LBSMT: Load Balancing Switch Migration Algorithm for Cooperative Communication Intelligent Transportation Systems

Himanshi Babbar, Shalli Rani, *Member, IEEE*, Ali Kashif Bashir, Raheel Nawaz

***Abstract*—We entered an era when the automotive industry is undergoing a digital revolution. Automobiles evolving into automated movable objects are using artificial intelligence capabilities. In contrast, cellular communications networks incorporate emerging technologies, such as SDN (software-defined networking) and NFV (network functions virtualization). Sophisticated software-defined communications networks virtualizes network functions and paving the way for the new design, monitoring, and management strategies. SDN is rising towards the application of load balancing for real time applications due to the heavy load of data on servers. When there is intra-communication between the various switches and domains; migration of switches takes place and the load over servers is imbalanced. An imbalance of the load will increase the response time and decrease the throughput. In intelligent transportation systems (ITS) balance on the servers should be maintained for the network sustainability. To provide a solution for the requirement of ITS, a dynamic QoS-aware load balancing switch migration algorithm (LBSMT) is proposed in this paper. As per the results validated in Python, after the migration LBSWT has improved CPU utilization, memory utilization, throughput and response time over server load, round robin, weighted round robin, LBBSRT and dynamic server algorithms.**

***Index Terms*—Software Defined Networking, Load Balancing, Switch Migration, Cooperative Communication, Intelligent Transportation System**

I. Introduction

THE Dutch Government of Infrastructure and Services, in collaboration with HERE, launched a way to solve testing and development in early 2016 and therefore, the assessment of a ”Cooperative” cellular network named ITS was initiated [1], [2]. As the ITS is increasing at a very fast pace, therefore vehicular networks are at the utmost advantage. These networks are presently facing a significant difficulty as a result of increased demand for services that make shipping safer, more effective, less expensive, and more friendly. New technologies namely SDN [3] and NFV have been added to the process. The cooperative ITS main goal is to have effective communication with each other and with the various domains, this optimizes the modes of transportation [4]. The architecture of transportation system with respect to SDN controller is depicted in Fig. 1, Every SDN controller is in charge of all the RSUs within its field of view, choosing the RSUs' destinations and monitoring the RSUs' operational status on a regular basis. For instance, whenever an RSU is undermined or has performance degradation, SDN delegates the RSU's function to the nearest RSU or to a virtual RSU [5].

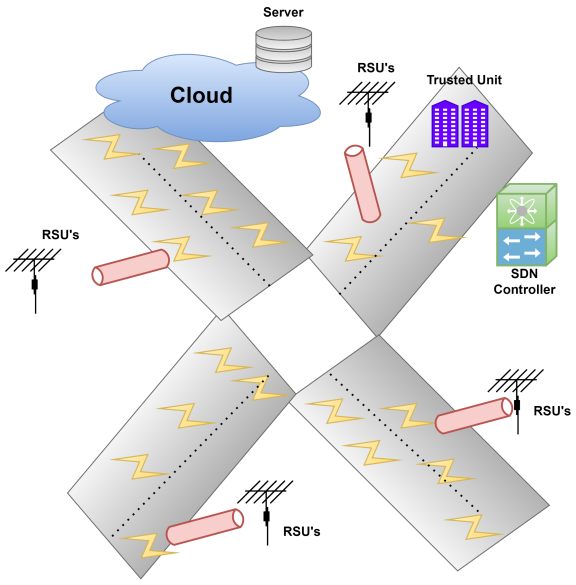

Fig. 1. Transportation System in SDN Controller

Traditional networks must be upgraded to handle the massive amount of network traffic created by the Internet of Things (IoT). A new form of networking architecture termed SDN has been presented as a result of the intervention for an optimal solution to this problem [6]. By breaking the vertical integration of network components, removing the control plane from the underlying data plane, encouraging (logical) centralization of network control, and offering the capability to design network operation [7], SDN transforms conventional network administration. In this, the single controller cannot handle the large amount of load with the increasing number of visits in the panel. So, to overcome this issue we have proposed multiple controllers to show the improvement of scalability and reliability in the control plane [8]. Switches in the domain interact with the controllers and hosts for the balancing of load between the various domains. As the number

H. Babbar and S. Rani are with the Chitkara University of Engineering and Technology, Chitkara University, Punjab, India (email: himanshi.babbar@chitkara.edu.in and shalli.rani@chitkara.edu.in).

A.K.Bashir is with with Department of Computing and Mathematics,Manchester Metropolitan University, Manchester, United Kingdom (email: dr.alikashif.b@ieee.org)

R.Nawaz is with Department of Operations, Technology, Events and Hospitality Management, Manchester Metropolitan University, UK. (email: R.Nawaz@mmu.ac.uk)

of visits varies, controllers and hosts in different domains handle the traffic in the network [9]. Dynamic migration of switches is a scheme that resolves the problem of imbalance load and migrates the switches from the overloaded domain to the underloaded one [10]. The existing scheme doesn't handle a large amount of traffic in the domains therefore, in this paper we have proposed the highly scalable and efficient "QoS-aware load balancing switch migration scheme" to achieve the maximum throughput, minimum response time with maximum utilization of CPU and minimum utilization of memory [11] [4].

A. Contributions of the paper

To balance the load after migrating the switches, the main contributions of the paper are:

1) We have designed the topology for the preliminary configuration of the network and proposed a topology for the intelligent communication in ITS, in which we have purchased three domains (.com, .org and .in) each consisting of one controller, one host and various switches.
2) In the preliminary network configuration for ITS, the load is unbalanced as it shows the number of visits at time t1. and in the proposed topology two algorithms are developed: i) In the first algorithm, the imbalance load is detected amongst the various domains which depends upon the number of visits and they should not be greater than threshold value for balanced switch. ii) In the second algorithm, the load that is detected imbalance will be migrated to another switch of a particular domain.
3) A new framework is designed for ITS which shows the flow of the switches that migrates load from one domain to another after checking the threshold values of switches' load.
4) The performance of various QoS metrics has been evaluated on the basis of the throughput, response time, CPU utilization and memory utilization.

The remaining paper is designed as: Section 2 shows the related work and motivation of the proposed scheme from the preliminary scheme by designing the topology. Section 3 shows the methodology, framework and system model for the proposed model. Section 4 presents the validations and simulations by analyzing the accuracy of the existing schemes. Lastly, section 5 shows the conclusion of the paper.

