internet of Things**, **mobile gaming**

## I. Introduction

### 배경과 문제

* 모바일·IoT 단말(센서, 스마트폰, 웨어러블)의 데이터 생성이 급증하는 반면, **연산·에너지 제약**이 큼.
* 이를 완화하기 위해 **클라우드 오프로딩**이 널리 사용되지만, 대규모 **원거리 데이터센터** 의존으로
* **높은 지연(latency)**
* **클라우드 구간 네트워크 부하 증가** 문제가 발생.

### 엣지/포그 컴퓨팅의 개념과 효과

* **정의**: 네트워크 **엣지(단말 인접 지점)**에 연산·저장 자원을 배치하여 단말과 **가까운 곳에서 처리**.
* **효과**
* **지연 감소**: 단말에서 **한 홉** 인근에서 전처리/분석 가능.
* **백홀 절감**: 대용량/빈번 데이터는 엣지에서 가공 후 핵심만 클라우드로 전송 → **대역폭 절약**.
* **이동성/지리적 분산 지원**: 단말 이동·분산 IoT 서비스에 적합(차량 콘텐츠, 모바일 데이터 실시간 분석, 환경 모니터링 등).

### CDN/ICN과의 차이

* **CDN/ICN**: **비대화형(정적/캐시 가능한) 콘텐츠**를 사용자 근처에 배치·라우팅 최적화.
* **엣지 컴퓨팅**: **연산 능력**을 제공하여 **대화형/실시간 애플리케이션**과 **모빌리티**까지 지원.
* 예: IoT 데이터는 캐시 가능하더라도, **엣지 서버에서 상호작용/추론/제어** 수행 가능.

### 프라이버시·에너지 관점 이점

* **프라이버시**: IoT 데이터가 **엣지 노드 내에서 저장·처리**되어 **민감 정보 사전 제거** 후 클라우드 전송 가능.
* **에너지**: **근거리 오프로딩**으로 단말의 **배터리 소모 감소**.

### 본 논문의 기여

1. **엣지 컴퓨팅 아키텍처·플랫폼 분류 및 서베이**, 그리고 **엣지에 적합한 IoT 애플리케이션 시나리오** 제시.
2. **모바일 게임**을 대표 사례로 **엣지·핵심 기술(예: 오프로딩)의 성능 평가** 수행 → **만족스러운 QoE 달성에 엣지가 필수**임을 실증.

* 모바일 게임은 **초저지연·신뢰 통신**과 **단말 센서 데이터**(위치 등)를 활용(예: Pokémon Go, Ingress).
* 더 넓게는 **복잡 3D 렌더링**이 필요한 **AR/VR** 등에도 일반화 가능.

### 논문 구성 안내

* **Section II**: 엣지 플랫폼 **분류/아키텍처**.
* **Section III**: 엣지의 **핵심 구현 기술(Enablers)**.
* **Section IV**: **IoT 요구사항**과 엣지의 **이점**.
* **Section V**: **모바일 게임 성능 평가 결과**.
* **Section VI**: 결과와 **현 기술과의 연계 논의**.
* **Section VII**: **결론 및 향후 연구 방향**.

## II. Edge Computing : Classes and Architectures

엣지 컴퓨팅 아키텍처는 ‘엣지’의 정의·참여 노드·용어가 문헌마다 다르다. 특성에 따라 **세 가지 클래스**로 분류한다.

### II-1. Resource-Rich Edge (Cloudlets)

* **핵심 아이디어**: 최종 단말에 1~몇 홉 거리의 네트워크 지점에 **고성능 서버(클러스터)**를 배치하여 저지연 처리를 제공.
* **대표 구현**
* **Cloudlet**(VM 기반): Wi-Fi AP/기지국 인근에 “**data center in a box**” 배치 → 다중 코어 연산·저장소·무선 LAN 제공. 인지 보조, 크라우드 비디오 분석 등 실시간 처리에 활용.
* **Micro Data Center / Micro Cloud**: 수천 대 서버를 가진 소형 데이터센터 또는 AP/기지국/펨토셀/차량에 소수 서버를 배치하여 인터랙티브 앱/콘텐츠 호스팅.
* **MEC (Multi-access Edge Computing)**: 이동통신 **RAN(기지국/컨트롤러)**에 IT/스토리지/가상화 자원을 통합. **실시간 무선 링크 정보**를 엣지 앱에 노출 가능. 모바일 외 **다중 액세스** 환경(실내 게이트웨이 등)으로 확장.
* **장점**: 낮은 지연, 높은 처리량, 무선 상태 정보 활용, 서비스 근접성.
* **고려사항**: 배치·운영 비용, 위치 선정, 규모 확장/오케스트레이션.

