Edge Computing in the Industrial Internet of Things Environment: Software-Defined-Networks-Based Edge-Cloud Interplay

Kuljeet Kaur, Sahil Garg, Gagangeet Singh Aujla, Neeraj Kumar, Joel J. P. C. Rodrigues, and Mohsen Guizani

The authors present an SDN-based edge-cloud interplay to handle streaming big data in IIoT environments, wherein SDN provides an efficient middleware support. In the proposed solution, a multi-objective evolutionary algorithm using Tchebycheff decomposition for flow scheduling and routing in SDN is presented.

Abstract

The emergence of the Industrial Internet of Things (IIoT) has paved the way to real-time big data storage, access, and processing in the cloud environment. In IIoT, the big data generated by various devices such as-smartphones, wireless body sensors, and smart meters will be on the order of zettabytes in the near future. Hence, relaying this huge amount of data to the remote cloud platform for further processing can lead to severe network congestion. This in turn will result in latency issues which affect the overall QoS for various applications in IIoT. To cope with these challenges, a recent paradigm shift in computing, popularly known as edge computing, has emerged. Edge computing can be viewed as a complement to cloud computing rather than as a competition. The cooperation and interplay among cloud and edge devices can help to reduce energy consumption in addition to maintaining the QoS for various applications in the IIoT environment. However, a large number of migrations among edge devices and cloud servers leads to congestion in the underlying networks. Hence, to handle this problem, SDN, a recent programmable and scalable network paradigm, has emerged as a viable solution. Keeping focus on all the aforementioned issues, in this article, an SDN-based edge-cloud interplay is presented to handle streaming big data in IIoT environment, wherein SDN provides an efficient middleware support. In the proposed solution, a multi-objective evolutionary algorithm using Tchebycheff decomposition for flow scheduling and routing in SDN is presented. The proposed scheme is evaluated with respect to two optimization objectives, that is, the trade-off between energy efficiency and latency, and the trade-off between energy efficiency and bandwidth. The results obtained prove the effectiveness of the proposed flow scheduling scheme in the IIoT environment.

Introduction

The need for on-demand state-of-the-art services (smart sensing, e-healthcare, smart transportation, etc.) and computing infrastructure has paved way to the powerful paradigm of cloud computing. Ever since its inception in 2000, the cloud computing paradigm has witnessed significant transitions in its overall usage, size, computational ability, and underlying technology [1]. This is evident from the widescale adoption of cloud computing infrastructure by the IT vendors. According to 451 Research, the cloud services would witness an overall increase in their worldwide market from $21.9 billion in 2016 to $44.2 billion by 2020 [2]. This tremendous popularity can be attributed to the cloud's "pay-per-use" model, wherein users utilize the available resources of storage, computation, and networking as per their demands. The National Institute of Standards and Technology (NIST) has listed five essential attributes of cloud computing:

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

These attributes are typically achieved by the underlying service oriented architecture (SOA) that supports services as per the respective enterprise model, that is, everything-as-a-service (EaaS or XaaS). These enhanced flexibility and reliability attributes offered by the cloud computing platform have led to its widespread popularity among academia and industry. However, with the emergence of the Internet of Things (IoT), the need for real-time data storage, access, and processing at the cloud has grown manifold. Moreover, the big data generated by the connected devices (smartphones, PDAs, wireless body sensors, smart meters, etc.) would be on the order of zettabytes in the near future. This is evident as per the latest statistics shared by Gartner in its annual report [3]. It advocates that nearly 50 billion devices would be connected to the Internet by 2020. Hence, the relaying of such huge data to the cloud infrastructure may create network bottlenecks in the future. Additionally, this would lead to latency issues which in turn may affect the overall quality of service (QoS) for various applications in this environment.

The Emergence of Edge Computing

In order to tackle the above limitations of the cloud platform, Cisco recently came up with the innovative concept of "edge computing." It is popularly known as the "cloud close to the

Digital Object Identifier:
10.1109/MCOM.2018.1700622

Kuljeet Kaur, Sahil Garg, Gagangeet Singh Aujla, and Neeraj Kumar are with Thapar University; Joel J. P. C. Rodrigues is with National Institute of Telecommunications (Inatel), Instituto de Telecomunicações, Portugal; ITMO University, and the University of Fortaleza (UNIFOR); Mohsen Guizani is with the University of Idaho.

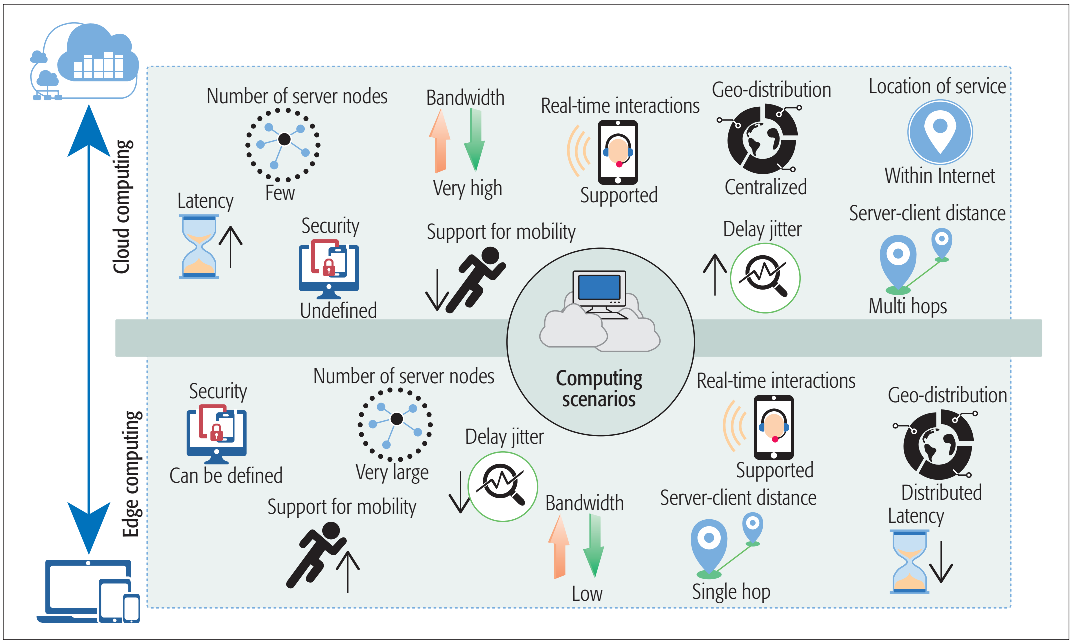

Figure 1. Comparison of cloud and edge computing.

