SDN-Enabled Energy-Efficient Routing Optimization Framework for Industrial Internet of Things

Faisal Naeem, *Member, IEEE*, Muhammad Tariq, *Senior Member, IEEE*, and H. Vincent Poor, *Life Fellow, IEEE*

***Abstract*—The traditional Internet architecture relies on the *best-effort* principle, which is not suitable for critical industrial Internet of Things (IIoT) applications such as healthcare systems with stringent quality-of-service (QoS) requirements. In this article, a software-defined network (SDN) based on an analytical parallel routing framework is proposed by using the massive processing power of a graphics processing unit (GPU) for dynamically optimizing multiconstrained QoS parameters in the IIoT. The framework considers three types of QoS applications for smart healthcare traffic: loss-sensitive, delay-sensitive, and jitter-sensitive. A QoS-enabled routing optimization problem is formulated as a max-flow min-cost problem, while a greedy heuristic that dispatches the path calculation task concurrently to the GPU for calculating optimal forwarding paths considering the QoS requirement of each flow is proposed. The results show that the proposed scheme efficiently utilizes the limited *bandwidth cost* in terms of energy and bandwidth while satisfying the QoS requirement of each flow with maximizing the network resources for future IIoT traffic flows. Comparative analysis of simulation results with shortest path delay, Lagrangian relaxation-based aggregated cost, and Sway schemes indicate a reduced violation in the service-level agreement by 17%, 19%, and 4%, respectively, by using the AttMpls topology, while it is 48%, 44%, and 7% when the Goodnet topology is used. Moreover, SEQOS is seen to be energy efficient and eight times faster than the benchmark algorithms in large IIoT networks.**

***Index Terms*—Energy efficient routing, industrial Internet of things (IIoT), NP-hard, software-defined networking.**

Manuscript received February 14, 2020; revised April 29, 2020 and June 3, 2020; accepted June 24, 2020. Date of publication July 3, 2020; date of current version May 3, 2021. This work was supported in part by the U.S. National Science Foundation under Grant CCF-0939370. Paper no. TII-20-0731. *(Corresponding author: Muhammad Tariq.)*

Faisal Naeem and Muhammad Tariq are with the Department of Electrical Engineering, National University of Computer and Emerging Sciences, Peshawar 25000, Pakistan (e-mail: faisal.naeem@nu.edu.pk; mtariq@princeton.edu).

H. Vincent Poor is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: poor@princeton.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2020.3006885

I. Introduction

THE INTERNET of Things (IoT) is a networking paradigm, in which billions of objects are connected by using the best-effort architecture [1]. Industrial Internet of Things (IIoT)-based healthcare monitoring is an IoT-based healthcare framework that connects patients to healthcare facilities by using communication networks. For example, heart rate, electrocardiography, diabetes, and vital body signs can be remotely monitored without the physical presence of patients [2].

An IIoT network generates a large volume of traffic from various devices that need new communication protocols to manage traffic [3]. However, the *best-effort* Internet architecture cannot handle the real-time IIoT traffic generated from billions of interconnected devices owing to their varied application-dependent requirements and limited bandwidth [4]. As a result, congestion may occur when using the existing architecture, which degrades the network performance. Because of this network congestion, the real-time IIoT analytical tasks for healthcare systems considering the quality of service (QoS) cannot be guaranteed by using the best-effort architecture [4].

IIoT applications in smart healthcare systems have varying QoS requirements in terms of delay, packet loss, jitter, and throughput [5]. About 90% of the traffic by 2019 was generated from multimedia streaming, which is expected to be increased by the medical applications' streaming [6]. The streaming of healthcare applications has become an essential part of life. The medical applications, such as interactive multimedia streaming, traffic generated from faulty sensors, and vital signs, can tolerate packet loss but have stringent delay requirements. In contrast, video streaming applications cannot tolerate jitter. Similarly, low-power devices are sensitive to packet loss, while the periodic physiological traffic of medical traffic can tolerate delay and jitter but not packet loss [7]. Hence, support is needed according to the nature of an application for minimizing the packet loss for resource-constrained devices. Thus, traffic flows in the IIoT based on a smart healthcare system are classified as *jitter-sensitive*, *delay-sensitive*, and *loss-sensitive*. The varied nature of the traffic requirements for different applications needs a communication paradigm to maintain QoS [8].

Software-defined industrial Internet of Things (SDIIoT) [9] is a communication architecture, in which logic from the

devices is shifted to a centralized control plane to simplify the management of the IIoT. Research studies [10], [11] have demonstrated that the proposed innovative SDIIoT architecture can significantly reduce the implementation complexity, provide end-to-end QoS guarantees, and realize fine-granularity resource allocation. However, with the rapid increase in the traffic flows in next-generation networks, the centralized software-defined network (SDN) controller needs to complete the online routing optimization process or path calculation in a short time [12]. To solve scalability issues in the SDN, parallel and efficient optimization techniques that can be integrated with the SDN to speed up the heuristics for real-time IIoT traffic are explored.

The effectiveness of an online routing optimization depends upon the calculation speed for finding the optimal paths in a communication network for a traffic engineering problem [13]. Several integer linear program (ILP) problems have been formulated for this purpose, but their calculation time for searching for optimal paths is too slow. In contrast, several heuristics [14]–[16] have been proposed for finding optimal paths in a reasonable time. Nevertheless, with the advent of IoT-based real-time environments, still higher optimization speeds are needed for satisfying the QoS requirement for different applications.