### II-2. Heterogeneous Edge Nodes

* **핵심 아이디어**: 서버·라우터·AP·셋톱박스·**스마트폰/차량 등 단말까지** 포함한 **이기종 자원**을 가상화로 묶어 협력 처리.
* **대표 구현**
* **Fog Computing**: 리소스 풍부 서버부터 경량 엣지 라우터·AP·단말까지 **높이 가상화된 계층**으로 구성. 다양한 **무선 액세스 기술**을 포괄.
* **단말 협력(Local Cloud)**: 인근 단말들이 **자발적 협력**으로 로컬 클라우드를 형성, **로컬 리소스 코디네이터**가 태스크 할당(단말/백엔드 클라우드로 분산).
* **저전력 미니 클러스터**: Raspberry Pi 등 **소형·저전력 노드**를 묶은 **휴대형 미니 클라우드**(실내/야외 배치).
* **장점**: 자원 활용 극대화, 배치 유연성, 네트워크/전력 제약 환경에서의 탄력성.
* **고려사항**: 이기종 관리 복잡도, 신뢰/보안, 이동성·가용성 변동, QoS 보장.

### II-3. Edge–Cloud Federation

* **핵심 아이디어**: **엣지 자원과 중앙 데이터센터**를 연합하여 위치·부하·지연·대역폭 조건에 따라 동적 오케스트레이션.
* **대표 구현**
* **Edge Cloud**: 엣지 앱이 **근거리 처리+원거리 클라우드**를 함께 사용(예: 실내 3D 위치/영상 모니터링).
* **프라이빗–퍼블릭 클라우드 연합**: 엣지 노드가 연합을 **동적 오케스트레이션**하여 실행 태스크 극대화.
* **분산 실행 존**: 인터넷 전반(라우터/AP/기지국 등)에 분산된 **지리적 실행 영역**에 서비스 배포.
* **미러링 기반 엣지 AI**: 공개 클라우드 서비스/레포지토리를 엣지에 **미러링**하여 **저지연·저대역폭** 이미지 분류 등 ML 작업 수행.
* **장점**: 지연·대역폭·비용·규모의 **동적 최적화**, 기존 클라우드 생태계 재사용.
* **고려사항**: 연합 간 **상호 운용성**, 데이터 이동/일관성, 보안/정책, 오케스트레이션 복잡도.

## III. Enabling Technologies

5G 전환과 함께 초저지연·고신뢰 통신, 스펙트럼 부족, 에너지 효율, 이기종 단말 데이터 폭증에 대응해야 한다. NFV/SDN은 네트워크 기능·서비스를 **프로그래머블**하고 **유연**하게 배치하게 하며, 이는 **엣지 컴퓨팅**의 핵심 기반이 된다.

### III-1. Virtualization

* **개념**: 단일 물리 서버에서 여러 독립 인스턴스(주로 VM)를 **격리** 실행.
* **VM vs 컨테이너**
* **VM**: 하이퍼바이저+게스트 OS → **강한 격리**, **높은 오버헤드**.
* **컨테이너**: 호스트 OS **공유**(커널 격리) → **경량**, **빠른 기동**, **우수한 성능**.
* **마이그레이션**
* VM/컨테이너를 타 서버로 이동해 **에너지 절감**, **부하 적응**, **사용자 이동성** 대응.
* **라이브 마이그레이션**으로 **중단 시간 최소화**.
* **엣지 의의**: 엣지 노드 간 워크로드 **유연 이동/확장** 가능. NFV/SDN의 **기반 기술**.

### III-2. Network Function Virtualization

* **NFV (Network Function Virtualization)**  
  네트워크 기능을 **범용 하드웨어**에서 **소프트웨어 모듈(VNF)**로 실행 → 전용 장비 의존 감소, **적절한 위치(엣지 포함)**에 유연 배치.
* **SDN (Software-Defined Networking)**  
  **제어 평면**과 **데이터 평면** 분리, **중앙 논리 컨트롤러**로 정책/경로를 **소프트웨어적으로 관리** → **유연성·민첩성** 향상.
* **조합 효과(엣지)**  
  수요 급증(예: 지역 IoT 트래픽)에 맞춰 **가상 리소스 자동 증설/오케스트레이션**, **정책·라우팅** 유연 변경 → **신규 서비스 신속 출시**, **운영비 절감**.

