Migration-driven Resilient Disaster Response Edge-Cloud Deployments

Xiaojie Zhang, Saptarshi Debroy
City University of New York
Email: *xzhang@gradcenter.cuny.edu, saptarshi.debroy@hunter.cuny.edu*

***Abstract*—Cloud based incidence response systems suffer from lack of network connectivity to offload compute intensive mission-critical applications to remote cloud. Thus, the next-generation incidence response solutions are becoming more edge-cloud based where computational resources are available closer to the disaster site. However, frequent and dynamic unpredictabilities or *fluctuations* generated in such edge-cloud deployments (e.g., using wireless spectrum in unlicensed manner) adversely impact the performance of mission-critical, real-time applications which often demand strict performance guarantees. Such *fluctuations* cause severe performance degradations to mission-critical applications that use such channels. In this paper, we propose an intelligent yet lightweight application task assignment (end-user device to edge-cloud) and application migration scheme (between edge-cloud resources) that can help mission-critical applications avoid impending *fluctuations* and improve system resilience. The proposed scheme implements Largest Bandwidth Largest Job-First Fit (LBLJ-FF) algorithm as a Unified Resource Broker (URB) service that optimizes transmission cost and migration overhead. The algorithm is light-weight and fast converging in reacting to sudden *fluctuations*. We demonstrate the performance of the proposed scheme through a realistic simulation that uses real *fluctuation* dataset. The results demonstrate the existence of an optimal trade-off point between transmission cost and migration overhead optimizations. The results also show the resilience of the proposed algorithm in improving job completion rate when the edge-cloud system is under intense *fluctuation*.**

***Index Terms*—Disaster response; edge-cloud; task assignment; application migration; unified resource broker.**

I. Introduction

In the wake of large-scale natural/man-made disasters, rapid situational awareness would help first responders to be more effective and efficient. In recent times, *edge-cloud* (i.e., mobile edge, fog, cloudlet) infrastructure based disaster incidence response deployments are being proposed over cloud-based deployments as unlike cloud, edge-cloud resources are available closer to the disaster site. Fig. 1 illustrates an exemplary edge-cloud based disaster incidence response scenario. Based on the mission objectives, raw sensory data are generated from end devices (e.g., humanoids, drones) that often require remote data processing and analysis (e.g., image and video processing) as the devices lack sufficient compute and storage resources. Thus, the sensory data is offloaded via wireless network to edge-cloud nodes hosted by multi-utility vehicles (MUVs) for processing where sufficient compute cores are available. If one such edge-cloud node does not possess sufficient compute resources to process the data, data is offloaded to other nodes in the pool to ensure successful processing. Finally, the processed data is fed back to first responders' handheld devices (e.g., laptops, tablets) to initiate, modify, and suspend incidence response operations based on the intelligence acquired from the processed data.

The success of incidence response missions and involved operations thus depends on the timely transfer and processing of the data applications which in turn depends on efficient

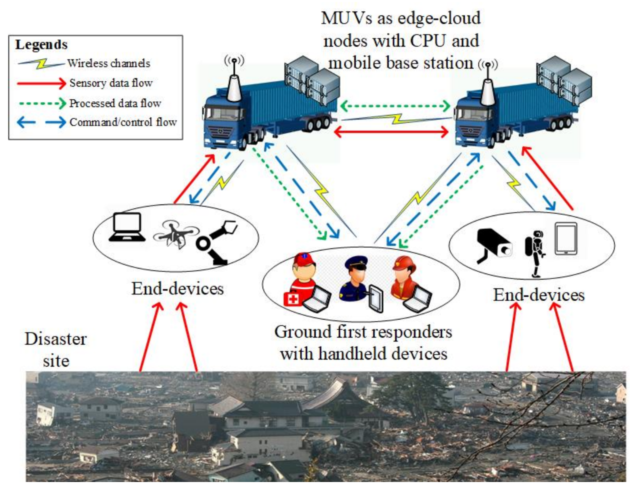

Fig. 1: An exemplary Edge-cloud deployment for next-generation disaster incidence response

network and compute resource provisioning by the edge-cloud system. However, edge-cloud resource provisioning in a large-scale disaster incidence response scenario is more challenging than cloud/edge resource management for other use cases. Firstly, data applications for large-scale disaster incidence response are unique due to their real-time and mission-critical nature. Secondly, inherent system unpredictabilities or *fluctuations* during disaster incidence response adversely impact the performance of mission-critical applications and make them vulnerable to faults. Finally, due to limited availability of edge-cloud nodes and compute cores within, traditional 'replications and redundancy' based cloud fault tolerance mechanisms cannot be applied to disaster response systems.

For example, edge-cloud systems use cognitive radio devices to utilize licensed wireless spectrum in an opportunistic manner [1] for data transfer between edge-cloud nodes and end-devices when wireless infrastructure (e.g., wifi, LTE) are partially or completely destroyed by the disaster [2], [3]. For such unlicensed access, the disaster response applications are assigned wireless channels currently unused by licensed transmission for data-transfer from end-devices to edge-cloud nodes. However, due to the temporal unpredictability of licensed transmission, the availability of channels for such mission-critical applications frequently change, thus making them vulnerable to performance degradation. At the same time, due to the limited availability of edge cloud nodes and wireless spectrum at such nodes, provisioning on-demand network and compute resources for reliable application performance without compromising system overhead is challenging.

