IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 20, 2019, accepted April 21, 2019, date of publication May 1, 2019, date of current version May 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2913150

Big Data in Motion: A Vehicle-Assisted Urban Computing Framework for Smart Cities

MURK[1], ASAD WAQAR MALIK[1,2], IMRAN MAHMOOD[1], NADEEM AHMED[1], AND ZAHID ANWAR[1,3]

[1]School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[2]Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia
[3]Department of Mathematics and Computer Science, Fontbonne University, St. Louis, MO 63105, USA

Corresponding author: Asad Waqar Malik (asad.malik@seecs.edu.pk)

ABSTRACT Smart cities are envisioned to facilitate the well-being of the society through efficient management of the Internet of Things resources and the data produced by these resources. However, the enormous number of such devices would result in unprecedented growth in data, creating capacity issues related to the acquisition, transfer from one location to another, storage, and finally the analysis. The traditional networks are not sufficient to support the transfer of this huge amount of data, proving to be costly both in terms of delay and energy consumption. Alternative means of data transfers are thus required to support this big data produced by smart cities. In this paper, we have proposed an efficient data-transfer framework based on volunteer vehicles whereby we employ vehicles to carry data in the direction of the destination. The framework promotes self-belonging, social awareness, and energy conservation through urban computing encouraging participation by citizens. The proposed framework can also facilitate the research community to benchmark their own route selection algorithms easily. Further, we performed an extensive evaluation of the proposed framework based on realistic models of vehicles, routes, data-spots, data chunks to be transmitted and the energy consumed. Our results show the efficacy of the proposed data transfer framework as the energy consumed through vehicles is significantly less than that consumed by transmission over the Internet thereby reducing the carbon footprint. The results also offer several insights into the optimal configuration of a vehicular data transfer network based on analysis of delay, energy consumption, and data-spot utilization.

INDEX TERMS Big-data transfer, data center, delay, energy, simulation, Anylogic.

I. INTRODUCTION

The role of smart cities and its services is increasing due to easy access to technology. However, to handle the concerns of end users, more pragmatism is required in research to keep it relevant for all the stakeholders [1]. Therefore, there is a need of structured research with a special focus on sustainable, global and social-aware policies for research in smart cities. Moreover, the emerging technologies such as Internet of Thing (IoT), context-aware computing, cognitive computing, 5G, and advanced distributed data-ware housing require greater attention to satisfy the end users and attract other stakeholders [2].

Internet of things (IoT) is becoming ubiquitous leading towards the proliferation of smart cities and smart vehicles [3]. In smart cities, the data generated from IoT devices can be further utilized to improve the urban environment and smart city services. This data is thus required to be aggregated and processed for effective decision making. However, the paradigm shift to smart cities has resulted in production of enormous data that requires handling, transfer (from the source to a data center for example) and storage. Thus, the main focus of smart cities becomes the efficient management of available resources and the data produced by these resources in order to improve the economy and the sustainability of societies. With the concept of smart cities, where all the resources are connected, monitored and utilized, it becomes imperative to efficiently handle the movement of this huge amount of data for its effective management.

The movement of this massive amount of data through the traditional networks is not sustainable as it overburden these networks consuming an enormous amount of energy at the intermediate nodes and incurring significant delays as well as monetary cost. In order to cater to the rapid growth of data

The associate editor coordinating the review of this manuscript and approving it for publication was Miltiadis Lytras.

produced by smart cities, alternative means are thus needed to transfer data to the nearest data center for further processing. Edge computing and fog computing are computing paradigms that have been introduced to reduce the amount of data that is required to be transferred to the data center in the cloud [4]. Data is thus processed near the source and only the processed data, that is a fraction of the raw data, is transferred to the back-end data center. Although these concepts are promising, currently we have limited edge and fog devices available to end users [5].

An interesting solution to this issue is to employ vehicles to carry data from a source to the destination. The number of vehicles is increasing, the highways are well constructed to support faster commute and people often travel using their own vehicles. Another advantage of utilizing vehicular traffic for data transfer is that the vehicles are available even in potentially destructive situations when all other media fail to survive. Intelligent Transportation Systems (ITS) [6] based on smart cars can thus be effectively utilized for data transfer. Therefore, using the daily commute routine, the vehicles can be used to transfer the data to the destination or to any close-by location. This can help alleviate the load on traditional networks. It also acts as an urban computing application in the perspective of smart cities. Citizens are thus encouraged to participate in activities for the overall well-being of the society. Participation in this data transfer through vehicles promotes self-belonging, green computing, energy conservation and social awareness among the participants to name a few.

As a case in point, Amazon provides a dedicated vehicle to transfer user's data into their data centers. Amazon provides three type of data carrying facilities termed as a snowball,[1] snowball edge,[2] and snowmobile.[3] Companies having Peta-bytes of data would take years to transfer this data using traditional network on the Amazon cloud but Amazon Web Services (AWS) shipping reduce the time to days and months. However, systems like AWS shipping suffer from some notable disadvantages. The vehicles with fitted devices are limited, and users have to book the service in advance, thus, adding a further delay in completing the data transfer. Moreover, depending on the type and volume of the data, the customers have to bear very high transfer costs.

In this paper, we propose a vehicle based framework that can plan and monitor the data transfer from any source to destination using the intermediate storage points referred as data-spots. The proposed framework is opensource[4] and can facilitate researchers to incorporate their own data transfer and route selection algorithms. In the proposed system, the data is kept encrypted so that it can't be accessed outside the system. The controller tracks the data chunks (parts of the data loaded on a vehicle) and request re-transmission if any vehicle fails to offload the data back into any system node. The data-spots and sender node maintains the backup which is only removed once the Acknowledgement from the destination node is finally received. Although, there exists a basic theoretical work that discusses the importance of data transfer through vehicles [7]; but as per our knowledge, no such framework exists that manages the complete control of the data transfer process.

A recent BusinessInsider report projects that there will be more than 55 billion IoT devices by 2025, up from about 9 billion in 2017. Moreover, it is estimated that there would be $15 trillion in aggregate IoT investment between 2017 and 2025 [8]. In an ecosystem supporting the use of smart cities and connected cars, IoT sensors and actuators for monitoring the movement of road vehicles precisely and facilitating traffic, vehicular communication is the information highway of the future. Therefore our research directly leverages emerging technology trends to reduce carbon emissions and improve reliable transmission of Big Data in motion. It also reduces the reliance on Internet gateways and routing infrastructure and contributes to alleviating the load on the traditional networks.

This paper is organized as follows. In Section II, we discuss our motivation towards the research problem, Section III covers related work. Proposed system design is presented in Section IV. Section V covers the evaluation case study. Results and discussions are presented in Section VI and VII respectively. Section VIII concludes the paper.

II. MOTIVATION

With the advancement of technology and increase in deployment of IoT devices resulting in massive amount of data being produced, there is a need to design efficient mechanism to transport this huge data. The processing, maintenance, privacy and security of the data are some other challenges related to massive data [9]. With the existing escalation rate, the amount of global traffic is predicted to grow over 100 trillion gigabytes in the next ten years [10].

According to Cisco report, a smart city with a population of 1 million could easily generate 180 million GB data per day or 42.3 ZB/month [7] and that the total usage would grow to about 49 EB/month by year 2021 (see Fig. 1).

With the development of high-speed networks, the data transfer rate has improved considerably. However, the overall energy consumption cost of big data transfer through the traditional network is still significant. The networking appliances consume a lot of electrical energy and emit carbon-dioxide. Therefore to save energy, an alternative means of transporting huge data is becoming a necessary requirement. In [11], authors show that the use of vehicular delay tolerant network is efficient compared to the emissions involved in data transfer through traditional networks.

We propose an overarching framework to transport data using vehicles. Data transfer using vehicles leverages the efficient utilization of energy already being consumed in vehicles for completion of their normal journeys. AWS provides

[1] https://aws.amazon.com/snowball
[2] https://aws.amazon.com/snowball-edge/
[3] https://aws.amazon.com/snowmobile/
[4] available on request

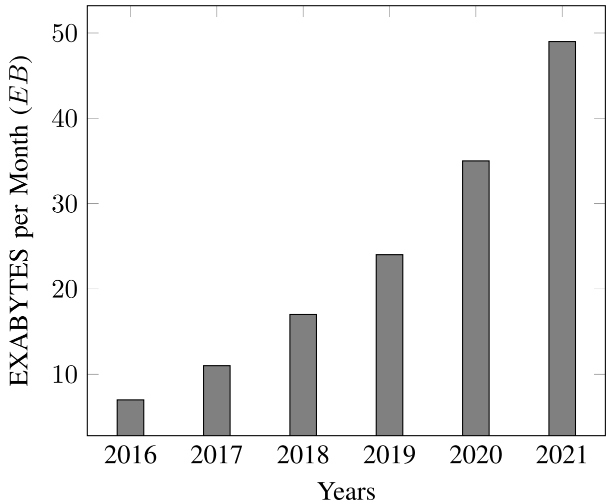

FIGURE 1. Amount of data-increase in Exabytes per month [7].

a shipping service for transfer of huge data into their data centers using vehicles. AWS send their company's vehicle to the customer, customer loads data onto the vehicle and the vehicle is driven back to AWS data center for uploading the data. However, there are monetary costs involved in this transfer. There is a cost to be paid by the customer (of about $200 minimum) if the transfer can be done in about 10 days otherwise the customer is charged even more. Moreover, the vehicle belongs to AWS, so the company has to deal with the expense of vehicles.

