smart cities

MDPI

Article

A New SDN-Based Routing Protocol for Improving Delay in Smart City Environments

Lamia EL-Garoui *, Samuel Pierre and Steven Chamberland

Department of Computer and Software Engineering, Ecole Polytechnique Montreal, Chemin de Polytechnique, Montreal, QC 2500, Canada; samuel.pierre@polymtl.ca (S.P.); steven.chamberland@polymtl.ca (S.C.)

* Correspondence: lamia.el-garoui@polymtl.ca

Received: 17 August 2020; Accepted: 6 September 2020; Published: 9 September 2020

Abstract: The smart city is an ecosystem that interconnects various devices like sensors, actuators, mobiles, and vehicles. The intelligent and connected transportation system (ICTS) is an essential part of this ecosystem that provides new real-time applications. The emerging applications are based on Internet-of-Things (IoT) technologies, which bring out new challenges, such as heterogeneity and scalability, and they require innovative communication solutions. The existing routing protocols cannot achieve these requirements due to the surrounding knowledge supported by individual nodes and their neighbors, displaying partial visibility of the network. However, the issue grew ever more arduous to conceive routing protocols to satisfy the ever-changing network requirements due to its dynamic topology and its heterogeneity. Software-Defined Networking (SDN) offers the latest view of the entire network and the control of the network based on the application's specifications. Nonetheless, one of the main problems that arise when using SDN is minimizing the transmission delay between ubiquitous nodes. In order to meet this constraint, a well-attended and realistic alternative is to adopt the Machine Learning (ML) algorithms as prediction solutions. In this paper, we propose a new routing protocol based on SDN and Naive Bayes solution to improve the delay. Simulation results show that our routing scheme outperforms the comparative ones in terms of end-to-end delay and packet delivery ratio.

Keywords: smart city; routing protocol; SDN; Naive Bayes; IoT

1. Introduction

IoT and 5G technologies promise a massively improved scalable and reliable connectivity. These technologies bring various innovative applications that offer high-quality services and efficient management of infrastructure in urban areas. The development of these technologies turns expectations of smart cities into reality. Their vision is to provide a better life quality for citizens and to enhance city management to react quickly to events and emergencies [1].

A smart city uses different devices, such as sensors, actuators, vehicles, and cameras for its various operations. The devices' deployment and operating compose a diversity of access networks that process automating data collection, provide necessary information used to manage infrastructures and resources efficiently, and meet applications' advanced requirements. This collection is analyzed and exploited in various systems, such as transportation, energy, education, and health systems [2]. With a common goal of enabling these systems to be intelligent, many applications have stringent real-time requirements in exchanging information between apparatus and everything around them.

The smart transportation system is mainly based on delay-sensitive applications. However, the high mobility and dynamic topology of the vehicle networks diminish data dissemination and bring more challenges in implementing communication protocols that guarantee efficient and reliable transmission [3], and resource management systems [4]. Moreover, the vehicle networks' heterogeneity

and scalability increase the complexity in designing new communication solutions and managing multifaceted technologies. This elevates the demand of create new communication initiatives, more specifically in developing routing protocols. Despite the significant number of existing routing protocols in the literature, we note that current routing protocols are relatively limited and cannot resolve a smart environment's new issues. Furthermore, the existing routing protocols are inherently distributed, since each node communicates with others by broadcasting information in its vicinity. They reflect partial visibility of the network, and they cannot know the real network conditions. There are still needs for novel approaches taken by the researchers in conceiving the original solving and considering significant heterogeneity challenges, scalability and new application requirements.

A programmable and dynamic solution named Software-Defined Networking (SDN) was proposed to reverse this situation, where new protocols can be implemented by software without needing any hardware modification on networks. The SDN separates the control and data planes to guarantee flexibility. The data plane includes the devices that forward the information. The control plane comprises a controller that generates and maintains a database containing network-wide knowledge such as network topology. The main aim of SDN is to improve rapid packet delivery, low latency, and overhead communication, best bandwidth utilization and selection of best routes, as well as to facilitate network programming.

Recent trends in smart cities reveal that SDN has been used in wired networks [5,6], making it unusable in the general case of an intelligent environment where the devices are ubiquitous and distributed. Therefore, SDN must be adapted to support the wireless nature of smart environment systems [7,8]. The intelligent environment complexity and SDN centralized management make the computation of routing solutions under various requirements of real-time applications even more difficult. However, Machine Learning (ML) algorithms afford a new possibility in solving such problems, which keeps computation more reasonable and manageable [9]. In this perspective, we explore using a ML algorithm and SDN technologies in proposing a new routing protocol to reduce end to end delay. We use the useful and practical machine learning technique to discover and extract knowledge for predicting the routing paths calculation. The main contribution of this paper is as follows:

- Develop a framework that uses the Naive Bayes algorithm. This supervised ML algorithm expects a dataset as input and delivers an output result, which is useful information in routing path construction.
- Create a large dataset based on the Montreal city open data website and using the Simulation of Urban MObility (SUMO) simulator and the calibrator framework. Then, process the transformation and label of the different features used in the Naive Bayes framework to predict nodes' information.
- Design and develop a routing protocol as an application in the SDN controller 'RYU' with integrating ML framework.
- Implement and evaluate the performance of our solution, then compare it to the literature routing solutions, Optimized Link State Routing Protocol (OSLR), SDN multipath routing and SDN Q-learning routing.

The remainder of the paper is organized as follows. Section 2 introduces the background knowledge of SDN and an overview of the related work. Section 3 gives a brief explanation of ML techniques and describes the generation and processing of the created dataset. Sections 4 and 5 present our network model and operation of our Naive Bayes based routing protocol. Performance evaluation is carried out in Section 6. The last section concludes.

2. Background and Related Work

In this section, we first introduce a brief background knowledge of SDN; secondly, we investigate recent work done about routing on SDN; and thirdly, we expose the Machine Learning-based routing solutions.

2.1. Background Knowledge of SDN

Despite the new technological progress achieved during the last decade, the SDN paradigm is still flourishing. It attracts more and more attention from the academic community to promote innovation works and improve the network communication protocols. We give a brief explanation of this emerged concept bellow. Mainly, SDN's idea is to decouple network control functions from forwarding data functions, which corresponds to the new network architecture [10,11]. The SDN architecture consists of three planes: the application plane, the control plane, and the data plane [12], as shown in Figure 1. There are SDN control protocols between every two planes, i.e., ones between application plane and control plane named Northbound protocols and others between data plane and control plane called Southbound protocols. The northbound side provides a program that manages the network from the application plane. Besides the northbound protocols, the southbound interfaces enable communication from the control plane to the data plane, which of these we cite the popular ones, including OpenFlow, OVSDB, NETCONF, etc. Nevertheless, the control plane is based on its operation on central equipment called a controller representing the brain of the network [13]. The controller enables the dynamically programming of the net to ensure efficient and flexible service of the physical and virtual hardware in the data plan. Moreover, the controller addresses the essential functionalities of the network used by data plane components. It provides rules from the application plane, such as routing, load balance of resources usage, network topology monitoring, and device configuration [14]. Several types of controllers are developed in the industry to ensure better network management with more flexibility and agile. We quote as an example Ryu controller, Onos controller, Opendaylight controller, etc. [13].

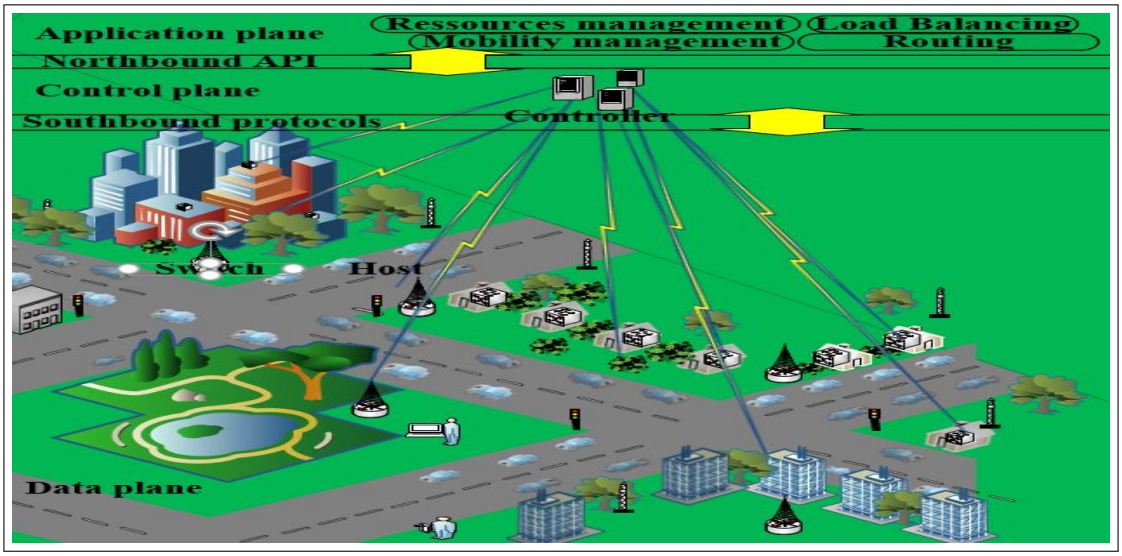

Figure 1. SDN architecture in Smart City Environment.