In this paper, we propose an intelligent network and compute resource provision scheme that improves disaster response edge-cloud system resilience induced by unlicensed spectrum access without compromising system overhead. We achieve this by borrowing strategies and best practices from cyber security which deals with volume cyber attacks, such as Distributed Denial of Service (DDoS) [4]. We divide the overall network and compute resource provisioning problem into sub-problems and propose: a global application task assignment technique to edge-cloud nodes, and a local job scheduling technique to compute cores within the edge-cloud nodes. We characterize the impact of unlicensed spectrum access induced fluctuations on applications' health and propose a prediction model to estimate such fluctuations' arrival caused by licensed transmission. Our proposed global assignment technique uses this prediction information to allocate wireless channels to incoming application tasks as well as, to proactively migrate run-time application tasks to other nodes in case of an impending fluctuation to a channel associated with a node. Our scheme also uses well known partitioned multiprocessor scheduling techniques for local job scheduling within edge-cloud nodes. The proposed scheme adaptively balances data transmission cost and migration overhead optimizations while maintaining an almost 100% job completion rate. The proposed scheme implements Largest Bandwidth Largest Job-First Fit (LBLJ-FF) algorithm as a Unified Resource Broker (URB) service that is light-weight and fast converging which are essential properties of resource management in a disaster incidence response scenario.

We evaluate the performance of the overall scheme using a realistic simulation that uses real dataset for unlicensed spectrum access based fluctuation information. The results demonstrate the existence of an optimal trade-off point called balance factor that balances mutually diverging optimizations for transmission cost and migration overhead. The results show the benefits of using fluctuation prediction driven migration in improving system resilience and yet maintaining a very high completion rate. Finally, we show how the proposed LBLJ-FF algorithm better balances transmission cost and migration overhead than other genetic algorithm and random global assignment based techniques commonly used for edge-cloud based resource provisioning.

The rest of the paper is organized as follows. Section II discusses the background and problem motivation. Section III presents the related work. Section IV discusses system model. Section V proposes the global assignment and migration strategy. Section VI presents simulation results. Section VII concludes the paper.

II. Background and Motivating Example

In this section, we discuss how edge-cloud resource management in a large-scale disaster scenario is unique in terms of mission-critical applications and fluctuations induced by unlicensed access.

A. Resource Management Challenges

Upon interacting with the Fire Department of New York City Office of Emergency Operations and investigating the case files of large scale disasters (e.g., 2011 Joplin tornado, 2012 hurricane Sandy in New York, and 2015 Nepal earthquake) [2], [3], we argue that the edge-cloud resource management for disaster response are different from edge-cloud deployments for other use cases due to the following reasons:

- The end-to-end applications generated for large-scale disaster incidence response are unique due to their: a) dynamic nature and properties, b) strict performance and energy requirements, c) real-time mission-critical characteristics. Thus, depending on the unique application requirements, on-demand compute and network resources need to be allocated.
- Incidence response for large-scale disaster scenarios is also unique due to frequent occurrence of system faults, unpredictabilities or *fluctuations* that need constant monitoring. At the same time, such *fluctuations* need to be dynamically addressed in order to prevent them from adversely impacting the performance of mission-critical applications.

B. Mission-critical Disaster Response Applications

Based on our initial discussions with the Fire Department of New York's (FDNY) Office of Emergency Operations, we realized that incidence response applications and involved data-flows are unique in terms of their nature, characteristics, and performance requirements. Here we outline FDNY's future fire rescue operations for multi-storied buildings using drones and explain how video processing applications from such operations can evolve during a large scale disaster. In this operation, the fire rescue personnel will use drones attached with 1080p at 25 FPS smartphone cameras to capture videos of the incident scene. This raw video will be sent to the central FDNY video processing server using AT&T and Verizon LTE networks for 3D reconstruction. The processed data will be uploaded to Amazon Web Services (AWS) that can be downloaded by ground first responders at the ground level of the disaster scene through apps on their compatible handheld devices. The entire end-to-end data-flow is real time, i.e., each video frame from collection to consumption has strict deadline. In a large-scale disaster, the applications are recurrent and mission-critical.

C. Unlicensed Access Induced Fluctuations

In a large-scale disaster scenario with limited or no LTE connectivity, sending data to a remote AWS data center for processing will be impossible. In such a scenario, first responder personnel have to rely on local edge-cloud nodes (hosted in MUVs) for live video processing and cognitive radio based devices to use vacant LTE bands for data transfer to such nodes. Therefore, the success of such fire rescue operations during a large scale disaster will depend on the edge-cloud system's ability to ensure compute cores for the application data processing and unlicensed wireless channel allocation for transferring the data to the compute location.

However, provisioning resources becomes even more challenging as large-scale disaster environments are fraught with unpredictable changes or *fluctuations*. One of the major fluctuations arise from disaster incidence response systems' opportunistic/unlicensed use of licensed spectrum by cognitive radio devices to network resources (i.e, wireless channels) [1]. For unlicensed access, regulation imposed by the FCC mandates that if such access interferes with licensed transmission on any spectrum channel, the former needs to suspend transmission immediately. Now the nature of licensed transmission being unpredictable [5], [6], mission critical applications that use cognitive radio based unlicensed access become vulnerable.

Fig. 2 shows the simulation results of the impact of such fluctuations on end-to-end data rates of applications that use unlicensed access for data transfer. We use publicly available data from RWTH Mobnets [7] dataset that records temporal spectrum usage pattern of licensed devices on different bands.

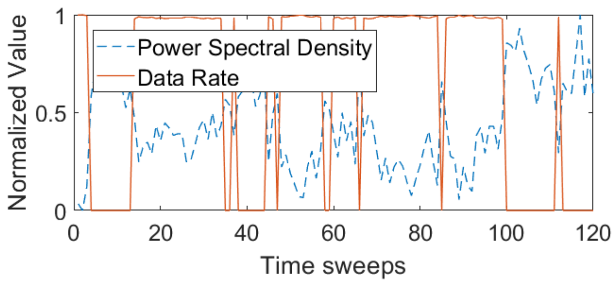

Fig. 2: Unpredictability of spectrum usage by licensed transmitters on a band and its impact on application's data rate (1 sweep = 1.8s)

Fig. 2 shows the unpredictability of power spectral density (PSD) of licensed usage over time for a particular spectrum band. The figure also shows application's data rate fluctuation caused by such unpredictability when the application uses the same band for data transfer.

III. Related Work

The related work can be broadly divided into two categories: Mobile edge computing applied for disaster response and service migration and job scheduling for mobile edge computing.