It is worth pointing out that the physical transport does consume a lot of energy and also emits carbon, we are arguing for the transfer of data by utilizing existing transport, or to put it simply, piggybacking data on vehicles that were going to cover the same route anyways. Moreover, with the lowering cost of storage today the added weight of storage devices will add minuscule additional fuel consumption if any. In this paper, our focus is on the design and evaluation of a framework to utilize vehicles as a replacement of the traditional networks. Finally, the optimal approach could be to use a combination of network and physical transport. We plan to work on a hybrid approach of transferring big data using vehicles as well as wired media, as a future extension work for this framework. Our framework provides a cost-saving approach, that would benefit customers, vehicle owners as well as the service providers such as AWS. Table 1 covers the abbreviations used in the rest of the paper.

TABLE 1. Nomenclature.

Symbols	Explanation
c	controller
d_c	data center
d_s	data-spot
v	vehicle
T_{vj}	vehicle's travel time to job
T_{dd}	chunk download time
T_{dest}	vehicle's travel time to reach destination
T_{du}	data upload time
V_{veh}	vehicle carrying capacity
V_i	number of vehicles
B_{int}	network bandwidth
e_{dd}	energy consumed in upload/download
S_c	System capacity
B_v	vehicles Bandwidth
GD_{tran}	Gross data transfer
D_{vol}	Data volume
T_r	Transfer rate
J	Job
chk	chunk
D_{chk}	Data chunk
D_{tr}	Data transfer rate
$Dist_{SD}$	distance btw source S and destination D
Avg_{speed}	average speed of a vehicle
D_{tr}	time taken to load or unload
C_v	storage capacity per vehicle
N	total volume of vehicles
ρ	probability of participating vehicles
F_e	fuel economy
W_{veh}	average weight of a vehicle
F_c	constant to convert liter in to joules
E_{veh}	total energy consumed by vehicular approach
e_u	energy consumed while unloading
e_o	energy consumed while offloading
e_t	transportation energy consumed by vehicle

III. RELATED WORK

Big data being produced through devices has numerous utilities. This includes health care data and weather forecasting to predict the future patterns [12]. Most of the research work done in this area is based on framework design and development, for efficient handling of the data. Frameworks such as MapReduce are commonly used for big-data. However, before the processing of the data using any of the frameworks, the first step is to transfer the data to the data center where the sufficient resources are available for analysis. This transfer mechanism is relatively less explored. In this paper, we have proposed a framework that uses vehicles to transfer the data from one location to another. Here, we briefly discuss the relevant work in this domain.

Today, data centers provide a significant amount of data storage on pay per use basis. Usually, multiple copies of data are stored across the data centers [13]. The transfer of non-time critical data from one location to another utilize a significant amount of energy and network bandwidth. Amazon Web Services (AWS) are the best example of providing alternative means to transfer data to their data centers; Digital Globe moved their entire data to AWS cloud using AWS snowmobile feature [14]. The Digital Globe is a global map provider to companies such as Uber and offers much more with its vast content of earth's imagery. On the other hand, Delay Tolerant Networks (DTNs) are focused towards physical transportation systems. Zhu *et al.* [15] describes impact of big-data on the field of Intelligent transportation systems(ITS) and explains that there are huge benefits to be obtained from rapid big-data growth by using big-data analytics techniques. Furthermore, Cheng *et al.* [16] proposed a research work claiming that the vehicles are the reliable and efficient resources which can be used for sharing bulk-data. Palma *et al.* [17] proposed a methodology to perform distant communications

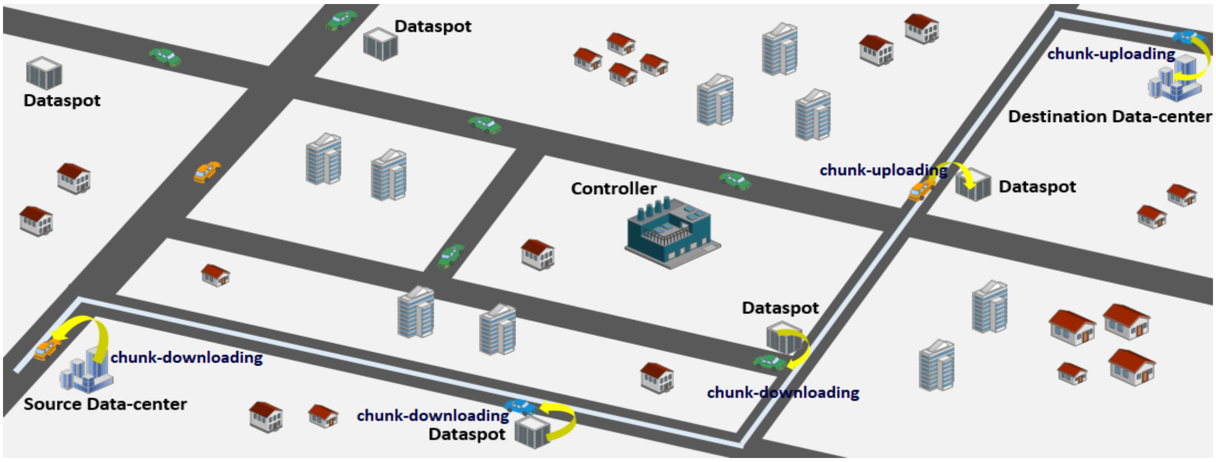

FIGURE 2. High level diagram showing data transfer through vehicles, the suggested route is also visible.

using unmanned aerial vehicles. Similarly, Usbeck *et al.* [18] and Hunjet *et al.* [19] proposed data dissemination approach via data ferries. In France, a pilot project is implemented where vehicles are used as data carriers. The objective is to reduce the overall network usage [20]. In this project, data is transferred through Dedicated Short Range Radio (SRR) waves; however, SRR is not suitable for transmitting big chunks of data.

In [21], Altintas *et al.* proposed a vehicle-based framework termed as Car4ICT. The Car4ICT is used to provide computing, storage, and information relay services. The user can discover services hosted through vehicles. The end user can interact with Car4ICT through any electronic gadget. In order to further enhance the support of compute-intensive applications, the vehicles are also connected with the cloud, where the data received from vehicles are stored, analyzed and used in various applications [22].

Dressler *et al.* [23] used the concept of parked vehicle as roadside units (RSUs) for data dissemination among other vehicles. The authors proposed a vehicle cord protocol, which enables parked vehicles to act as an RSU. The system can thus easily scale with a number of parked vehicles. Marincic and Foster [11] analyze the data transfer by means of physical shipping of data-disk. The authors conclude that the physical shipping was more energy efficient compared to data transfer using traditional networks. The energy consumption is always been a critical area of research. With the tremendous growth of data, the data transfer also consumes a significant amount of energy. Moreover, the data center itself consume a significant amount of energy. Therefore, energy consumption inside the data center has been studied well over the years. As per the recent study conducted on German data-centers estimated that by 2020, the energy consumption due to the IT equipment may increase up to 14 billion kWh and will reach to 16.4 billion kWh by 2025 [24]. Therefore, various techniques and schemes are adopted to reduce the network energy [25]. Munjal *et al.* [26] performed data transfer through public buses. Thus, saving a significant amount of energy consumed through core networks. We next cover the proposed framework design in detail.

Discussion – In terms of big data, most of the work has been done on data storage and efficient processing using techniques such as HDFS and MapReduce. However, data transfer through other mechanism is less explored. Moreover, in some cases, the authors have discussed the data transfer mechanism using ferries, vehicles and short-range radio communication. However, these techniques are theoretically explored with focus on data transfer between vehicle to vehicle [21] [23]. Moreover, in all the reported work, delay and energy has not been considered for data transfer. Thus, no extensive work exists that provides the complete framework to support bulk data transfer and benchmark in terms of transfer delay and energy consumption. Moreover, we have also compared the vehicular data transfer with traditional Internet mechanism.

The proposed framework managed data transfer through central control, who continuously track the data chunk movement from the source center to the destination. The controller also monitors the chunks on intermediate data-spots, where the received chunks are validated and re-transmission is requested in case of invalid chunks. As the data is transferred through private vehicles, therefore scenario of undelivered chunks are also catered through the controller.

IV. PROPOSED SYSTEM DESIGN

The proposed system has been designed to leverage the capabilities of smart city and smart vehicles. There are four major components in this architecture; centralized controller, source & destination data centers, data-spots and the smart vehicles (as shown in Fig. 2). The smart vehicles are connected through a cellular network and are assumed to have variable data carrying capacities. The request for data transfer to a specific destination data center is initiated by the source data center. The controller performs route planning and schedules the transfer. Smart vehicles are used to transfer the data from the source data center and they carry it to either the final destination (one hop transfer) or offload the data to any of the data-spots along the route (multi-hop transfer). The data is eventually received at the destination data center.

The *controller* is the main module of the proposed framework. It is responsible for efficient route planning so that data can be transferred to the destination with minimum delay and through best available routes. The source data center first communicate with the controller, sharing the destination location and size of the data that need to be transferred. Based on the received information, the controller selects the transfer mechanism i.e. through traditional network or employing vehicles. In case of data transfer through vehicles, the controller calculates the best possible route. The route information is transferred back to the source center. Moreover, the controller also informs intermediate data-spots and the destination node about the proposed activity. The complete functionality of controller is presented in Algorithm 1.