2.2. SDN-Based Routing Algorithms

Since routing is the fundamental key in any communication network, several prior works have been designed for a wide selection of systems and applications with various challenges, such as minimizing delay transmission, maximizing packets delivery, enhancing reliability, etc. According to the listed works below, the structure of SDN provides a global view of the network, which eases computing optimal routing paths and guarantee efficient transmission. Besides, this section highlights the new routing contributions using the SDN paradigm.

Venkatraman et al. [15] proposed an SDN-enabled connectivity-aware geographic-aware routing protocol (SCGRP), in which the SDN controller calculates routing measurements in real-time using cellular networks. SCGRP performance scores an enhancement in link deterioration, transmission delay, and packet delivery ratio. Moreover, SCGRP optimizes the transmission of data packets using an extended SDN architecture to vehicular networks in urban areas. The proposed solution aims to take

advantage of a global view of the network topology and heterogeneous traffic conditions maintained by SDN.

Sudheera et al. [16] suggested a new routing protocol for Software-Defined Vehicular Networking (SDVN) that optimizes the routing scheme with a source routing based flow instantiation (FI) operation. The routing protocol analyzes the link quality fluctuations due to dynamic topology that causes short lifetime links and then packet losses. Based on an incremental packet allocation scheme, the solution solves the routing problem in a less time complexity by finding multiple shortest paths. The built routes are collectively stable, which guarantees delivery of packets without any loss.

Singh et al. [17] designed a new approach by combining the SDN with the small-cell deployments. They implemented the Multipath TCP connections to establish link reliability between vehicles, increase throughput, and reduce packet loss. Similarly, Zhu et al. [18] considered using SDN in a Vehicular Ad hoc NETwork (VANET) setting as the best key to ensure a routing solution's efficiency. The SDN controller builds better paths based on a global view over the network. They used a new metric named "minimum optimistic time" to switch between multi-hop forwarding and carry-and-forward models according to the network density. The routing protocol aims to decrease delivery delay time and overhead.

Other work introduced a new approach to resolve the routing problem [19]. Abbas et al. [19] put forward the routing solution that used a road-aware approach by selecting paths using road segment id to ensure path durable. The SDN controller computes path estimation model using speed, position, and road id information for finding the shortest routes. The data is gathered by a dedicated controller, i.e., the edge controller, which processes the real-time data coming from the vehicles. The usage of this controller aims to reduce the response time and packet overhead. The cellular network is employed to transmit packets between the SDN controller and vehicles with less bandwidth and low latency.

Abugabah et al. [20] proposed a new method to build multiple paths and to simplify the traffic construction procedure. The protocol used a new criterion based on average path channel load, channel loading, path length, and channel length to select a path from the available routes. The solution employs a modified channel state wave algorithm in path selection that allows a more uniform capacity of information transmission. The protocol manages the shortest paths between the intermediate and final nodes to weed out the recalculation of new routes in the transmission process.

Consequently, the solutions presented in this section are based on the broadcasting approach to discover the destination if the controller did not know it beforehand. This target discovery will generate an additional delay in building routes and data transmission. As well, The roads' construction uses information collected by dedicated equipment, which presents extra processing time to accomplish the routing task. Besides, the suggested protocols used the cellular network in exchanging the packets between the data plane and the control plane, which represents a limited option given the exponential growth of the equipment deployed in future cities, particularly with the scope of the IoT. The wireless devices spread will lead to a scarcity of resources in terms of medium allocation. As well, the arrival of 5G will overthrow the current cellular networks, with the deployment of innovative decentralized architectures based on hybrid clouds and supporting IoT applications. Such advancement will revolutionize existing solutions by requiring their update to provide a high quality of service (QoS) for users, and high quality of experience (QoE) [21]. In contrast, our protocol anticipates the destination discovery step and avoid it by using a prediction mechanism to reduce the transmission delay. Besides, we are adopting the WiFi 802.11P technology in network communication, as this type of technology is most suitable in high mobility communications.

2.3. Machine Learning-Based Routing Algorithms

Currently, Machine Learning is a widely used Artificial Intelligence (AI) method for a broad assortment of tasks, like classification, in a variety of application spheres such as computer vision, security and network communications. In this section, we present the relevant research works designed for routing strategies using ML algorithms.

Tang et al. [22] proposed a mobility prediction-based routing scheme to ensure reliable and timely data dissemination in VANETs. The SDN controller predicts the vehicle arrivals and connections using the artificial neural network. Road Side Unit (RSU)/Base Station (BS) uses the prediction to integrate a stochastic urban traffic model for estimating the successful transmission and average delay. Depending on the location of the source vehicle and destination vehicle, the controller or the RSU/BS built routing paths with minimum delay. Vehicle-to-Infrastructure (V2I), as well as Vehicle-to-Vehicle (V2V) channels, can transmit simultaneously via different spectrum access technologies. On the other hand, Azzouni et al. [23] suggested a new routing solution called neural network-based routing (NeuRoute). The routing protocol uses Deep Feed Forward Neural Network (DFFNN) to learn and build routes. It inputs collected path decisions history, the network state, and the predicted traffic matrix to train step of neural routing network. The network state consists of all links costs and their available capacities. After training the model, the protocol can route new arriving flows based on the traffic matrix and the network state as an input.

Chen and Zheng [24] suggested a new routing solution entitled A Machine Learning-based routing preplan for SDN. This solution works on three phases: flow feature extraction, user requirement prediction, and route selection. As indicated in the solution's name, the process considers planning routing based on relevant features extracted from users' history data and clustered using a semisupervised clustering algorithm. The core idea is to predict the user's requirements, then plan frontwards routing policies that ensure reducing delay. The solution employs the extreme learning method to optimize the network structure and discards old data post the training. Li et al. [25] incorporated a Naive Bayes classifier in the Least Loaded (LL) routing algorithm. The ML algorithm provides extra information obtained from record historical network state information and the link traffic load information to LL routing that decides the best candidate route. Another usage case was proposed by Baz [26], which integrated the Bayesian Machine Learning algorithm in Flow Prediction in SDN Switches. The algorithm allows switches to predict the traffic to reduce unnecessary communications between switches and the controller. Therefore, the switches determine processes generating flows and use them to assign a packet with the unknown flow to match it with the appropriate one.

To resolve the Routing and Wavelength Assignment (RWA) problem in the context of optical Wavelength-Division Multiplexing (WDM) networks, Martin et al. [27] transformed this issue into an ML-based classification problem where ML classifier provides RWA solution based on an input traffic matrix. Moreover, Sun et al. [28] introduced the TIme-relevant DEep reinforcement learning (TIDE) protocol as an optimized routing strategy on SDN modified architecture that contained three logic planes: data plane, control plane, and AI plane. In the AI plane, an RNN-based deep reinforcement learning method operates to output a near-optimal routing strategy according to the varying traffic distribution. However, a smart agent can output a near-optimal routing solution for the underlying network without any prior knowledge. Yao et al. [29] proposed a load balancing routing scheme boosted by ML. The protocol uses network state information (NSI) in the form of queue length to train the neural network and to make route predictions. In another type of networks, Liu et al. [30] introduced a Q-learning based multi-objective Unmanned Aerial Vehicles (UAV) routing algorithm to reduce energy consumption and delay. Moreover, Zhang et al. [31] presented a range of Deep Learning algorithms used in routing solutions.