Mobile Edge Computing (MEC) based wide-area analytic solutions have been recently proposed for offloading computationally heavy tasks to mobile edge nodes to reduce device energy consumption. Among these, [8], [9] are notable for exploring resource allocation methods for heterogeneous and computation-heavy tasks, such as image and video processing. LAVEA [8] proposes task offloading to edge nodes driven by local and remote execution time and network transmission delay. An energy-efficient offloading framework has been studied in [9] where each task is either local-only or offloadable. However, both papers address whether tasks should be executed remotely or locally without considering heterogeneity of edge cloud nodes and service migration caused by such heterogeneity. Works such as [10], [11], [12], [13], [14] propose wide-area analytics for emergency operations using edge resources. However, these mainly focus on edge vs. cloud computation under mostly static environments and may not be extended for disaster response operations due to mostly real-time data-flow requirements and presence of fluctuations.

In order to counter the dynamism of a system environment, service migration methods in MEC have been proposed. Most of the notable works in this category [15], [16] study Markov Decision Process (MDP) based dynamic service migration. SEGUE [17] proposes a QoS-aware service migration that uses MDP to solve the "when" and "where" of service migration problem. Authors in [18] propose a novel mobility-aware online service placement framework that uses long-term cost budget constraint to control decisions. Whereas, authors in [19] propose a multi-component application placement and component migration technique in MEC suing a heuristics algorithm. However, none of these works consider realistic computational constraints caused by resource provisioning and scheduling in highly dynamic environment and may fail when deployed under high-load systems. Although, sporadic task scheduling especially on multiprocessors [20], [21] has been extensively studied with proven efficiency, most of these were proposed before the advent of MEC. Thus, such works fail to reproduce promising results when put under highly dynamic environment with frequent system fluctuations.

IV. System Model

In this section, we discuss the architecture of the proposed edge computing framework along with system, application and fluctuation model. Let $\mathcal{K} = \{1, 2, 3, ..., K\}$ denote the set of edge-cloud nodes that are equipped with M identical compute cores $\mathcal{M} = \{1, 2, 3, ..., M\}$. They provide remote processing resource to a set of end-user nodes $\mathcal{N} = \{1, 2, 3, ..., N\}$ (handheld devices of first responders, drones, robots etc.) which do not have sufficient processing capabilities. The entire system is managed by a URB whose job is to assign user applications to edge-cloud nodes for remote execution. In the absence of wireless infrastructure, such information exchange is achieved by utilizing predefined control channels. The end-devices offloads data to edge-cloud locations selected by the URB using unlicensed spectrum assigned by the URB. We define assignment O_π produced by policy π as a combination function:

$$O_\pi(G, L) : \mathcal{K} \times \mathcal{N} \times \mathcal{M} \rightarrow \{0, 1\} \tag{1}$$

where G is global assignment (i.e., application to edge-cloud node mapping) and L is local scheduling (i.e., individual application jobs to core within the edge-cloud node). A single element $o_{n,k,m}$ inside the matrix O_π is set to 1 if the user application τ_n is executed on core m within edge-cloud node k; otherwise $o_{n,k,m} = 0$.

A. Application Model

Due to the real-time and recurrent nature of disaster response applications, we use Sporadic Task Model [20] to model compute-intensive applications that are repetitive and have strict yet similar deadlines. In this paper, a user application is described as a task τ_n which contains a set of recurrent jobs (e.g., a video is divided into multiple frames and each job is is processing one frame). Each task is represented as the tuple (E_n, W_n, T_n, D_n) which is called the task profile, where E_n denotes the worst-case job execution time, W_n denotes job data size, T_n denotes minimum interval between recurrent jobs, and D_n denotes the minimal latency requirement (deadline) of each job. For the rest of the paper, task and application will be used inter-changeably.

B. Edge-cloud Node

In this model, edge-cloud nodes are equipped with a M-core computing platform that can run M jobs at a time (i.e. same type of MUVs are equipped with M identical computing resources). For disaster response, we assume that end-devices and edge-cloud nodes (mounted on MUVs with mobile base stations) use unlicensed channels for data transmission when license transmission is absent on those channels. In this paper, we use Orthogonal Frequency Division Multiple Access (OFDMA) communication scheme. Specifically, each edge-cloud node is pre-allocated with a channel of fixed bandwidth B that can be equally divided into multiple orthogonal sub-channels. The coordination between the end-devices and edge-cloud nodes for channel decision is configured by other OFDMA-based Dynamic Spectrum Access (DSA) models [22] that is beyond the scope of this paper. Thus, given such coordination, the objective of the global assignment is to minimize the transmission cost of using unlicensed sub-channels for application data transfer between end-user and edge-cloud node pair. However, only a few sub-channels are available at any instance based on the status of licensed transmission on those sub-channels at that instance. We define $\theta_{t,k}$ as the number of available sub-channels at any time instance t at an edge-cloud node k where,

$$b \times \overbrace{\sum_{f \in \mathcal{F}_k} \underbrace{\gamma_{t,k,f}}_{\textbf{availability indicator}}}^{\theta_{t,k}} \leq B \tag{2}$$

Here $\mathcal{F}_k$ denotes the set of such sub-channels and b denote the bandwidth of each sub-channel. To simplify the model, we assume that the transmission power of end-devices and the channel gain over sub-channels are the same. We also apply the concept of channel aggregation where generated tasks are allocated a collection of sub-channels (during global assignment) that are associated with one edge-cloud node and at the same time are not being used by licensed transmission. This, the aggregated data rate for individual end-device is defined as,

$$R_{t,k} = \frac{\theta_{t,k}}{\sum_{k=1}^{K} \sum_{m=1}^{M} o_{n,k,m}} \times b \log_2(1 + SNR) \tag{3}$$

Evidently, with the arrival of licensed transmission, $R_{t,k}$ will decrease. This will trigger the URB to de-allocate the unavailable sub-channels and allocate new sub-channels to the end-device either associated with the same edge-node or a different edge-node resulting in migration. We allow edge-cloud nodes to suspend a run-time task when network bandwidth and/or computing resource become scarce. Details of the transmission cost optimization will be discussed in Section V.