The *data-spots* are the intermediate stations placed at different locations to facilitate the data transfer process. In our proposed framework, the data-spots are large in number, usually available on multiple routes between the data centers. The ideal location of such spots are near gas service stations and rest areas. Data-spots have docking stations equipped with wired connections i.e. USB 3.2 (having 20Gbps data-rate) that can be used to transfer data between vehicles and data-spots. The purpose of a wired connection for data transfer is to reduce the transfer time. Later on, with 5G communication enabled, the wireless transfer can also be put in place. These data-spot also checks the data integrity of any received data chunk through an error detection mechanism such as checksum [27]. The spot discards any data that fails the sanity check with information to the controller for retransmission. In smart cities with hybrid/electrical vehicles, an ideal place for data-spots is at vehicles charging stations. Algorithm 2 states the functionality of data-spots.

The *vehicles* are the backbone of the proposed framework where these are used as a carrier to transfer data from one location to another. The proposed framework is primarily based on volunteer participation from vehicles, although it supports an incentive mechanism by offsetting the charges for services such as electrical vehicles re-charge, meal/coffee breaks for drivers etc. In the proposed framework, vehicle drivers are not forced to deviate from their original directions just to deliver the data chunks, thus, the role of data-spots comes into play. The vehicles can offload data to nearby data-spot, before leaving the proposed path. Moreover, the vehicles can also take their own path either to the destination or select any of the data-spot to offload chunks. Every vehicle offers a variable amount of data storage capacity. The functionality is also stated through Algorithm 3.

The *data centers* are usually the source or a destination of the data transfer process. However, any data-spot can also become a source or destination. The source initiates the transfer process by sending the request along with supporting information to the controller. The proposed framework can be utilized to move archived data from one data center to another.

Algorithm 1 Controller ▷ This Function Represents the Central Controller Whose Task is to Check the Queue for Incoming Tasks, Calculate Routes and Oversee the Transfer of Data Chunks

```
msg ← null
Check MessageQueue
while (MessageQueue! = Empty) do    ▷ Check for tasks
    msg ← MessageQueue_pop
    if (msg.Type == routeRequest) then
        call CalcRoute(msg)
    else if (msg.Type == dataRetransmitReq) then
        Locate(missingChunk)
        Send(retransmitRequest)
    else
        Send(acknowlegement)
    end if
end while

function calcRoute(routeReq)
    counter ← 0
    srcLocation ← routeReq.srcLoc
    destLocation ← routeReq.destLoc
    datasize ← routeReq.datasize
    Queue tempQueue[] ← Empty
    Queue spotsInRoute[] ← Empty
    distance1 ← 0
    distance2 ← 0
    Read globalSpotsQueue[]
    tempQueue[] ← globalSpotsQueue[]
    firstPoint ← srcLocation
    lastPoint ← destLocation
    spotsInRoute.push(firstPoint)    ▷ Use dataspots in route calculation
    nearestPoint ← firstPoint
    while (nearestPoint! = lastPoint) do
        distance1 ← getDistance(nearestPoint, lastpoint)
        nearestPoint ← getNearestSpot(tempQueue)
        if nearestPoint == lastPoint then
            EndCurrentLoop
        else
            distance2 ← getDistance(nearestPoint, lastPoint)
            if (distance1 > distance2) then
                spotsInRoute.push(nearestPoint)
                counter ← counter + 1
            else
                tempQueue.remove(nearestPoint)
                nearestPoint ← spotsInRoute[counter]
            end if
            tempQueue.remove(nearestPoint)
        end if
    end while
    spotsInRoute.push(lastPoint)
    decision ← call compCost(spotsInRoute, datasize)
    if (decision == spotsInRoute) then
        Send(decision, spotsInRoute)
    end if
    return decision
end function

function compCost(route, datasize)    ▷ Prefer the communication type (Internet or via vehicle) with least delay
    decision ← null
    calculate Delay_int
    calculate Delay_tr
    if (Delay_int > Delay_tr) then
        decision ← route
    else
        decision ← transferViaInternet
    end if
    return decision
end function
```

Algorithm 2 Data-Spot ▷ Call Vehicle for Data-Pickup if Data-Spot Task Queue Is Not-Empty

```
v_i ← null
D_chk ← null
check dataQueue
while (dataQueue! = Empty) do
    D_chk ← dataQueue_pop
    if (D_chk.isCorrupted() == False) then
        v_i ← call vehicle.pickUpData()
        Load(D_chk, v_i)
    else
        Send(retransmitReq, Controller)
    end if
    check dataQueue
end while
```

Algorithm 3 Vehicle ▷ Manages Vehicle Data Pick-Up and Dropoff

```
D_chk ← null
check uploadsQueue
while (uploadsQueue! = Empty) do
    D_chk ← uploadsQueue_pop
    offload(D_chk, NearestSpot)
end while

function pickUpData
    return vehicleInfo
end function
```

Moreover, the framework also supports transfer of users data to the data center, but for that purpose, the user needs to transfer its data to the nearest data-spot. The framework manage the rest of the transfer process. The source sends the request to the controller, and receive the proposed data transfer path from the controller. The controller also passes this information to the data-spots and the receiving data center. The receiving data center is also responsible to ensure that all the data chunks have been successfully received. Otherwise, it informs the controller for the re-transmission of missing chunks. Further, data integrity is also checked on the entire received data at the destination as well. The functionality of source and destination center is also stated in Algorithm 4 and 5.

The framework has been developed to handle *data* that is usually significant in size. The system splits the data into fixed size chunks/blocks, and tracks the movement of these data chunks along their journey to the destination, from one data-spot to another. The system is thus aware of all data chunks that are in motion or are stored at any intermediate data-spot.

Fig. 3 shows the complete flow diagram of the proposed framework. The execution is triggered by the source data center through its data transfer request to the controller. The controller computes route based on shortest path algorithm.

Algorithm 4 Sender Data-Center ▷ A Datacenter Calls the Controller for Chunk Transmission Tasks

```
bigData ← null
decision ← null
msg ← null
vehicle v ← null
D_chk ← null
check dataQueue
while (dataQueue! = Empty) do
    bigData ← dataQueue_pop
    decision ← call controller.calcRoute(request)
    if (decision == Route) then
        v_i ← call v_i.pickUpData()
        D_chk ← pickChunkOf(bigData)
        Load(D_chk, v_i)
    end if
    check dataQueue
end while
check msgQueue
while (msgQueue! = Empty) do
    msg ← msgQueue_pop
    if (msg.Type == decision) then
        if (decision == Route) then
            vehicle ← call v_i.pickUpData()
            dataChunk ← pickChunkFrom(bigdata)
            Load(D_chk, v_i)
        end if
    else if (msg.Type == retransmission) then
        v_i ← call v_i.pickUpData()
        Load(D_chk, v_i)
    else
        Send(acknowledgement)
    end if
end while
```

Algorithm 5 Receiver Data-Center ▷ Receiving Data-Center Checks for Missing Data and Requests Retransmission If Necessary

```
D_chk ← null
check globalDataQueue
while (globalDataQueue! = Empty) do
    D_chk ← globalDataQueue_pop
    if (D_chk.Status==missing) then
        Send(retransmitReq, Controller)
    end if
    check globalDataQueue
end while
```

The calculated routes also include the intermediate data-spots. The controller sends the computed information to all the data-spots and the destination node. As soon as the source center receives the route information from the controller, it starts generating transfer request messages. The nearby vehicles receive the message and based on its availability

