EdgeSys: A Decentralized Cloudlet Framework for Edge Computing

USC Viterbi School of Engineering
Ming Hsieh Department of Electrical and Computer Engineering

Yu-An Chen, John Paul Walters and Stephen Crago
University of Southern California/Information Sciences Institute
chen116@usc.edu, jwalters@isi.edu, crago@isi.edu

Introduction:

In edge computing, local compute resources and gateways form cloudlet clusters [1] that provides computation offloading for mobile devices. Many works have been done in resource allocation and task scheduling for static cloudlet clusters, but there is a need for a scalable edge computing framework that is resilient to machine failures and cluster reconfiguration [2]. EdgeSys is designed to be a decentralized cloudlet framework that can adjust accordingly to the dynamic behavior of a cloudlet cluster to maintain applications' throughput and latency requirements.

Challenges:

- Load Balancing
- Application Management
- Fault Tolerance

Cloudlet Cluster Attributes:

- Device Mobility
- Dynamic Deployment
- Cloudlet Failures

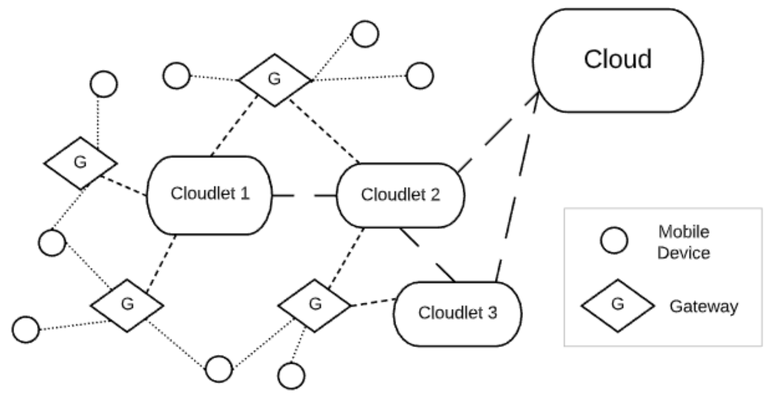

Figure 1: Example Cloudlet Cluster

[1] Mahadev Satyanarayanan. 2017. The Emergence of Edge Computing. Computer 50, 1 (January 2017)
[2] Yousefpour, Ashkan et al. "All One Needs to Know About Fog Computing and Related Edge Computing Paradigms: A Complete Survey." Journal of Systems Architecture 98 (2019)

Queue-Based Application Model:

- Offloaded device data forwarded to the application's input queue based on topic.
- Application throughput can be adjusted through horizontal elasticity.
- Queues' statistic can be used for load balancing and application managements.

The EdgeSys Framework focuses on reducing the time between the device data arrives at the gateway and the application, denoted as *Waiting Time* in Figure 3.

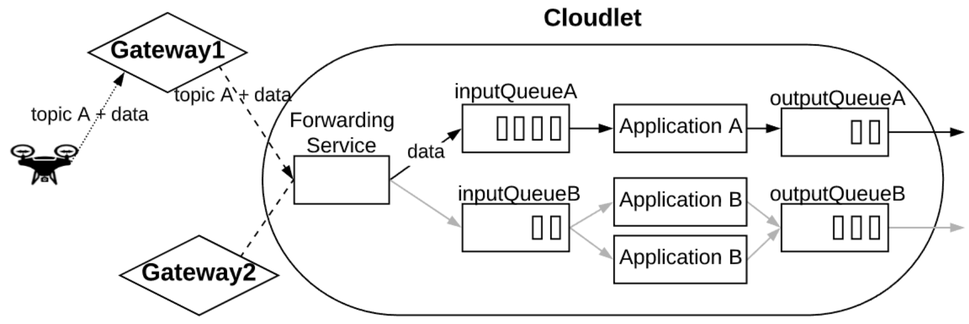

Figure 2: Queue-Based Application Deployed in a Cloudlet

Decentralized EdgeSys Framework:

Load Balancing:

- Each cloudlet and gateway uses randomized max weight algorithm to determine which cloudlet to forward the incoming device data.