However, the "No Free Lunch" theorem [32] states that there is not just one ML algorithm that can solve any problem, and that always performs better than all other algorithms because it depended on many factors. Furthermore, several factors are involved in choosing the ML algorithm that provides a given situation's proper learning model. We mention the main ones which are the dataset dimension and structure, the type and complexity of the problems, the prediction speed, the precision of the forecasts, the duration of the training step, the size of trained data, the straightforward implementation, and the storage space of training data. Indeed, the works presented in this section opt for various ML solutions, except that these solutions do not meet the real-time requirements of the routing problem in the intelligent environment. Because they take a long time to compute the prediction, and they

require massive storage resources due to the network's dynamic scalability. Whereas The Naive Bayes algorithm provides fast training and prediction speed, which meets real-time operating, it performs well with small and big instances; it can handle lots of features and easy to implement. Moreover, it can provide high accuracy predictions using different types of datasets [33,34]. Thus, this candidate algorithm is a suitable approach that fulfills our expectations to resolve the smart cities' routing issue.

3. Machine Learning Algorithms

The scalability of the network and the heterogeneity of equipment and the full range applications deployed in smart cities make the SDN controller tasks complicated and tedious. Therefore, network management and data routing mechanisms produce a large number of overhead packets and long transmission delays, thus disrupting the systems' functionalities based on real-time applications.

ML algorithms are introduced in SDN to overcome these challenges [35]. This provides robust computation and optimized processing to predict essential information in the routing paths construction or the network topology discovery either in the allocation resources. Mainly, the advantage of using Machine Learning techniques consists of reducing transmission delay and control packets significantly in routing applications. ML algorithms rely on real-time and historical network data to operate. They analyze them and then estimate and predict a future situation or outcome that can bring intelligence to the SDN controller. The controller will be able to optimize network configuration and automatically deliver new network services in real-time.

ML techniques are divided into four approaches: supervised, unsupervised, semi-supervised, and reinforcement learning. Supervised learning is based on pairs of input–output observations to learn a function explaining the relationship between them by looking for patterns in the value labels assigned to the data points [36]. This algorithm predicts an outcome variable given a set of input variables, and it creates a system model that matches inputs to target outputs. Once the training process has reached the level of precision, the model can be used to achieve the desired result. There are various models of supervised learning, such as neural networks, regression, decision tree, nearest k-neighbor, logistic regression, support vector machine, and Naive Bayes, etc. However, our work focuses on the Naive Bayes supervised learning approach that we investigate in this section.

The concept of Bayesian machine learning depends on the conditional probability of calculating the probability that an instance will occur given a particular set of attributes corresponding to previous knowledge, which might be related to it. This learning method classifies a new event or incident into a set of possible outcomes provided by previous training data. The classifier Naive Bayes tries to find the most likely value for a current instance given its known attributes. Bayesian learning uses the most probable result, often called the maximum a posteriori to classify the hypothesis. It is based on the Bayesian theorem defined in Equation (1) as follows [37]:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \tag{1}$$

where $P(C)$ is the prior probability of hypothesis C, $P(X)$ is the prior probability of training data X, $P(X|C)$ probability of X given C, and $P(C|X)$ probability of C given X.

In the next section, we explain the creation and transformation of the dataset used in our ML framework, and we introduce the proposed model and formulation problem.

3.1. Problem Formulation and Proposed Modeling

Our routing application tends to reduce transmission delay and overhead packets in an intelligent transport environment using SDN to manage vehicle communication in smart cities efficiently. We integrate the supervised approach given by the Naive Bayes classifier in the routing solution. We use the Naive Bayes classifier giving consideration to its simple implementation with low computational complexity and a real-time prediction process. This ML algorithm predicts locating a vehicle in a specific area of the road network assigned to a particular RSU assuming the role of a virtual switch in

SDN. Therefore, the controller will not broadcast packets-out to discover the vehicle's location, which will reduce the control packets exchanged between the controller and the various switches/RSU in the network. Localization of the node allows establishing a particular candidate route between a pair of nodes, depending on the current link delay. Based on the least current delay on each candidate route, we select the one with the best value, i.e., the lowest delay.

X $[x_1,\ldots, x_n]$ vector indicates a problem instance where $x_1,\ldots, x_n$ are n characteristics related to the vehicle information defined and discussed in the next section. We assume that there is k class, $C_1,\ldots, C_k$ corresponding to areas including RSU/ Switch S_j with $j = 1..k$. Applying the Naive Bayes theorem defined in Equation (1) to our problem, allows us to define the conditional probability of our model, represented by $P(C_k \mid \boldsymbol{X})$, which denotes the instance probability of C_k given X, in the Equation (2) as follows:

$$P(C_k \mid \boldsymbol{X}) = P(C_k \mid x_1,\ldots, x_n) = \frac{P(C_k) \cdot P(x_1,\ldots, x_n \mid C_k)}{P(x_1,\ldots, x_n)} \tag{2}$$

According to the conditional independence hypothesis of the theorem, i.e., the features are mutually conditionally independent, then the posterior probability for each class becomes:

$$P(C_k \mid \boldsymbol{X}) = \frac{P(C_k) \cdot \prod_{j=1}^{n} P(x_j \mid C_k)}{\prod_{j=1}^{n} P(x_j)} \tag{3}$$

Before executing the proposed approach in a real-time process, it should perform a training step in an offline manner by examining the set of data in the learning stage. In the first step, the model clams a dataset to learn the function that maps the future events with past knowledge occurrences. The creation and implementation of such a dataset are developed in the following section.

3.2. DataSet Generation and Processing

To collect the vehicle data, we simulate the traffic in a real-life scenario on a testbed built using the SUMO simulator and a real map of downtown Montreal extracted via the OpenStreetMap framework [38–40].

The most important information that helps us in the creation of the dataset is the configuration of the road network with the identification of roads and trails, traffic infrastructure (e.g., traffic lights) and traffic demand discussed in [39], which provides the number of vehicles present at each intersection of the road network, as shown in Figure 2. We simulate the scenario several times and calculate the statistical data using the Traci and Calibrator package provided by SUMO [41].

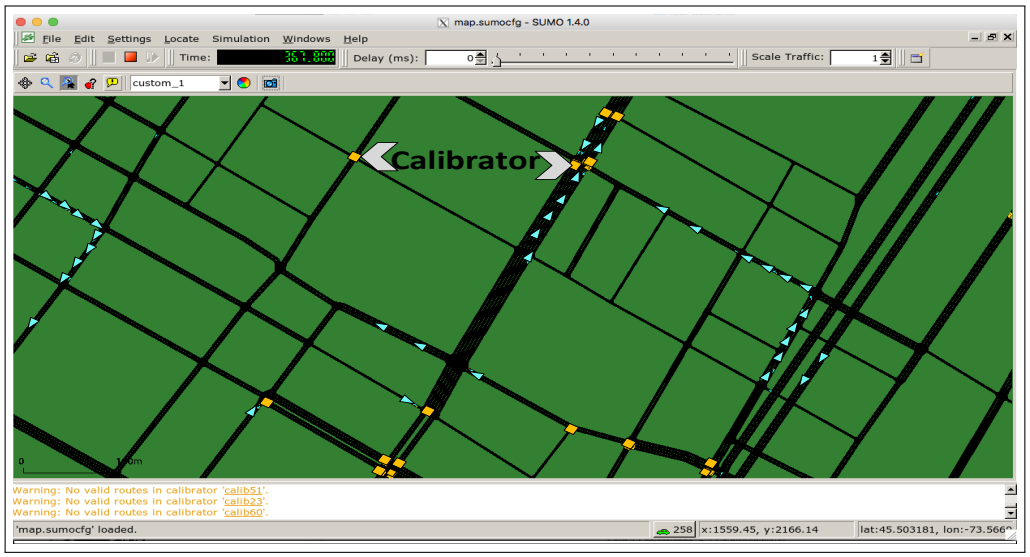

Figure 2. Montreal Real Scenario Testbed built using Sumo and OpenStreetMap.

As indicated in [41], we count the number of vehicles passed over a road segment by implementing a calibrator object at each intersection. The calibrator controls and calculates crossing vehicles at the junction. If the computed value reaches a specified amount by Montreal's openData portal, the calibrator framework removes the fleet exceeding the specified target level. Besides, if the traffic request does not match the specified value, the calibrator inserts new vehicles.

