A Fog-Based IOV for Distributed Learning in Autonomous Vehicles

Pawan Subedi(✉), Beichen Yang, and Xiaoyan Hong

Computer Science, The University of Alabama, Tuscaloosa, USA
{psubedi,byang12}@crimson.ua.edu, hxy@cs.ua.edu

Abstract. Internet of Vehicles (IoVs) consist of connected vehicles and connected autonomous vehicles. With fog computing built within the IoV, it becomes promising for federated learning to be used in vehicular environments. One important application of such a fog computing system is distributed deep learning for decision-making tasks in autonomous driving. In this paper, a distributed training system building on top of the Named-Data Networking (NDN) architecture is introduced in order to combat the mobility challenges to the underlying network. The paper presents analyses on critical latency issues pertained to soliciting the worker CVs and collecting the partial updates. Further, the advantages of using NDN for the IoV are evaluated with comparisons to IP network through simulation. The results show promising performance gains for the evaluation cases.

Keywords: Internet of Vehicles · Fog computing · Connected vehicles · Federated learning · Autonomous driving

1 Introduction

With great progresses in wireless communications supporting vehicular networks, the Internet of Vehicles (IoVs) has been closer to reality. Connected vehicles (CVs), and more recently, connected autonomous vehicles (CAVs), all benefit from IoVs for various data and application services for enhancing safety, efficiency and autonomy in driving. The networked vehicles often sense, generate and consume massive amount of data for these services. At the same time, for information dissemination in vehicular fog environment, the multi-hop vehicular ad hoc network (VANET) and encounter based vehicular delay tolerant network (VDTN) have been extensively studied. The recent easiness in cloud accesses for both storage and computations have further motivated this trend. On the other hand, some of these services are the application scenarios that directly

The work is supported partly by the National Science Foundation under Grants No. 1719062. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

C. T. Calafate et al. (Eds.): MONAMI 2021, LNICST 418, pp. 227–242, 2022.
https://doi.org/10.1007/978-3-030-94763-7_18

impact driving and utilize the vehicles to forward messages generated by them. With the latter, the networking are challenged in terms of connectivity, latency, bandwidth, throughput, and security requirements.

Along with the easiness of the cloud infrastructure, the architectures of Mobile Edge Computing and Fog Computing also have been developing for IoVs, which would enable the edge servers or the CVs (Fog Computing) to carry out equivalent service tasks traditionally carried out by the cloud. Building on vehicle networking technology, Fog Computing brings computing into the network so to further reduce the overhead from communication and networking.

On the other hand, the development of autonomous vehicles is expected to help alleviate various transportation management issues including safety, congestion, energy conservation, etc. Autonomous driving involves various critical decision-making tasks requiring artificial intelligence [1]. These sophisticated tasks require continual evolution of the models used in the AI system to guide the ever-changing scenarios of driving conditions. Therefore, the AI models need to be constantly trained and retrained. The massive aggregation of the training data and computing are mostly carried out at the central cloud.

At the same time, the recent research in distributed machine learning brings new opportunity for autonomous driving to use local and emerging new data in decision making. Federated Learning (FL), a distributed training process for Deep Neural Networks (DNN) [2], can be performed in the fog of the vehicles [3]. With federated learning, the vehicles in the fog system participate in the training process with their local data. It will bring two advantages to autonomous driving: quickly access richer dataset from distributed data sources because there is no need to transfer data to the cloud; and built-in privacy preservation because each distributed training process uses own set of data and a participating vehicle's data never leaves the owner. On top of the improvements on federated learning, the recent advances in incremental (continual) learning further lay the foundation for taking local data and enhancing an existing model [4,5]. With incremental learning, a trained model can be trained again using new data, without loosing the learned knowledge. This enables the federated learning to be more useful for the autonomous driving scenario as the learned knowledge can be integrated in an asynchronous manner without losing the previously learned knowledge of the model [6].

We envision a system that integrates federated learning with the continuum of the fog of the vehicles, the edge network, and the cloud such that the retraining of the model can happen in the fog of the vehicles, and the involvement of the edge network, and the cloud enables the aggregation of the updates coming from the fog [7–9].