### III-3. Computation Offloading

* **개념**: 리소스 제약 단말의 연산/저장을 **클라우드 또는 엣지**로 위임하고 결과만 수신.
* **효과**
* 단말 **배터리 소모 감소**, 고복잡 연산을 단말 사양과 무관하게 수행.
* **엣지 오프로딩** 시 지연·에너지 비용 **추가 절감**.
* **활용 예**: 모바일 게임, m-러닝, **자연어 처리(NLP)**, 모바일 헬스케어 등 **고연산 애플리케이션**을 저사양 단말에서도 구동.

## IV. Edge Computing for IoT Applications

### IoT와 엣지의 궁합

IoT 단말(센서·스마트폰·웨어러블·머신)은 **연산·전력 제약**이 크며, 생성 데이터는 점점 **대역폭 집약적**이고 **지리적으로 분산**됩니다. 엣지 컴퓨팅은 단말 **근접 위치**에서 데이터를 처리·저장해 **지연을 줄이고** **백홀 부하를 완화**하며 **이동성**을 지원합니다.

### IoT 특성별 엣지의 필요성

* **저지연 통신 필수**: 커넥티드카, 모바일 게임, 원격 헬스, 물류, 산업 제어 등은 **단말 생성 데이터의 실시간 처리**가 핵심.
* **대역폭 집약 데이터 근접 처리**: 감시카메라·순찰차·사용자 단말의 **영상/센서 데이터**를 **한 홉 인근**에서 분석 → 클라우드 전송량 감소, 예) **재난·공공안전**에서 위험 지역 영상을 현지 처리.
* **지리적 분산 처리**: 센서 네트워크 기반 사례(예: **도로변 RSU의 충돌 회피**)는 차량·보행자 **위치/속도/가속도**를 **로컬**에서 처리해야 저지연 달성.
* **단말 이동성**: 단말 이동에 맞춰 **가상화 자원 마이그레이션**을 수행, **QoE** 유지.

### 상호작용형(인터랙티브) 애플리케이션의 부상

* **디지털–물리 세계의 실시간 결합**: 센서/사용자 입력과 **인공 3D 시나리오**의 융합(예: 사람의 제스처/동작을 가상 환경에 실시간 반영).
* **지각–행동 루프의 시간 제약**: 사람의 반응시간은 상황에 따라 **수 ms 수준**까지 요구될 수 있음. **AR/VR**은 특히 **극저지연·고신뢰**가 필수.
* **모바일 AR 게임 사례**: Pokémon Go 등은 **사용자 위치·센서**를 활용, 향후 **웨어러블 센서** 연계를 통해 더 **몰입·상황인지형**으로 진화. **멀티유저**일수록 지연 영향이 큼.

### 왜 클라우드만으로는 부족한가

* 클라우드는 연산·저장 자원이 풍부하지만 **물리적으로 멀어** **왕복 지연**이 커지고, **네트워크 링크 부담**이 증가 → **인터랙티브·실시간** 용도에 **QoE 저하** 발생.

### 엣지가 제공하는 해법

* **사용자 근접 연산/저장**으로 **지연 단축**.
* **근원지 전처리**로 **대역폭 절감** 및 **프라이버시 보호**(민감정보 선제 제거 후 전송 가능).
* **가상화·오프로딩·마이그레이션**을 통해 **이동성**과 **지리적 분산**을 견딤.

## V. Use Case : Mobile Gaming

* **목적**: 엣지 컴퓨팅이 **상호작용형 3D 애플리케이션(모바일 게임, AR/VR)**의 **지연** 요구를 충족하는지 실험적으로 검증.
* **게임**: **Neverball**(오픈소스 3D 아케이드) — 단말이 입력(기울기 등)을 서버로 전송 → 서버가 렌더링 후 **비디오 스트리밍**으로 단말에 전송.
* **핵심 지표**: **Response Delay = PD(서버 처리) + OD(클라이언트 디코드/재생) + ND(네트워크 RTT)**.