The majority of IT giants, including Cisco and IBM, have initiated the process to push the computing capabilities to the edge of the networks, encapsulating the world with a range of edge devices in the form of devices, routers, and sensors. However, the most common problem with cloud computing is the bandwidth limitation.

ground," as it provides computational and processing facilities at the edge of the network. This is achieved by leveraging the machine-to-machine and device-to-device interactions supported by the edge devices in the form of nano-data centers (nDCs) and micro-DCs (mDCs). These devices are widely distributed to support real-time data processing. With this approach, the need to relay device data to the core has decreased to a large extent. The relative advantages of edge computing over cloud computing are shown in Fig. 1.

The majority of IT giants, including Cisco and IBM, have initiated the process to push the computing capabilities to the edge of networks, encapsulating the world with a range of edge devices in the form of devices, routers, and sensors. However, the most common problem with cloud computing is the bandwidth limitation. Per statistics shared by the World Economic Forum, the United States is ranked 35th worldwide with respect to bandwidth consumption per user. This implies that even the biggest competitors in the cloud computing era have bandwidth issues when it comes to relaying every bit of data to the core computing platform. This issue is expected to further proliferate with the rapid advances in IoT with heterogeneous physical entities connecting wirelessly to the cloud in the near future.

The layered architecture of the edge computing paradigm is depicted using Fig. 2. As shown in the figure, the computing architecture comprises seven layers, namely core computing devices, objective functions, networking devices, edge computing devices, access technology, applications, and actors. Here, the smart devices (mobiles, PDAs, etc.), vehicles, and individuals play the role of actors who try to access the different on-demand applications, including smart sensing, smart education, e-healthcare, smart transportation, and so on. To access these services, the actors need to connect to the computing platform using the available access technologies such as third generation (3G), 4G, 5G, WiFi, and Bluetooth [4]. The access technology then relays the user application/service requests to the nearby nDCs and mDCs, which in turn form the integral part of the edge computing devices layer. These devices have limited computational ability; hence, they forward the computing-intensive requests to the core computing layer. The requests are routed to this innermost layer via the networking devices like routers and switches for further processing. The entire computing architecture tends to address different objectives as part of the service level agreement (SLA). These objectives range from minimization of energy, latency, and cost to higher availability and throughput [1].

However, edge devices may consume high energy if energy efficiency is not taken into consideration. Moreover, in order to provide low latency and high data rates, energy consumption of nDCs may surge. For example, Wang *et al.* [5] highlighted that edge computing can gradually meet the requirements of large-scale mobile services in terms of lower latency, geo-distribution, and mobility support.

EDGE-CLOUD INTERPLAY

Edge computing provides dual functionality:
- Single-hop communication to mobile users to provide them services without assistance of cloud
- Connection to the cloud to leverage its high-end functionality and tools [6]

In this direction, Jalali *et al.* [7] presented a comparative study of energy consumption of nDCs and centralized DCs. The authors highlighted various factors in system design, such as type of access networks and server time utilization, that force nDCs to consume lower energy than centralized DCs. One of its major findings was that efficient cooperation between cloud and edge may lead to energy saving.

In this direction, Deng *et al.* [8] proposed an optimal workload allocation scheme in edge-cloud computing with respect to trade-off between delay

The dependence of traditional networks based on standard protocols and architecture may act as a bottleneck to address challenges in front of edge-cloud interplay. Hence, a programmable, scalable, flexible, vendor-independent, and reconfigurable network architecture is required to handle the huge amount of traffic between edge and cloud devices.

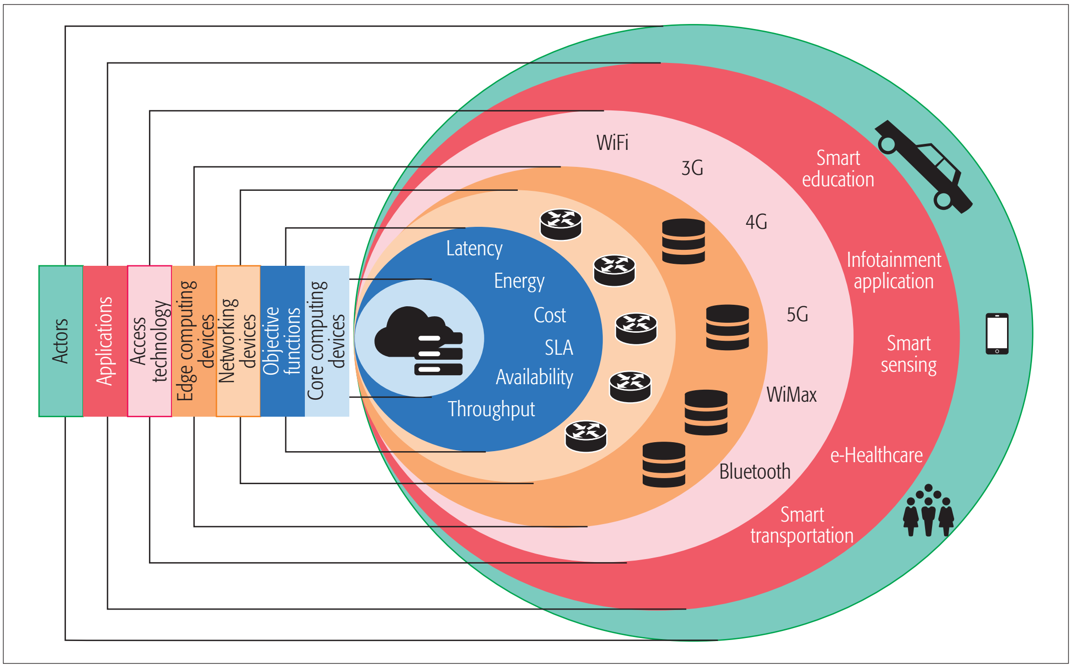

Figure 2. Layered architecture of the edge computing paradigm.