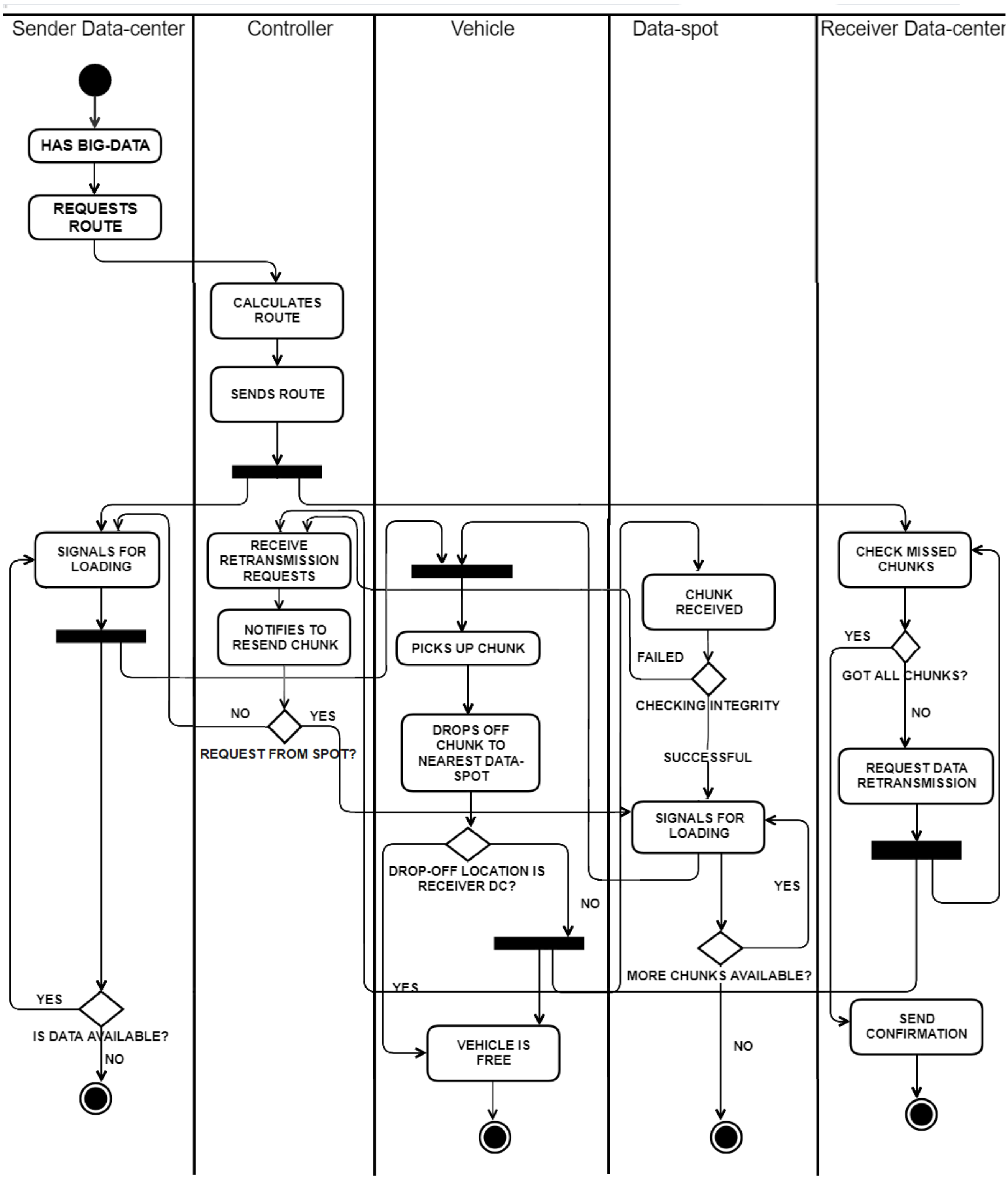

FIGURE 3. Flow diagram of proposed framework.

can accept the request by sending transfer accept message. Once the request is confirmed, the vehicle drives to the source data center docking station and the data chunks are uploaded from the source center through a wired connection. The data is upload depending upon the storage capacity of the vehicle, which may vary from vehicle to vehicle. Moreover, the route information is also shared with the driver, which is used to identify the closest data-spot from the driver's own destination. The vehicle delivers the data to the selected data-spot once it is about to deviate from the data path. The transfer process at data-spot checks the data integrity while downloading the data from the vehicle.

Moreover, at each data-spot, there are fixed number of docking stations where vehicles can park to upload or offload data and the data storage capacity at each data-spot is also limited. However, there may be cases when storage capacity at data-spot is fully utilized, not available at a particular time or a vehicle leaves the effective path without offloading data due to a personal emergency. In such cases, missing data chunks are identified at the destination node through missing sequence numbers. Therefore, the destination node sends the retransmission request to the controller for the missing chunks. Note that all data being transported through this framework is first encrypted and the chunks are uploaded in a random order to make sure that a rogue vehicle cannot get useful information out of the chunks being transferred. Further, the destination node sends acknowledgement to the controller after receiving all the data chunks associated with a job. On receiving the acknowledgment, controller informs the source node about the successful data transfer.

The proposed framework can be used to reduce the overall energy consumption and total delay in data transfer. In a typical scenario, the controller computes the best possible path with minimum energy to transfer the data. In this case only those vehicles can be selected that are moving close to the final destination of the data. Whereas, in another scenario, any vehicle can be selected for data transfer that promise to deliver to any of the intermediate data-spot. Additionally, the transfer can be supplemented by utilizing the existing traditional network from the source to the destination or from any data-spot to the destination.

A. SYSTEM MODEL FORMULATION

In this subsection, we formulate our system model for vehicular networks with respect to delay, capacity, bandwidth and energy consumption. The total time to transfer data from the source center to the destination is based on the geographical distance between source and destination, data upload and offload time to vehicles, number of data-spots utilized in the route and road speed limits along the route.

The total delay in transferring the data from source to destination using vehicles is shown in eq. 1.

$$Delay = \sum_{i=1}^{n}(T_{vj} + T_{dd} + T_{dest} + T_{du}) \tag{1}$$

where T_{vj} ($T_{vj} = PickupDistance/AvgSpeed$) represents the vehicle travel time to the assigned pickup location i.e. data center or data-spot. T_{dd} represents the chunk download time to vehicle. The T_{dest} ($T_{dest} = DestinationDistance/AvgSpeed$) represents the travel time to reach the destination spot and the T_{du} represents the chunk upload time from the vehicle. Here n represents the number of participating vehicles.

Every communication involves multiple data transfers for example for single hop, the source node transfers data chunk to a vehicle and vehicle offloads it directly to the destination data center. Each additional data-spot involves data transfers, vehicle time to reach the pick-up location and travel time to the destination. Thus, the more number of data-spots are involved, the delay increases accordingly.

The total volunteer vehicles V_{veh} with carrying capacity (C) can be calculated as:

$$V_{veh} = \sum_{v=0}^{N} V_i \times \rho \tag{2}$$

where ρ represents the probability of vehicles to participate in data transfer and N represent the total vehicle density. Using eq.2, we can compute the system capacity as:

$$S_c = \sum_{v=0}^{V_{veh}} V_i C_i \tag{3}$$

Two terms in the total data transfer delay in eq. 1 (T_{dd} and T_{du}) depends on the data transfer capacity available at source and data-spots. We assume a data transfer rate of 20 Gbps typical for USB 3.2. For the worst case analysis, we consider that all data is sequentially loaded on the available vehicles (neglecting any operation that can be done in parallel). We thus have

$$GD_{tran} = \frac{D_{vol}}{T_r} \tag{4}$$

where D_{vol} represents the total data transfer volume. T_r represents the data transfer rate and GD_{tran} represents the time taken for gross data transfer. In case of data transfer through Internet, let's assume B_{int} is the available bottleneck bandwidth of the network, thus the delay (inclusive of processing, transmission, propagation and queuing delays) can be calculated as follows:

$$Delay_{int} = \frac{D_{vol}}{B_{int}} \tag{5}$$

Similarly, the total energy consumption in data transfer via vehicles is based on multiple factors that include data upload, offload, and energy required to carry the data (vehicle) from one location to another. The energy consumption in moving data from one location to another location is stated in Equation. 6 below.

$$E_{veh} = e_u + e_o + e_t \tag{6}$$

where
e_u is the energy consumed during data upload
e_o is the energy consumed during data offload
e_t is the energy consumed by the vehicle moving from source to destination. Therefore, the potential objective is to minimize the E_{veh} and efficiently utilize the available capacity. We can compute e_o and e_u as shown below:

$$e_u = e_o = \left[\sum_{i}^{V_{veh}} V_i\right] \times e_{dd} \tag{7}$$

where V_{veh} is the number of participating vehicles and e_{dd} is the amount of energy consumed while transferring data chunks from data-spot to vehicle device or from vehicle to the data-spot. For V_{veh} participating vehicles, the transport energy e_t can be computed as follows:

$$e_t = [F_c \times \sum_{i=0}^{V_{veh}} \frac{V_i Dc_i}{Fe_i}] + \omega \tag{8}$$

where Dc is the total distance traveled by the vehicle (reaching the source and from source to destination), ω is the delay factor, Fe_i is the fuel economy of the vehicle and F_c is the conversion factor. If the vehicles are voluntarily participating in the data transfer, then the total system cost only include cost for loading and offloading the data. In this case, we can reduce eq. 6 as shown below:

$$E_{veh} = e_u + e_o \tag{9}$$

V. CASE STUDY FOR EVALUATION

In order to represent the complex dynamics and to effectively represent structure and behavior of different entities, our proposed framework uses the potential of agent based modeling (ABM) paradigm. The use of ABM helps to express the states and the interactions of different entities at a micro-level. Our framework further uses Anylogic Simulation Software to implement our framework. Anylogic provide support for modeling complex road networks using spatial elements that help represent realistic traffic scenarios. Our framework therefore provides a novel and flexible implementation of a dynamic, agent-based, hierarchical, multi-resolution modeling, simulation and visualization tool to support vehicle assisted bulk data transfer. We have evaluated the performance of our proposed framework by employing the metrics of utilization, delay incurred, failure rate etc. We have made a comparative analysis of the data transfer through Internet versus vehicular data transfer evaluating all these aforementioned metrics. The topology used for the evaluation is shown in Fig. 4. Here, we have considered multiple data-centers located in different cities. The average distance between the source and destination is around 340kms. The entire system is also connected through a cellular network. The connectivity is to ensure communication with the central controller. This can help to track the data chunk position in terms of data-spots and allow source and destination nodes to easily communicate with the controller.