II. Related Work and Motivation

A. Related Work

[1] shows the outline of critical information, difficulties, and possibilities for the telecom industry to enable people and commodities mobility in ITS. [12] presents the problems to enable vehicle-to-x communication; for example, connected automobiles, which would be the first machines to have a significant impact on millions of people's everyday lives. We look at the 5G architecture created with Software Defined Networking and its function in Diverse Networks in general. [13] designed an efficient algorithm for the load balancing switch migration which shows that the protocol used for this has 25% accuracy in migration time and 10% in migration buffer size which ensures consistency amongst all controllers. [14] provides a switch and controller selection method for switch migration that is based on deep learning. Load balancing reinforcement learning (SAR-LB). The usage ratio of different sources is used by SAR-LB. Both controllers and switches can be used as neural network inputs. Switches are also taken into account as RL agents to decrease the learning action space while considering all migration scenarios. [15] provides an SDN-based load balancing (SBLB) solution for data centers that maximize resource efficiency while reducing user response time. An application module that operates in front of an SDN controller and server pools that link to the controller via Openflow are the components of the current technique. [16] described in terms of space and processing power, vehicle energy, and network management and administration, cooperative communication between cars and other devices poses a variety of issues. Security is a significant part of the Internet of Vehicles, and it is essential to safeguard connected vehicles against theft and disasters. [6] new flexible LB method is implemented that combines the genetic algorithm (GA) with Ant colony optimization (ACO) to boost the effectiveness of SDN. It takes advantage of the advantages of GA's quick global search and ACO's effective search for the best solution. [17] proposed as an effective solution to the problem. When the network size increases, the simulation results demonstrate that EASM improves baseline methods by decreasing controller response time by roughly 21.9 percent, boosting controller throughput by 30.4 percent on average, creates a stable load balancing rate, and lowering migration costs and time. [18] designed load balancing approach based on the prediction that accurately detects overloaded controllers and chooses target underloaded controllers to move partial load from one of the overloaded ones. Simulation results demonstrate that EXPRL allows the network to significantly boost network throughput while also lowering network latency and migration costs compared to its state-of-the-art competitors. [19] address the switch migration problem, a heuristic strategy with solution shaking is shown. Within a search strategy, shift and swap actions are included. Every action is assessed in terms of how beneficial it'll be to both immigration and outmigration control officers. When contrasted to some of the most recent methodologies, the experimental results suggest that the proposed methodology can outperform state-of-the-art methodologies and enhance load balancing results by up to 14 percent in specific cases. black The comparative analysis of existing load balancing algorithms have been explained in Figure I:

black Our proposed scheme is

1) We have proposed 3 domains and each domain consists of 1 controller, 1 host switch, and various switches.
2) Load of all the domains is balanced by migrating the switches from the heavily loaded domain to the lightly loaded domain.
3) The switches are assigned the weight in Kbps, and the

TABLE I
COMPARATIVE ANALYSIS OF VARIOUS LOAD BALANCING ALGORITHMS

Author Name / Year	Proposed Technique	Parameters	SDN Controller	Topology	Advantages	Limitations
[20] / 2018	Online Controller Load Balancing	Response time, Number of switches migration, Standard deviation	NOX Controller	Fat-tree (number of switches=320, controllers=6)	minimize the average controller response time	exploring the response diversity of different flow requests to design a more scalable control plane.
[21] / 2018	SMCLBRT, a load balancing strategy of multiple SDN controllers based on response time	Controller loads, Load ratio, Number of switches migration	SDN multiple controller load-balancing	load-balancing framework based on SMCLBRT	Achieves load balancing of multiple SDN controllers effectively and quickly	Better balancing approach of multiple overloaded controllers by considering the migration cost.
[22] / 2018	Wardrop Load-Balancing algorithm	Throughput, average latency	Open daylight		Dynamically balancing the requests of the switches among the SDN controllers to avoid congestion	Improving the convergence properties of the algorithm.
[23] / 2019	Dynamic fractional-level assignment and heuristic algorithm	average number of iterations, average coefficient of variation, average min-max	number of controllers=6 and switches=24		increases the load balancing accuracy slightly and at the same time improves the mapping stability	Design of a migration protocol for the fractional switch migration, a network state synchronization paradigm.
[24] / 2020	experience and prediction based load balancing strategy	average controller throughput, percentage of cascaded migration, average cost of load balancing	Floodlight	B-cube(4,1)	significantly high network throughput at the cost of much reduced load information message exchange	Number of cascaded migrations and overloaded controllers.
[25] / 2020	Efficient, failure Resilient, and Consistent load migration protocol	Buffer size, protocol running time	Three controllers (C0, C1, and C2) connected to 15 switches	ERC network topology	liveness, serializability and safety along with consistency and failure resiliency	Specific controller acts as the coordinator for such scheduling tasks trying to utilize a series of sequential and parallel switch migrations.

threshold value is fixed. E.g. let's suppose the threshold value is 40. Now we need to check if the weight of switches is greater than 40, then migrate that switch to make the particular domain balanced.

4) Computation of the domain is done in both ways: Intra domain and Inter-domain. Later, we need to calculate the distance between switches and controllers.
5) The QoS metrics based on Throughput, Response time, Average CPU utilization, and Average Memory utilization are analyzed for optimization through the new algorithm.

black

B. Motivation

Every switch in a distributed controller architecture is managed by a controller in the local domain. Controllers handle traffic flows, and as the internet traffic rises, the mapping of flow rules between controller and switch becomes overloaded, while a few controllers become underloaded [26] [27]. Due to reduced throughput and lengthy response time, such imbalance affects the effectiveness of the SDN network. Switch migration is most commonly used to change the allocation of controller loads by moving the switch from an overloaded to an underloaded controller [28]. Existing migration approaches, on the other hand, make it challenging to achieve good load balancing effectiveness and low migration costs.