C. Local Scheduling

Local scheduling of jobs to compute cores with an edge-cloud node follows Partitioned Multiprocessor Scheduling algorithm proposed in [20] and [21] that has proved to be one of the most efficient algorithms to solve sporadic task scheduling problems. Applying this algorithm, we assign each job of a task to a compute core (within an edge-cloud node) and ensure that all the following jobs belonging to the same task can be finished by the same core before each job's deadline. When the job scheduling fails, the global assignment is considered as invalid and a new assignment of the task/application to an edge-cloud node will be required. This partitioning between global assignment and local scheduling allows us to reduce the problem size from $|(K \times M)^N|$ to $|K^N|$. In general, task utilization $u_n = E_n/T_n$ is widely used for job scheduling with *implicit-deadline model* (i.e., $T_n = D_n$) that follows the constraint $\sum_{n=1}^{N} u_n \leq 1$. However, Fig. 3 illustrates that due to the uncertainty of data transmission with unlicensed channels, recurrent tasks in remote execution will have arbitrary remaining completion time D_n^* (called *dynamic task profile*). Thus, we use an alternative approach to use task density $\lambda_n = E_n/min(T_n, D_n^*)$ as a scheduling measure used in [20]. Using this approach, the value of *dynamic task profile* can be modeled as,

$$D_n^* = D_n - W_n/R_{t,k} - max(0, W_n/R_{t,k} - T_n) \tag{4}$$

where $W_n/R_{t,k}$ is the data transfer time between end-device and edge-cloud node.

D. Spectrum Fluctuation and Prediction Model

In order to fairly study the impact of unlicensed spectrum usage induced uncertainty or fluctuations, we assume

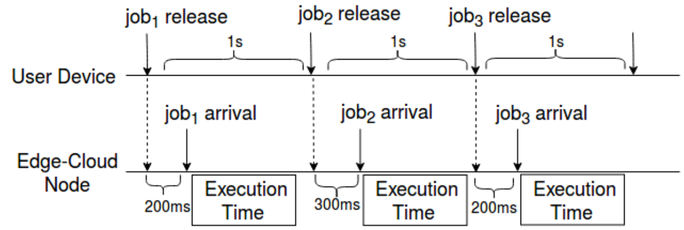

Fig. 3: Example of sporadic task in remote execution where $T_n = D_n = 1s$ that shows the delay (200ms and 300ms) in data transfer caused by channel uncertainty

the queue buffers in all edge-cloud nodes to be infinitely large and the computing capacity of each core to be the same. Furthermore, we assume that inter-edge communication utilizes reserved high bandwidth wired connectivity to transmit and receive data. This allows us to focus on uplink spectrum channels between end-devices and edge-cloud nodes, and we can treat inter-edge communication costs as invariants.

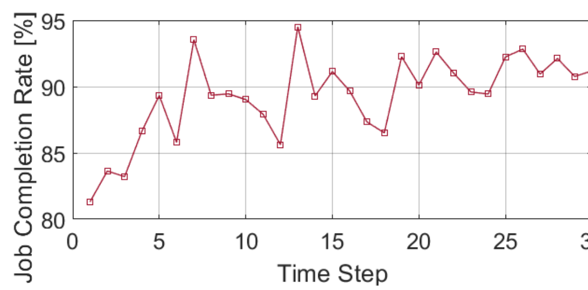

Fig. 4: Job completion rate when system orchestration do not address spectrum fluctuation for uplink data transmission

For typical sporadic task running locally, task profiles are static. However, the time cost of data transmission between end-user node and edge-cloud node using unlicensed spectrum must be considered in remote execution. Fig. 4 illustrates that the transmission delays caused by spectrum fluctuations varies over time and if not addressed, can heavily impact the task/job completion rate that in turn can cause considerable end-user experience degradation. Therefore, precise dynamic task profiles guarantee the success of job execution. However, it is impractical to build functions for $\theta_{t,k}$ in uncertain environments and subsequently to get data transfer rate and dynamic task profile. In order to make more accurate task profiles, the "Spectrum Prediction" model [23] is introduced to predict the number of available unlicensed sub-channels of a given bandwidth in near feature. Task profiles are updated after every prediction instance based on predicted $\theta_{t,k}$. Then, the URB starts a future validation process for the current assignment by checking the accumulated task density and gathering the information of local job scheduling at each edge-cloud node. In our model, a valid assignment is the one where all local job scheduling in all edge-cloud nodes succeed; otherwise, there exists some future validation failures among computing resources. Fig. 5 shows the proposed URB design and assignment process.

V. Global Assignment and Migration Strategy

The failure of future validation and the discovery of better assignments will trigger the URB to select a new global assignment and we denote such reassignment as task migrations. This is similar to how applications are migrated among virtual machines in a cloud environment when targeted by a DDoS attack. However, similar to defense against DDoS

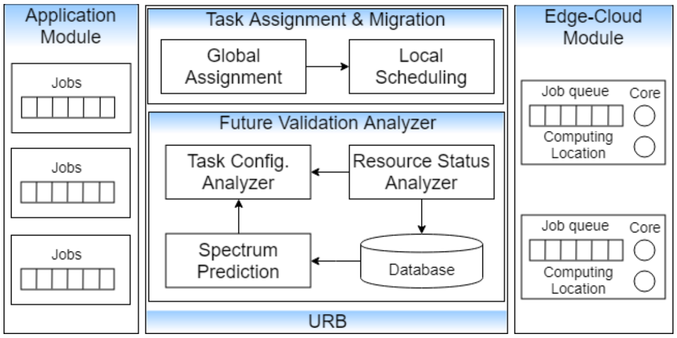

Fig. 5: Application to compute core assignment within edge-cloud nodes through URB

in cloud environments, task migrations that are too frequent can add to substantial management cost to the system. At the same time, frequent migrations will adversely impact the short term performance that might be detrimental for involved task and job success. Whereas, the alternate approach of too infrequent migrations may leave the task vulnerable to impending spectrum fluctuations. Thus, there is a need for migration optimization.