Recognizing the challenges coming from the host-based network architecture, e.g., uneasy to obtain addresses, hard to maintain long-term and stable connections, dynamic network topology, routing protocol overhead, etc., the fog computing will use an information-centric networking (ICN) approach to overcome the mobility and intermittent connectivity challenges. Being on realization of ICN, Named-Data Networking (NDN) has been recognized as a promising solution for ad hoc networks and for vehicular networking [10–12]. In NDN, communications are based on data names, instead of host-to-host connections. To request data, a consumer sends an Interest packet with a data name to the network. Any provider who has the name-matching data will send it back.

The sharing nature of NDN communication model via its connection-free feature, such as multicast and in-path caching, allows effective use of wireless broadcast channel, leading to lower communication overhead. In addition, the name based data addressing scheme of NDN eliminates the burden for learning and managing the addresses of the CAVs and CVs in the vehicular scenario with constant topology changes and network disruptions [11,12]. The advantages and enhancements of NDN over VANET and VDTN are also being discussed in earlier work [10,13–16]. In addition, with NDN's name-data integrity, any participated nodes can verify that the received packets are tampered or not during transmissions. This offers security assurance to the CAVs.

In this paper, we present extensive analyses about the impact of vehicle mobility on the latency in tasks involving FL in vehicular fog. Understandings on the latency is extremely important because it reveals whether it would be feasible using FL for model enhancement under certain vehicular environments. Although many works have been done analyzing the network performance relating to mobility, most of these works do not directly address problems involving tasks in federated learning over the fog continuum. Similarly, we also present simulation results comparing IP networks with NDN for such systems.

The analyses show closed form latency distributions for the tasks in vehicular fog. Similarly, the simulation results show NDN to have better network performance in comparison to IP networks for such systems. At the same time, the simulation results also exhibit that the inclusion of fog continuum into the NDN based system as envisioned in this paper further improves the network performance.

The rest of the paper is organized as follows. Section 2 describes the networking and computing system over IoV Fog and cloud continuum. Section 3 presents the main analyses. The advantages of using NDN over IP network is presented in Sect. 4 through simulation results. Section 5 concludes the paper.

2 System Model

The vehicular fog to cloud continuum mainly consists of CVs and CAVs. It builds on top of the underlying network technologies of NDN over multihop vehicular ad hoc network (NDN-VANET) and encounter based vehicular delay tolerant network (NDN-VDTN) [10,13–15]. The vehicular fog to cloud continuum is also supported by edge servers. The edge servers serve as distributed distribution centers for the training models, which not only host initial models, but also receive model updates and forward them to the cloud. The cloud servers are responsible for the model aggregation.

On the other hand, the distributed training process involves several steps such as a CAV must have the model at the first place, be able to transfer the

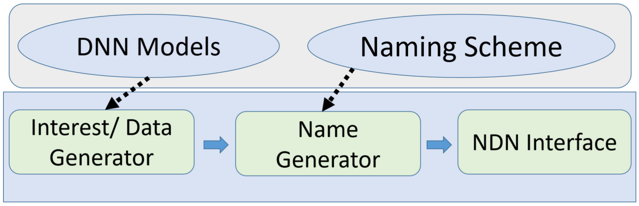

Fig. 1. System modules

model to at least one CV worker and collect at least one new update. With a proper ratio of CVs and set of NDN-VANET and NDN-VDTN protocols, a CAV can send an interest to solicit CV workers in proximity to help train a model. Multiple CVs can join as workers. The CVs, after receiving the model from the CAV, train the model so that both the environmental conditions as well as the actions taken by the driver in a CV based on these conditions can be fed to the model. After completing the training, the CAV can collect these updates. In a more general scenario, many CAVs can participate in soliciting more CVs to increase the capability of the training process. Or, many CAVs can directly benefit from the partial updates available at the workers. In addition, when considering mobility, a CAV who previous expressed an interest may move increasing away from a worker. Yet, the update may be useful for another CAV. To facilitate these situations, our proposed protocol is for a worker to send an interest to announce the available update. Interested CAVs can then collect the updates. Such a system increases the chances of finding a worker and also increases the computing efficiency by maximizing the sharing of the updates. Further, the CAV can submit the partial updates to an edge server. The edge servers are connected via internet to share the partially updated models and aggregate multiple of them. The enhanced model will be hosted by the edge servers for subsequent requests from the CAVs.