The existing optimization-based techniques [14]–[17] are known as sequential algorithms that are designed to be simulated in the CPU environment. The massive increase of traffic with a future IoT-based network needs fast heuristics to speed up the computation tasks, as sequential algorithms consume significant amounts of CPU time to search for optimal paths [13], [18]. Different from conventional optimization-based approaches, a novel and practical context-aware learning-based computation task offloading approach was proposed for edge-computing-empowered IIoT in [19], which can provide guaranteed performance with a bounded deviation based on only local and causal information. It creatively incorporates context awareness into the learning process to reduce the learning regret, which has opened up a new direction for learning-based context-aware optimization. The existing techniques cannot have satisfactory performance in reducing service-level agreement (SLA) violations with the arrival of a massive volume of traffic in a short time at the SDN controller for the delay-sensitive and real-time healthcare applications. Moreover, the existing schemes are not energy efficient and cannot utilize the limited available bandwidth for the IIoT traffic flows expected in future applications.

In this study, a QoS-enabled efficient parallel routing optimization scheme, named SEQOS, is proposed for an IIoT-based smart healthcare system using the SDN network paradigm that considers the diverse QoS requirements for medical applications. The proposed approach is different from the benchmark schemes in several aspects. First, the centralized programmatic functionality of the SDN is utilized to forward IIoT-based traffic for smart healthcare systems by considering application-dependent requirements, such as *jitter-sensitive*, *loss-sensitive*, and *delay-sensitive*, or combinations of any QoS traffic types. Second, to efficiently utilize the limited bandwidth and energy efficiency of resource-constrained devices, the *bandwidth distance* (BD) and *bandwidth energy* (BE) capacities are taken into account to forward the IIoT traffic to the destination while selecting the paths that can efficiently utilize the bandwidth and energy, by satisfying the QoS requirement to maximize the *bandwidth cost* (BC) capacity for IIoT traffic flows. Finally, an efficient parallel routing optimization technique, which runs on a graphics processing unit (GPU) with massively parallel computation capability, is proposed to speed up the computational speed of the proposed heuristic.

To the best of authors' knowledge, research related to multiconstrained QoS parameters using energy and bandwidth-efficient parallel routing optimization technique in IIoT-based smart healthcare networks has not been explored significantly. The proposed work outperforms prior work when taking into account the different QoS parameters, computational speed, and rule-capacity constraints.

The key contributions of this article are summarized as follows.

1) We propose an energy-efficient and QoS-aware parallel routing optimization algorithm for SDIIoT based on healthcare systems that decouple the path calculation task concurrently on the GPU to speed up the path calculation task.
2) We formulate the multiconstrained QoS-aware optimization problem as a max-flow min-cost problem from the perspective of the SDN in an IIoT-based smart healthcare network, while selecting optimal routes for maximizing the BC capacity in terms of bandwidth and energy by satisfying the QoS requirement for IIoT traffic flows.
3) We propose a scalable routing scheme for a large IIoT network that is eight times faster than the existing schemes.

The rest of this article is organized as follows. Section II describes the network architecture, while Section III presents the formulation of the problem. Section IV presents the proposed online parallel routing optimization algorithm. Section V provides the simulation results. Finally, Section VI concludes this article.

II. Network Architecture

The proposed end-to-end system architecture for the IIoT based on a smart healthcare system is provided in Fig. 1. The model architecture consists of patients at a hospital, home, or office, in which sensors are worn so that they can generate the traffic related to health parameters, such as Smart e-Health gateways acting as an SDN switch or forwarding device, SDN controller responsible for calculating the optimized routes satisfying the QoS constraints, and servers or cloud from where the patient medical history can be viewed. The wireless body area network (WBAN) connects various IIoT-based medical devices to collect information related to health using the heterogeneous networks that are connected to the Internet using the SDN gateway. The components of the system model are as follows.

The sensors are worn on the human to transmit health-related parameters such as blood pressure, sugar level, electrocardiography, and heart rate. The SDN switch acts as a Smart e-Health gateway for forwarding the traffic to the Internet. The personal

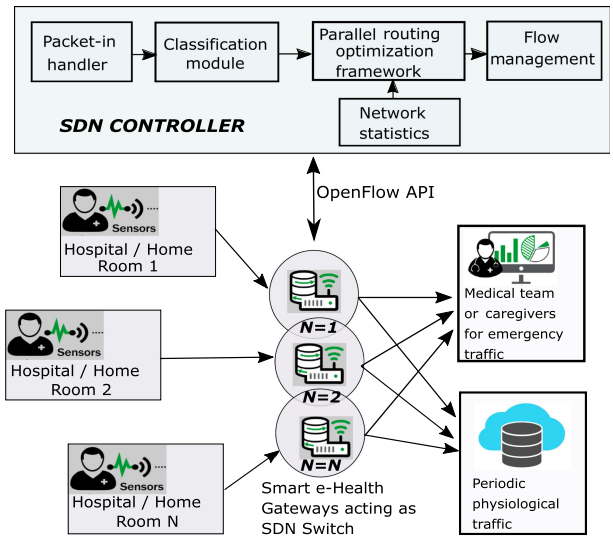


Fig. 1. SDN-based QoS optimized architecture for the smart healthcare system.

digital assistant (PDA) can be best configured as an SDN switch that can perform the functionality of linking IIoT-based WBAN sensors to external networks using 4G/5G cellular networks if a patient is on the move or using Wi-Fi routers while the patient is at home/office. The PDA in the proposed architecture also performs the functionality of protocol conversion for heterogeneous networks such as IPv4–IPv6 and conversion of WBAN protocols to the next-hop network device protocol. The SDN controller can be configured in any external location to perform intelligent functionalities such as routing, scheduling, and flow control. The functionality of the modules of the SDN controller in the architecture is based on the following functionalities. The traffic flows having QoS requirements are forwarded from the patients to an SDN controller using a packet-in message. The classification module implemented in the controller classifies the flow into delay-sensitive, loss-sensitive, or jitter-sensitive flows. The classified flow is forwarded to the parallel routing optimization module, which calculates the shortest paths concurrently on the GPU. The optimal paths assuring the QoS requirement of the flows with efficiently utilizing the bandwidth and energy are forwarded to the SDN switch using a flow management module.