We save the data simulation in CSV file to clean and process the dataset, as shown in Figure 3, then we extract the necessary features used in our solution. We label each column of the CSV file, and we remove non-valid values, then we erase columns to reduce unnecessary data. We label the columns to identify the features by: node_id: Vehicle_id, TT: travel_time, RSTh: historical road segment travel, CT: current time, RT: trajectory time on the segment of road, TD: trajectory direction, ST: speed travel, XP: X-axis position, YP: Y-axis position, AN: Area number and RSU-Addr: roadside unit address. It should be noted that the Naive Bayes algorithm is based on strong (naive) independence assumptions among the various features. For this reason, we review and assess each feature pair's correlation matrix, including Pearson's correlation coefficients, as illustrated in Figure 4. The matrix correlation shows the level of dependency between the attributes to keep only the most independent.

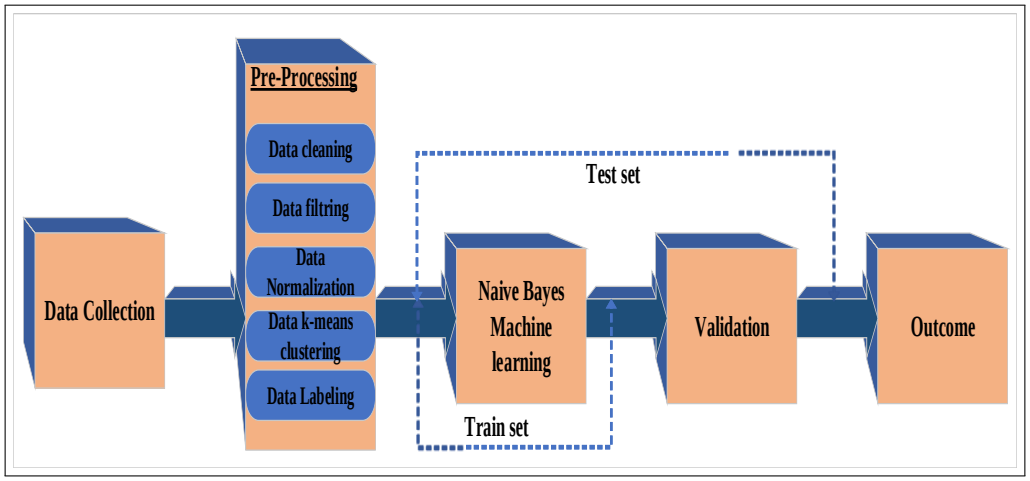

Figure 3. Dataset processing and Model testing stages [42].

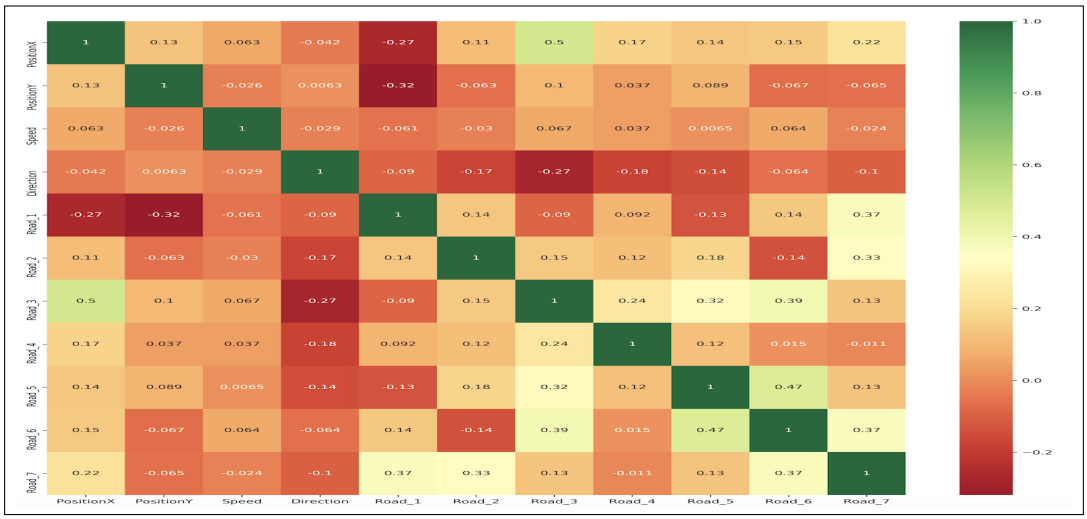

Figure 4. Matrix correlation reporting Pearson's correlation coefficients of each pairs of Features Dataset.

Pearson's correlation coefficients show mainly that the features are closer to zero, which means that no linear relationship is present in the data. Besides, the negative values indicate that one variable tends to decrease as the other increases. On the other hand, the positive coefficients highlight that as one feature value increases, the other tends to increase. By the end of dataset processing that contains more than 1,000,000 instances, we test the ML model to evaluate its accuracy by varying the number of occurrences fed to the train part and validation part. The obtained results in Table 1 help us choosing the percentage of splitting the dataset. We observe that the accuracy of the model is advantageous and is over 90% for all dataset size variation. The efficiency for 70%–30% splitting and 80%–20% are close, but the third column is suitable for the choice of dividing the dataset while it shows the best accuracy results.

Table 1. Accuracy evaluation and splitting dataset ratio selection.

Dataset Size	50% Train Stage 50% Validation Stage	60% Train Stage 40% Validation Stage	70% Train Stage 30% Validation Stage	80% Train Stage 20% Validation Stage
10,000	0.9164	0.9102	0.9056	0.901
50,000	0.9077	0.9083	0.9092	0.9078
150,000	0.9142	0.9147	0.9164	0.9179
250,000	0.9162	0.9171	0.9181	0.9170
500,000	0.9147	0.9137	0.9136	0.9128
1,000,000	0.9298	0.9296	0.9301	0.9303

The ML algorithm works is a very dynamic environment, which induces a high variance of the data, and herby, after some time, the model's performance may deteriorate. Therefore, the ML algorithm must be retrained. Still, excessive retraining generates more computation time and adds complexity to the model implementation by including new modules such as that monitors the performance of the system. While, at each time a new routing request is received, the controller retrained the Naive Bayes method to maintain its efficiency; thus, it must use fewer resources.

4. Network Model

This section describes the proposed network model used to design the SDN-Naive Bayes-based routing protocol in a smart city environment. Mainly, our work focuses on intelligent transportation systems that represent the lifeblood of smart cities. The network and their operation components take advantage of the SDN paradigm and ML to offer efficient communication and aid to bring more

flexibility and agile to the system. Our network is composed of three types of equipment. The following components are required for deploying the system.

- Controller. It plays a central role in the network as the heart and concretize the control plane. The controller has the global intelligence provided by the overview of the whole system and knowledge of all devices in the network. This equipment controls all the network devices. In this paper, we choose the Ryu controller for its suitability to integrate a large selection of applications. Ryu controller development matches very well with ML programming API, since they use the same language, i.e., Python.
- Roadside units (RSU) considered virtual switches. This material is the relay between the controller and the host nodes. They transmit all the rules and modifications in the network.
- Vehicles. They are the fundamental part of the network and represent SDN Wireless Nodes. The cars communicate actively with their neighbourhood and act as the end-users and forwarding element. They are the first source of data in smart transportation systems. The big challenge with these devices is their high mobility, frequent arrival and departure in the network.

The RSU and vehicles constitute the data plane. They operate with the OpenFlow protocol to communicate with the controller. The proposed network model works in a heterogeneous network environment in which the forwarding infrastructure uses wireless and wired technologies to deliver information; the WiFi 802.11P to connect a vehicle to RSU infrastructure (V2I), and the optical technology to connect the RSU to the controller. Figure 5 highlights the network components and used communication technologies. As displayed, the communication V2V and V2I employ wireless connection. Each vehicle is equipped with the WiFi 802.11P interface to exchange messages with RSU in mesh connection. However, the RSU is equipped with two interfaces, one dedicated to wireless communication to transmit information to vehicles. The second interface used to communicate with the controller to forward the rules to the hosts. In the next section, we describe the suggested routing algorithm assisted by the supervised Naive Bayes classifier to reduce delay transmission and network overhead.