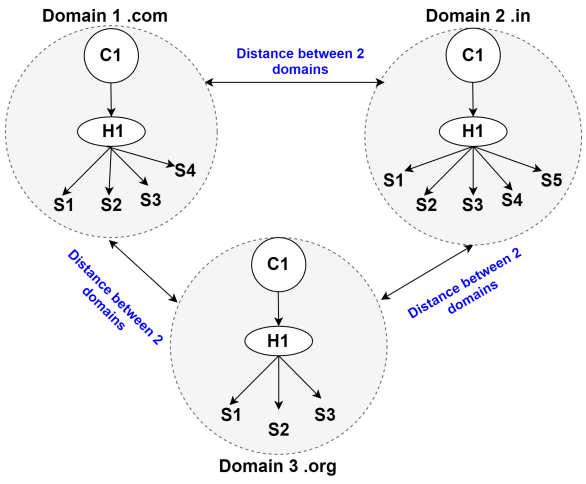

Fig. 2. Preliminary Network Configuration

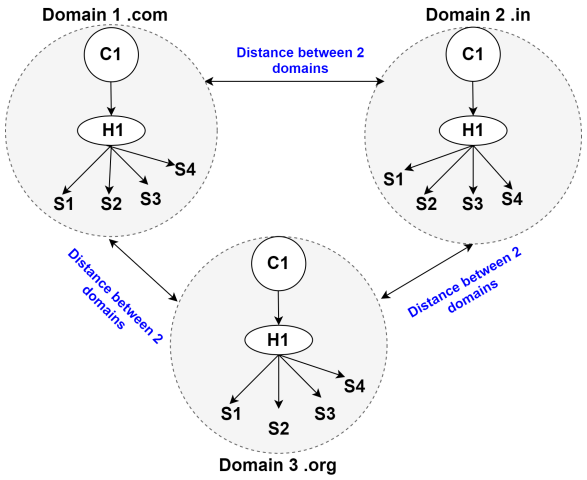

Fig. 3. Proposed Topology

In Fig 3, the SDN network comprises three controllers, hosts. The network has 3 different domains namely domain1.com, domain2.in and domain3.org, in which each domain consists of several switches and is managed by the controller and host [29]. The incoming visits of each switch are shown as time t1. At time t1, the overall flow rates of a domain are used to indicate the domain's load, as well as the standardized load variance, is used to indicate the load balancing rate (LBR) as depicted in eq 2, 3 and 4 for different domains domain1.com(D1.com), domain2.in(D2.in) and domain3.org(D3.org) respectively in which the overall load and average load is computed in eq 1:

$$\frac{1}{n}(l1 + l2 + l3 + \ldots\ldots\ldots + ln)$$

$$Totalload(count) = \sum_{i=1}^{n} l_i$$

$$Averageloadof Domains = \frac{1}{n}\sum_{i=1}^{n} l_i \quad (1)$$

li is the load of a particular switch, and $1/$n L1 is the average load of the switches. More the LBR makes the distribution of load-balanced.

$$LBRfordomain1.com = \frac{\frac{1}{n}\sum_{i=1}^{n}(l_i - \frac{1}{n}L1)}{\sqrt{\sum_{i=1}^{n}(l_i - \frac{1}{n}L1)}} \quad (2)$$

$$LBRfordomain2.in = \frac{\frac{1}{n}\sum_{i=1}^{n}(l_i - \frac{1}{n}L1)}{\sqrt{\sum_{i=1}^{n}(l_i - \frac{1}{n}L1)}} \quad (3)$$

$$LBRfordomain3.org = \frac{\frac{1}{n}\sum_{i=1}^{n}(l_i - \frac{1}{n}L1)}{\sqrt{\sum_{i=1}^{n}(l_i - \frac{1}{n}L1)}} \quad (4)$$

The controllers are initialized with imbalanced loads, the average load of all the domains and the load balancing rates of three domains are calculated by using the following:

Preliminary network configuration for load imbalance (A1)

Load$_{A1}(D1.com)$ = 20 + 40 + 35 + 27 = 122 KB/sec
Load$_{A1}(D2.in)$ = 37 + 34 + 34 + 34 + 34 = 173 KB/sec
Load$_{A1}(D3.org)$ = 23 + 34 + 34 = 91 KB/sec
Average load of D1.com = 30.5
Average load of D2.in = 34.6
Average load of D3.org = 30.3

In the above Fig 2, the underloaded domains are D1.com and D3.org while the overloaded domain is D2.in. In this, each domain has one controller and one host which plays three roles: master, equal and slave [30]. In the master domain, the controller is required for refining the number of visits sent by the switches; As a backup, equal and slave domains are utilized [31]. Each switch is interconnected with one master and various slave domains. In Fig. 1b the proposed topology named "QoS-aware load balancing switch migration scheme" is used. In this, the load of all the domains are been assigned the threshold values in KB/sec and if the number of visits in one domain of a particular switch has crossed the threshold value then the load will be migrated to another switch of that domain. In case all the visits have been fulfilled in the domain1.com then the load will be migrated to another domain2.in and if all the switches have crossed the threshold value in the domain2.in then the load will be migrated to domain3.org. black

In this, the load in D2.in of switch S5 in Fig 2 has been migrated to the D3.org i.e 34 in Fig 3, as the D2.in was overloaded in the preliminary. After the switch migration is done, the load of the domain and average load is computed as:

QoS-aware load balancing switch migration (QOSALBSM) scheme

Load$_{A1}(D1.com)$ = 20 + 40 + 35 + 27 = 122 KB/sec
Load$_{A1}(D2.in)$ = 37 + 34 + 34 + 34 = 139 KB/sec
Load$_{A1}(D3.org)$ = 23 + 34 + 34 +34 = 125 KB/sec
Average load of D1.com = 30.5
Average load of D2.in = 34.7
Average load of D3.org = 31.2

The number of visits that are expected to be in three different domains for switches is shown in Table II at time t1.

black The comparison and contrast of two different scenarios are discussed in Table III which shows the preliminary network configuration for load imbalance and the proposed scheme improves the balancing of load and average load of all the domains. The table III intimates the switch migration to improve the performance of the domains which describes: migration of switches, the effectiveness of the interaction between the various domains and evaluates the switch migration.