The proposed architecture is considered to be an SDN-enabled simple directed graph, i.e., $N = (S, E)$, where S denotes the SDN switches and E indicates the edges in the network. The edges $E = (i, j) \in S, i \neq j$, represent the set of all the edges that belong to the set of SDN switches in an SDN network, where the source and destination in the graph should be unique. The SDN controller uses an OpenFlow (OF) API communication protocol to communicate with the SDN switches. The northbound interface is used to communicate between the controller and the application plane. Each application has varying QoS requirements depending upon the type of the traffic of IIoT based on the smart healthcare system. The QoS requirements of the traffic can be defined using the application plane.

III. Problem Formulation

The QoS routing problem is formulated as max-flow min-cost in the software-defined industrial internet of health things (SDIIoT). The objective of the QoS routing is to maximize the flows while minimizing the BD and BE costs considering the delay, packet loss, and jitter SLA requirement. Furthermore, the switch rule-capacity constraint must also be satisfied. The set of flows for the given QoS routing problem is defined as

$$F_L = \{f_m | m \in N\}, \quad f_m := \{x_m, z_m, q_m, t_m\} \tag{1}$$

where N represents the neighborhood of a network with a set of OF switch $i \in S$ that is adjacent or directly connected to an SDN switch. The individual flow f_m represents the source x_m, destination z_m, QoS demand q_m, and type of traffic t_m (either ls, js, and ds) for the mth flow f_m. An identity function $\alpha_m = (i, j)$ is defined to show whether a flow f_m can be forwarded on edge $(i, j) \in E$

$$\alpha_m(i, j) = \begin{cases} 1, & \text{if } f_m \text{ is routed on } (i, j) \in E \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

$N(i)$ represents the neighborhood of an SDN switch $i \in S$. A flow f_m can be forwarded on an edge (i, j), only if flow rule is defined at OF switch i to any other OF switch $j \in N(i), i \neq j$. Thus, the maximum number of flow tables at OF switch $i \in S$ associated with flow f_m can be expressed as $R_m(i) = \sum_{j \in N(i)} \alpha_m(i, j)$. Thus, the number of flow rules defined at specific SDN switch can be defined as

$$R_i = \sum_{f_m \in F_L} R_m(i). \tag{3}$$

The total delay, jitter, and packet loss associated with a flow f_m can defined as

$$f_m(d) = \sum_{(i,j) \in E} d(i, j)\alpha_m(i, j) \tag{4}$$

$$f_m(j) = \sum_{(i,j) \in E} j(i, j)\alpha_m(i, j) \tag{5}$$

$$f_m(l) = \sum_{(i,j) \in E} pl(i, j)\alpha_m(i, j). \tag{6}$$

The capacity of a path from the bandwidth consumption rule can be mathematically defined as

$$C(f_m) = \min_{(i,j) \in E} c(i, j)\alpha_m(i, j). \tag{7}$$

where $Cr(i, j)$ is the remaining bandwidth routed through the edge, $(i, j) \in E$ after all flows $f_m \in F_L$ with its associated specifications. Mathematically, it can be defined as

$$C_r(i, j) = c(i, j) - \sum_{f_m \in E} q_k^{\text{bandwidth}} \alpha_m(i, j). \tag{8}$$

A. QoS-Aware Max-Flow Min-Cost Algorithm

The objective of using the optimization problem function is to maximize the flows while minimizing the cost in terms of energy and bandwidth associated with each edge satisfying the SLA requirement for the healthcare applications. The SDN-enabled

graph $N(S, E)$ have limited network resources known as BC efficiency defined as

$$\text{BC} = \sum_{i=1}^{K} \sum_{i=1}^{K} C(i,j).\beta(i,j). \tag{9}$$

Given a traffic flow (x_m, z_m), the optimal routing path needs to minimize the BC capacity with selecting the optimal paths that efficiently utilize the energy and bandwidth resources of the network for future IIoT traffic flows. The BD and BE capacities are considered as constraints for minimizing the BC capacity.

BC utilization by a commodity f_m over an edge E with a rate $C(i,j)$ is defined as

$$\text{BC}(E, C(i,j)) = \sum_{(i,j)\in E} C(i,j).\beta(x_{(i,j)}, z_{(i,j)}) \tag{10}$$

where $x_{(i,j)}$ and $z_{(i,j)}$ represent the source and destination of the edge E, respectively.

The problem of satisfying the SLA requirement of the healthcare system is formulated as the max-flow min-cost problem by

$$\max_{f_m} \min_{(i,j)} \sum_{f_m \in F_L} \sum_{(i,j)\in E} f_m \beta(i,j)\alpha(i,j) \quad \forall (i,j) \in E \tag{11}$$

s.t.