### V-1. Testbed Setup

* **플랫폼**: **GamingAnywhere**(오픈소스 클라우드 게이밍).
* **클라이언트**: Google **Nexus 5** (Android 5.1.1).
* **서버(로컬 워크스테이션)**: Intel Xeon E3-1230(4C), RAM 16GB, **NVIDIA Quadro 2000 ×2**.
* **네트워크**: **Wi-Fi** / **LTE(NetLeap 4G, Nokia)** — 동일 대학망 사용.
* **서버 배치 시나리오(3)**  
  1) **Edge**: LTE 기지국 **공위치(co-located)** 또는 동일 무선망 **근거리** 워크스테이션.  
  2) **전용 클라우드(CSC cPouta, Kajaani)**: OpenStack 기반 **핀란드** 전용 연구 클라우드.  
  3) **퍼블릭 클라우드(AWS EC2)**: **프랑크푸르트·아일랜드** 리전(GPU 인스턴스).
* **가상화 구성(3)**: **Bare Metal(B)** / **Container(C, Docker 1.10.3)** / **VM(V, QEMU 2.5.0)** — **GPU 1개 패스스루**.
* **EC2 인스턴스**: g2.2xlarge(1 GPU, 8 vCPU) / g2.8xlarge(4 GPU, 32 vCPU), **Dedicated Instances** 사용.
* **방법론**: 섹션별 4회 반복, **사전 기록된 입력 재생**으로 일관성 확보, **30 FPS / 4.5 Mbps**로 스트리밍.

### V-2. Experimental Results

#### 1) 네트워크 지연(ND)

* **핀란드 내(엣지·전용 클라우드)**: **< 25 ms**, 엣지(LTE)에서는 **< 20 ms** 달성(현행 무선 통신의 상한선 수준).
* **퍼블릭 클라우드(원격 리전)**: **≥ 50 ms** (최소 2배 ↑). 거리(약 **1,500–2,000 km**)가 커질수록 ND 증가.
* **Wi-Fi vs LTE**: 평균 **Wi-Fi가 더 짧지만 분산(지터) ↑**, **LTE는 지터 ↓**로 **스트리밍**에 더 안정적.

#### 2) 처리 지연(PD) & 재생 지연(OD) vs 해상도 & 가상화

* **컨테이너 ≈ 베어메탈**: 해상도와 무관하게 **동일 수준** 성능.
* \*\*VM은 PD 약 \**+30%*\* 증가\*\* → 1280×720에서 **컨테이너는 30 FPS 유지** 가능, **VM은 미달**.
* **1920×1080(Full HD)**: **모든 구성**에서 **목표 30 FPS 불가**.
* **OD(클라이언트)**: FHD 미만에서 **PD와 비슷한 크기**, 해상도↑에 따라 증가하나 **평균 25 ms 이내**.
* **분산은 OD > PD**, 해상도↑ 및 **VM 사용 시** 분산 더 큼.

#### 3) 클라우드의 추가 연산자원이 ND를 상쇄할 수 있는가?

* **EC2 g2.2xlarge**: 800×600, 1280×720에서 **베어메탈보다 PD↓**, 그러나 **1920×1080에서는 +5 ms**(베어메탈 대비).
* **EC2 g2.8xlarge**: 모든 해상도에서 **베어메탈보다 PD↓**.
* **핵심 원인**: PD의 대부분이 **인코딩 지연**(렌더링보다)에서 발생 → GPU 늘려도 **체감 이득 제한적**.
* 예) 800×600에서 g2.2xlarge는 **베어메탈 대비 −12 ms**, 1920×1080에서 **−8 ms** 수준.

### 해석 및 시사점

* **근접 배치(엣지)**가 **ND를 압도적으로 단축** → **총 응답지연**에서 결정적.
* **경량 가상화(컨테이너)**는 **베어메탈급** 성능, **하이퍼바이저형 VM**은 **유의미한 오버헤드**.
* **FHD 스트리밍은 병목(인코딩·디코딩)**으로 **30 FPS 달성 어려움** → **해상도/코덱/파이프라인 최적화** 필요.
* **더 강력한 클라우드 GPU**도 **원격 ND**를 **완전히 상쇄**하지 못함 → 상호작용형 3D/AR/VR에는 **엣지 우선 배치**가 유리.