III. System Model

In this section, the notations are used for the designing of the proposed scheme for reliable and effective load balancing. The topology of SDN consists of the graph G which has

TABLE II
NUMBER OF VISITS OF A PARTICULAR SWITCH IN THE DIFFERENT DOMAINS

Switches	**S1**	**S2**	**S3**	**S4**	**S1**	**S2**	**S3**	**S4**	**S1**	**S2**	**S3**	**S4**
Number of Visits (KB/sec)	**20**	**40**	**35**	**27**	**37**	**34**	**34**	**34**	**23**	**34**	**34**	**34**

TABLE III
CONTRAST BETWEEN TWO DIFFERENT SCENARIOS

Domains (KB/sec)	**Preliminary Network Configuration**	**QoS-aware load balancing switch migration**
Domain1.com	122	122
Domain2.in	173	139
Domain3.org	91	125
Average load of Domain1.com	30.5	30.5
Average load of Domain2.in	34.6	34.7
Average load of Domain3.org	30.3	31.2

vertices V and links between controller, host and switches; E i.e. $G =< V, E >$, D1.com $\rightarrow$S1, S2, S3, S4;$D2.in$ $\rightarrow$S1, S2, S3, S4;$and D3.org$ $\rightarrow$S1, S2, S3, S4.

The domain's load is highly dynamic. Each domain has one controller which handles the load of the switches and one host which communicates with the switches for the migration process and various switches that are controlled by a domain. The overall framework of the proposed scheme is shown in Fig 4 which explains and computations of domains load which results in various network overheads are given below:

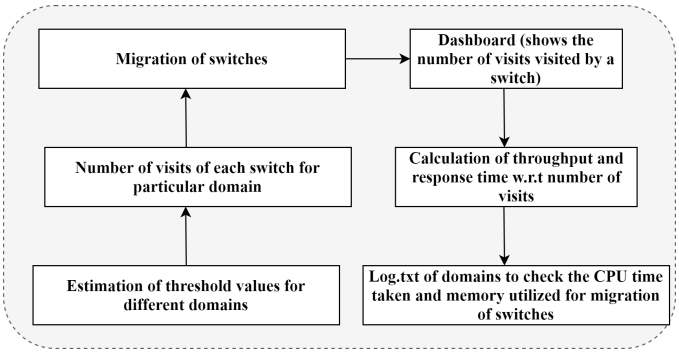

Fig. 4. Framework of QoS-aware load balancing switch migration scheme

Some theories have been framed while defining the proposed scheme:

- **Theory 1:** To balance the load of the domain in the control plane, the SDN network can allow the switches to migrate from one domain to another domain.
- **Theory 2:** If the particular switch has chosen to be migrated, then the switch that is migrated cannot return to the previous domain until and unless the new domain has finished the process.
- **Theory 3:** One controller and one host have been assigned to one domain with several switches, in which overloaded domain will serve as master and the other will serve as slaves.
- **Theory 4:** All the domains cannot be said to be overloaded at the same time t1.

A. Proposed Methodology

We devise a modest but efficient method for determining if the controller loads in the network are balanced is shown in eq 5 to 7.

$$D1.comL1 = \sum_{i=1}^{n} l_i \tag{5}$$

$$D2.inL2 = \sum_{j=1}^{n} l_j \tag{6}$$

$$D3.orgL3 = \sum_{p=1}^{n} l_p \tag{7}$$

Step 1: The matrix is developed for the domains load is defined in eq 8 as

$$d_i(L_i, L_j) = \frac{d_i(L_i)}{d_i(L_j)}$$

$$D_i n * n = \begin{bmatrix} d_{i(L1,L1)} & d_{i(L1,L2)} & d_{i(L1,L3)} \\ d_{i(L2,L1)} & d_{i(L2,L2)} & d_{i(L2,L3)} \\ d_{i(L3,L1)} & d_{i(L3,L2)} & d_{i(L3,L3)} \end{bmatrix} \tag{8}$$

where $d_i(L_m, L_n)$ is the distance between the three domains, to evaluate the distance between the domains, the estimation of threshold values for different domains and trigger factor are calculated in eq 9 and eq 10 respectively:

black

$$Threshold, T = \frac{maxD_{in*n} - minD_{in*n}}{maxD_{in*n}} \tag{9}$$

Where $maxD_{in*n}$ is the highest load on the particular domain and $minD_{in*n}$ is the minimum load on that particular domain.

black

$$\phi_{i(mn)} = |d_{i(L_m,L_n)} - d_{i(L_n,L_m)}| > T \tag{10}$$

where $\phi_{i(mn)}$ is the trigger factor set. If the trigger factor set is greater than the T then the detection of imbalance load occurs means a particular switch of the domain has crossed the threshold value and therefore, that particular switch should be migrated. Let's take an illustration to detect the imbalance load among the domains from our proposed scheme. In Fig. 1b we have L1 = 122 KB/sec, L2 = 139 KB/sec and L3 = 125

KB/sec, now we calculate the distance between the domains in the form of a given matrix in eq 8:

$$D_{i3*3} = \begin{bmatrix} d_{i(L1,L1)} & d_{i(L1,L2)} & d_{i(L1,L3)} \\ d_{i(L2,L1)} & d_{i(L2,L2)} & d_{i(L2,L3)} \\ d_{i(L3,L1)} & d_{i(L3,L2)} & d_{i(L3,L3)} \end{bmatrix} = \begin{bmatrix} 1.0 & 0.8 & 0.9 \\ 1.1 & 1.0 & 1.1 \\ 1.0 & 0.8 & 1.0 \end{bmatrix}$$

black Where $T = \frac{1.1–0.8}{1.1} = 0.2$; therefore T = 0.2 and $\phi_i(12) = |0.8–1.1| > 0.2 \Rightarrow 0.3¿0.2 \Rightarrow$ load imbalance, is detected which means migration of the particular switch is required. Hence justified. black

B. Algorithm for detecting the imbalance load for intra-domain

In this section, the algorithm is described for imbalanced load among the domains. Firstly, we will check the threshold values of the switches in all the domains, secondly, for each domain lets say for domain1.com the load of S1 is 40 and S1 has completed with the 40 visits in the domain1.com and if 41 load visits the dashboard it will generate the overloaded switch of the domain. For this, we will compute the matrix Din*n and the distance between the various switches. Thirdly, the trigger factor set will be checked for different switches and then make a comparison with the threshold value. The trigger factors which cross the threshold value will be inserted into the new trigger factor set. The algorithm for the same has been shown in below-given table IV.