and power consumption. The authors suggested that it is important to study the impact of interplay and cooperation between cloud and edge in order to minimize energy consumption. Similarly, Borylo *et al.* [6] presented a dynamic resource provisioning scheme for energy-aware interplay between cloud and edge. The authors highlighted the use of software defined networking (SDN) in oder to minimize the energy consumption of the underlying networks between edge and cloud.

After analyzing the above discussions, some of the major findings and challenges of edge-cloud interplay are listed as follows:

- Limitation of resources at the edge devices requires shifting of major load on cloud DCs.
- Overloading of resources at an edge node may lead to higher energy consumption.
- Classification of delay-sensitive and resource-oriented requests among edge and cloud DCs improves performance and reduce energy consumption.
- Mobility of edge devices is a challenge for energy consumption due to re-traceability of lost links.
- The huge number of migrations and communications between edge and cloud devices leads to additional burden on underlying networks.
- The underlying network plays a major role in providing low-latency services to end users.

Proposed Edge-Cloud Interplay Architecture: To handle the above listed challenges, an edge-cloud interplay architecture is presented in this proposal. The edge-cloud interplay architecture consists of three layers:

1. The cloud computing layer
2. The edge computing layer
3. The network layer

Figure 3 shows the architecture of edge-cloud interplay along with various characteristics. The overall architecture relies on the middleware supported by the network layer, which is the core toward successful implementation of edge-cloud interplay. An inefficient network can lead to excessive energy consumption and high latency. Moreover, the dependence of traditional networks based on standard protocols and architecture may act as a bottleneck to address challenges in front of edge-cloud interplay. Hence, a programmable, scalable, flexible, vendor-independent, and reconfigurable network architecture is required to handle the huge amount of traffic between edge and cloud devices in order to improve performance and reduce energy consumption.

Emergence of SDN

To provide a flexible and scalable architecture for handling network congestion and complexity in edge-cloud environments, a recent out-of-the-box network technology, SDN, has emerged. In SDN, the data and control planes are decoupled from each other in order to reduce network congestion and complexity [9, 10]. Thus, SDN is the most viable network technology in the multi-edge-cloud environment. Generally, the communication infrastructure in SDN works according to standards designed by the Open Networking Foundation (ONF). OpenFlow protocol is used to handle the flow of traffic in SDN [11]. In the proposed edge-cloud interplay architecture, the three decoupled planes — data, control, and application — are shown in Fig. 3. The network backbone in the considered setup could range from 3G, 4G, and 5G to WiFi and Bluetooth. A detailed description of the communication protocols is provided in [11]. Accordingly, the transmission rates of the underlying network varies with respect to the communication standard chosen. The working of these planes is discussed below.

Data Plane: All the forwarding devices (FDs) such as OF switches, routers, and gateways are located in this plane. All these FDs behave according to the forwarding decisions made by the controller residing in the control plane. All such decisions are configured into flow tables of the FDs using a data-control plane interface. All the flow

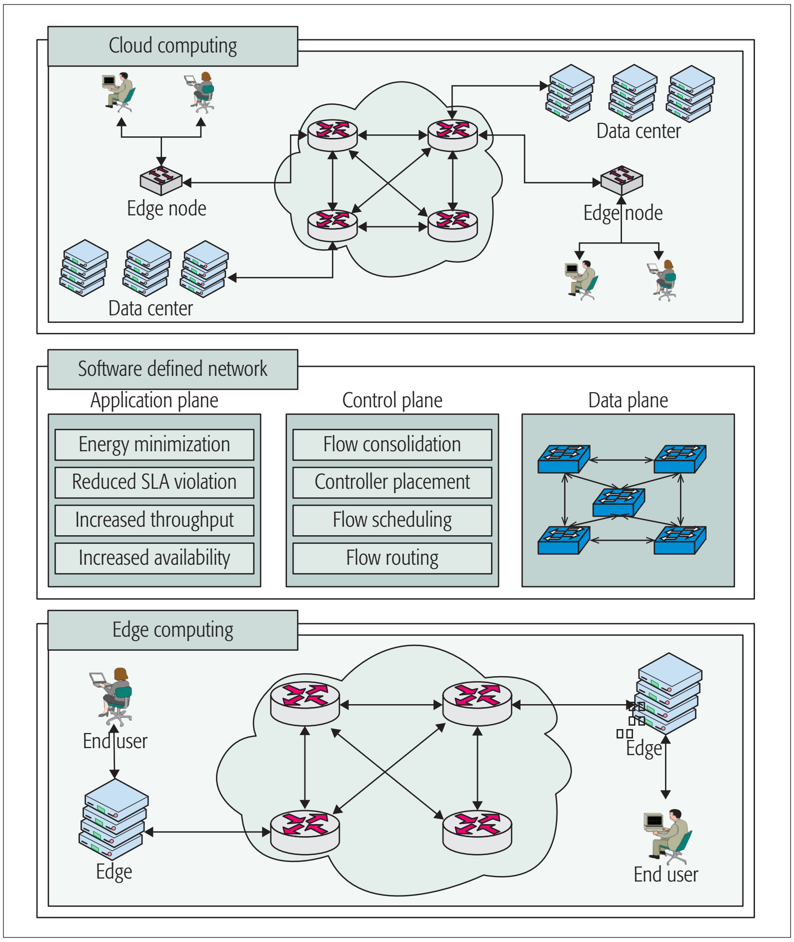

Figure 3. Different layers of edge-cloud interplay architecture.

tables work according to the instructions added in the instruction set available with the controller [9].