A. Assignment and Migration Process

In our model, a policy π of task assignment is selected by the URB when it receives a collection of applications/tasks $\{\tau_1, \tau_2, ...\tau_N\}$ through "Task Config. Analyzer" component described in Fig. 5. For the global task assignment process, the URB collects the static task profile of applications and gathers the statistical analysis of spectrum usage at each edge-cloud node from "Spectrum Prediction" component. Using the predicted value of $\theta_{t,k}$, the URB estimates the data transfer rate and generates dynamic task profile using equations Eqs. (3) and (4). Next, the URB calculates task density λ_n to test future validation and try to find better assignments. Both are performed by "Task Assignment & Migration" component within the URB (as shown in Fig. 5). Whenever a migration decision is made, notifications of new assignment information are sent to involved end-user devices and edge-cloud nodes through control channels. All entities that receive the notification initiate task migration. Such migrations also use high bandwidth dedicated proprietary network between edge-cloud nodes to transfer intermediate data and task run-time information. Such management operations lead to proprietary network resource consumption and delays which in this paper are defined as migration overhead.

B. Transmission Cost and Migration Overhead Optimization

In this paper, we propose a migration strategy that is carried out by the optimizations of transmission cost and migration overhead. Given a task migration that involves two consecutive assignments O^t and O^{t+1}, the transmission improvement can be measured as follows,

$$IP(O^t, O^{t+1}) = \frac{1}{N} \sum_{n=1}^{N} \frac{\Delta T}{T_n} \cdot [\underbrace{TF(O^t)}_{\textbf{old}} - \underbrace{TF(O^{t+1})}_{\textbf{new}}] \quad (5)$$

where $\sum_{n=1}^{N} \Delta T/T_n$ is the estimated number of jobs released during a prediction window and $TF(\cdot)$ is the sum of transmission cost per job per task that is represented as,

$$TF(O^t) = \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{m=1}^{M} o_{n,k,m}^t \cdot \frac{W_n}{R_{t,k}} \quad (6)$$

The corresponding migration overhead can be formulated as,

$$RF(O^t, O^{t+1}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{m=1}^{M} (o_{n,k,m}^t \oplus o_{n,k,m}^{t+1}) \cdot rf \quad (7)$$

where rf is a constant value that estimates the inter-edge communication cost for task migration. We define the score function of O^{t+1} as follows,

$$S(O^{t+1}) = \alpha \underbrace{IP(O^t, O^{t+1})}_{\textbf{improvement}} - (1-\alpha) \underbrace{RF(O^t, O^{t+1})}_{\textbf{migration cost}} \quad (8)$$

where balance factor α $(0 \leq \alpha \leq 1)$ denotes the balance between transmission cost and migration overhead. This balance factor α characterizes the trade-off between too frequent and too infrequent migrations discussed earlier. Based on the proposed score function, the best assignment O^{t+1} for task migration is the one with highest score $S(O^{t+1})$ as it maximizes transmission improvement and minimizes migration overhead, yet satisfies the job scheduling constraints discussed in Section IV-C. In the case of the highest $S(O^{t+1}) <= 0$, our strategy only allows migrations in the event of future validation failures. Thus, the overall optimization problem can be represented as,

$$\begin{aligned} \underset{o_{n,k,m}^{t+1}}{\text{Maximize}} \quad & S(O^{t+1}) \\ \text{subject to} \quad & \sum_{k=1}^{K} \sum_{m=1}^{M} o_{n,k,m}^{t+1} = 1, n \in \mathcal{N}, \\ & \sum_{n=1}^{N} o_{n,k,m}^{t+1} \cdot \frac{E_n}{min(T_n, D_n^*)} \leq 1 \\ & , k \in \mathcal{K}, m \in \mathcal{M} \end{aligned} \quad (9)$$

The first constraint in Eq. (9) indicates that application is only assigned to one edge-cloud node and running on one computing core. The second constraint guarantees the success of local job scheduling.

C. Heuristic Algorithm

Inspired by Multiple Knapsack Problem (MKP) [25], a NP-hard combinatorial optimization problem, we consider edge-cloud nodes as K knapsacks that have the same capacity (measured by M compute cores) and treat tasks as a set of N items. Each item has a weight w (task density). The value (*score function*) defines the benefits of service migration where the highest positive score indicates the best migration solution. Compared to traditional MKP, we have strict constraints and the way we calculate the weight and the score of each combination change dynamically. In addition, in this work we consider a time series for spectrum fluctuations where repeated search for best solution is necessary. Thus, to reduce the computation complexity and decision response in disaster scenarios, we introduce a lightweight heuristic approach to generate high quality solutions for the proposed optimization problem.

In Eq. 6, the transmission cost is strongly correlated with job data size and the allocated fair bandwidth. Inspired by this

Algorithm 1: Pseudocode for LBLJ-FF

```
Input: sorted edge nodes K, sorted user tasks N
Output: O^{t+1}
total_load ← sum(N.job_size)
total_bandwidth ← sum(K.bandwidth)
for each k ∈ K do
    k.workload ← total_load · k.bandwidth / total_bandwidth
unassigned ← N
equal_load ← True
while unassigned ≠ empty do
    saturate ← True
    for each k ∈ K do
        for each n ∈ unassigned do
            add n to k's service list
            m ← EDF-FF(n, k)
            if m ≥ 0 and (k.workload ≥ n.job_size or
             not equal_load) then
                o^{t+1}_{n,k,m} ← 1
                remove n from unassigned
                k.workload ← k.workload − n.job_size
                saturate ← False
                if not equal_load then
                    break
            else
                o^{t+1}_{n,k,m} ← 0
                remove n from k's service list
    if not equal_load and saturate then
        break
    else
        equal_load ← False
compute score of O^{t+1} based on Eq. (8)
if score > 0 or O^t not valid then
    return O^{t+1}
return O^t
```