TABLE IV
DETECTING THE IMBALANCE LOAD FOR INTRADOMAIN

Input: SDN Network $G =< V, E >$
Output: Set Trigger Factor Set TRF_i for all domains
Begin:
for each domain_i $\Rightarrow domain\_n$
for each controller in domain_i
get $Ld_i^n(S_m)$ and $dist_{d_i}^n(S_m, S_n)$
construct matrix D_{in*n} and compute T
while$(D_{i(n*n)} = \phi)$
compute $\delta_i(mn)$
if$(\delta_i(mn) > T)$
detect load imbalance
Add $\delta_i(mn)$ to set TRF_i
end if
$D_{in*n} = D_{in*n} - dist_d(S_m, S_n), dist_d(S_n, S_m)$
end while
end for
end for

The complexity for the load imbalance detection is $O(n^2)$, where O is big O notation and n is the number of domains.

C. Layout of migration of switches

To balance a load of a particular domain fastly, we set the overloaded domain as the migrating domain. As per the distance between the inter domains and intra switches, we have calculated the trigger factor between all the domains. Therefore, we obtain the $d_i(Li) > d_i(Lj)$ and then set Li as the migrating switch. We can discover that migration switch and inter domains selection have a significant impact on load balancing rate and migration cost by examining and evaluating the scenarios in motivation as discussed in section 2. The algorithm for the migration of switches is described in table V and VI:

TABLE V
INTER DOMAIN SELECTION

for each domain $dist_{di} = n$
while (TRF$_i^n \phi$)
select $\delta_i(mn) fromTRF_i$
if $Ld_i(S_m) > Ld_i(S_n)$
Add $S_{mi} intoC_{mi}$
end if
$TRF_i = TRF_i - \delta_i(mn)$
end for

TABLE VI
INTER-SWITCH SELECTION

Procedure: Intra Switch Selection
Get migration efficiency T_i
Compute γ_{Sm} of each switch managed in D_i.
Migrating switches $S_{im} = argmax\gamma_{Sm}$
$count_i^n = 0$
for each domain D_i^n
for each switch S_i^n
if $Ld_i > T$
$count_i = count_i + 1$
end if
end for
for count = i to n
if $(count_i == 0)$
if $(count_{i+1}0)$
migrate $Ld_i fromD_i toD_{i+1}$
else
goto: Intra Switch Selection procedure
end if
end if
end for
end for

The complexity of the inter-switch selection is $O(n^2)$. Thus, in this section, we will focus on calculating the migration cost with migration request cost and cost of load change of domains. There are described terms used for the calculation of migration cost and LBR:

Description 1: Migration cost (MC): When the switch l_i is migrated from L1 to L2 or domain1.com to domain2.in then the cost of migration will be generated. Therefore, the migration of switches generates the cost of the network. The migration cost consists of two parts: a. migration request cost and b. cost of load change of domains.

1) **Migration request cost (MRC):** During the process of switch migration, firstly the switch will check the number of visits taking place. If the switch has crossed the load assigned to it then the switch will request the host and controller for the migration of load to another switch l_i. The cost taken to migrate will be termed as MRC.

$$\frac{1}{n}\sum_{i=1}^{n} t_i \tag{11}$$

When the switch l_i is migrated from D1.com to D2.in or D2.in to D3.org then the MRC will be computed as

eq 11, where t_i is the total load of the switch. In this, we are taking the fuzzy variable (x): X[0, 1], which indicates the connection between the switch l_i and domain L1. Therefore, we compute the value as U_i/t_i, based on the distance between the domains, where Ui is the used load and its value lies between 0 and 1. The formula for MRC is shown in eq 12:

$$MRC = (\frac{1}{n}\sum_{i=1}^{n} t_i) * \sum_{i=1}^{n} U_i/t_i \tag{12}$$

2) **Cost of load change of domains (LCD):** If the migration of the overloaded switch is accepted by the other switch or domain then the switch load will be managed by the host and the controller of that particular domain. This will result in the change of load in the domains and it is known as LCD. Let's assume that the load of the switch is similar to domains load and the path length from the switch to the domain is calculated in eq 13:

$$Numberofvisitsperswitch * Path_length$$

$$Path_length = d_i c_1 h_1 + d_i h_1 D_1 + d_i D_1 s_1 \tag{13}$$

Therefore, the total migration cost is computed in eq 14 as:

$$MC = MRC + LCD$$

$$MC = [(\frac{1}{n}\sum_{i=1}^{n} t_i) * \sum_{i=1}^{n} U_i/t_i] +$$
$$[Path_length = d_i c_1 h_1 + d_i h_1 D_1 + d_i D_1 s_1] \tag{14}$$

Description 2: LBR between the controllers before migration between host and domain

This describes the variance of load to evaluate LBR and the average load of the domain. Before migrating the switches the LBR is computed in eq 15 to 17 as:

$$\rho 1 = \frac{1}{n}\sum_{i=1}^{n}(l_i - \frac{1}{n}L1)^2 \tag{15}$$

$$\rho 2 = \frac{1}{n}\sum_{i=1}^{n}(l_i - \frac{1}{n}L2)^2 \tag{16}$$

$$\rho 3 = \frac{1}{n}\sum_{i=1}^{n}(l_i - \frac{1}{n}L3)^2 \tag{17}$$

Case 1: After migrating the switches between the two domains L1 and L2 is specified in eq 18:

$$\rho 12 = \frac{1}{n}\sum_{i=1}^{n}[L1^* + L2^*] \tag{18}$$

$$L1* = L1 – numberofvisits * Path_length$$

$$L2* = L2 + numberofvisits * Path_length$$

Case 2: In case we have three domains, after the migration of switches the average load of all the domains are specified in the eq 19: black

$$\rho 12 = \frac{1}{n}\sum_{i=1}^{n}[L1^* + L2^* + L3^*] \tag{19}$$

$$L1^* = L1 – numberofvisits * Path_{length}$$

$$L2^* = L2 – numberofvisits * Path_{length}$$

$$L3^* = L3 + numberofvisits * Path_{length}$$

black

IV. Simulation Environment

A. Validation and Simulation

In this section, we will compute the proposed scheme using the real-time environment. We have purchased one server and three domains namely qosalbsma.com, qosalbsma.in, qosalbsma.org and used PyCharm community as a coding platform in which each domain is defined with 4 switches; S1, S2, S3 and S4. The server is used to test the migration of switches. The code settings are done on the server. The configuration is done on the VPS server and vCPU having 2GB RAM, 50GB SSD Disk, 500GB bandwidth, SSL certificate and a dedicated IP (internet protocol). The load is handled by the cPanel WHM, CentOS 7.6 operating system compatible with Webuzo and cPanel. The VPS performance optimization is done with the resource monitoring dashboard having a CSF firewall installed and configuration to be optimized on the webserver and fine-tuning.