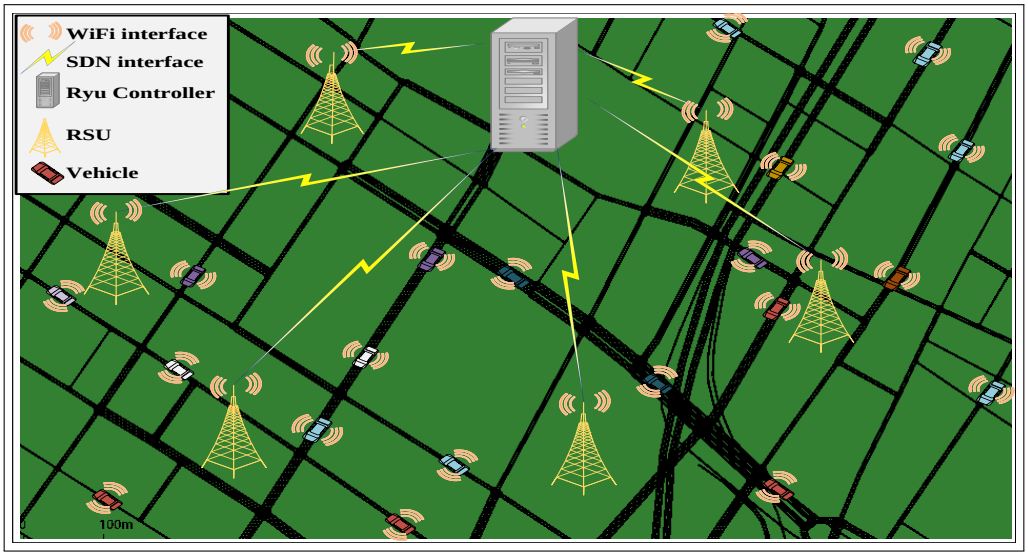

Figure 5. Network Model.

5. SDN-Naive Bayes-Based Routing Protocol Working

Considering the previously depicted network model, in this section, we elaborate on the evoked solution's operation steps. The controller performs all the work, from discovering the elements and their needs in the network until the control and maintenance of the communication, to ensure the system's optimized operation. To understand the working of the new routing protocol, we explain how to achieve a communication attempt between two vehicles, as shown in Figure 6.

Let's assume that the car noted Source Host wants to send a data to the vehicle named Destination Host. Therefore, the Source Host looks for the Destination Host in its flow table. If the node did not find the wanted entry, it sends a request, represented by arrow 1 in Figure 6, to the RSU relied on the area of node's location. In our case, the vehicle asks RSU1 that covered the Area_1. Since there is no previously installed rule for this flow, the RSU1 forwards the request to the controller by sending a packet-in (arrow 2) using OpenFlow protocol. Ones the packet arrived at the Ryu controller, the controller identifies the origin of the message by extracting source address and check, if there is a previous request of this node, which means its presence in the table of flows. Next, RYU responds by packet modification to update the node's flow table (arrow 3).

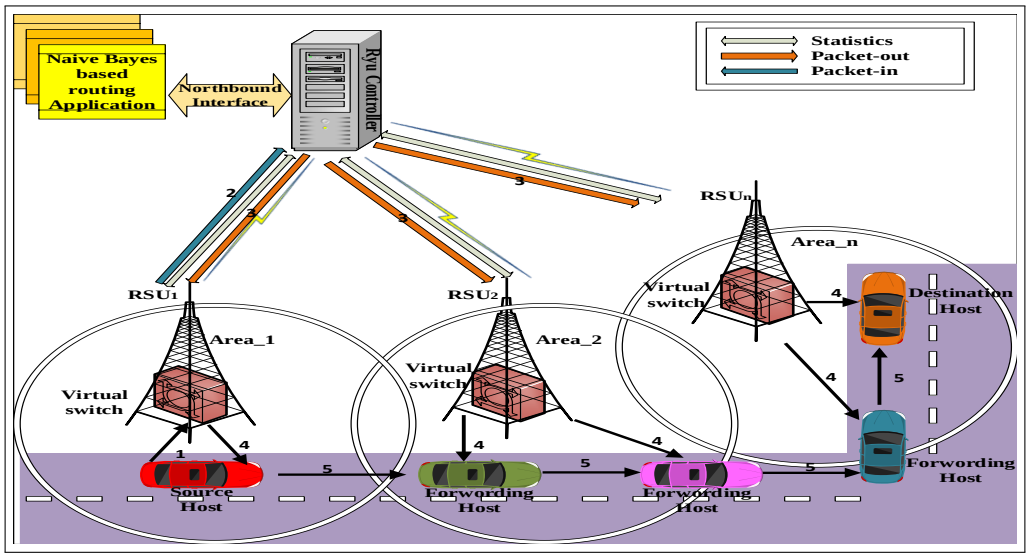

Figure 6. SDN-Naive Bayes-based Routing Protocol Operation.

Otherwise, the controller discovers that it does not have any previous knowledge of the source. Then, it adds it to its flow table and launches the process for finding the destination host attachment points. Nevertheless, if the destination is known, the controller sends a packet-out (arrow 3) to the source host with the routing path as rules for including them in the source's flow table. That is an optimistic assumption that the controller recognizes the destination. Generally, that is not the case, thus in other routing solutions, the controller floods the network with a request packet until reaching the target. However, our solution avoids this practice because it costs in packet overhead and increases the transmission delay.

Our protocol uses the Naive Bayes model to predict the area where the destination node rests. The RYU furnishes the feature's value needed in the working of the ML algorithm that computes the corresponding RSU associated with the destination node online. When it identifies the corresponded RSU of the destination location zone, the prediction model provides the outcome to the controller. The controller sends a packet-out comprising the routing rules between the source and destination to nodes for installing the appropriate controls. However, the controller constructs the routing paths based on the delay cost of links that constitute constructed routing paths. The controller applies the shortest path routing algorithm to select the best route. As the delay is the primary information upon which our suggested solution depends, our protocol retrieves the network's statistics to provide link delay in real-time for the system's whole links. The controller performs statistics periodically based on frequency time, offering newer delay values and a detailed network overview. Immediately the routing paths constructed and sent to the RSUs, which in turn transmit them to the vehicles included in this routing (arrow 4). Each vehicle establishes a connection with the gateway vehicle to initiate the transfer of packets between a source and destination cars (arrow 5).

6. Performance Evaluation

To assess the performance of the SDN-Naive Bayes-based routing protocol, we developed a simulation testbed, a virtual network created through virtual machines configured with ubuntu 18.10 Operating system, Mininet-WiFi 2.7 a wireless SDN emulator [43], SUMO 1.4.0 mobility simulator and OpenFlow 1.3 SDN protocol. The vehicles move on road network size 1500 $\times$ 2000 m^2 of Montreal downtown. In such topology, we consider all the nodes connected constitute a mesh network. Moreover, to generate traffic, we install the Iperf and ping applications to carry out the data transmission between vehicles. Table 2 summarizes the remaining emulation parameters of our experimentation; we repeat the evaluation of the performance ten times with a confidence interval of 95%.

Table 2. Emulation Parameters.

Parameter	Value
Network size	10–130 vehicles, 10 RSU and one controller
Simulation area	1500 $\times$ 2000 m^2
Propagation loss	Two-Rays Ground
Mobility	SUMO real scenario
Simulation time	500 s
Packet size	512 bytes
transmission range	250 m, 500 m
Transport protocol	UDP
Mac protocol	802.11p

To analyze the performance of the proposed routing protocol, we compare it to three routing protocols of the literature, multipath routing protocol [44], Q-learning protocol [45], and OLSR protocol [46]. The multipath routing protocol and Q-learning protocol are SDN routing protocols, except that OLSR is a Mobile Ad-Hoc Network (MANET) routing protocol. We decide to consider OLSR as comparative routing, since the main operation of this MANET routing is close to SDN working paradigm. OLSR apply centralized routing management executed on multipoint relays nodes that are the responsible nodes in control packet transmission. The following performance metrics measure the efficiency of the evaluated protocols [47]:

- PDR (Packet Delivery Ratio): the rate of data packets successfully delivered to data packets sent;
- Throughput describes an amount of data transfer by time unit. The compute is the ratio of packets size to the time difference between its sent and its reception;
- Packet Jitter Derivation (Jitter): is a variation in delay between packets. It defines the consistency and stability of the proposed wireless network. Components in the communication path introduce jitter in the system;
- Loss Ratio is the number of packets lost divided by the packets sent during simulation time;
- End to end delay: average time is taken by a data packet to reach the destination.