Control Plane: The control plane is the core of the SDN architecture and acts as a decision making plane. This plane works according to the control logic provided to the controller. All the forwarding decisions are decided by the controller and added to the instruction set in this plane. Moreover, the controller can also use the network operating system to create a virtual controller using a hypervisor. One of the most important characteristics of SDN is that the control logic can be programmed and reconfigured according to different environments [12]. Thus, utilizing this property of SDN in this work, a flow management scheme is presented that optimizes the trade-offs between energy efficiency and bandwidth, and energy efficiency and latency in the proposed edge-cloud interplay architecture.

In this edge-cloud interplay, the incoming traffic flow is divided into three categories and added to the respective queues, as shown in Fig. 4. Three categories of flows are:

1. Active
2. Waiting
3. Suspended

This is done in order to map a flow to a particular link. Initially, a flow path is matched for incoming traffic using flow tables. If a flow path exists for the incoming traffic, the flow is added to a waiting queue. The flow becomes active when it reaches the top of the queue. However, if no flow path exists for incoming traffic, the queue is suspended and sent back to the controller to reconfigure the flow paths [13].

Application Plane: All applications such as energy efficiency, throughput, SLA, and availability are provided to end users through this plane. Using this plane, the controller receives the feedback of applications provided to the end user.

HANDLING STREAMING APPLICATIONS DATA AT THE EDGE-CLOUD INTERPLAY

This section presents the proposed technique for energy-aware and QoS-guaranteed flow scheduling and routing in SDN. The entire methodology of the proposed scheme is represented using the sequence diagram in Fig. 5. As is evident from the figure, the overall process can be segregated into the following phases:

- Flow classification
- Selection of control logic by SDN's control plane
- Execution of the control logic by SDN to achieve energy-driven flow scheduling and routing

The overall middleware support in the considered setup is provided by SDN. Apart from this, the connectivity preservation can be effectively ensured by using data offloading and network selection presented in our previous work [11].

The detailed technical description of the above mentioned phases in presented as follows.

PHASE I: FLOW CLASSIFICATION

In the initial phase, flows are classified into two categories: batch processing and stream processing workflows. The former considers the quality bandwidth of network services more crucial; the

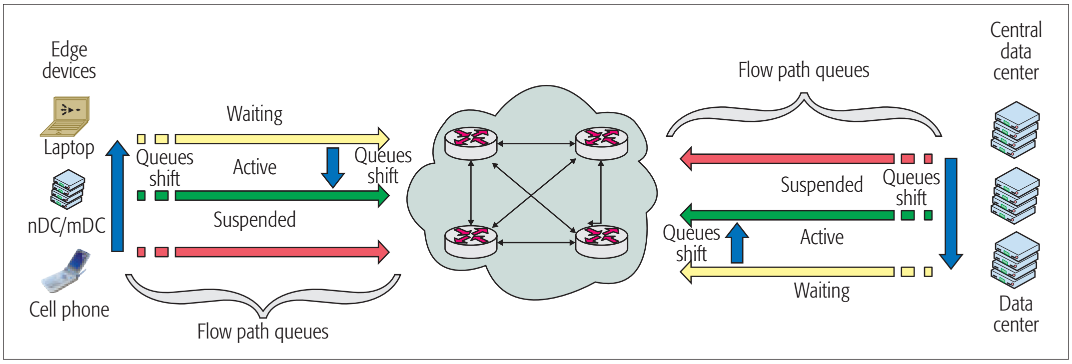

Figure 4. Flow management in edge-cloud interplay using SDN.

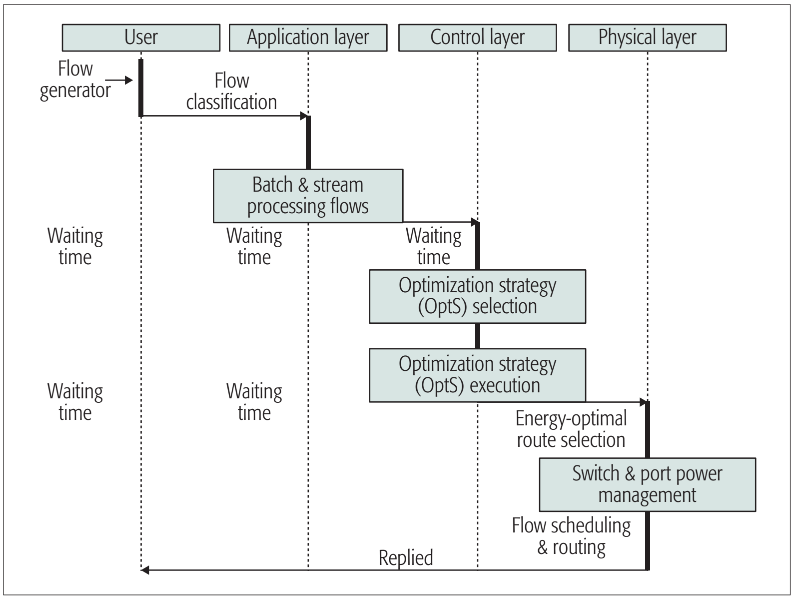

Figure 5. Sequence diagram of the proposed scheme.

latter comprises real-time applications, which are particularly latency-sensitive. Keeping this in view, AND the heterogeneity of the incoming and outgoing workflows the servers usually cater to, the proposed scheme tends to adapt according to the real-time workflows to avoid performance degradation issues. Additionally, the acquired data is also preprocessed and normalized for further processing, as done in our previous work [14].

The QoS parameters considered for both scenarios are widely separated. Therefore, considering a single QoS parameter for evaluation of the proposed scheme would lead to a compromise of the quality of experience (QoE) of the underlying services or applications. Hence, to bridge this gap, the article presents an adaptable control strategy to minimize energy consumption of data center networks (DCNs) using the SDN platform; they are specifically configured to cater to the QoS requirements of the varied set of workflows. To the best of the our knowledge, this is the first work that aims to amalgamate the energy minimization of DCNs based on SDN with adaptable QoS parameters.