B. Setting QoS metrics

We have used the framework named **flask, CSS, PyCharm** and the libraries of the flask. To produce the load to be transferred and count the number of visits on the switches; a **pusher** is used for the simulations. The total load requests on domain1.com are 122KB/sec, domain2.in is 139KB/sec and domain3.org is 125KB/sec which is the maximum count of the load that the particular domain can handle. The throughput and response time of the scenario is said to be finite therefore, we define the number of switches to be managed by one domain is from S1 to S4.

black

C. Comparison between various schemes

To evaluate the improvement and effectiveness of the proposed scheme, we will make a comparison amongst the various schemes. The various schemes are showing the improvement of the proposed scheme with Round-Robin, Weighted Round-Robin, LBBSRT, Dynamic Server, GLBA and Server Load. The given table VII and Fig 5 shows the improvement of proposed over existing schemes which shows that the response time of the proposed scheme is 98% effective than the server load scheme, 93% improved from Weighted round-robin,

TABLE VII
TOTAL IMPROVEMENT OF THE PROPOSED SCHEME OVER EXISTING SCHEMES

	Server Load(%)	Weighted Round Robin(%)	Round Robin(%)	LBBSRT(%)	Dynamic Server(%)
Response Time	98	93	96	96	97
Throughput	45	17	23	26	3.6
CPU Utilization	14	25	31	31	34
Memory Utilization	10	3.1	12	14	12

96% from Round robin, 96% from LBBSRT and 97% from Dynamic server; throughput is 45% effective than server load, 17% from the weighted round-robin, 23% from the round-robin, 26% from LBBSRT, 3.6% from Dynamic server; CPU utilization is maximum achieved from Dynamic server with 34%; and memory utilization is achieved best with a minimum number of servers at 14% from LBBSRT. black

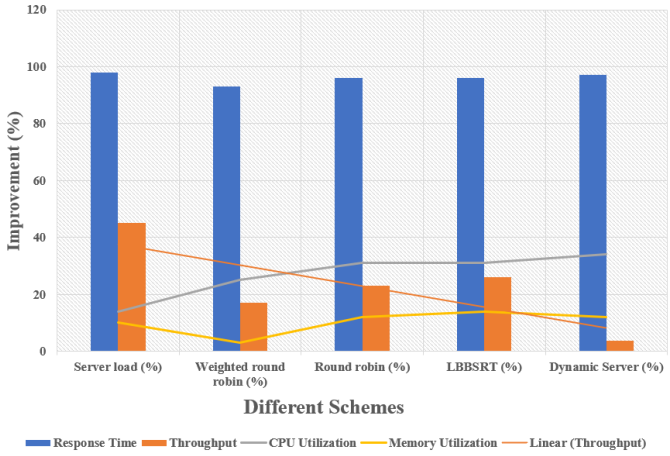

Fig. 5. Improvement of the proposed scheme from the existing schemes

D. Performance Evaluation of the proposed scheme based on QoS metrics

1) **Response time of a domain:** The minimum response time taken by the switches to migrate is one of the most important metrics to evaluate the performance. It is the amount of time to process the request by the user when the request is achieved by the client. Long delays can be avoided in the response time. To avoid long delays like jitter the load has to be balanced by minimizing the response time. The response time is calculated as: the difference between the time taken by the number of visits of a second switch and the number of visits of the first switch gives the response time. Fig 6 is the given graph showing the response time of the proposed scheme:
The time taken by the one visit of a switch is 2.56seconds and the time taken by the two visits of a switch is 2.5 seconds, therefore, the difference is 2.56 – 2.50 = 0.06 seconds.

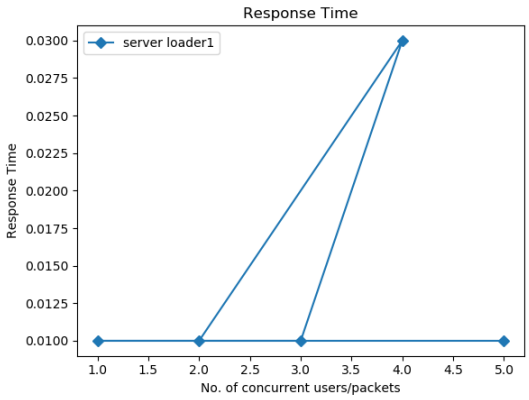

Fig. 6. Response Time

2) **Throughput of a domain:** The maximum throughput taken by the switches to migrate is one of the second important metrics to evaluate the performance. It is defined as the amount of load that can be delivered in a predefined frame. Maximum throughput can be achieved by balancing the load of a domain. In case one server is overloaded then the throughput will be affected. As compared to the existing schemes, the proposed scheme achieves the maximum throughput. The throughput is calculated as the time taken by the switch to visit the dashboard. It is shown in the below-given fig 7: The time taken by the 3 switches

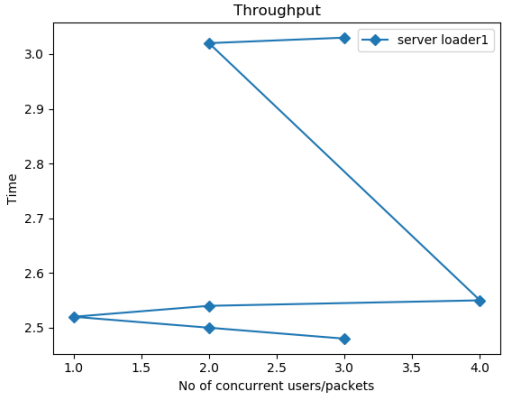

Fig. 7. Throughput

to be visited is 2.48 seconds, 2 switches take 2.50 seconds, 1 switch takes 2.52 seconds and so on.