Figure 7 illustrates the packet delivery ratio transmitted by vehicles to a destination. The overall appearance of curves (a) or (b) shows a decrement in the delivery rate. Therefore, the more the number of vehicles in the network increases, the more the delivery rate decreases. This situation is the result of using a single controller to handle all network requests, leading to controller overload.

Overall, the Naive Bayes-based protocol shows better results than OLSR, multipath and Q-learning protocols. The following figures confirm the degradation of communication in the presence of a dense network. However, Figure 8 shows a loss ratio of more than 50% of the packets sent when the number of nodes exceeds 50 vehicles. Nevertheless, beware of our solution's objective, which brings an

improvement compared to other comparative solutions. Our protocol performs better than others with less packet loss, because it uses prediction based on the area where the destination is located, which provides enough coverage to ensure a routing path between the source-destination pair. On the other hand, the Multipath, Q-learning, and Naive Bayes protocol's throughput are very similar and better than OLSR. This result proves the efficiency of SDN solutions, except that the consumption of the bandwidth is not optimal and does not even arrive at half level, as shown in Figure 9. The efficiency of a routing protocol lies in the network's stability in terms of communications with the least possible interruption. Jitter is the performance metric that allows us to distinguish the most reliable routing solutions with the most stable routes. In Figure 10, the protocols using ML algorithms, i.e., Q-learning and Naive Bayes, show good results compared to OLSR and multipath, especially in low densities.

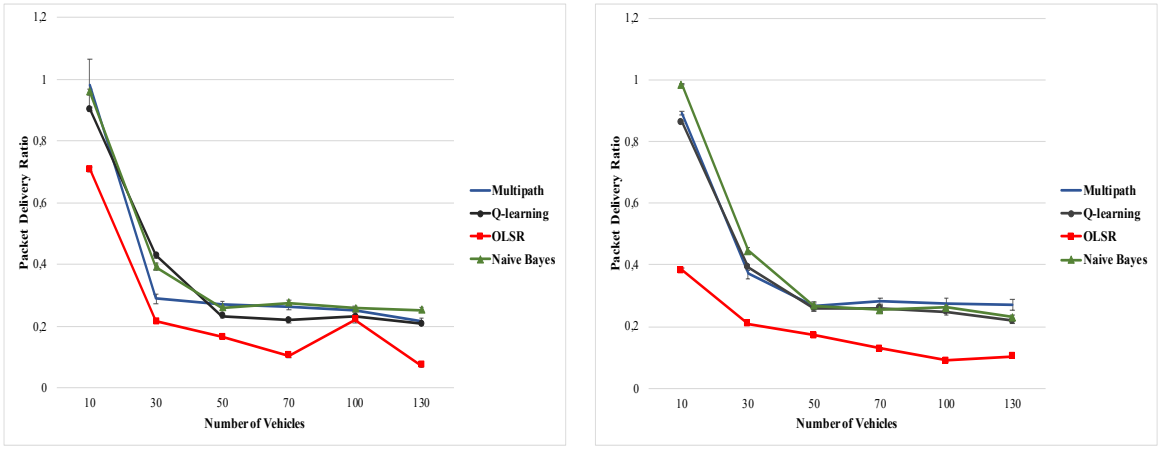

Figure 7. Packet Delivery Ratio vs. number of vehicles (**a**) range 250 m, (**b**) range 500 m.

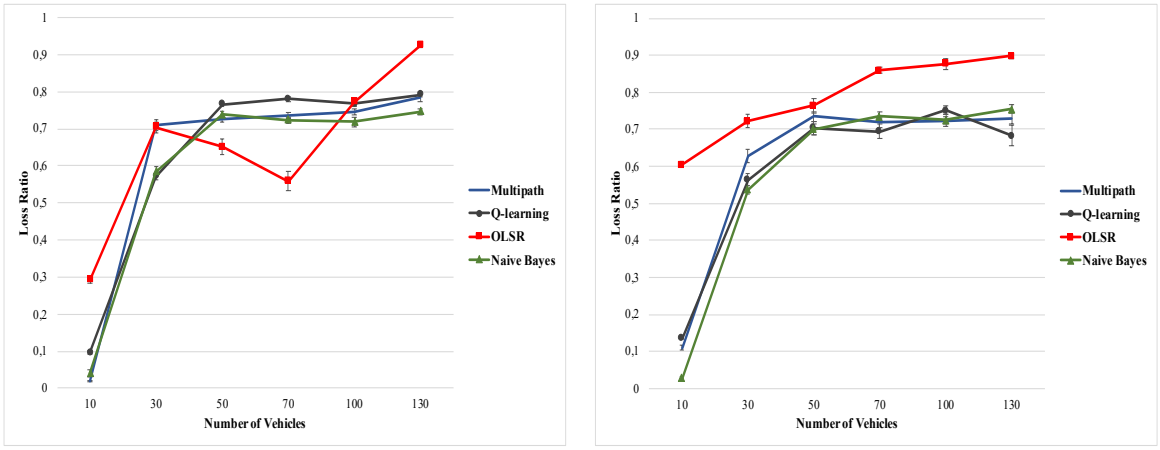

Figure 8. Loss Ratio vs. number of vehicles (**a**) range 250 m, (**b**) range 500 m.

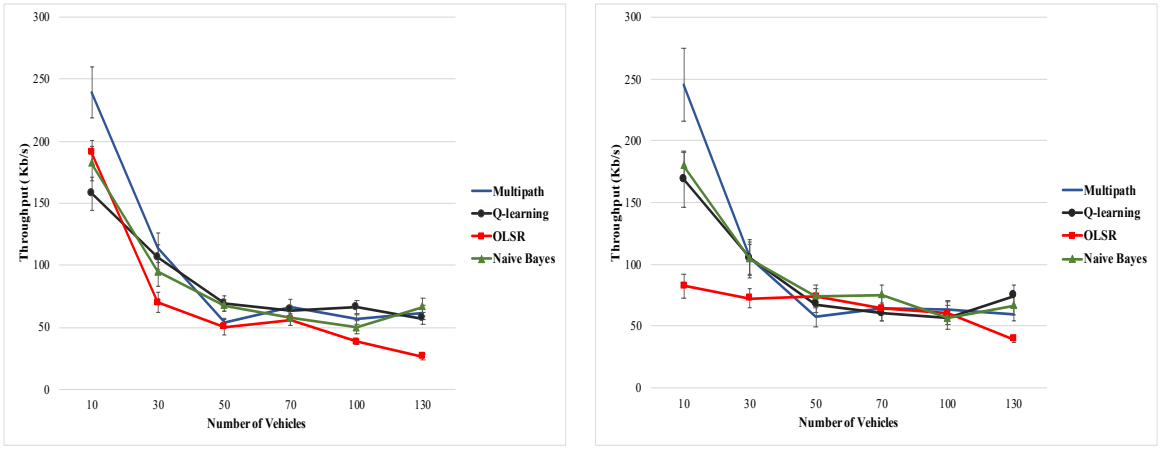

Figure 9. Throughput vs. number of vehicles (**a**) range 250 m, (**b**) range 500 m.

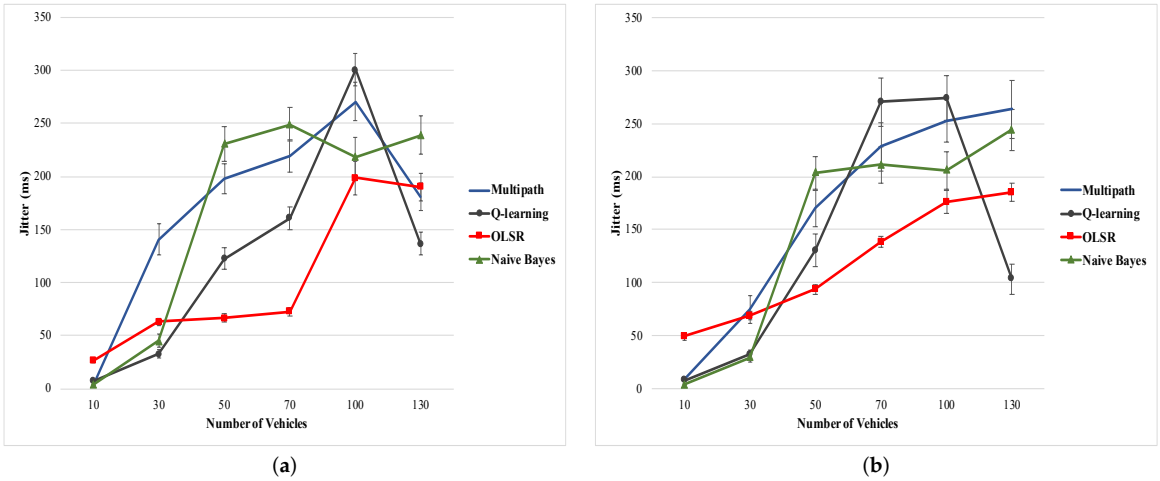

Figure 10. Packet Jitter Derivation (Jitter) vs. number of vehicles (**a**) range 250 m, (**b**) range 500 m.