The parameters used to evaluate our proposed framework are listed in Table 2. Here, we have considered two cases i) vehicles with fixed storage capacity and ii) vehicles with heterogeneous storage capacity. In case of fixed capacity, we have assumed one tera-byte (TB) storage; whereas, in the heterogeneous capacity model, every vehicle have different capacity in the range 100-1000 GB depending upon their type. The results are obtained in terms of the number of vehicles used to complete the data transfer task, data chunks created, delivered or undelivered chunks and time required to complete the job. Here, we have considered that the source

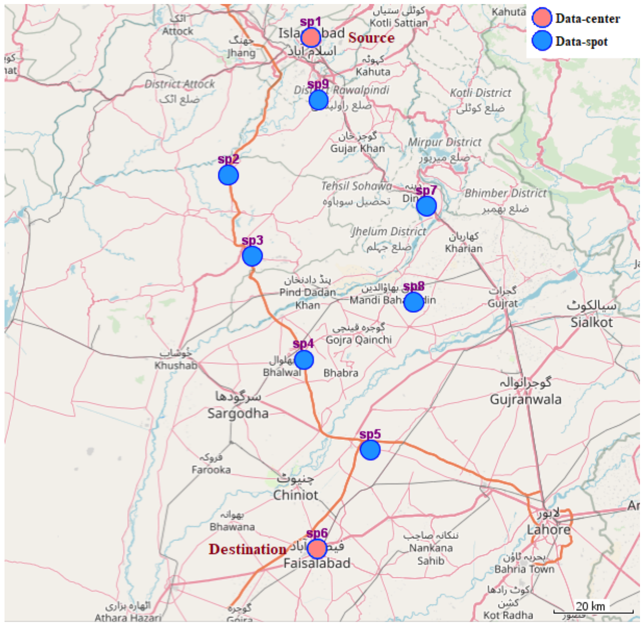

FIGURE 4. Topology diagram used for evaluation.

TABLE 2. Simulation parameters.

Parameter	Value
Simulator	Anylogic PLE version 8.2.3
Vehicle storage capacity (GB)	Fixed/Variable
Vehicle density	1000
Volunteer Vehicles participated	650
Data-spots Storage capacity (GB)	20,000
Number of Data-spots	09
Data transfer rate (Gbps)	20
Data block size (GB)	Fixed/Variable
Total data volume (GB)	20,000
Average vehicle speed (Km/hr)	80
Distance from src to dest(Km)	340
Vehicular frequency per Km	8
Storage Device	Samsung Portable SSD T1
F_c(J/L)	37624722.29

center (spot1/sp1) has two independent jobs $j1$ and $j2$ of data transfer from source to destination (spot6/sp6). The size of total data in both j_1 and j_2 is 10 TB each. The destination location is the same for both the jobs.

VI. RESULTS & EXPLANATION

Fig. 5 shows the data-spot usage in terms of data transfer. As it can be seen in Fig. 5a, the data transfers is initiated at spot1 and its usage declining with the passage of time as data is transferred to vehicles while usage at the destination spot6 is increasing as data is finally downloaded at the destination. Intermediate spots also participate in the data transfer (multi-hop transfer) depending on their respective locations. With fixed size vehicle capacity (1 TB), both the jobs are completed within 3.2 days time as shown in Fig. 5b; whereas, with the variable size chunks it takes more than 20 days. In variable size vehicle storage capacity, the overhead of data offload and upload can easily be seen in Fig. 5a. As more vehicles approach the data-spot, more are the

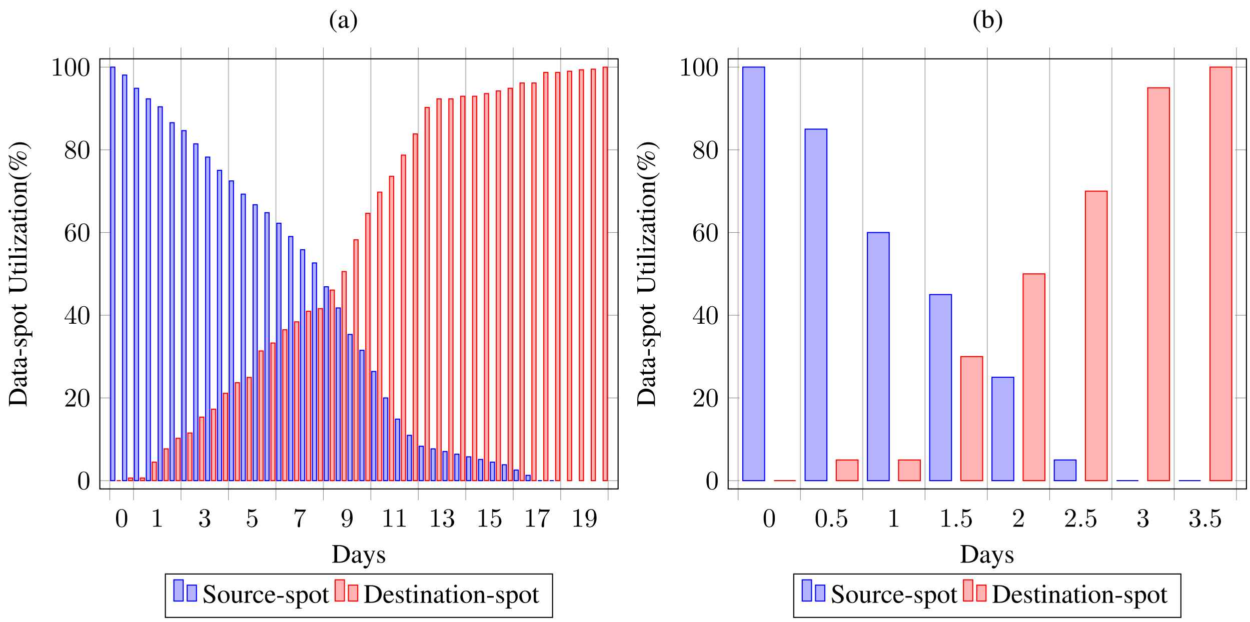

FIGURE 5. Data-spot utilization. (a) Vehicles with variable carrying capacity. (b) Vehicles with fixed carrying capacity.

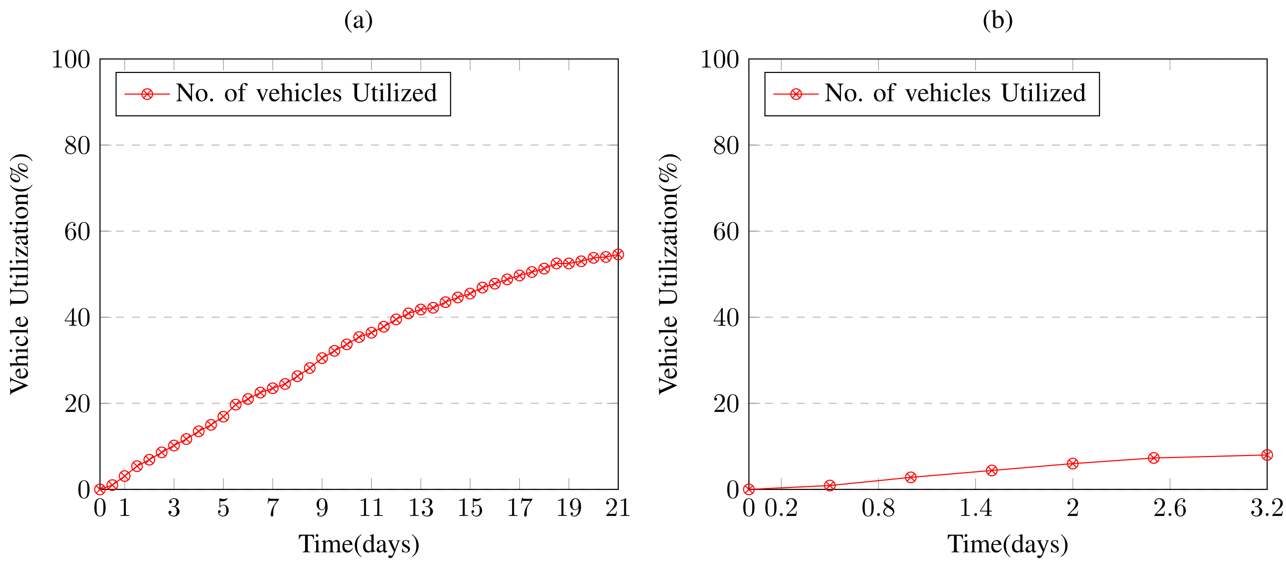

FIGURE 6. Vehicle utilization. (a) Vehicles with variable carrying capacity. (b) Vehicles with fixed carrying capacity.

chances that vehicles leave the data-spot without the offload due to unavailability of docking stations. Thus, this can also increase the number of unsuccessful jobs.

Fig. 6 show the number of vehicles that participated in data transfer activity. The figure also shows the number of vehicles increasing with the passage of the time, this is due to the fact that initially there is one source center and after the data chunk is delivered to a data-spot, it also becomes a source. Thus, the number of vehicles is increasing with increase in data spots. Moreover, in Fig. 6a, around 50% of the available vehicles are used and with fixed carrying capacity only 10% vehicles are utilized as shown in Fig. 6b. In the fixed capacity case, as the individual storage capacity for each vehicle is quite significant, thus, the entire data move very efficiently in significantly less time involving only 10% of the available vehicles.