## VI. Discussion

### 결론

* **엣지 배치가 QoE 달성의 사실상 유일한 해법**: 사용자와 **근접(접속망 엣지)** 에 연산 자원을 두어야 상호작용형 서비스의 응답지연을 만족.
* **지연 임계값**: 일반 인터랙션은 **< 150 ms** 수용 가능하지만, **빠른 상호작용은 70 ms 초과 시 체감 품질 악화**. 퍼블릭 클라우드(원격 리전) 기반에선 해당 목표 달성이 어려움.
* **엣지 vs 클라우드의 본질적 차이**: 클라우드의 **추가 연산능력**으로 **처리지연(PD)** 을 줄여도, **네트워크지연(ND)** 을 상쇄하기에 **불충분**. 반면 **모바일 네트워크 엣지** 배치는 HD 해상도에서도 **< 70 ms** 달성이 가능.

### 확장성·멀티유저 시나리오에서의 함의

* **세션 격리 모델**: 모바일(클라우드) 게이밍과 유사하게 **사용자당 1 VM/컨테이너** 모델을 엣지에 적용해도 기본 관찰치는 동일하게 유지.
* **데이터센터(엣지) 내부 지연 최적화**: **플로우 제어**로 마이크로서비스 간 서비스 지연을 축소 가능.
* **스케일 아웃 전략**: **가상화·NFV·SDN**을 통해 이용자 증가에 따라 **게임 모듈(VNF) 자동 배치**, **네트워크 정책/라우팅 동적 제어**, **인코딩 파라미터 자동 튜닝** 수행.
* **오케스트레이션**: 게임 요구사항에 맞춰 **고성능 VM/컨테이너**를 선택 기동, **라이브 마이그레이션**으로 이동 사용자에 대한 세션 연속성 유지.

### 인프라 트렌드와 과제

* **엣지 노드의 증가**: 통신사·콘텐츠/서비스 사업자의 엣지급 연산 장비가 빠르게 확산 → **노드 선택·관리**가 핵심 이슈로 부상(배치 최적화, SLA, 비용).
* **자원할당 최적화의 복잡성**: 지연·대역폭·비용·에너지·이동성 제약을 동시에 만족해야 하는 **실시간 최적화** 문제로 고도화.

### 기술적 한계와 5G의 목표

* **현재 한계**: 무선/컴퓨팅 기술의 한계로 **10 ms 이하 응답시간**은 근본적 혁신 없이 달성 곤란.
* **5G 목표**: **1 ms** 저지연을 지향하지만, **접속망 이후 구간**의 지연 요인 최소화에는 여전히 불확실성 존재.
* **확실한 사실**: 이러한 야심 찬 목표에 접근하기 위해서는 **엣지 컴퓨팅이 필수**.

## VII. Conclusion

* **결론**: 엣지 컴퓨팅은 빠르게 상호작용하는 모바일 게임 같은 신흥 IoT 애플리케이션에 **필수적**이다.
* **지연의 본질**: 지역 데이터센터는 네트워크 지연을 *상당히* 줄여주지만, **사용자 체감 품질(QoE)** 을 만족시키기엔 부족하다.
* **클라우드 스케일의 한계**: 클라우드 서버의 연산 성능을 늘려도 **네트워크 지연 증가를 상쇄할 수 없다**.
* **실무적 시사점**: **엣지에 제한적 자원만 배치**해도 대상 사례(모바일 게임)의 **체감 품질이 유의미하게 개선**된다.
* **향후 연구 방향**
* 엣지 컴퓨팅 기반 모바일 게임에 대한 **대규모 실증 평가**, 특히 **온라인 멀티플레이** 환경을 포함한 연구
* **애플리케이션 시나리오별 다양한 엣지 아키텍처** 간 **성능 비교 연구**
* **연구 기대**: 본 연구가 **엣지 컴퓨팅 기반 응용 분야의 후속 연구를 촉진**하길 기대한다.

## Contributions — Authors’ Statement

* **분류·서베이**: 엣지 컴퓨팅 **아키텍처와 플랫폼**을 분류·조사하고, 엣지의 이점을 얻는 **핵심 IoT 응용 시나리오**를 서술.
* **실증 평가**: **모바일 게이밍**을 대표 사용례로, 엣지 및 그 **기반 기술**의 실험적 평가 수행.