PHASE II: SELECTION OF CONTROL LOGIC BY SDN'S CONTROL PLANE

In this phase, the control logic to be adopted by the control plane is selected based on the workflow classification executed in the previous phase. This is done to explicitly select the optimization strategy (OptS) to achieve the optimal trade-off between energy efficiency and QoS assurance. For instance, for batch processing workflows, OptS targets the bandwidth of network services as the QoS parameter. On the other hand, OptS for the stream processing workflows has latency as the desirable QoS parameter. This choice helps to adapt the network according to the real-time workflows and ensure increased SLA targets. It is evident that these resources are directly linked to the revenue of the utility providers and the end users. Hence, energy usage and QoS have been considered as the set of incentives in the proposed work.

PHASE-III: EXECUTION OF THE CONTROL LOGIC BY SDN TO ACHIEVE ENERGY-DRIVEN FLOW SCHEDULING AND ROUTING

The control logic execution as per the outcomes of the next phase is achieved by executing the OptS as follows.

OptS-I: Trade-off between Energy Efficiency and Bandwidth: The considered OptS takes into account the trade-off between energy efficiency and bandwidth of network services as per the following multi-objective optimization function ($\mathbb{F}_1(x)$), wherein the parameter $\delta_{k,i}(t)$ is the binary decision variable under consideration. It represents the optimal path selected by the $\mathbb{F}_1(x)$ for the kth flow via the ith switch of the SDN's data plane.

$$\min \mathbb{F}_1(x) = f(\mathbb{E}(\delta_{k,i}(t)), -\mathbb{B}(\delta_{k,i}(t))) \qquad (1)$$

$$\mathbf{C1}: \mathbb{E}(\delta_{k,v}(t)) = \Bigg(F \times \sum_k (t_k^{end} - t_k^{start}) \times \delta_{k,vi}(t)\Bigg)$$

$$+\Bigg(D \times \sum_k (t_k^{end} - t_k^{start} \times \delta_{k,vi}(t) \times \delta_{k,vl}(t) \times a_{i,l})\Bigg)$$

$$\mathbf{C2}: \mathbb{B}(\delta_{k,v}(t)) = \Bigg(\sum_{i \in w} \sum_{j \in |P(i)|} \sum_k B_{i,j} \times \Big(t_k^{end} - t_k^{start}$$

$$\times \delta_{k,vi}(t) \times \delta_{k,vl}(t) \times a_{i,l}\Big)$$

$$\mathbf{C3}: F_{vi,vl} = \{ {}_j | \delta_{k,vi}(t) = 1 \,\&\, \delta_{k,vl}(t) = 1; \forall v_i, v_l \in w \}$$

$$\mathbf{C4}: \delta_{k,v}(t) \in 0,1; \forall k, v, t$$

In the above equation, function $\mathbb{E}(\delta_{k,i}(t))$ represents the energy computation of the DCN (**C1**), which can be attributed to the total energy consumption of the active switches. It is worth mentioning here that the energy consumption of the individual switches is not proportional to the traffic load but on the dynamic part of the switches. Ideally, the total energy utilized by a switch (EnUti) can be attributed to its fixed part and dynamic part. Here, the former represents the total energy utilized to power the fans, chasis, and so on. On the other hand, the dynamic part is proportional to the number of active ports on the switch. Mathematically, it can be represented as follows: EnUti(Switch) = fixed part + dynamic part. The parameters F and D denote the fixed and dynamic energy consumption values of the DCN switches. Moreover, in the considered setup, a total of w switches have been considered, which are indexed using i and represented using V_i. The adjacency between the two switches, say V_i and V_l, is denoted using the binary variable $a_{i,l}$. Moreover, the index j is used to denote the port of a switch, and every switch is assumed to be equipped with $|P(i)|$ ports. Additionally, k flows are assumed to be available at timestamp t, wherein t_k^{start} and t_k^{end} refers to the start and end of the flow on the DCN.

Constraint C2 denotes the computation of DCN's bandwidth ($\mathbb{B}(\delta_{k,i}(t))$) for the chosen route at time t. Here, the variable $B_{i,j}$ denotes the bandwidth of the jth port of the ith switch. The next constraint, C3, depicts the flow restriction in the considered setup, wherein flow via the two switches ($F_{vi,vl}$) always selects a single route. Finally, constraint C4 enforces the integrality restriction of the decision variable $\delta_{k,i}(t)$.

OptS-II: Trade-off between Energy Efficiency and Latency: In contrast to the above mentioned OptS , the present OptS tends to achieve the optimal trade-off between energy efficiency and network latency for stream processing flows. It is achieved in accordance with the below mentioned objective function (i.e., $F_2(x)$).

$$\min \mathbb{F}_2(x) = f(\mathbb{E}(\delta_{k,i}(t)), \mathbb{L}(\delta_{k,i}(t))) \tag{2}$$

$$\mathbf{C1:} \mathbb{E}(\delta_{k,v}(t)) = \left(F \times \sum_k (t_k^{end} - t_k^{start}) \times \delta_{k,vi}(t) \right)$$

$$+ \left(D \times \sum_k \left(t_k^{end} - t_k^{start} \right) \times \delta_{k,vi}(t) \times \delta_{k,vl}(t) \times a_{l,k} \right)$$

$$\mathbf{C2:} \mathbb{L}(\delta_{k,v}(t)) = \left(\sum_j \frac{d_{vi,vl} \times \delta_{k,vi}(t) \times \delta_{k,vl}(t) \times a_{i,l}}{\mathbb{P}r(t)} \right)$$

$$+ \left(\sum_{i \in w} \sum_{j \in |P(i)|} \frac{\mathcal{P}_{i,j}(t)}{B_{i,j} \times O_{i,j}(t)} \right)$$

$$+ \left(\sum_{i \in w} \sum_{j \in |P(i)|} \frac{|\mathbb{Q}_{ready}(t)|}{B_{i,j} \times O_{i,j}(t)} \right)$$