$$\sum_{(i,j)\in(i,j)_{\text{out(i)}}} |f_{m_{i,j}}| = \sum_{(i,j)\in(i,j)_{in(i)}} |f_{m_{i,j}}| \quad \forall i,j \in E \setminus \{x_m, z_t\} \tag{12}$$

and

$$\sum_{(i,j)\in(i,j)_{\text{out}(x_m)}} |f_{m_{i,j}}| = \sum_{(i,j)\in(i,j)_{in(z_t)}} |f_{m_{i,j}}| = |F_L| \tag{13}$$

$$f^{q_m}[f_m(d), f_m(j), f_m(l)] \leq [q_m^{\text{delay}}, q_m^{\text{jitter}}, q_m^{\text{loss}}] \tag{14}$$

$$R(i) \leq R_m \quad \forall i \in S \tag{15}$$

$$C(f_m) \leq q_k^{\text{bandwidth}} \quad \forall f_m \in F_L \tag{16}$$

$$C_r(i,j) \geq 0 \quad \forall (i,j) \in E. \tag{17}$$

The primary objective of the proposed problem is to maximize the number of flows while minimizing the BD in terms of bandwidth and energy cost with satisfying SLA constraints as provided in (11), while maximizing the BC capacity for future traffic demands. Equations (12) and (13) define the conservation of flow constraints such that there is only one source and destination node for every flow f_m belonging to F_L. Equation (14) represents the application-dependent requirements of the flow while forwarding the flow in a network. The tuple $\langle f_m(d), f_m(j),$ and $f_m(l)\rangle$ represents the delay, jitter, and loss requirement of flow, and it should be satisfied while choosing an edge (i,j) to route the flow. The values of flows in our scenario are defined according to the classification of the traffic as delay-sensitive, jitter-sensitive, or packet-loss sensitive. Equation (15) defines the rule-capacity constraints in our model. The demand constraint $q_k^{\text{bandwidth}}$ defined in (16) represents the QoS requirement of the flow f_m with the bandwidth value of the network. Equation (17) is associated with the capacity constraint with each edge $(i,j) \in E$. The routing configuration can be supported at the edge when the residual capacity is positive. Similarly, if the residual capacity constraint is negative, then the routing configuration cannot be supported on the edge. The QoS-aware max-flow min-cost algorithm consists of two ILPs, which can be formulated in the following.

B. Maximum-Flow ILP

The objective of the first ILP is to maximize the aggregate flows while considering the rule-capacity and link-capacity constraints. Mathematically, we have

$$\max \sum_{f_m \in F_L} \sum_{(i,j)\in E} f_m \alpha(i,j)$$

$$\text{s.t. } (12) - (15).$$

C. Minimum-Cost ILP

The objective of the second ILP function is to minimize the BD cost in terms of energy and bandwidth while satisfying the SLA constraints to maximize the remaining BC capacity available for future traffic flows. The second ILP is expected to satisfy the QoS requirement and to utilize the resources efficiently for future traffic flows

$$\min \sum_{f_m \in F_L} \sum_{(i,j)\in E} f_m \, C(i,j)\beta(i,j)$$

$$\text{s.t. } (12) - (15).$$

IV. Yen's k-Shortest Algorithm Implementation on GPU

The Compute Unified Device Architecture (CUDA) programming model consists of a set of instructions that are executed on threads in a GPU device. The threads are grouped in blocks. The implementation of the Bellman–Ford algorithm in this study is based on [20], in which a relaxation operation is used in the threads of blocks for sources and links. The value of pointer changes to 1 if there is any relaxation process on the link, while it terminates if there is no relaxation operation (pointer is set to 0). The Yen's algorithm uses the shortest path algorithm internally for finding the k-shortest paths.

The PyCUDA [21], a CUDA API for python, is implemented in the POX controller, as shown in Algorithm 1. The threads are created asynchronously on the GPU that can incur the synchronization issue due to the updating of a predecessor and distance of a node in a kernel. For solving the synchronization issue, two kernels, i.e., kernel-distance-update and kernel-predecessor-update, are implemented for updating the predecessors and distances, as proposed in [20].

A. QoS-Aware Online Algorithm

The ILP problem is NP-hard in general. The online optimization heuristic is implemented in the SDN controller that forwards the path calculation task in parallel to GPU for finding the optimal routing paths, considering the QoS nature of the

Algorithm 1: Calculating K-shortest paths in parallel using PyCUDA.

Input: The traffic flows f_m, each flow having its QoS requirement with sets of Edges E
Output: The shortest paths for each traffic flow f_m

The traffic flows f_m are forwarded to set S
$Pointer \leftarrow 1$
while $Pointer > 0$ **do**
    $Pointer > 0$
    Call Kernel update distance(S, E, $Distance$)
end while
Call Kernel update predecessor (S, E, $Distance$, $Previous$)
Calculate the shortest paths for traffic flows f_m based on predecessor information in the $Previous$, and forward the paths to P
return P

Algorithm 2: QoS routing algorithm for searching optimal route for different flows.

Input: Directed Graph , N
Input: Set of flows $\in F_L$, each flow f_m has its QoS requirement according to the application associated with smart health-care traffic
Input: The priority of flows i.e., ds flows, js flows, ls flows, and $Bandwidth - Distance$ with its associated QoS requirement. ▷ User-defined
Input: The maximum rule-capacity of the switch $R_m(i)$
Output: The set of edges $\in E$ or routes that can forward the different QoS requirement of the flows

for *each* $j \in N$ *and* S **do**
    flow-rules(j) $\leftarrow R_m$ ▷ rule-capacity initialization
$k, k \leftarrow 1$
while *all flows* $f_m \in F_L$ *are not forwarded* **do**
    ▷ fair allocation of ds , js and ls flows.
    if ∋ *ds flows not forwarded* **then**
        for $m \leftarrow 1$ *to* C_1 **do**
            route ← SEARCH-OPTIMAL ROUTE(x_m, z_m, q_m, t_m)
            $k \leftarrow k + 1$ ▷ QoS satisfied for ds flows
    if ∋ *js flows not forwarded* **then**
        for $n \leftarrow 1$ *to* C_2 **do**
            route ←SEARCH-OPTIMAL ROUTE(x_m, z_m, q_m, t_m)
            $j \leftarrow j + 1$ ▷ QoS satisfied to js flows
    if ∋ *ls flows not forwarded* **then**
        for $q \leftarrow 1$ *to* C_3 **do**
            route ← SEARCH-OPTIMAL ROUTE(x_m, z_m, q_m, t_m)
            $p \leftarrow p + 1$ ▷ QoS satisfied to ls flows

IIoT based on medical traffic. The proposed parallel routing optimization model calls the parallel Yen's k-shortest path calculation function, as explained in Algorithm 1, to calculate the shortest paths in parallel on GPU.