Using the NDN architecture requires a thoughtful design of the naming scheme because it determines how a piece of named-data may be retrieved through the interest-data packet pair. It can impact the overall network performance [17]. The naming scheme for the system primarily includes three major fields, reflecting hierarchical information within the names.

$$\langle model\_fields\rangle \,/\, \langle src\_id\rangle \,/\, \langle seq\_i\rangle$$

Here, the first **model_fields** is to identify a specific model. This field can contain multiple subfields. Three common subfields are: (i) *model_id:* It is the identifier of the training model; (ii) *model_version:* A training process involves multiple rounds/epochs. This field identifies the round of the model with *model_id*. It can be the incremental versioning of the trained model; and (iii) *part_number:* It identifies the part of the model that the packet carries. The second **src_id** is an optional identifier of the node looking out for the data and could include vague geographic location to help forward the packet. The last **seq_id** is the sequence number of this packet when it was created, serving as version control.

The IoV system is layered according to the dependence of functionalities. The layering consists of the function blocks for an application, here, the DNN model and the naming scheme, building on top of NDN primitives. The interplay of these function blocks and primitives is shown in Fig. 1. For the NDN primitives, *Interest/Data Generator* module handles new data pieces, be it a message, a model or a partial update. The module submits the data piece along with the metadata (e.g., version number, model Id, etc.) to the *Name Generator* for the further processing of the interest or data packet. *Name Generator* is the component responsible for creating the names for the packet in consideration. It uses the name convention when doing so. It then attaches the generated name to the data or interest piece submitted by the Data Generator module to create a complete packet. It then submits the packet to the NDN Interface Module. The *NDN Interface* is the NDN network interface that handles the forwarding and receiving of both Interest and Data packets.

3 Numerical Analysis

In this section we present analyses about the latency occurred during worker solicitation process and update collection process. The worker solicitation process decides how the computation task originated from CAVs would be distributed to the worker CVs. Meanwhile, the update collection process decides how the training results generated by CVs would be returned to CAVs. Understandings about the latency of the two processes are vital to the performance of the purposed system. Our analyses consider an area with N CAVs, M worker CVs that are willing to participate in the training process, and other CVs in the connected vehicular environments.

CAVs and CVs can communicate when they are within the transmission range of one another. The IoV fog networking environment can occur in a mix of NDN-VANET and NDN-VDTN. The former will disseminate information in a short time period via multi-hop packet relays. But with the latter, communication opportunities occur only when CVs or CAVs encounter one another. In this section, we analyze the impact of mobility of the more challenging NDN-VDTN scenario. There are certain similarity between worker solicitation process and update collection process. However, two different mathematical approaches are used in the analyses. This is due to the two different problem settings. With the worker solicitation process, the analyses starts with the case that all the CAVs have the same model waiting to be distributed, whereas with the update collection process, only a subset of the CVs have the updates ready.

3.1 Worker Solicitation Delay

Worker solicitation delay measures the time for a worker CV to receive an interest packet sent by a CAV looking for CVs to join distributed training. A worker CV receiving the interest is called a holding vehicle. After receiving the interest, it then follows up to retrieve the model. The solicitation delay is the time needed for a CAV to meet one of the holding vehicles. The delay mainly constitutes of the time needed for encountering a holding vehicle, which may take multiple encounters because only a part of the vehicles are the holding vehicles.