Figures 11 and 12 show the end-to-end delay of the four protocols. Mainly our solution presents the shortest end-to-end delay, whether for 250 m or 500 m transmission range. Moreover, when we vary the size of the transmitted packets, we notice that always our protocol presents the best end-to-end delay compared to Q-learning, multipath, and OLSR protocols.

Except that for the case of 500 m transmission range, the Q-learning learning solution shows the best performance in term of the end to end delay. We explain this to the fact that the supervised learning analyzes a massive amount of data to form a system and produces a generalized formula. The quality of the training data influences this ML method's prediction results, so the results obtained are approximated. Whereas in Q-learning solution which uses reinforcement learning model, the reinforcement core is defined in the Markov model decision process. Reinforcement learning is trained as a learning agent where it functions as a reward and action system. The learning system itself creates data by interacting with the environment to observe its basic behaviour in discrete steps. The approach investigates every time the background and receives a reward for each observation. Finally, the goal is to collect as many bonuses as possible for doing more observations. Furthermore, this type of learning machine uses the Markov decision process that provides a mathematical framework for modeling. Sequential decision-making occurs, and the next entry depends on the learner's decision or the system. Based on the mathematical formulation, the reinforcement learning model generates an exact result available from the initial state. Moreover, in this scenario, our system's size is relatively small, which allows Q-learning to make several observations in a short

time, which improves its performance. In contrast, if the network becomes large enough and complex, the Q-learning response time will become tedious, influencing, and deteriorating the end to end delay.

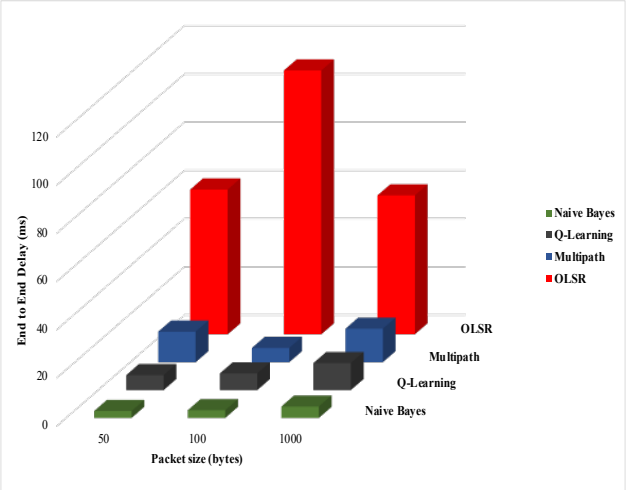

Figure 11. End to End delay vs. Packet size.

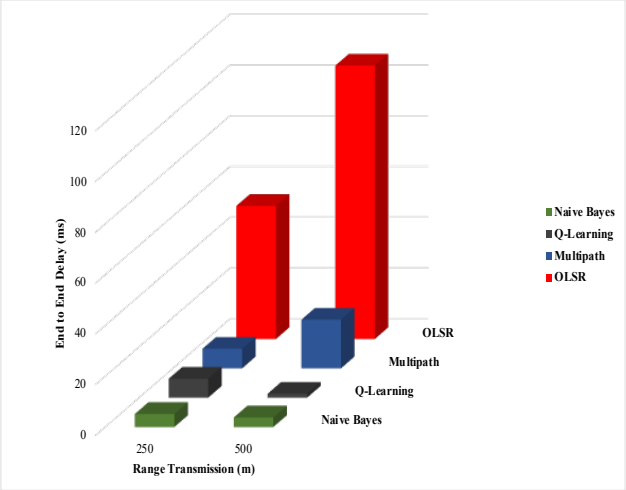

Figure 12. End to End delay vs. Range transmission.

7. Conclusions

In this article, we proposed a new routing protocol based on the SDN and Naive Bayes solution to improve the delay. Combining the SDN's flexibility with Naive Bayes prediction makes this routing protocol more reliable and optimal. The controller executes the Naive Bayes framework to predict the destination node location, avoiding the flooding process to discover the destination in the network. For building routing paths, our solution collects statistics of each node in the network periodically. It retrieves the links' delay information to use it as a routing metric to choose intermediate nodes included in the constructed route. The controller is responsible for providing a stable path with low delay by minimizing the path duration between the source and destination. The Naive Bayes framework uses a realistic dataset that we created based on Montreal traffic information using the SUMO simulator. Simulation results show the enhanced performance of Naive Bayes-based routing protocol in end-to-end delay, packet delivery ratio than the Q-learning, Multipath, and OLSR routing protocols. As part of future work, we intend to address the issue of using a single controller by proposing a new architecture based on cooperation between multi-controllers and LoraWan IoT technology.

Author Contributions: Conception and methodology, L.E.-G., S.P.; supervision, S.P. and S.C.; writing—original draft, L.E.-G.; writing—review and editing, L.E.-G., S.P. and S.C.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received funding from The Natural Sciences and Engineering Research Council of Canada (NSERC) and Flex Groups.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Kumar, H.; Singh, M.K.; Gupta, M.P.; Madaan J. Moving towards smart cities: Solutions that lead to the smart city transformation framework. *Technol. Forecast. Soc. Chang.* **2020**, *153*, 119281. [CrossRef]
2. Chiariotti, F.; Condoluci, M.; Mahmoodi, T.; Zanella, A. SymbioCity: Smart cities for smarter networks. *Trans. Emerg. Telecommun. Technol.* **2018**, *29*, e3206. [CrossRef]
3. Ding, H.; Li, X.; Cai, Y.; Lorenzo, B.; Fang, Y. Intelligent data transportation in smart cities: A spectrum-aware approach. *IEEE/ACM Trans. Netw.* **2018**, *26*, 2598–2611. [CrossRef]
4. Cordeschi, N.; Amendola, D.; Shojafar, M.; Naranjo, P.G.V.; Baccarelli, E. Memory and memoryless optimal time-window controllers for secondary users in vehicular networks. In Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems, Chicago, IL, USA, 26–29 July 2015; pp. 1–7. [CrossRef]
5. Hakiri, A.; Berthou, P.; Gokhale, A.; Abdellatif, S. Publish/subscribe-enabled software-defined networking for efficient and scalable IoT communications. *IEEE Commun. Mag.* **2015**, *53*, 48–54. [CrossRef]
6. Rehmani, M.H.; Davy, A.; Jennings, B.; Assi, C. Software-defined networks-based smart grid communication: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2637–2670. [CrossRef]
7. Hakiri, A.; Gokhale, A. Work-in-Progress: Towards Real-time Smart City Communications using Software Defined Wireless Mesh Networking. In Proceedings of the IEEE Real-Time Systems Symposium, Nashville, TN, USA , 11–14 December 2018; pp. 177–180. [CrossRef]
8. Patil, P.; Hakiri, A.; Barve, Y.; Gokhale, A. Enabling software-defined networking for wireless mesh networks in smart environments. In Proceedings of the IEEE 15th International Symposium on Network Computing and Applications, Cambridge, MA, USA, 31 October–2 November 2016; pp. 153–157. [CrossRef]
9. Wang, Y.; Martonosi, M.; Peh, L.S. A supervised learning approach for routing optimizations in wireless sensor networks. In Proceedings of the 2nd international Workshop on Multi-Hop ad Hoc Networks: From Theory to Reality, Florence, Italy, 26 May 2006; pp. 79–86. [CrossRef]
10. Xia, W.; Wen, Y.; Foh, C.H.; Niyato, D.; Xie, H. A survey on software-defined networking. *IEEE Commun. Surv. Tutor.* **2014**, *17*, 27–51. [CrossRef]
11. Haque, I.T.; Abu-Ghazaleh, N. Wireless software-defined networking: A survey and taxonomy. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2713–2737. [CrossRef]
12. Prabakaran, P.; Isravel, D.P.; Silas, S. A Review of SDN-Based Next-Generation Smart Networks. In Proceedings of the IEEE 3rd International Conference on Computing and Communications Technologies, Chennai, India, 21–22 February 2019; pp. 80–85. [CrossRef]
13. Bannour, F.; Souihi, S.; Mellouk, A. Distributed SDN control: Survey, taxonomy, and challenges. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 333–354. [CrossRef]
14. Okay, F.Y.; Ozdemir, S. Routing in fog-enabled IoT platforms: A survey and an SDN-based solution. *IEEE Internet Things J.* **2018**, *5*, 4871–4889. [CrossRef]
15. Venkatramana, D.K.N.; Srikantaiah, S.B.; Moodabidri, J. SCGRP: SDN-enabled connectivity-aware geographical routing protocol of VANETs for urban environment. *IET Netw.* **2017**, *6*, 102–111. [CrossRef]
16. Sudheera, K.L.K.; Ma, M.; Chong, P.H.J. Link stability based optimized routing framework for software defined vehicular networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2934–2945. [CrossRef]
17. Singh, P.K.; Sharma, S.; Nandi; S.K.; Nandi, S. Multipath TCP for V2I communication in SDN controlled small cell deployment of smart city. *Veh. Commun.* **2019**, *15*, 1–15. [CrossRef]
18. Zhu, M.; Cao, J.; Pang, D.; He, Z.; Xu, M. SDN-based routing for efficient message propagation in VANET. In Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications, Qufu, China, 10–12 August 2015; pp. 788–797. 77. [CrossRef]