observation, we propose the hypothesis that equal-load task assignment based on job size provides an overall transmission cost that is close to optimal assignment if the jobs with large data size are assigned to edge-cloud nodes that have large available bandwidth. Based on this, we choose well-known Earliest Deadline First-First Fit (EDF-FF) algorithm [20] for job scheduling and propose a lightweight Largest Bandwidth Largest Job-First Fit (LBLJ-FF) Algorithm 1 for global assignment and migration. We introduce the concept of equal-load when each edge-cloud node gets the amount of workload proportional to their bandwidth as long as their local job scheduling do not fail. In this algorithm, edge-cloud nodes and tasks are sorted in non-increasing order by bandwidth and job size, respectively. The higher positions in the list will be considered and paired first. The algorithm performs equal-load assignment that follows an equal-load strategy where each edge-cloud node are assigned tasks from the sorted list until it reaches its maximum workload or its local job scheduling fails. The algorithm also performs constraint assignment that removes the equal-load limitation, but only allows edge-cloud nodes to select one unassigned task for each iteration until all local job schedules are saturated (i.e., density = 1). The remaining unassigned tasks are forced to wait for the next assignment turn.

VI. Results

In this section, we evaluate the performance of the proposed migration scheme through a realistic simulation environment. We design disaster response applications generated at end-devices that are assigned unlicensed wireless channels for data transfer to edge-cloud nodes. The behavior of spectrum fluctuations induced by licensed transmission is modeled from RWTH Mobnets [7] dataset. To measure the maximum bandwidth for unlicensed usage by the applications, we use ON-OFF licensed occupancy model [24]. This model allows us to convert raw power spectral density (psd) values of licensed transmission on a channel into availability scalar (0 for available or 1 for busy) for unlicensed access by comparing the psd values against a predefined decision threshold δ. The simulation parameters are described in Table I.

TABLE I: Simulation Parameters

Name	Value
spectrum span	20MHz – 1500 MHz
power spectral density threshold δ	-107 dbm/200Khz
max bandwidth B	40 – 60 MHz
bandwidth of sub-channel b	200 KHz
SNR	0.68
prediction window ΔT	30 s
result time step	300 s
test duration	3000 s
job size W_n	3 – 6 MB
job execution E_n	500 – 1000 ms
job interval T_n	3 – 8 s
job deadline D_n	3 – 8 s
number of edge-cloud nodes K	10
migration overhead / per task rf	100 – 2000 ms
number of cores per edge-cloud node M	2
number of applications N	30 – 120
number of runs	50

A. Trade-off Analysis

To study the trade-off between migration overhead and transmission cost and evaluate optimal α, we define *migration-to-transmission* ratio $r = rf/tf$ where rf is the migration cost per task and tf is the estimated average transmission cost per job. We identify optimal α by characterizing these two metrics against α for different values of r. For a constant r, an α value is optimal if for any α greater than the optimal value, there is a sudden or significant increase in the growth of migration overhead and at the same point the transmission cost per job starts to slowly improved.

Fig. 6 shows the ideal α for different balance factor r. One of the two special cases, $\alpha = 0$ indicates that after the first assignment, no service migration will be triggered until future validation fails. This leads to high transmission cost and increases job failure probability. Whereas, for the case with $\alpha = 1$, greedy task migration will try to minimize transmission cost, however frequent migration will cause the overhead to explode. Fig. 6 shows the existence of optimal α somewhere in between these two extreme cases which is a function of *migration-to-transmission* ratio r.

B. Job Completion Rate

To fairly evaluate how global assignments impact job execution, we define application *workload* $= \frac{N}{K}$ and observe the

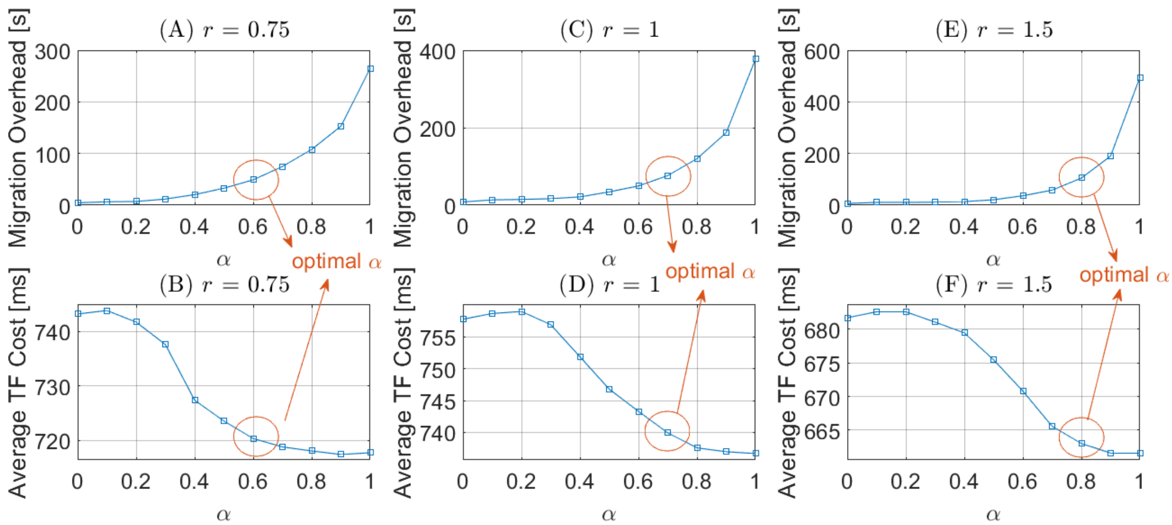

Fig. 6: Migration overhead and average transmission cost per job characteristics against α for different r values showing existence of optimal α in each case

changes in job completion rates for different *workloads*. The job completion rate is calculated as the ratio of number of jobs processed and number of jobs released for all tasks.