Algorithm 2 takes the set of flows with its SLA requirement, such as packet loss, delay, jitter, and BD, as an input considering the application plane. The algorithm outputs an optimal path if it exists. The ls, ds, and js flows are application dependent and are served in a round-robin fashion to ensure fairness among the traffic types. The QoS flows are calculated using the SEARCH-OPTIMAL ROUTE algorithm that is discussed

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Topology	Goodnet , AttMpls [22]
SDN-enabled switches	17 (Goodnet) , 25 (AttMpls)) [22]
Network links	31 (Goodnet) , 57 (AttMpls) [22]
Delay flows	1 - 100 ms
Loss flows	0 - 30 %
Jitter flows	0 - 50 ms
Avg. packet size	94-699 bytes [23]
Mean rate	562-516,540 bps bytes [23]

Algorithm 3: SEARCH-OPTIMAL ROUTE using parallel Yen's k-shortest path Algorithm in GPU.

function `SEARCH-OPTIMAL ROUTE`(*x , z , q ,t*)**:**
    if *flow-class= ds* **then**
        for *route in parallel Yen's k-shortest-routes (x, z, f_m)* **do**
            if *Search-QoS-route(route ,w)* **then**
                optimal-route ← route
    else if *flow-class= js* **then**
        for *route in parallel Yen's k-shortest-route (x, z, f_m)* **do**
            if *Search-QoS-route(route ,w)* **then**
                optimal-route ← route
    else if *flow-class= ls* **then**
        for *parallel Yen's k-shortest-route (x, z, f_m)* **do**
            if *Search-QoS-route(route ,w)* **then**
                optimal-route ← route
    return optimal-route
function `Search-QoS-route`(*route , demand*)**:**
    if *ls* ***and*** *js* ***and*** *ds* ***and*** *FLOWRULES* ***and*** *BANDWIDTH-DISTANCE* **then**
        return True
    return False

in Algorithm 3. The OF rule capacity of the SDN switches is initialized by using (3). Similarly, the remaining bandwidth of each link is also updated by using (7).

The function of SEARCH-OPTIMAL ROUTE is provided in Algorithm 3, as it forwards the path calculation task to GPU for finding the optimal paths. The parallel Yen's k-shortest-routes function calls the parallel shortest path calculation function, as shown in Algorithm 1, which calculates the shortest paths in parallel on GPU. The classified traffic, such as ds, ls, and js flows, is used to find the k-shortest paths considering the cost functions for different flows defined in (4)–(6). The loss, jitter, delay, and BD constraints are defined according to the type of application.

V. PERFORMANCE EVALUATION

The proposed scheme is evaluated by using the Mininet network emulator and the POX controller on Nvidia GPUs implemented on the PyCUDA (API for python) for calculating the shortest path in parallel on GPU. The simulation parameters for the experiment are provided in Table I. For the SLA requirement, the delay, jitter, and loss flow values are assigned according to

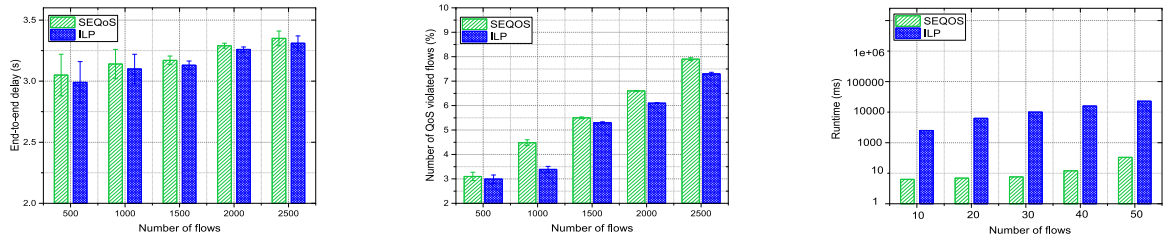


Fig. 2. Offline comparison: tradeoff between optimal ILP solution versus SEQOS. (a) Average end-to-end delay with different number of flows. (b) Number of violated SLA flows (%). (c) Runtime.

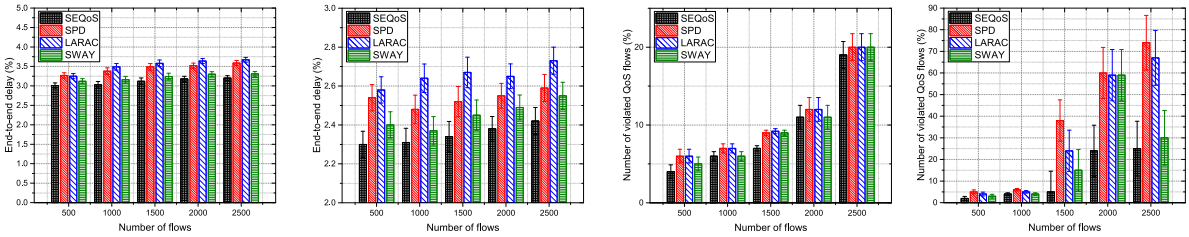


Fig. 3. End-to-end delay with SLA requirement and performance comparison of SLA (%) of health flows. (a) End-to-end delay with SLA requirement using the AttMpls topology. (b) End-to-end delay with SLA requirement with using the Goodnet topology. (c) Satisfied SLA using the AttMpls topology. (d) Satisfied SLA Goodnet topology.

health-related requirements. The proposed scheme is compared with the shortest path delay (SPD) [24], which considers a single metric delay as a cost metric for forwarding the traffic to the destination. Besides, the Lagrangian relaxation-based aggregated cost (LARAC) [15] method using a genetic algorithm to find the optimal QoS route and a greedy heuristics method [6] based on a sequential algorithm is also considered to simulate on CPU environment, which is not scalable for the massive flux of traffic generated from IIoT devices.