$$+ \left(\sum_{i \in w} \mathbb{P}_i(t) \times \sum_k \left(t_k^{end} - t_k^{start} \right) \times \delta_{k,vi}(t) \right)$$

$$\mathbf{C3:} F_{vi,vl} = \left\{ f_j \middle| \delta_{k,vi}(t) = 1 \,\&\, \delta_{k,vl}(t) = 1; \forall v_i, v_l \in w \right\}$$

$$\mathbf{C4:} O_{i,j} = c_1 \times \delta_{k,v}(t)$$

$$\mathbf{C5:} \delta_{k,v}(t) \in 0,1; \forall k, v, t$$

In the above equation, $\mathbb{L}(\delta_{k,i}(t))$ denotes the latency computation function. Constraint C2 highlights the detailed computational process involved in the estimation of $\mathbb{L}(\delta_{k,i}(t))$. The first part of C2 depicts the propagation delay. Here, the variable $d_{vi,vl}$ depicts the distance between the switches, while $\mathbb{P}r(t)$ represents the medium propagation delay. In addition to this, the second part of C2 denotes the transmission delay wherein packet size and occupation ratio are denoted using $\mathcal{P}_{i,j}(t)$, and $\mathcal{O}_{i,j}(t)$, respectively. The third part of C2 represents the queuing delay wherein the number of flows in the ready queue are denoted using $|\mathbb{Q}_{ready}(t)|$. Finally, the last part of C2 refers to the processing delay of the node, and $\mathbb{P}_i(t)$ refers to the processing delay induced by the ith node. The rest of the constraints follow the same notation as discussed.

It is worth noting that the proposed scheme handles the spatial-temporal characteristics of data in the IIoT environment. It is evident from the use of the parameters t and $d_{vi,vl}$. The former represents the timestamp across which the flows are scheduled and routed. On the contrary, $d_{vi,vl}$ plays an essential role in latency computation. Constraint C2 defined above signifies that greater distances between the switches lead to higher network latencies and vice versa.

MOEA/D and Tchebycheff Decomposition: The OptS-*I* problems defined above are solved using a multiobjective evolutionary algorithm based on decomposition (MOEA/D). It decomposes the considered multi-objective problem into multiple scalar optimization subproblems (SPs). The obtained SPs are then solved simultaneously using the population of solutions, which tend to evolve in every generation phase. In every generation, the population consists of the best solution corresponding to each SP. In addition to this, the decomposition method chosen in the present scheme is *Tchebycheff decomposition* due to its inheritant advantages. It is a widely accepted technique for constructing aggregation functions and deals with both convex and non-convex problems. Additionally, it is known for enhanced convergence performance and search region.

Mathematically, the objective function of a particular SP, say the lth SP, in accordance with *Tchebycheff decomposition* is defined as per the following equation:

$$g^{te}\left(x \middle| \lambda^l, z^\star \right) = \max_{1 \le i \le m} \left\{ \lambda_i^l \left| fi(x) - z_i^\star \right| \right\} \tag{3}$$

Here, $\lambda^1, \lambda^2, \ldots, \lambda^N$ are evenly spread out weight vectors, and the value of λ^j is equivalent to $(\lambda_1^j, \lambda_2^j, \ldots, \lambda_m^j)^T$. The parameter z^* considered is the reference point.

OBSERVATION AND ANALYSIS

SIMULATION SETUP

This section investigates the impact of the proposed SDN-based scheme for flow scheduling and flow routing on DCNs' overall energy utilization while keeping the SLA in a reasonable boundary. It is implemented using i3-6100U CPU @ 2.30 GHz with 4 GB of RAM on MATLAB R2016a.

In order to validate the efficacy of the proposed scheme (scheme I), it has been compared with two existing schemes (schemes II and III). The existing schemes do not take the workflow classification into account. The QoS adaptivity feature is also not considered in these schemes. Scheme II is an extension of OptS -*I* and suggests the energy plus bandwidth-aware flow scheduling and routing in the considered cloud-edge setup amalgamated with SDN capabilities. Along similar lines, scheme III is an extension of OptS-II presented in the previous section. Hence, it takes into account the latency-sensitive and energy-aware route selection in SDN setup.

For the purpose of evaluating the proposed scheme, a 4-ary Fat-Tree network topology comprising of 4-port switches was taken into account. The simulation tests were conducted 25 times across a 24-hour simulation period. The considered simulation parameters with respect to energy consumption of network switches is as under. The fixed power of the active switches was set to 36 W, while 1 W of power utilization was taken as the standard value for an active port. Additionally, to evaluate the performance of the three schemes based on the above simulation setup, workflow traces acquired from a Google Compute cell were used [15]. The performance evaluations of the considered scheme was carried out in accordance with the following set of parameters.

Average number of SLA violations: The SLA metric is an important metric in the considered setup, wherein higher SLA violations depict poor QoS, and reduced SLA violations lead to satisfactory QoS. As the QoS in the considered setup is

The QoS parameters considered for both scenarios are widely separated. Therefore, considering a single QoS parameter for evaluation of the proposed scheme would lead to a compromise of the Quality-of-Experience (QoE) of the underlying services or applications.