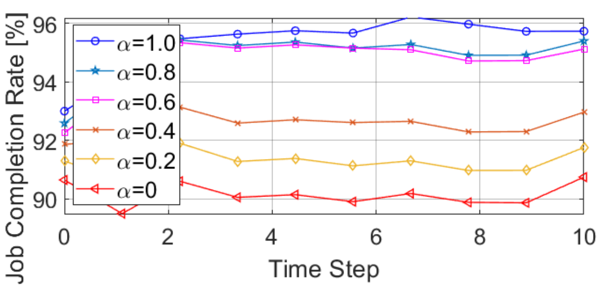

Fig. 7: Time steps of job completion rate for $r = 0.75$ and *workload*= 8

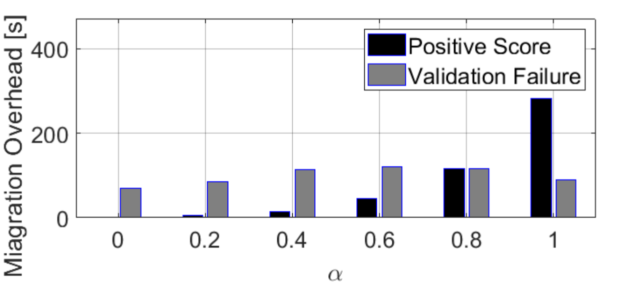

Fig. 8: Migration overhead characteristics against α under different policies for $r = 0.75$ and *workload*= 8

Fig. 7 and 8 show the time steps of job completion rate and migration overhead characteristics against α under different policies, both for $r = 0.75$ and *workload*= 8. Being consistent with Fig. 6, the results show that the closer α is to 0, lower is the migration frequency and higher is the failure rate. We presents two types of migration costs; the first type of migration is triggered by new assignments with $score > 0$ and the second type is forced migration caused by future validation failures. This figure confirms the optimal α to be 0.6 (with $r = 0.75$) as it achieves a higher completion rate that is close to greedy task migration policy and a relatively low migration overhead.

Application task execution for disaster incident response has very strict completion rate requirement, and failure to complete even a single job can lead to serious consequences. Therefore, our approach is designed to achieve near 100% completion rates for tasks that are assigned to edge-cloud nodes. Fig. 9 shows how that is achieved by dynamic task profile and validation of job scheduling. Compared to static task profiles, dynamic task profiles can avoid certain amount of unnecessary migrations and keep 100% completion rate by providing accurate task density as shown in the figure. This proves the accuracy of our model and verifies the correctness of the formulation where we used updated task density as an input to the LBLJ-FF algorithm.

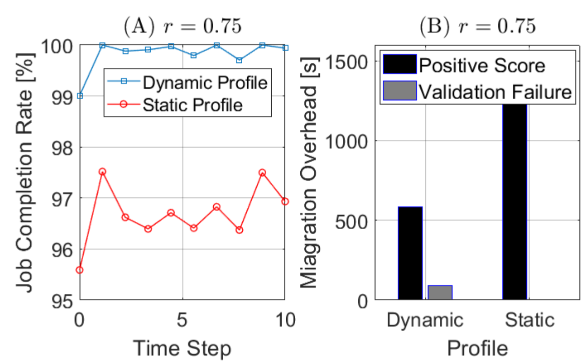

Fig. 9: (A) Time steps of job completion rate of all tasks that are assigned for different task profiles, (B) Migration overhead characteristics of all tasks that are assigned for different task profiles

C. Performance Comparison

Next, we compare the performance of our proposed LBLJ-FF algorithm against other heuristics based approaches, such as genetic algorithm (GA) and random global assignment (RND) strategy. GA approach uses Eq. (8) as its fitness function and also applies our migration strategy. Whereas, RND approach only enables local job scheduling and randomly selects global assignments when future validation fails.

As RND is not subjected to any constraints, the success of local job scheduling cannot be guaranteed which can result

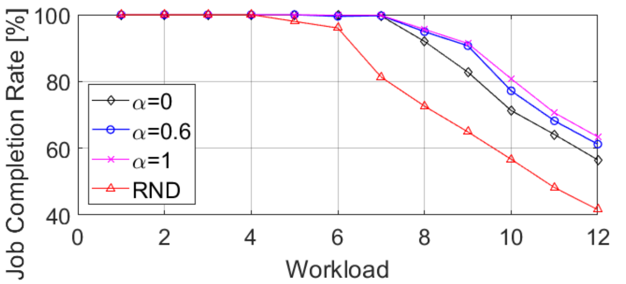

Fig. 10: Job completion rate comparison against different workloads

in very low job completion rate when workload increases. Compared to RND, the proposed LBLJ-FF scheme provides intelligence to task migration and is superior to RND strategy in all cases as shown in Fig. 10. At a $workload = 8$, the improvement rate is the maximum at 25%. At the same time, Fig. 11 shows that the RND strategy has a significantly high transmission cost caused by uncertainty of spectrum fluctuations that leads to many unnecessary migrations. While by setting α to 0.6, LBLJ-FF and GA strategies generate very little migration overhead while achieving transmission cost that is close to the best case scenario (i.e., with $\alpha = 1$). Our lightweight LBLJ-FF algorithm makes it easier to achieve the same overall performance compared to GA that requires extensive computations for its mutation, crossover and fitness functions, thus proving to be a better choice for disaster incidence response edge-cloud system.

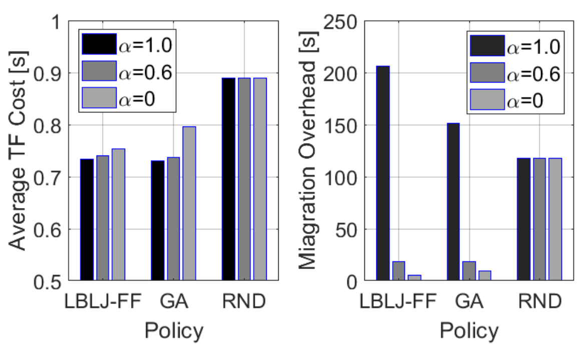

Fig. 11: (A) Average data transmission cost per job under different policies, (B) Migration overhead characteristics against α under different policies

VII. Conclusions and Future Work

In this paper, we explored how migrations can help disaster response edge-cloud systems to be resilient against spectrum *fluctuations*. We showed how our proposed scheme with global assignment and local scheduling techniques tried to strike a balance between too frequent and too infrequent migrations guided by optimizations. Simulation results showed that our scheme can significantly improve overall job completion rate, yet strike a balance between transmission cost and migration overhead, creating a highly adaptive resource provisioning method.