A. Performance Comparison of the Offline Approach

The Gurobi Optimizer [25] is used to solve the ILP formulated in Section IV. The offline comparison of optimization framework and proposed SEQOS is shown in Fig. 2. A low computation time about SEQOS and moderate variation from the optimal solution in contrast to ILP is observed. Hence, the SEQOS provides a tradeoff between consideration of SLA criteria and runtime.

B. Performance Comparison of the Online Approaches

1) End-to-End Delay: The end-to-end delay of SEQOS is provided in Fig. 3(a) and (b). The performance of the proposed scheme is outperformed in the benchmark scenarios, even with the increasing number of flows in the network. The proposed scheme achieves 11%, 13%, and 4% reduction, respectively, in end-to-end delay on average compared to the benchmark SPD, LARAC, and Sway schemes using the AttMpls topology, as provided in Fig. 3(a). Similarly, the Goodnet topology uses provided end-to-end delay reduction of 14%, 8%, and 6% on average compared to LARAC, SPD, and Sway schemes in Fig. 3(b). The proposed scheme forwards the k-shortest path calculation task to GPU to speed up the computational task in parallel. The value of k is further increased in the heuristic to search the optimal paths. Hence, less end-to-end is achieved due to a smaller number of violated SLA flows. In contrast, the benchmark schemes forward the traffic by using best-effort routing. They also use sequential algorithms to find the optimal paths, while the number of SLA violated flows is higher in number resulting in high end-to-end delay.

2) Violated SLA Flows: The violated SLA flows are ones that are forwarded by using best-effort and have no feasible path. The comparison of the percentage of violated SLA flows of the SEQOS and the benchmark scheme is provided in Fig. 3(c) and (d). The figures indicate that the SEQOS outperforms the benchmarks in terms of reduction by violated SLA flows. The proposed scheme achieves 17%, 19%, and 4% reductions in the SLA flows compared to SPD, LARAC, and Sway schemes when the AttMpls topology is used, as depicted in Fig. 3(c), while 48%, 44%, and 7% reductions are observed when the Goodnet topology is used, as shown in Fig. 3(d). The benchmark LARAC, SPD, and Sway schemes have a sudden increase in the percentage of violated SLA constraints with 2000 and 2500 flows in the network with a saturation of ternary content-addressable memory (TCAM) occurrences. Although the SEQOS forwards the path calculation task in parallel to GPU for finding the optimized routes in less time, it also considers the TCAM rule-capacity constraint of the switches, while consequently performing better with a significant reduction in the percentage of violated SLA flows. The performance of SEQOS is scalable

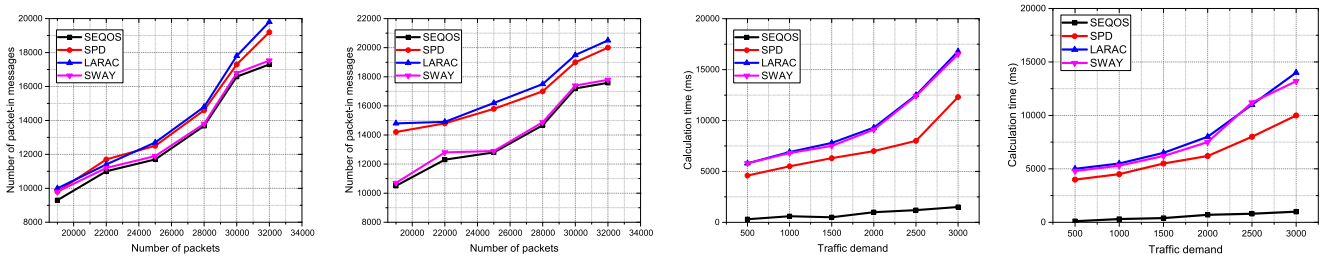


Fig. 4. Effect of rule-capacity constraint and computation time using AttMpls and Goodnet network topologies. (a) Effect of rule-capacity constraint with using the AttMpls topology. (b) Effect of rule-capacity constraint with using the Goodnet topology. (c) Computation time using the AttMpls topology. (d) Computation time using the Goodnet topology.

TABLE II
EFFECT OF BD CAPACITY WITH HIGH TRAFFIC LOAD

Algorithm	SLA satisfied (%)	Edge Load (%)	Bandwidth before routing (GB/s)	Bandwidth after routing (GB/s)	Energy cost
SPD	83	71	920k	1098k	500 J
LARAC	86	69	920k	960k	450 J
SWAY	90	68	920k	940k	300 J
SEQOS	97.4	64	920k	910k	230 J

and good in terms of reduction in SLA when compared to benchmark schemes. Hence, the proposed scheme is scalable for IIoT-based healthcare systems with different QoS requirements of the application.