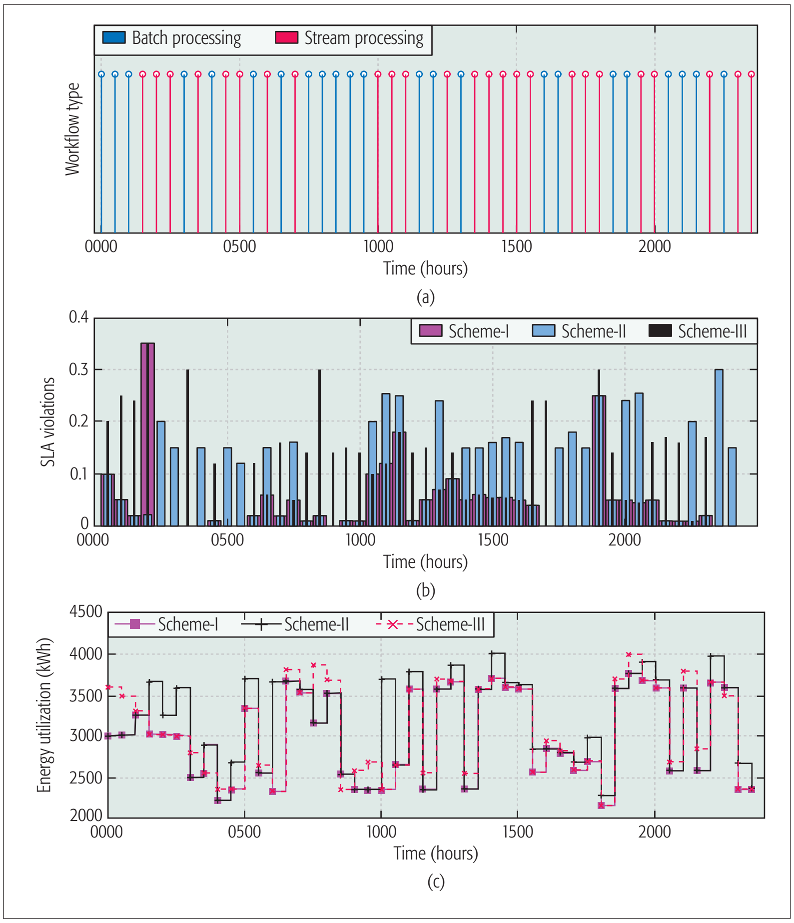

Figure 6. Comparison between the proposed and existing schemes with respect to different performance metrics: a) workflow classification among batch processing and stream processing; b) average number of SLA violations observed during flow scheduling and routing; c) total energy consumption of DCNs with respect to time.

centered around network latency and bandwidth, the SLA violations are defined accordingly. SLA is considered to be violated under two scenarios:

- If the SDN setup fails to process the flows within the threshold level of latency, which is dependent on the flow size and type
- If the SDN setup fails to allocate the required bandwidth of the considered flow.

Total energy consumption (kWh): The cumulative energy utilization in the considered setup refers to the sum of energy utilized by the network switches that form the integral part of the underlying DCN.

RESULTS

The proposed scheme aims to adapt to real-time traffic workflows by considering their flow scheduling and routing under different QoS parameters. Hence, the proposed scheme classifies the workflows into two broad categories based on their size and other characteristics. The related classified workflows across batch and stream processing is depicted in Fig. 6a. As evident from the figure, the considered simulated environment takes into account random arrival of different workflows after every 30-minute time interval. For instance, at 0500 h the workflow under reference was of stream processing type. Hence, the SDN control plane quickly adapted to OptS-II to produce energy optimal routes for data transfer while guaranteeing the required bandwidth.

The related results on the bandwidth and latency assurance for the two types of workflows have been represented in terms of SLA violations. The related results have been highlighted using Fig. 6b, wherein higher violations across schemes II and III are clearly evident. Overall, the proposed scheme is found to operate with the least number of SLA violations across varied workflow types. For example, at 0500 h the SLA violations of schemes I and III were almost negligible, while those of scheme II led to SLA violations of almost 12 percent.

The proposed scheme takes into account adaptable QoS selection based on workflow classification. For instance, it adapts to OptS *-I* in the case of batch processing workflows and to OptS-II for processing stream processing workflows. This classification and subsequent adaptation makes the proposed scheme suitable in the context of energy-aware flow scheduling and routing. Hence, the related energy-centric results obtained using the three schemes is depicted using Fig. 6c. As evident from the figure, scheme I is found to be the most energy-efficient in comparison to the other schemes. For instance, schemes II and III lead to overall energy utilization of 3695 and 3321 kWh at 0500 h. The proposed scheme, on the other hand, achieved the most energy-efficient results of 3321 kWh.

In summary, the comparative evaluations of the three schemes can be described as below. The proposed scheme achieves an overall improvement of 1.4 and 1.6 percent relative to scheme II and scheme III, respectively, in terms of SLA violations. Additionally, the proposed scheme also depicts an overall improvement of 6 and 3.5 percent over the other schemes with respect to overall energy utilization of the DCN. These improvements can be attributed to the adaptive decision making capability of the middleware supported by SDN in accordance with different workflow types.

CONCLUSION

In this article, an SDN-based edge-cloud interplay is presented to deal with flow scheduling among edge and cloud devices. In this regard, a multi-objective evolutionary algorithm using Tchebycheff decomposition is designed for flow scheduling in SDN. The proposed scheme is evaluated with respect to two optimization problems:

1. Trade-off between energy efficiency and latency
2. Trade-off between energy efficiency and bandwidth

In order to validate the efficacy of the proposed scheme, it is compared to existing schemes using different evaluation parameters. The results obtained prove the effectiveness of the proposed flow scheduling scheme in the IIoT environment.

ACKNOWLEDGMENTS

This work is sponsored by the research project from the Council of Scientific and Industrial Research (CSIR), New Delhi with reference no. 22/717/16/EMR-II, by FCT — Fundação para a Ciência e a Tecnologia through the UID/

EEA/500008/2013 Project, the Government of the Russian Federation, Grant 074-U01, and Finep, with resources from Funttel, Grant No. 01.14.0231.00, under the *Centro de Referência em Radiocomunicações* — CRR project of the *Instituto Nacional de Telecomunicações* (Inatel), Brazil.

References

[1] K. Kaur *et al.*, "Container-as-a-Service at the Edge: Trade-off between Energy Efficiency and Service Availability at Fog Nano Data Centers," *IEEE Wireless Commun.*, vol. 24, no. 3, June 2017, pp. 48–56.

[2] B. Daly, Mar. 2016, "After a Decade of Dominance, What's Next for AWS?" 451 Research; https://blogs.the451group.com/techdeals/infrastructure-software/after-a-decade-of-dominance-whats-next-for-aws/, accessed May. 2017.