Fig. 7 shows the behavior of vehicles in variable or fixed capacity environment. The figure shows that out of total available vehicles, the percentage of vehicles that completed the job by successfully delivering the data chunk to any of the data-spots, the number of vehicles not used for data transfer and vehicles that quitted tasks i.e., got the data but failed to deliver it back to any system node. Fig. 7a shows that in variable storage capacity, out of 1000 vehicles, 54.6% of the vehicles are used and 5.4% of the vehicles failed to perform the assigned tasks. This means that a total of 540 vehicles

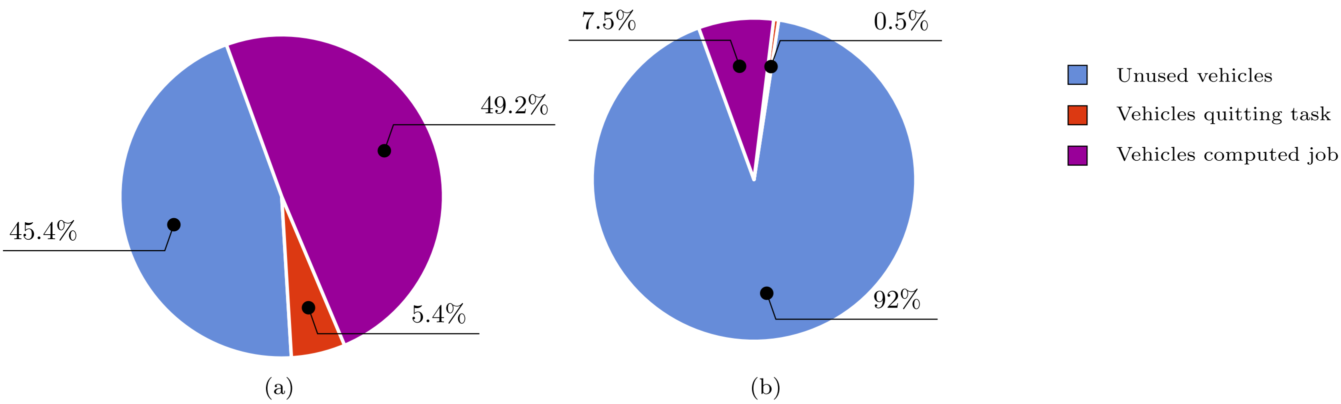

FIGURE 7. Overall performance showing completed and incomplete tasks. (a) Vehicles with variable carrying capacity. (b) Vehicles with fixed carrying capacity.

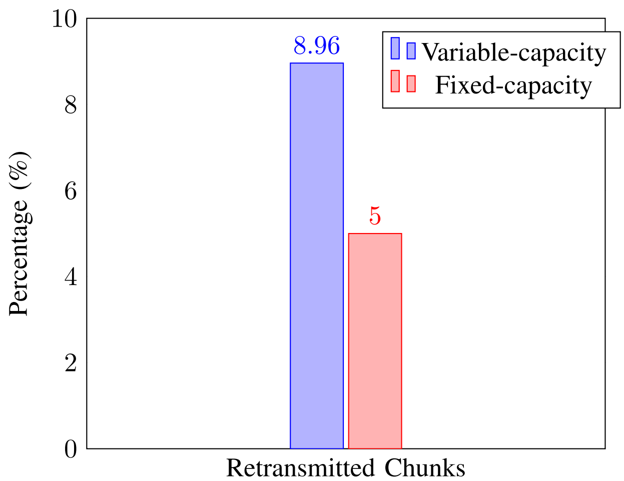

FIGURE 8. Number of re-transmitted chunks for both the jobs.

are assigned tasks, out of which 492 successfully completed their tasks while 54 vehicles failed, either due to unavailable storage space at a data-spot or driver's personal choice. On the other hand, in fixed capacity case large blocks are transferred with few vehicles. Total 8% of the vehicles participated and 0.5% of failed to deliver chunks back to the system i.e. from 1000 vehicles only 80 vehicles are used for data transfer, only 5 vehicles left their tasks incomplete, which later on were successfully completed by other vehicles. Thus, in a variable carrying capacity case, the number of retransmission requests are also more, thus, causing an overhead for the entire system.

Fig. 8 shows the amount of data failed to deliver due to either non-availability of a docking spot, fully utilized storage capacity or vehicle deviated due to other reasons. Therefore, in that case, the chunk is re-transmitted either from the previous data-spot or the source location. In Fig. 8, with variable capacity vehicles, a large number of vehicles are used to transfer a data, thus, that increases the chances for unsuccessful delivery compared to fixed capacity, as a result, larger number of chunks are re-transmitted.

Fig. 9a and Fig. 9b shows the amount of time it takes to complete both the jobs successfully. For variable capacity, it takes almost 21 days, whereas, for fixed capacity, it took around 3.2 days to complete both j_1 and j_2. These graphs depict the overall data received at the destination data center per day accounting for the data lost in transit and its recovery through the re-transmission process.

In Fig. 10, we have analyzed the ideal scenario, where we have assumed that there is no data loss and the storing capacity at data-spots is unlimited. Based on these assumptions, result shows that in order to deliver 20 Tera-bytes, it takes a little less than 3 days. Whereas, with all the constraints intact, we have achieved the data transfer in 3.2 days approximately. This includes re-transmission and data-spot capacity overflow. Through experimental results, we can conclude that the data transfer via high fixed size storage capacity provides better results as compared to vehicles with variable capacities.

These experimental results show that data transfer through vehicles is a viable solution in terms of energy and delay. It highlights that the vehicle carrying capacity has a significant impact on the transfer. In case of fixed carrying capacity, the entire transfer can be completed much earlier as compared with varying capacity vehicles. The transfer also depends on the density of the vehicles in a particular area and volunteers participating in the transfer.

VII. GENERAL DISCUSSION

Table 3 shows the comparison of our proposed vehicle based data transfer framework with other existing works. Most of these work (except the Amazon) use radio communications for uploading/downloading data to/from the vehicles. Using wireless communications is less secure compared to our approach. Moreover, the data and error rates through wireless is also a challenge that require careful consideration. A large body of work is based on Unmanned Aerial Vehicles (UAVs) for data transfer which has inherent battery limitations; therefore, UAVs can only cover a limited area. Moreover, UAVs are expensive and usually limited in number. Using UAVs for data mule also raise security questions. Amazon snowball and snowmobile require prior scheduling and user has to bear the cost of data transfer. For massive data transfer through snowmobile, such scheduling takes tens

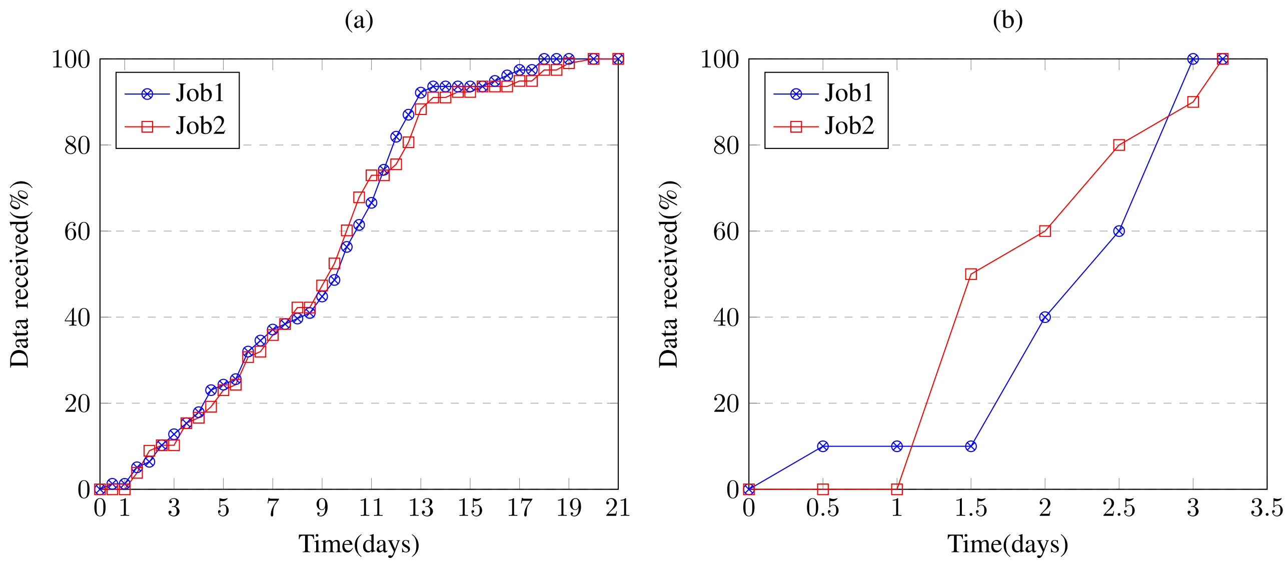

FIGURE 9. Amount of time it takes to deliver no. of chunks to the receiver. (a) Vehicles with variable carrying capacity. (b) Vehicles with fixed carrying capacity.