3) **CPU utilization of a domain:** This is one of the essential factors to evaluate the load on the server. We have considered the utilization of CPU, QoS metric computes the server's load. As per the definition, QoS is used in computing the load on the server, CPU is not balanced so to balance the load researchers have to balance the load by minimizing the number of servers in the cPanel. The below-given fig 8 the utilization of CPU:

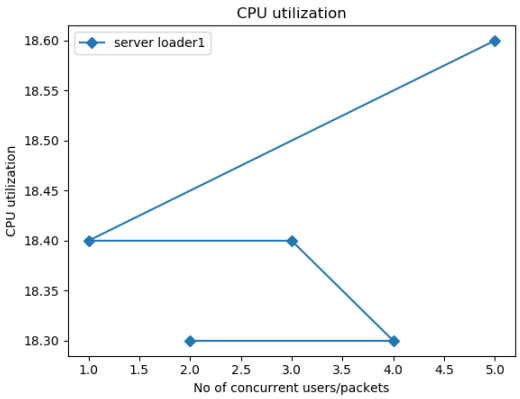

Fig. 8. CPU Utilization

If 1.0 visits are there on the server, the utilization of CPU is 18.40%; 2.0 visits, CPU utilization is 18.30%, 4.0 visits, CPU used is 18.30% and so on.

4) **Memory utilization of a domain:** This is again one of the important factors of QoS metrics for the switch migration. The load when utilization of memory takes place is unbalanced on the servers. SDN focuses on the overall maximization of the load to improve the overall utilization of memory. As the proposed scheme balances the load by maximizing the servers and due to which overall utilization of memory is achieved. The below-given fig 9 depicts the memory utilization as per the number of visits:
In case of 2.0 visits are there on the server, the memory utilization will be 13.7%; 4.0 visits on the server depict the memory utilization of 13.4% and so on.

V. Conclusion

In this paper, we have proposed the QoS-aware load balancing switch migration scheme to efficiently improve the throughput, response time, CPU utilization and memory utilization for balancing the load between the domains for communication in ITS. The main goal of proposed scheme is to show the switch migration to find the migration cost and LBR. In this, computations are done in the both ways: intra and inter domain where the weight of the switches is assigned in Kbps and the threshold value is fixed for each

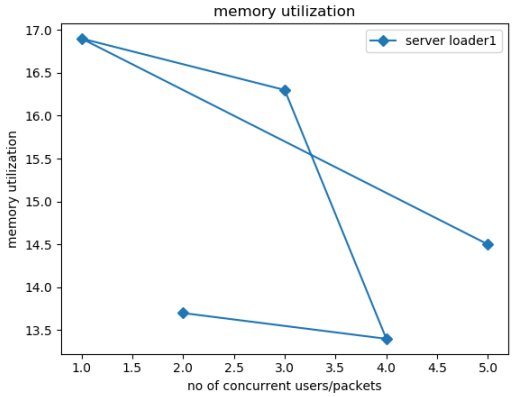

Fig. 9. Memory Utilization

switch and domain. Proposed approach has improved response time 2 times than server load, weighted round robin, round robin , LBBSRT and dynamic server; has improved throughput 1 time than server load, 0.3 times than weighted round robin, 0.4 times than round robin, 0.5 times than LBBSRT and 0.06 times than dynamic server; has improved CPU utilization 0.3 times than server load, 0.5 times than weight round robin, 0.6 times than round robin, LBBSRT and dynamic server; has improved memory utilization 0.2 times server load, 0.06 times than weighted round robin, 0.2 times than round robin, LBBSRT and dynamic server. These outcomes proves that proposed scheme is suitable for ITS communication even when load on the servers is very high. In future, SDN can be merged with artificial intelligence (AI) to improve the prediction of QoS metrics of the proposed scheme.

References

[1] Z. Kljaic, P. Skorput, and N. Amin, "The challenge of cellular cooperative ITS services based on 5G communications technology," *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2016 - Proceedings*, pp. 587–594, 2016.

[2] S. Manzoor, Z. Chen, Y. Gao, X. Hei, and W. Cheng, "Towards qos-aware load balancing for high density software defined wi-fi networks," *IEEE Access*, vol. 8, pp. 117 623–117 638, 2020.

[3] H. Babbar and S. Rani, "Emerging prospects and trends in software defined networking," *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 10, pp. 4236–4241, 2019.

[4] K. Dev, P. K. R. Maddikunta, T. R. Gadekallu, S. Bhattacharya, P. Hegde, and S. Singh, "Energy optimization for green communication in iot using harris hawks optimization," *IEEE Transactions on Green Communications and Networking*, 2022.

[5] W. Wang, M. H. Fida, Z. Lian, Z. Yin, Q.-V. Pham, T. R. Gadekallu, K. Dev, and C. Su, "Secure-enhanced federated learning for ai-empowered electric vehicle energy prediction," *IEEE Consumer Electronics Magazine*, 2021.

[6] H. Xue, K. T. Kim, and H. Y. Youn, "Dynamic load balancing of software-defined networking based on genetic-ant colony optimization," *Sensors (Switzerland)*, vol. 19, no. 2, 2019.

[7] S. Manzoor, H. Manzoor, M. Manzoor, and A. Mahmood, "A load balancing mechanism for a multi-controller software defined wifi network," in *International Conference on Artificial Intelligence and Security*. Springer, 2021, pp. 3–14.

[8] S. Singh and R. K. Jha, "A Survey on Software Defined Networking: Architecture for Next Generation Network," *Journal of Network and Systems Management*, vol. 25, no. 2, pp. 321–374, 2017.