19. Abbas, M.T.; Muhammad, A.; Song, W.C. SD-IoV: SDN enabled routing for internet of vehicles in road-aware approach. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 1265–1280. 2-019-01319-w. [CrossRef]
20. Abugabah, A.; Alzubi, A.A.; Alfarraj, O.; Al-Maitah, M.; Alnumay, W.S. Intelligent Traffic Engineering in Software-Defined Vehicular Networking Based on Multi-Path Routing. *IEEE Access* **2020**, *8*, 62334–62342. [CrossRef]
21. Trakadas, P.; Nomikos, N.; Michailidis, E.T.; Zahariadis, T.; Facca, F.M.; Breitgand, D.; Gkonis, P. Hybrid clouds for data-Intensive, 5G-Enabled IoT applications: An overview, key issues and relevant architecture. *Sensors* **2019**, *19*, 3591. [CrossRef]
22. Tang, Y.; Cheng, N.; Wu, W.; Wang, M.; Dai, Y.; Shen, X. Delay-minimization routing for heterogeneous VANETs with machine learning based mobility prediction. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3967–3979. [CrossRef]
23. Azzouni, A.; Boutaba, R.; Pujolle, G. NeuRoute: Predictive dynamic routing for software-defined networks. In Proceedings of the 13th International Conference on Network and Service Management, Tokyo, Japan, 26–30 November 2017; pp. 1–6. [CrossRef]
24. Chen, F.; Zheng, X. Machine-learning based routing preplan for SDN. In Proceedings of the International Workshop on Multi-disciplinary Trends in Artificial Intelligence, Fuzhou, China, 13–15 November 2015; pp. 149–159.
25. Li, L.; Zhang, Y.; Chen, W.; Bose, S.K.; Zukerman, M.; Shen, G. Naive Bayes classifier assisted least loaded routing for circuits witched networks. *IEEE Access* **2019**, *7*, 11854–11867. [CrossRef]
26. Baz, A. Bayesian machine learning algorithm for flow prediction in SDN switches. In Proceedings of the IEEE 1st International Conference on Computer Applications and Information Security, Riyadh, Saudi Arabia, 4–6 April 2018; pp. 1–7. [CrossRef]
27. Martin, I.; Troia, S.; Hernandez, J.A.; Rodriguez, A.; Musumeci, F.; Maier, G.; Alvizu, R.; de Dios, O.G. Machine learning-based routing and wavelength assignment in software-defined optical networks. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 871–883. [CrossRef]
28. Sun, P.; Hu, Y.; Lan, J.; Tian, L.; Chen, M. TIDE: Time-relevant deep reinforcement learning for routing optimization. *Future Gener. Comput. Syst.* **2019**, *99*, 401–409. [CrossRef]
29. Yao, H.; Yuan, X.; Zhang, P.; Wang, J.; Jiang, C.; Guizani, M. Machine learning aided load balance routing scheme considering queue utilization. *IEEE Trans. Veh. Technol.* **2019**, *68*, 7987–7999. [CrossRef]
30. Liu, J.; Wang, Q.; He, C.; Jaffres-Runser, K.; Xu, Y.; Li, Z.; Xu, Y. QMR: Q-learning based multi-objective optimization routing protocol for flying ad hoc networks. *Comput. Commun.* **2020**, *150*, 304–316. [CrossRef]
31. Zhang, C.; Patras, P.; Haddadi, H. Deep learning in mobile and wireless networking: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2224–2287. [CrossRef]
32. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
33. Liangxiao, J.; Lungan, Z.; Chaoqun, L.; Jia, W. A Correlation-Based Feature Weighting Filter for Naive Bayes. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 201–213. [CrossRef]
34. Osisanwo, F.Y.; Akinsola, J.E.T.; Awodele, O.; Hinmikaiye, J.O.; Olakanmi, O.; Akinjobi, J. Supervised machine learning algorithms: Classification and comparison. *Int. J. Comput. Trends Technol. (IJCTT)* **2017**, *48*, 128–138. [CrossRef]
35. Xie, J.; Yu, F.R.; Huang, T.; Xie, R.; Liu, J.; Wang, C.; Liu, Y. A survey of machine learning techniques applied to software-defined networking (SDN): Research issues and challenges. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 393–430. [CrossRef]
36. Zhu, G.; Zan, J.; Yang, Y.; Qi, X. A supervised learning-based QoS assurance architecture for 5G networks. *IEEE Access* **2019**, *7*, 43598–43606. [CrossRef]
37. Mitchell, T.M. *Machine Learning*; McGraw-Hill Higher Education: New York, NY, USA, 1997.
38. OpenStreetMap (OSM). Available online: https://www.openstreetmap.org (accessed on 1 October 2019).
39. SUMO User Documentation. Available online: https://sumo.dlr.de/userdoc/index.html (accessed on 1 October 2019).
40. Lopez, P.A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flotterod, Y.P.; Hilbrich, R.; WieBner, E. Microscopic traffic simulation using sumo. In Proceedings of the IEEE 21st International Conference on Intelligent Transportation Systems, Maui, HI, USA, 4–7 November 2018; pp. 2575–2582. [CrossRef]

41. Open Data Portal Montreal City Website. Available online: http://donnees.ville.montreal.qc.ca/dataset/comptage-vehicules-pietons (accessed on 1 October 2019).
42. Bratsas, C.; Koupidis, K.; Salanova, J.M.; Giannakopoulos, K.; Kaloudis, A.; Aifadopoulou, G. A Comparison of Machine Learning Methods for the Prediction of Traffic Speed in Urban Places. *MDPI Sustain.* **2020**, *12*, 142. [CrossRef]
43. Fontes, R.D.R.; Mahfoudi, M.; Dabbous, W.; Turletti, T.; Rothenberg, C. How far can we go? towards realistic software-defined wireless networking experiments. *Comput. J.* **2017**, *60*, 1458–1471. [CrossRef]
44. Wildan M.S. Multipath Routing with Load Balancing Using RYU OpenFlow Controller. Available online: https://github.com/wildan2711/multipath (accessed on 1 October 2019).
45. Zuo X. A Holistic Testbed/Emulator for the Book: Computing in Communication Networks: From Theory to Practice. Available online: https://git.comnets.net/public-repo/comnetsemu (accessed on 1 October 2019).
46. Clausen, T.; Jacquet, P.; Adjih, C.; Laouiti, A.; Minet, P.; Muhlethaler, P.; Viennot, L. Optimized link state routing protocol (OLSR). In Proceedings of the IEEE International Multi-Topic Conference, Lahore, Pakistan, 30 December 2001. [CrossRef]
47. Jany, M.H.R.; Islam, N.; Khondoker, R.; Habib, M.A. Performance analysis of OpenFlow based software-defined wired and wireless network. In Proceedings of the IEEE 20th International Conference of Computer and Information Technology, Dhaka, Bangladesh, 22–24 December 2017; pp. 1–6. [CrossRef]