C. BD Capacity

The effect of BD capacity with different traffic loads is provided in Table II. The SPD, LARAC, and Sway schemes have 6%, 7%, and 10% higher BD utilization compared to the proposed SEQOS with less satisfied SLA requirement of the flows. The data show that the SEQOS achieves minimum BD utilization compared to the benchmark scheme at higher traffic loads. The proposed scheme achieves a higher percentage of SLA satisfied flows with less amount of resources utilized compared to the state-of-the-art schemes. Hence, the proposed scheme performs satisfactorily with maximizing the BC capacity for future IIoT based on health traffic.

1) Energy Efficiency: The analysis of the SEQOS is performed in terms of energy efficiency compared to the benchmark scheme, and its results are provided in Table II. Each edge of the network costs an amount of energy for forwarding the traffic. The proposed algorithm always tends to forward the traffic on the shortest energy paths while satisfying the QoS constraints. Thus, SEQOS utilizes 230 J of energy on average for forwarding the traffic flows in the network. In contrast, SPD does not consider energy constraints and utilizes 500 J for forwarding the same traffic flows. The LARAC utilizes 450 J as it uses an optimization technique for forwarding but does not consider the energy constraint while forwarding the traffic. Finally, the Sway scheme considers a link activation and sleeping mechanism while forwarding the traffic and achieves less energy utilization compared to SPD and LARAC. Thus, SEQOS utilizes less energy compared to the existing schemes making the network green.

2) Packet-in Message: Fig. 4(a) and (b) shows the effect of rule-capacity constraints in the SEQOS scheme when compared to the existing schemes. The proposed scheme reduces packet-in messages sent to the controller by 7% and 8% compared to SPD and LARAC schemes using the AttMpls topology and 16% and 18% reduction using the Goodnet topology. The Sway and SEQOS schemes use rule capacity as a constraint; hence, both schemes have similar packet-in messages. For this reason, the proposed scheme is beneficial in reducing the controller overhead and congestion for IIoT-based smart healthcare systems.

3) Calculation Time for Optimal Paths: The time complexity of Algorithms 2 and 3 is analyzed. There are $|F|$ iterations of the while loop in Algorithm 2 with a function call to SEARCH-OPTIMAL ROUTE at each iteration. The Yen's k-shortest route algorithm utilizes the shortest path algorithm that uses GPU in parallel for calculating the shortest paths. The SEARCH-OPTIMAL ROUTE function in Algorithm 3 has a set of neighborhood nodes in the network graph, since its running time can be upper bounded by $O(\frac{K|S||E|}{C} + |S|\log|S| * C)$, where C is the number of GPU cores used for finding the shortest paths. Thus, the proposed QoS-aware heuristic has pseudopolynomial time complexity of $O((|F|_*(\frac{K|S||E|}{C} + |S|\log|S| * C)))$. Fig. 4(c) and (d) shows the calculation time for achieving optimal paths using the parallel processing power of GPU with different loads by using AttMpls and Goodnet topologies. A denser network topology compared to Goodnet is presented in Fig. 4(d), where the calculation time of SEQOS is much less than that of the benchmark schemes. The calculation time of the sparse Goodnet topology also has less computational time compared to that of benchmark schemes. Hence, the SEQOS is scalable and fast compared to the benchmark schemes.

VI. CONCLUSION

In this article, a fast parallel online routing optimization framework has been proposed for IIoT-based smart healthcare systems using the SDN approach. The proposed scheme considers the heterogeneous requirement of smart healthcare flows as *delay-sensitive*, *jitter-sensitive*, and *loss-sensitive*. The proposed heuristic is calculated on a GPU considering the SLA

requirement with the TCAM constraint. The SEQOS scheme was determined to be capable of satisfying the SLA requirement and utilizing the bandwidth as well as energy resources of the network efficiently for future IIoT traffic. It was also seen to be eight times faster compared to the state-of-the-art schemes, which makes it very promising for the next-generation smart healthcare systems. The next-generation IoT networks are expected to be highly dynamic and complex. Thus, in future work, we plan to consider machine learning techniques such as deep reinforcement learning techniques for satisfying the QoS of the network.

REFERENCES

[1] T. Umer, M. H. Rehmani, A. E. Kamal, and L. Mihaylova, "Information and resource management systems for Internet of Things: Energy management, communication protocols and future applications," *Future Gener. Comput. Syst.*, vol. 92, pp. 1021–1027, 2019.

[2] M. S. Hossain and G. Muhammad, "Cloud-assisted industrial Internet of Things (IIoT)-enabled framework for health monitoring," *Comput. Netw.*, vol. 101, pp. 192–202, 2016.

[3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[4] S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato, "A survey on network methodologies for real-time analytics of massive IoT data and open research issues," *IEEE Commun. Surv. Tut.*, vol. 19, no. 3, pp. 1457–1477, Jul.–Sep. 2017.

[5] F. Al-Turjman and S. Alturjman, "Context-sensitive access in industrial Internet of Things (IIoT) healthcare applications," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2736–2744, Jun. 2018.

[6] S. Sahhaf, W. Tavernier, D. Colle, and M. Pickavet, "Adaptive and reliable multipath provisioning for media transfer in SDN-based overlay networks," *Comput. Commun.*, vol. 106, pp. 107–116, 2017.

[7] A. Stanford-Clark and H. L. Truong, *MQTT for Sensor Networks (MQTT-SN) Protocol Specification*, version 1.2. New York, NY, USA: IBM Corporation, 2013.

[8] L. Skorin-Kapov and M. Matijasevic, "Analysis of QoS requirements for e-health services and mapping to evolved packet system QoS classes," *Int. J. Telemed. Appl.*, vol. 2010, 2010, Art. no. 628086.

[9] J. Wan *et al.*, "Software-defined industrial Internet of Things in the context of Industry 4.0," *IEEE Sens. J.*, vol. 16, no. 20, pp. 7373–7380, Oct. 2016.