[9] H. Babbar, S. Rani, M. Masud, S. Verma, D. Anand, and N. Jhanjhi, "Load Balancing Algorithm for Migrating Switches in Software-Defined Vehicular Networks," *Computers, Materials and Continua*, vol. 67, no. 1, pp. 1301–1316, 2021.
[10] H. Sufiev, Y. Haddad, L. Barenboim, and J. Soler, "Dynamic SDN Controller Load Balancing," pp. 1–21, 2019.
[11] H. Babbar and S. Rani, "Performance evaluation of QoS metrics in software defined networking using ryu controller," *IOP Conference Series: Materials Science and Engineering*, vol. 1022, no. 1, 2021.
[12] F. Camacho, C. Cárdenas, and D. Muñoz, "Emerging technologies and research challenges for intelligent transportation systems: 5G, HetNets, and SDN," *International Journal on Interactive Design and Manufacturing*, vol. 12, no. 1, pp. 327–335, 2018.
[13] M. A. Beiruti and Y. Ganjali, "Load Migration in Distributed SDN Controllers," *Proceedings of IEEE/IFIP Network Operations and Management Symposium 2020: Management in the Age of Softwarization and Artificial Intelligence, NOMS 2020*, 2020.
[14] S. Yeo, Y. Naing, T. Kim, and S. Oh, "Achieving balanced load distribution with reinforcement learning-based switch migration in distributed SDN controllers," *Electronics (Switzerland)*, vol. 10, no. 2, pp. 1–16, 2021.
[15] A. A. Abdelaziz, E. Ahmed, A. T. Fong, A. Gani, and M. Imran, "SDN-Based load balancing service for cloud servers," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 106–111, 2018.
[16] R. Sahbi, S. Ghanemi, and R. Djouani, "A Network Model for Internet of vehicles based on SDN and Cloud Computing," *Proceedings - 2018 International Conference on Wireless Networks and Mobile Communications, WINCOM 2018*, pp. 1–4, 2019.
[17] L. Yao, T. Hu, and J. Lan, "Sequenced Switch migration for balancing controller loads in large-scale software-defined networking," *International Journal of Performability Engineering*, vol. 14, no. 8, pp. 1848–1856, 2018.
[18] A. Banerjee and D. M. Hussain, "EXPRL: experience and prediction based load balancing strategy for multi-controller software defined networks," *International Journal of Information Technology (Singapore)*, 2020. [Online]. Available: https://doi.org/10.1007/s41870-019-00408-5
[19] F. Al-Tam and N. Correia, "On load balancing via switch migration in software-defined networking," *IEEE Access*, vol. 7, pp. 95 998–96 010, 2019.
[20] S. Zhang, J. Lan, P. Sun, and Y. Jiang, "Online load balancing for distributed control plane in software-defined data center network," *IEEE access*, vol. 6, pp. 18 184–18 191, 2018.
[21] J. Cui, Q. Lu, H. Zhong, M. Tian, and L. Liu, "A load-balancing mechanism for distributed sdn control plane using response time," *IEEE transactions on network and service management*, vol. 15, no. 4, pp. 1197–1206, 2018.
[22] A. Pietrabissa, L. R. Celsi, F. Cimorelli, V. Suraci, F. D. Priscoli, A. Di Giorgio, A. Giuseppi, and S. Monaco, "Lyapunov-based design of a distributed wardrop load-balancing algorithm with application to software-defined networking," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 5, pp. 1924–1936, 2018.
[23] F. Al-Tam and N. Correia, "Fractional switch migration in multi-controller software-defined networking," *Computer Networks*, vol. 157, pp. 1–10, 2019.
[24] A. Banerjee and D. A. Hussain, "Exprl: experience and prediction based load balancing strategy for multi-controller software defined networks," *International Journal of Information Technology*, pp. 1–15, 2020.
[25] M. A. Beiruti and Y. Ganjali, "Load migration in distributed sdn controllers," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020, pp. 1–9.
[26] R. D. Devapriya and S. I. Gandhi, "Enhanced Load Balancing and QoS Provisioning Algorithm for a Software Defined Network," *International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE 2020*, pp. 1–5, 2020.
[27] D. Liu, Y. Zhang, W. Wang, K. Dev, and S. A. Khowaja, "Flexible data integrity checking with original data recovery in iot-enabled maritime transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
[28] A. A. Neghabi, N. J. Navimipour, M. Hosseinzadeh, and A. Rezaee, "Load balancing mechanisms in the software defined networks: a systematic and comprehensive review of the literature," *IEEE Access*, vol. 6, pp. 14 159–14 178, 2018.
[29] Y. Zhou, Y. Wang, J. Yu, J. Ba, and S. Zhang, "Load balancing for multiple controllers in SDN based on switches group," *19th Asia-Pacific Network Operations and Management Symposium: Managing a World of Things, APNOMS 2017*, pp. 227–230, 2017.
[30] P. V. A. V. Atienza and W. E. S. Yu, "A framework for performance analysis of various load balancing techniques in a software-defined networking environment," *Lecture Notes in Electrical Engineering*, vol. 514, pp. 67–74, 2019.
[31] S. Manzoor, K. B. Bajwa, M. Sajid, H. Manzoor, M. Manzoor, N. Ali, and M. I. Menhas, "Modeling of wireless traffic load in next generation wireless networks," *Mathematical Problems in Engineering*, vol. 2021, 2021.

Himanshi Babbar is Assistant Professor- Research in CSE, working in Chitkara University, Rajpura, Punjab, India. She has 2 years of teaching experience, CGC, Landran, Mohali, Punjab. She received an MCA (Master's in Computer Applications) degree from Chitkara University, Punjab Campus in 2015 and completed her Ph.D. in Computer Applications from Chitkara University, Punjab Campus in 2021. She is pursuing PostDoctoral Fellowship from UAE since 2021.

Shalli Rani is Associate Professor in CSE with Chitkara University (Rajpura (Punjab)), India. She has 14+ years of teaching experience. She received an MCA degree from Maharishi Dayanand University, Rohtak in 2004 and the M.Tech. degree in Computer Science from Janardan Rai Nagar Vidyapeeth University, Udaipur in 2007 and a Ph.D. degree in Computer Applications from Punjab Technical University, Jalandhar in 2017. Her main area of interest and research is Wireless Sensor Networks, Underwater Sensor networks and Internet of Things.

Ali Kashif Bashir received the B.Sc. degree (Hons.) in computer forensics and security from the Department of Computing and Mathematics, Manchester Metropolitan University, U.K., and the Ph.D. degree in computer science and engineering from Korea University, South Korea. His past assignments include an Associate Professor of ICT with the University of the Faroe Islands, Denmark; Osaka University, Japan; Nara College, National Institute of Technology, Japan; the National Fusion Research Institute, South Korea; Southern Power Company Ltd., South Korea; and Seoul Metropolitan Government, South Korea

Raheel Nawaz currently the Director of Digital Technology Solutions and a Reader in analytics and digital education with Manchester Metropolitan University (MMU). He has founded and/or headed several research units specializing in artificial intelligence, data science, digital transformations, digital education, and apprenticeships in higher education. He has led on numerous funded research projects in U.K., EU, South Asia, and Middle East. He has held adjunct or honorary positions with several research, higher education, and policy organizations, both in U.K., and overseas.