[3] D. Evans, "The Internet of Things: How the Next Evolution of the Internet is Changing Everything," Cisco Internet Business Solutions Group, Apr. 2011, accessed May 2017.

[4] A. Dua, N. Kumar, and S. Bawa, "A Systematic Review on Routing Protocols for Vehicular Ad Hoc Networks," *Vehicular Commun.*, vol. 1, no. 1, 2014, pp. 33–52.

[5] S. Wang *et al.*, "Cachinmobile: An Energy-Efficient Users Caching Scheme for Fog Computing," *IEEE/CIC Int'l. Conf. Commun. in China*, July 2016, pp. 1–6.

[6] P. Borylo *et al.*, "Energy-Aware Fog and Cloud Interplay Supported by Wide Area Software Defined Networking," *IEEE ICC*, May 2016, pp. 1–7.

[7] F. Jalali *et al.*, "Fog Computing May Help to Save Energy in Cloud Computing," *IEEE JSAC*, vol. 34, no. 5, May 2016, pp. 1728–39.

[8] R. Deng *et al.*, "Optimal Workload Allocation in Fog-Cloud Computing toward Balanced Delay and Power Consumption," *IEEE Internet of Things J.*, vol. 3, no. 6, 2016, pp. 1171–81.

[9] H. Huang *et al.*, "Green Datapath for TCAMbased Software-Defined Networks," *IEEE Commun. Mag.*, vol. 54, no. 11, Nov 2016, pp. 194–201.

[10] G. S. Aujla *et al.*, "SDN-Based Data Center Energy Management System Using RES and Electric Vehicles," *IEEE GLOBECOM*, Dec 2016, pp. 1–6.

[11] G. S. Aujla *et al.*, "Data Offloading in 5G-Enabled Software-Defined Vehicular Networks: A Stackelberg-Game-Based Approach," *IEEE Commun. Mag.*, vol. 55, no. 8, Aug. 2017, pp. 100–08.

[12] G. Xu *et al.*, "Bandwidth-Aware Energy Efficient Routing with SDN in Data Center Networks," *IEEE 17th Int'l. Conf. High Performance Computing and Commun.*, 2015, Aug 2015, pp. 766–71.

[13] G. Xu *et al.*, "Bandwidth-Aware Energy Efficient Flow Scheduling with SDN in Data Center Networks," *Future Generation Computer Systems*, vol. 68, 2017, pp. 163–74.

[14] A. Jindal *et al.*, "Decision Tree and SVM-Based Data Analytics for Theft Detection in Smart Grid," *IEEE Trans. Industrial Informatics*, vol. 12, no. 3, 2016, pp. 1005–16.

[15] J. Wilkes, "More Google Cluster Data," Google Research Blog, Nov. 2011; http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html, accessed Apr. 2017.

Biographies

KULJEET KAUR [S'14] (kuljeet0389@gmail.com) received her B. Tech. degree in computer science and M.E. degree in information security. She is currently working towards a Ph.D. degree with the Department of Computer Science and Engineering, Thapar University, Patiala, India. Her main research interests include radio frequency identification, cloud computing, and vehicle-to-grid.

SAHIL GARG [S'16] (garg.sahil1990@gmail.com) received his B.Tech degree from Maharishi Markandeshwar University, Mullana, Ambala, India, in 2012, and his M.Tech degree from Punjab Technical University, Jalandhar, India, in 2014, both in computer science and engineering. He is currently working toward a Ph.D. degree in computer science and engineering from Thapar University. His research interests include machine learning, big data analytics, knowledge discovery, game theory and vehicular ad hoc networks.

GAGANGEET SINGH AUJLA [S'15] (gagi_aujla82@yahoo.com) is pursuing a Ph.D.from Thapar University. He received his B.Tech degree in computer science and engineering from Punjab Technical University in 2003 and HIS M.Tech degree from the same university in 2013. He has many research contributions in the area of smart grid, cloud computing, vehicular ad hoc networks, and software-defined networks.

NEERAJ KUMAR [M'16, SM'17] (neeraj.kumar@thapar.edu) is working as an associate professor in the Department of Computer Science and Engineering, Thapar University. He received his M.Tech. from Kurukshetra University, Kurukshetra (Haryana) followed by his Ph.D. from SMVD University, Katra (J&K). He was a postdoctoral research fellow at Coventry University, United Kingdom. He has more than 200 research papers in leading journals and conferences of repute. He is an Associate Technical Editor of *IEEE Communications Magazine*, an Associate Editor of *IJCS*, Wiley, *JNCA*, Elsevier, and *Security & Communication*, Wiley.

JOEL J. P. C. RODRIGUES [S'01, M'06, SM'06] (joeljr@ieee.org) is a professor and senior researcher at the National Institute of Telecommunications (Inatel), Brazil, and a senior researcher at the Instituto de Telecomunicações, Portugal. He is leader of the Internet of Things research group (CNPq), a member of the IEEE ComSoc Board of Governors as Director for Conference Development, an IEEE Distinguished Lecturer, Past Chair of IEEE ComSoc Technical Committees on eHealth and Communications Software, and a Steering Committee member of the IEEE Life Sciences Technical Community. He is Editor-in-Chief of three international journals, and a co-author of over 550 papers, three books, and three patents.

MOHSEN GUIZANI [S'85, M'89, SM'99, F'09] (mguizani@ieee.org) received his B.S, M.S., and Ph.D. from Syracuse University. He is currently a professor and ECE Department Chair at the University of Idaho. His research interests include wireless communications/mobile cloud computing, computer networks, security, and smart grid. He is the author of nine books and more 450 publications. He was Chair of the IEEE Communications Society Wireless Technical Committee. He served as an IEEE Computer Society Distinguished Speaker.

The proposed scheme aims to adapt to real-time traffic workflows by considering their flow scheduling and routing under different QoS parameters. Hence, the proposed scheme classifies the workflows into two broad categories based on their size and other characteristics.