The ideas and results presented in this paper are elements towards proposing a wider paradigm shift about how edge-cloud resources should be managed in disaster incidence response scenarios. We will build upon the ideas proposed in this paper to develop broader URB services that will implement robust resource orchestration policy algorithms that can optimize system utility within a finite time horizon under a fluctuating environment unlike traditional long-term cloud system performance with steady-state analysis. The outcomes of our proposed research will benefit: disaster management efforts by the first responders; incident response management planning and policy makers; cloud, cyber-infrastructure, network management, and future Internet research communities.

References

[1] Y. Sun, K. R. Chowdhury, "Enabling emergency communication through a cognitive radio vehicular network," *IEEE Communications Magazine*, vol. 52, no. 10, pp. 68-75, 2014.

[2] M. Kobayashi, "Experience of infrastructure damage caused by the Great East Japan Earthquake and countermeasures against future disasters," *IEEE Communications Magazine*, vol. 52, no. 3, pp. 23-29, March 2014.

[3] J. Parajuli, K. E. Haynes, "The earthquake impact on telecommunications infrastructure in Nepal: a preliminary spatial assessment", *Regional Science Policy & Practice*, vol. 8, no. 3, pp. 95-109, 2016.

[4] S. Debroy, P. Calyam, M. Nguyen, A. Stage, V. Georgiev, "Frequency-Minimal Moving Target Defense using Software Defined Networking," *Proc. of IEEE ICNC*, 2016.

[5] S. Debroy, S. Bhattacharjee, M. Chatterjee, "Spectrum Map and Its Application in Resource Management in Cognitive Radio Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 1, no. 4, pp. 406-419, 2015.

[6] J. Riihijarvi, P. Mahonen, "Exploiting spatial statistics of primary and secondary users towards improved cognitive radio networks," *Proc. of CrownCom*, 2008.

[7] RWTH Aachen University Static Spectrum Occupancy Measurement Campaign - https://download.mobnets.rwth-aachen.de/

[8] S. Yi, Z. Hao, Q. Zhang, Q. Zhang, W. Shi, Q. Li, "LAVEA: Latency-Aware Video Analytics on Edge Computing Platform," *Proc. of IEEE ICDCS*, 2017.

[9] Z. Dong, Y. Liu, H. Zhou, X. Xiao, Y. Gu, L. Zhang, C. Liu, "An energy-efficient offloading framework with predictable temporal correctness" *Proc. of ACM/IEEE SEC*, 2017.

[10] M. S. Elbamby, M. Bennis, W. Saad, "Proactive edge computing in latency-constrained fog networks," *Proc. of EuCNC*, 2017.

[11] C. Liu, M. Bennis, H. V. Poor, "Latency and Reliability-Aware Task Offloading and Resource Allocation for Mobile Edge Computing," *Proc. of IEEE Globecom Workshops*, 2017.

[12] C-C. Hung, G. Ananthanarayanan, P. Bodk, L. Golubchik, M. Yu, P. Bahl, M. Philipose, "VideoEdge: Processing Camera Streams using Hierarchical Clusters," *Proc. of ACM SEC*, 2018.

[13] C-C. Hung, G. Ananthanarayanan, P. Bodk, L. Golubchik, M. Yu, M. Zhang, "Wide-area analytics with multiple resources," *Proc. of EuroSys*, 2018.

[14] R. Gargees et al.,"Incident-Supporting Visual Cloud Computing Utilizing Software-Defined Networking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 1, pp. 182-197, 2017.

[15] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, K. K. Leung, "Dynamic service migration in mobile edge-clouds," *Proc. of IFIP Networking*, 2015.

[16] A. Ksentini, T. Taleb, M. Chen, "A Markov Decision Process-based service migration procedure for follow me cloud," *Proc. of IEEE ICC*, 2014.

[17] W. Zhang, Y. Hu, Y. Zhang, D. Raychaudhuri, "SEGUE: Quality of Service Aware Edge Cloud Service Migration," *Proc. of IEEE CloudCom*, 2016.

[18] T. Ouyang, Z. Zhou, X. Chen, "Follow Me at the Edge: Mobility-Aware Dynamic Service Placement for Mobile Edge Computing," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2333-2345, 2018.

[19] T. Bahreini, D. Grosu, "Efficient Placement of MultiComponent Applications in Edge Computing Systems," *Proc. of ACM/IEEE SEC*, 2017.

[20] S. Baruah, N. Fisher, "The partitioned multiprocessor scheduling of sporadic task systems," *Proc. of IEEE RTSS*, 2005.

[21] J. M. Lopez, M. Garcia, J. L. Diaz, D. F. Garcia, "Worst-case utilization bound for EDF scheduling on real-time multiprocessor systems," *Proc. of Euromicro RTS*, 2000.

[22] L. Zhang, T. Jiang, K. Luo, "Dynamic Spectrum Allocation for the Downlink of OFDMA-Based Hybrid-Access Cognitive Femtocell Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1772-1781, 2016.

[23] X. Xing, T. Jing, W. Cheng, Y. Huo, X. Cheng, "Spectrum prediction in cognitive radio networks," *IEEE Wireless Communications*, vol. 20, no. 2, pp. 90-96, 2013.

[24] S. L. Castellanos-Lopez, F. A. Cruz-Perez, G. Hernandez-Valdez, "Performance of cognitive radio networks under ON/OFF and Poisson primary arrival models," *IEEE PIMRC*, 2011.

[25] P. C. Chu, J. E. Beasley. "A genetic algorithm for the multidimensional knapsack problem," *Journal of Heuristics*, vol. 4, no. 1, pp. 63-86, 1998.