Application Management:

- Each cloudlet executes burstiness aware horizontal elasticity to adjust number of application instances based on workload.

Fault Tolerance:

- Each connection is actively monitored, and device data is rerouted to other cloudlets when a connection or cloudlet failure is detected.

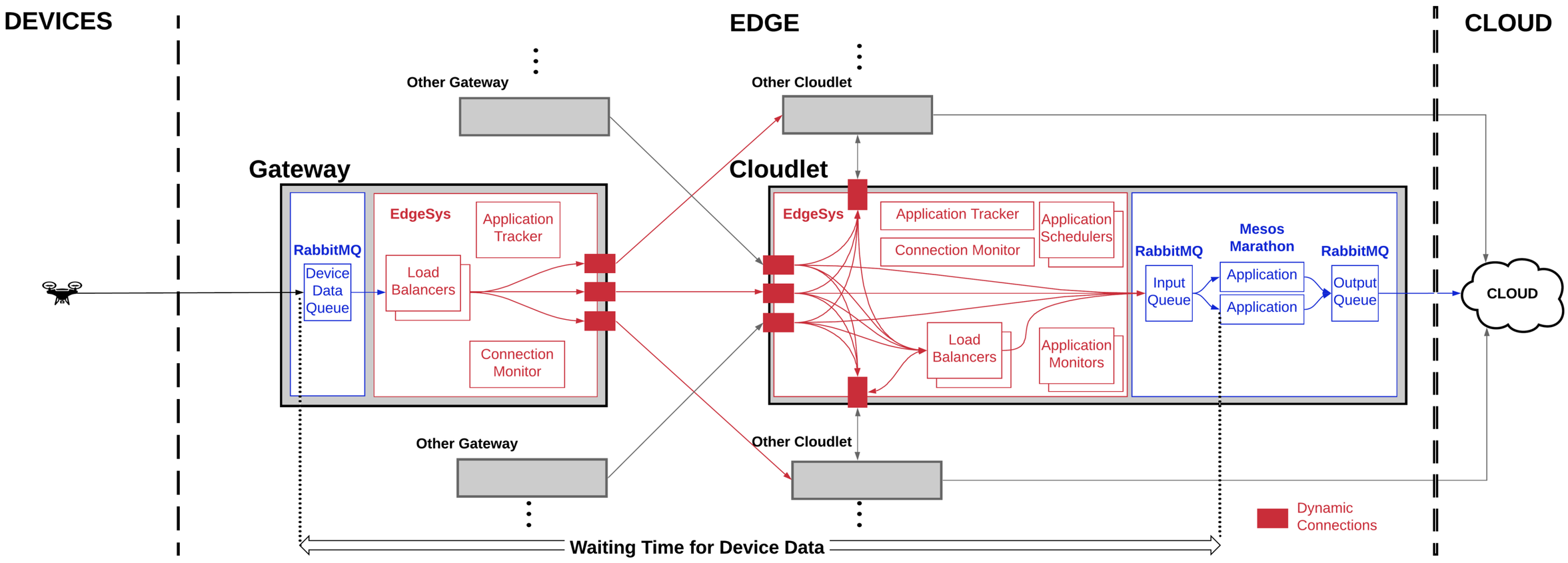

Figure 3: Device Data Path in EdgeSys

Experiment Result:

Drone Fleet Operation:

- **Applications:** 2 synthetic applications
- **Simulated Devices:** 30 drones' flight traces [2]
- **Cluster Topology:** 7 gateways and 6 cloudlets

[2] Vásárhelyi, Gábor, et al. "Optimized flocking of autonomous drones in confined environments." *Science Robotics* 3.20 (2018)

Methodology:

1. At t=0 second, deploy 15 devices which continuously offload data to the 2 applications at the rate of 30Hz and 15Hz, respectively.
2. At t=70 seconds, force shutting down cloudlet 3.
3. At t=140 seconds, restart cloudlet 3.
4. At t=200 seconds, deploy additional 15 devices.