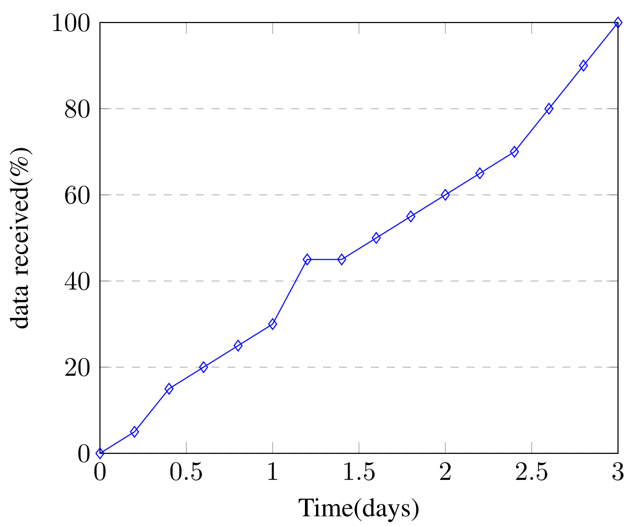

FIGURE 10. An ideal case - no chunk was missed.

of days for an appointment. Other relevant contributions are not open-source and hence not available for researchers to benchmark their own route or vehicle selection algorithms.

Our proposed framework is based on vehicles to transfer data from source to destination either in single hop or multiple hops using the intermediate data-spots. With the popularity of smart vehicles and smart cities, the role of data transfer can thus be transferred to vehicles to save network communication energy. Transfer of a large amount of data through communication network can not only chock the entire network but also affect other applications. The framework would cause the maximum delay when the data is repeatedly offloaded and reloaded at each of the data-spots or when large number of vehicles leave the system without offloading the data chunks back into the system. The average upload and download data transfer rates in traditional network varies from continent to continent. Fig. 11 shows the upload time of 20TB in each of the mentioned continents using the data rate mentioned in [29]. For example, in Africa, it would take around 38 months to transfer 20TB of data and whereas in Australia it takes around 4 months. Note, here we are only considering the upload time involved, neglecting the transfer time and download time.

In the proposed framework, the controller is responsible to discover the optimal route between the source and the destination. It also makes a decision that whether the data should be transmitted through traditional or vehicular networks. This decision is taken based on transfer time and the availability of volunteer vehicles near the source to begin with and near data-spots afterwards. In future, we are interested to add more sophisticated algorithms to compute the transfer path from source to destination.

Energy – Traditionally, Internet is widely used for transferring data. The energy associated with the communication through the Internet can be categorized as the end-user cost which is around 38% of the total energy consumption and the rest of the 62% energy is consumed by the network nodes i.e. routers, switches, repeaters, servers and the data-centers [30].

Based on the parameter values of core network devices used in Internet data transfer, Costenaro and Duer [30] estimated the energy consumed by the Internet in kWh per GB is approx. 5.12 kWh/GB. Considering the estimation to calculate the 20TB data transfer, total 102.4 MWh energy is consumed by the Internet data. On the other hand, to compute energy consumed using vehicles for the data transfer, we have considered the average power consumption of a server equal to 145 watts per hour following the research work of Pries *et al.* [31]. Using USB 3.2 for the 1TB data transfer between the vehicle and the infrastructure, it takes 7 minutes, hence, the average power consumed by

TABLE 3. Benchmark comparison with existing frameworks.

Authors	Storage transfer capacity	Transfer Frequency	Data Trans. mode	Route requirment	Reliability of Trans. tasks	Framework	Comm. between devices	Security
Amazon Snowmobile	100PB	One time and Scheduled on demand	Large container truck and specialized driver	Limited specialized regions	×	Amazon Platform	Wired	✓
Amazon Snowball	<100TB	One time and Scheduled on demand	Large container truck and specialized driver	Limited specialized regions	×	Amazon Platform	Wired	✓
Naseer et al. [7]	0.5TB to 1TB	LTE D2D limits	Vehicles can participate	Limited to routes close to cellular networks	×	Theoretical work	WiFi	×
Cheng et al. [16]	unspecified	unspecified	Unmanned Aerial Vehicles	Actual flight paths	×	- -	Radio comm.	✓
Palma et al. [17]	unspecified	Scheduled own resources	Unmanned Aerial Vehicles	Fixed route	×	Emulated Environment	Radio comm.	×
Usbeck et al. [18]	unspecified	Scheduled own resources	Unmanned Aerial Vehicles	unspecified networks	×	Android Platform	Radio comm.	×
Hunjet et al. [19]	unspecified	Scheduled own resources	Unmanned Aerial Vehicles	Tactical network	×	MASON framework	Radio comm.	×
Rashmi et al. [26]	2TB	Scheduled own resources	Public Buses	Based on User Profiles	×	ONE	WiFi-direct	×
Coutinho et al. [28]	unspecified	Scheduled own resources	Boats	unspecified network	×	ONE and NS2	Radio comm.	×
Proposed System	Varies. Split into chunks	Continuous and flexible limits	Any vehicle/driver can participate	Flexible Routes	✓	Open-source framework	USB 3.2	✓

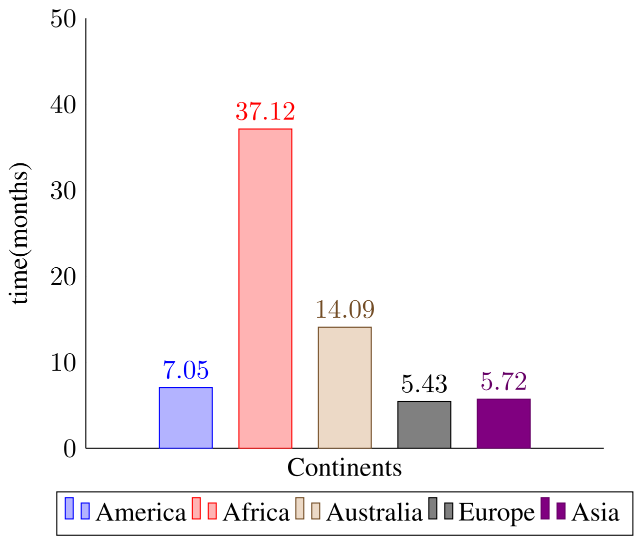

FIGURE 11. Average upload time for 20TB in sub-continents.

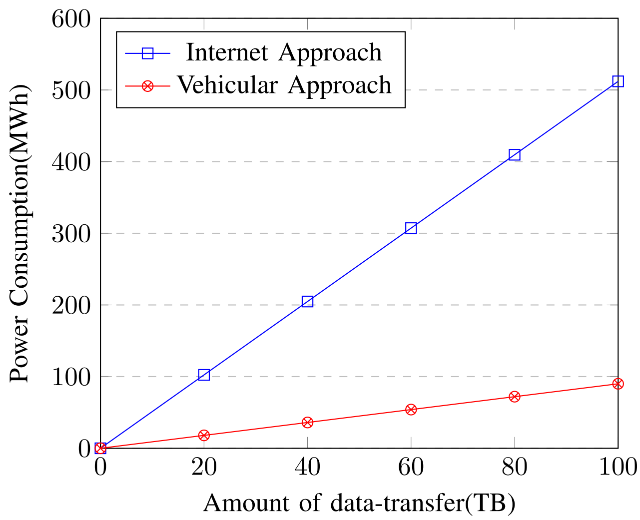

FIGURE 12. Average energy consumption of Internet vs. Vehicular approach.

a server for this (7 minutes) duration is around 17watt/min approximately. Similarly, using eq. 7 we can get the entire cost of 20 TB data upload or offload, which is equal to 571.2kWh. Furthermore, using eq. 8, the transportation cost of a vehicle comes out to be 16723kWh approximately. Therefore, the total vehicular energy consumed is equal to 17.864MWh. In order to compute the total cost in both the cases, we have considered the average electricity cost i.e. $0.112 per Kilowatt hour (kWh) as standardized by U.S. Energy Information Administration [32]; therefore, based on the energy consumption for data transfer through Internet, the cost of system energy for 20 TB is estimated as $11469 approximately, whereas for the proposed vehicular approach the cost is $2000 approx. Based on the above discussion, Fig. 12 shows the energy consumption by varying the amount of data transferred using the Internet or proposed vehicular approach.Thus, for both energy as well as delay, our vehicular based data dissemination approach performs better than Internet transfer.

Our approach to big data transfer via vehicular networks presents a significant level of improvement in terms of information security as compared to transfer via the Internet. Since data transfer via vehicular networks can be considered a case of *data at rest* it presents a significant reduction in terms of attack surface. Data theft requires relying on physical attacks such as robbery or burglary of the storage equipment installed in vehicles. The attacker cannot as easily remain covert or stealthy as in the case of eavesdropping or tampering attacks on *data in transit* in networks. A second major contribution is the leveraging of advantages offered by the Internet of Things (IoT) paradigm. The leaps and bounds in the development of low-cost and smart storage and computing devices will enable our framework to utilize economic and smart storage devices. Rapid advances in smart vehicle technology that allow for remote vehicle diagnosis, self-routing and collision avoidance in cars amongst other features means that we rely on technology that is going to get very robust and reliable in the future.

VIII. CONCLUSION

In this paper we have proposed a framework that utilizes vehicular networks for big data transfer between data centers. The framework supports urban/social computing through participation of volunteer vehicles and helps in reducing the carbon footprint associated with movement of Exabyte scale of data. Simulation results show that this approach is efficient in terms of energy savings as well as the utilization of resources. The framework also facilitates the researchers to incorporate different algorithms to optimize the data transfer mechanism. In future, we aim to extend our framework by incorporating Artificial Intelligence and machine learning techniques to select the suitable vehicle for data transfer; thus, this can improve the task delivery ratio further. Moreover, hybrid techniques, where part of data is transfer through vehicle and remaining through Internet can also be incorporated. The proposed framework has set the platform for the researchers to incorporate their own vehicle routing algorithms to improve energy and reduce the delay for data transportation in smart cities.