[10] Z. Zhou, J. Gong, Y. He, and Y. Zhang, "Software defined machine-to-machine communication for smart energy management," *IEEE Commun. Mag.*, vol. 55, no. 10, pp. 52–60, Oct. 2017.

[11] F. Naeem, G. Srivastava, and M. Tariq, "A software defined network based fuzzy normalized neural adaptive multipath congestion control for the Internet of Things," *IEEE Trans. Netw. Sci. Eng.*, to be published, doi: 10.1109/TNSE.2020.2991106.

[12] I. Cisco, "Cisco visual networking index: Forecast and methodology, 2011–2016," *CISCO White Paper*, vol. 518, 2012.

[13] K. Kikuta, E. Oki, N. Yamanaka, N. Togawa, and H. Nakazato, "Effective parallel algorithm for GPGPU-accelerated explicit routing optimization," in *Proc. IEEE Global Commun. Conf.*, 2015, pp. 1–6.

[14] K. Venkatesh, L. Srinivas, M. M. Krishnan, and A. Shanthini, "QoS improvisation of delay sensitive communication using SDN based multipath routing for medical applications," *Future Gener. Comput. Syst.*, vol. 93, pp. 256–265, 2019.

[15] H. E. Egilmez, S. Civanlar, and A. M. Tekalp, "An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 710–715, Apr. 2013.

[16] Y. Liu, D. Niu, and B. Li, "Delay-optimized video traffic routing in software-defined interdatacenter networks," *IEEE Trans. Multimedia*, vol. 18, no. 5, pp. 865–878, May 2016.

[17] N. Saha, S. Bera, and S. Misra, "Sway: Traffic-aware QoS routing in software-defined IoT," *IEEE Trans. Emerg. Topics Comput.*, to be published, doi: 10.1109/TETC.2018.2847296.

[18] S. Agarwal, M. Kodialam, and T. Lakshman, "Traffic engineering in software defined networks," in *Proc. IEEE INFOCOM*, 2013, pp. 2211–2219.

[19] H. Liao *et al.*, "Learning-based context-aware resource allocation for edge computing-empowered industrial IoT," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4260–4277, May 2020.

[20] X. Wang, Q. Zhang, J. Ren, S. Xu, S. Wang, and S. Yu, "Toward efficient parallel routing optimization for large-scale SDN networks using GPGPU," *J. Netw. Comput. Appl.*, vol. 113, pp. 1–13, 2018.

[21] A. Klöckner, N. Pinto, Y. Lee, B. Catanzaro, P. Ivanov, and A. Fasih, "PyCUDA and PyOpenCL: A scripting-based approach to GPU run-time code generation," *Parallel Comput.*, vol. 38, no. 3, pp. 157–174, 2012.

[22] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet topology zoo," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 9, pp. 1765–1775, Oct. 2011.

[23] A. Sivanathan *et al.*, "Characterizing and classifying IoT traffic in smart cities and campuses," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2017, pp. 559–564.

[24] J. M. Llopis, J. Pieczerak, and T. Janaszka, "Minimizing latency of critical traffic through SDN," in *Proc. IEEE Int. Conf. Netw., Archit. Storage*, 2016, pp. 1–6.

[25] *Gurobi Optimizer Reference Manual*, Gurobi Optimization, Beaverton, OR, USA, 2015.

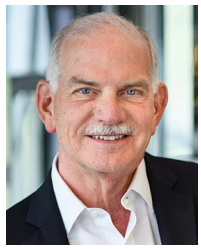
Faisal Naeem (Member, IEEE) is currently working toward the Ph.D. degree in electrical engineering with the National University of Computer and Emerging Sciences, Peshawar, Pakistan.

Since 2018, he has been a Research Fellow with the Department of Electrical Engineering, National University of Computer and Emerging Sciences. His current research interests include Internet of Things (IoT), software-defined networking, and deep reinforcement learning techniques for optimizing the resources of IoT-based networks.

Muhammad Tariq (Senior Member, IEEE) received the M.S. degree in electronics & computer engineering from Hanyang University, Seoul, South Korea, in 2009, and the Ph.D. degree in wireless communication systems from Waseda University, Shinjuku, Japan, in 2012.

He is an Associate Professor and the Director of the National University of Computer and Emerging Sciences, Peshawar, Pakistan, where he was the Head of the Department of Electrical Engineering. He was a Postdoctoral Researcher with Princeton University as a Fulbright Scholar under the supervision of Prof. H. V. Poor in 2016.

Dr. Tariq has received various rewards for his work and delivered research talks as a guest/invited/keynote speaker at various forums and Universities in Pakistan, China, Saudi Arabia, and the U.S.



H. Vincent Poor (Life Fellow, IEEE) received the Ph.D. degree in electrical engineering & computer science from Princeton University, Princeton, NJ, USA, in 1977.

From 1977 to 1990, he was on the faculty of the University of Illinois at Urbana-Champaign, Urbana, IL, USA. Since 1990, he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor of Electrical Engineering. From 2006 to 2016, he served as Dean of Princeton's School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently with the University of California, Berkeley and the University of Cambridge. His current research interests include information theory, machine learning and network science, and their applications in wireless networks, energy systems, and related fields. Among his publications in these areas is the forthcoming book is *Advanced Data Analytics for Power Systems* (Cambridge, U.K.: Cambridge Univ. Press).

Prof. Poor is a Member of the U.S. National Academy of Engineering and the U.S. National Academy of Sciences, and a Foreign Member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. Recent recognition of his work includes the 2017 IEEE Alexander Graham Bell Medal and a D.Eng. *honoris causa* from the University of Waterloo awarded in 2019.