Figure 4: Drones Flight Path

Figure 5: Gateway/Cloudlet Topology

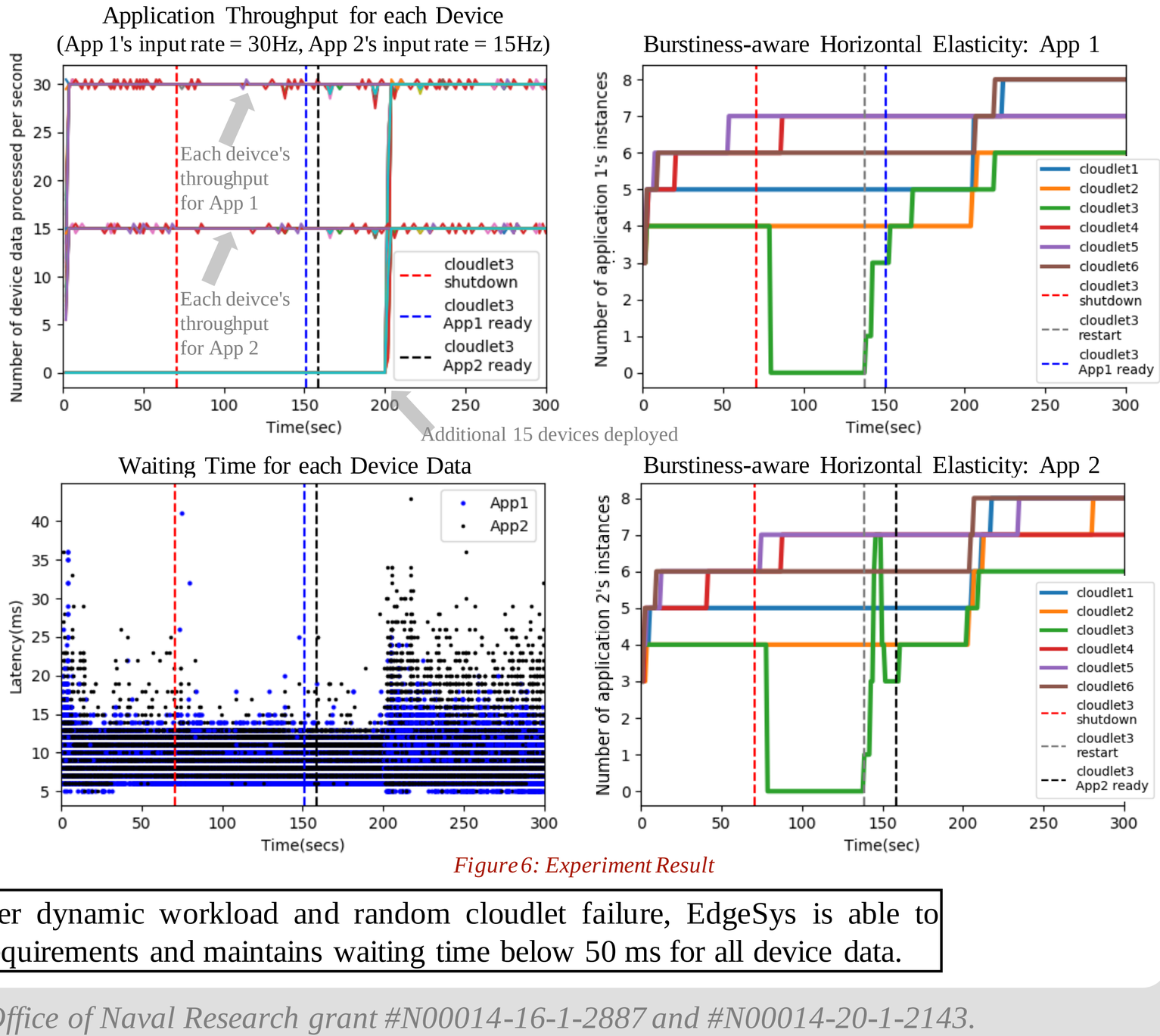

Figure 6: Experiment Result

The experiment shows that under dynamic workload and random cloudlet failure, EdgeSys is able to satisfy applications' throughput requirements and maintains waiting time below 50 ms for all device data.

This work was supported by the Office of Naval Research grant #N00014-16-1-2887 and #N00014-20-1-2143.