REFERENCES

[1] M. D. Lytras and A. Visvizi, "Who uses smart city services and what to make of it: Toward interdisciplinary smart cities research," *Sustainability*, vol. 10, no. 6, p. 1998, Jun. 2018.

[2] A. Visvizi, M. D. Lytras, E. Damiani, and H. Mathkour, "Policy making for smart cities: Innovation and social inclusive economic growth for sustainability," *J. Sci. Technol. Policy Manage.*, vol. 9, no. 2, pp. 126–133, Jul. 2018.

[3] H. Habibzadeh, Z. Qin, T. Soyata, and B. Kantarci, "Large-scale distributed dedicated- and non-dedicated smart city sensing systems," *IEEE Sensors J.*, vol. 17, no. 23, pp. 7649–7658, Dec. 2017.

[4] W. Yu *et al.*, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.

[5] W. Chen and T. Zhang, "Fog computing," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 4–6, Mar. 2017.

[6] C. Gosman, T. Cornea, C. Dobre, F. Pop, and A. Castiglione, "Controlling and filtering users data in intelligent transportation system," *Future Gener. Comput. Syst.*, vol. 78, pp. 807–816, Jan. 2018.

[7] S. Naseer, W. Liu, N. I. Sarkar, P. H. J. Chong, E. Lai, and R. V. Prasad, "A sustainable vehicular based energy efficient data dissemination approach," in *Proc. 27th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2017, pp. 1–8.

[8] (2018). *IoT Report: How Internet of Things Technology is Now Reaching Mainstream Companies and Consumers*. Accessed: Oct. 2018. [Online]. Available: https://www.businessinsider.com/internet-of-things-report

[9] S. M. Idrees, M. A. Alam, and P. Agarwal, "A study of big data and its challenges," *Int. J. Inf. Technol.*, vol. 4, pp. 1–6, Aug. 2018.

[10] (2015). *What Happens in an Internet Minute? How to Capitalize on the Big Data Explosion?* Accessed: Oct. 2018. [Online]. Available: https://www.excelacom.com/resources/blog/what-happens-in-an-internet-minute-how-to-capitalize-on-the-big-data-explosion

[11] I. Marincic and I. Foster, "Energy-efficient data transfer: Bits vs. atoms," in *Proc. 24th Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, Sep. 2016, pp. 1–6.

[12] C. Zhu, G. Cheng, and K. Wang, "Big data analytics for program popularity prediction in broadcast TV industries," *IEEE Access*, vol. 5, pp. 24593–24601, 2017.

[13] D. Sitaram and G. Manjunath, "Managing the cloud," in *Moving to the Cloud*, D. Sitaram and G. Manjunath, Eds. Boston, MA, USA: Syngress, 2012, pp. 1–449.

[14] (2018). *Digital Globe Company Supported by Amazon Web Services*. [Online]. Available: https://aws.amazon.com/solutions/case-studies/digitalglobe/

[15] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2018.

[16] C.-M. Cheng, P.-H. Hsiao, H. Kung, and D. Vlah, "Maximizing throughput of UAV-relaying networks with the load-carry-and-deliver paradigm," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2007, pp. 4417–4424.

[17] D. Palma, A. Zolich, Y. Jiang, and T. A. Johansen, "Unmanned aerial vehicles as data mules: An experimental assessment," *IEEE Access*, vol. 5, pp. 24716–24726, 2017.

[18] K. Usbeck *et al.*, "Data ferrying to the tactical edge: A field experiment in exchanging mission plans and intelligence in austere environments," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2014, pp. 1311–1317.

[19] R. Hunjet *et al.*, "Data ferrying with swarming UAS in tactical defence networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Aug. 2018, pp. 6381–6388.

[20] B. Baron, P. Spathis, H. Rivano, M. D. de Amorim, Y. Viniotis, and M. H. Ammar, "Centrally controlled mass data offloading using vehicular traffic," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 2, pp. 401–415, Jun. 2017.

[21] O. Altintas, F. Dressler, F. Hagenauer, M. Matsumoto, M. Sepulcre, and C. Sommery, "Making cars a main ICT resource in smart cities," in *Proc. IEEE Conf. Comput. Commun. Workshops*, May 2015, pp. 582–587.

[22] B. Li, Y. Pei, H. Wu, Z. Liu, and H. Liu, "Computation offloading management for vehicular ad hoc cloud," in *Algorithms and Architectures for Parallel Processing*. New York, NY, USA: Springer, 2014, pp. 728–739.

[23] F. Dressler, P. Handle, and C. Sommer, "Towards a vehicular cloud-using parked vehicles as a temporary network and storage infrastructure," in *Proc. ACM Int. Workshop Wireless Mobile Technol. Smart Cities*, Aug. 2014, pp. 11–18.

[24] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: A measurement study and implications for network applications," in *Proc. 9th ACM SIGCOMM Conf. Internet Meas.*, Nov. 2009, pp. 280–293.

[25] B. Kim, B. Gu, J. Kim, C. Lim, and J. Hong, "Energy efficient power management for multi-mode wireless network devices," in *Reliable and Autonomous Computational Science* (Autonomic Systems), S. Shin, R. Gantenbein, T. W. Kuo, and J. Hong, Eds. Basel, Switzerland: Springer, 2011, pp. 223–243.

[26] R. Munjal, W. Liu, X. J. Li, J. Gutierrez, and M. Furdek, "Sustainable massive data dissemination by using software defined connectivity approach," in *Proc. 27th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2017, pp. 1–6.

[27] Y. Chen, L. Li, and Z. Chen, "An approach to verifying data integrity for cloud storage," in *Proc. 13th Int. Conf. Comput. Intell. Secur. (CIS)*, Dec. 2017, pp. 582–585.

[28] M. M. Coutinho, A. Efrat, T. Johnson, A. Richa, and M. Liu, "Healthcare supported by data mule networks in remote communities of the Amazon region," *Int. Scholarly Res. Notices*, pp. 1–8, Oct. 2014.

[29] (2018). *Speedtest Market Reports of Calculating Upload and Download Data Rates in 2017–18*. Accessed: Oct. 2018. [Online]. Available: http://www.speedtest.net/reports/

[30] D. Costenaro and A. Duer, "The megawatts behind your megabytes: Going from data-center to desktop," in *Proc. ACEEE Summer Study Energy Efficiency Buildings*, Washington, DC, USA, 2012, pp. 13–65.

[31] R. Pries, M. Jarschel, D. Schlosser, M. Klopf, and P. Tran-Gia, "Power consumption analysis of data center architectures," in *Proc. Int. Conf. Green Commun. Netw.* New York, NY, USA: Springer, 2011, pp. 114–124.

[32] (Feb. 2018). *Annual Energy Outlook 2018, US Energy Information Administration*. Accessed: Oct. 2018. [Online]. Available: https://www.eia.gov/outlooks/aeo/pdf/aeo2018.pdf

MURK received the bachelor's degree in software engineering from the Mehran University of Engineering and Technology (MUET), Jamshoro, Sindh, Pakistan. She is currently pursuing the M.Sc. degree in information technology from the National University of Sciences and Technology (NUST), Islamabad, Pakistan. Her interests include modeling and simulation, the IoT, and advanced database concepts.

ASAD WAQAR MALIK received the Ph.D. degree in parallel and distributed simulation/systems from the National University of Science and Technology (NUST), Pakistan, in 2012. He is currently an Assistant Professor with the Department of Computing (DOC), School of Electrical Engineering and Computer Science (SEECS), NUST. Besides, he is also a Senior Lecturer with the Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. His primary area of interest includes distributed simulation, cloud/fog computing, and the Internet of Things.

IMRAN MAHMOOD received the master's degree in software engineering and the Ph.D. degree in software and computer systems from the School of Information and Communication Technology (ICT), KTH-Royal Institute of Technology, Sweden, in 2007 and 2013, respectively. He is currently an Assistant Professor with the Department of Computing (DoC), School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology, Pakistan, and also a Director with the Center for Research in Modeling and Simulation (CRIMSON). His research interests include applied modeling and simulation, distributed artificial intelligence, multi-agent systems; analysis and formal verification of complex systems, and has several years of experience in the respective fields.

NADEEM AHMED received the M.S. and Ph.D. degrees in computer sciences from the University of New South Wales (UNSW), Sydney, Australia, in 2000 and 2007, respectively. His research interests include wireless sensor networks, data center networking, vehicular cloud computing, and software defined networks.

ZAHID ANWAR received the M.S. and Ph.D. degrees in computer sciences from the University of Illinois at Urbana-Champaign, in 2005 and 2008, respectively. He was a Software Engineer and a Researcher at IBM, Intel, Motorola, National Center for Supercomputing Applications (NCSA), xFlow Research, and CERN on various projects related to information security, operating systems design, and data analytics. He held a postdoctorate position with Concordia University. He was a Faculty Member with the University of North Carolina at Charlotte. He is currently a Faculty with the National University of Sciences and Technology and an Associate Professor with Fontbonne University.