The encounter opportunity of two mobile nodes is one important mobility property. Earlier work has shown that the time for any pair of nodes to meet follows a Poisson distribution and thus the inter-meeting time of the pair follows an exponential distribution, Robin et al. [18]. The work shows that such a result applies to the Manhattan mobility, the random waypoint model, or simply random direction. In addition, various empirical studies on real-life mobility traces of the vehicles have also shown that exponential distribution is followed by the inter-meeting times. Thus, we assume the same for the vehicles in our scenario. Suppose $0 \leq t_{i,j}(1) < t_{i,j}(2) < ...$ are the series of the time points when two vehicles i and $j (i \neq j)$ meet. The processes $\{t_{i,j}(n), n \geq 1\}, 1 \leq i, j \leq T, i \neq j$, is the independent Poisson processes with parameter λ. λ describes how many times can a pair of vehicle meet with each other in unit time. Further, let $\rho_{i,j}(n) = t_{i,j}(n+1) - t_{i,j}(n)$ be the n-th inter-meeting time of vehicle i and j. The random variables $\{\rho_{i,j}(n)\}_i, j, n$ are independent with each other and follow the exponential distribution with mean $1/\lambda$. $1/\lambda$ is the expected inter-meeting time before any pair of vehicles meet again with each other. A Markov Chain model is used in the analysis considering the case of a single worker CV (named W) receiving the solicitation interest in the area with total $R+1$ CVs. Let a state denotes the number of vehicles in the area that have the interest. Here we assume N AV nodes initially hold the interest, i.e., the initial state being N. Let F be the absorbing state when W receives the interest. Let $N+i$ represents the state that ith non-worker CVs have received the interest. State $N+i$ transitions to the next non-absorbing state $N+i+1$ when the interest is transmitted to another non-worker CV. State $N+i$ transitions to the absorbing state F when the interest is transmitted to W. The solicitation process finishes when W has the interest. Thus, the state space, S of the Markov chain includes $R+1$ non-absorbing states and one absorbing state F, denoted as $S = \{N, N+1, N+2, ..., N+R-2, N+R-1, N+R, F\}$.

Starting from the initial state N, one CAV encounters one of the R CVs at the rate $R\lambda$, and thus the aggregate transition rate from the state N to $N+1$ is $NR\lambda$. On the other hand, if W receives the solicitation from one of the CAVs, the aggregate rate from state N to the absorbing state F is $N\lambda$. In general, let $b_i = N+i$ be the number of vehicles having the interest at the state $N+i, i = 0, 1, ...R$, and let $d_i = (R-i)$ be the number of remaining non-worker CVs. The aggregate transition rate from ith state $N+i$ to the $(i+1)th$ state $N+i+1$ is $b_i d_i \lambda$, and the aggregate rate from state $N+i$ to the absorbing state F is $b_i\lambda$. These transition rates are shown in Fig. 2.

Let a series of state transitions from $N \to N+1 \to \cdots \to N+i \to F$ be a complete trajectory ending at the absorbing state for W. The random variable X such that $X = x_i$ describes the ith trajectory which transitioning from the

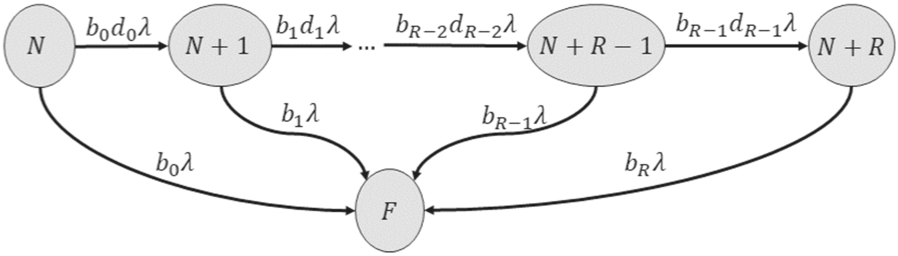

Fig. 2. Markov chain for a single worker solicitation

last state $N+i$ to F. Further, let p_{x_i} be the probability of the trajectory $X = x_i$, i.e., $P(X = x_i)$. For example, $X = x_0$ means a transition from the state N to F, and $X = x_1$ refers to the transitions from the states N to $N+1$ then to F.

At state $N+i$, there are $N+i$ number of holding vehicles, and $R-i$ non-worker CVs without the solicitation. As seen from Fig. 2, two types of transitions would happen to them. The first type happens when one of the former meets one of the latter, hence, transits to state $N+i+1$. There are $(N+i)(R-i)$ possible occurrences of this type. The other happens when one from the former meets the worker W, leading to the absorbing state. There are $N+i$ possible occurrences. Thus, the total possible events are $(N+i)(R-i)+(N+i)$. As such, the probability for transition from state $N+i$ to $N+i+1$ is $(N+i)(R-i)/(N+i)*(R-i)+(N+i) = (R-i)/(R-i+1)$, whereas the probability for transitioning from state $N+i$ to F is $(N+i)/(N+i)(R-i)+(N+i) = 1/(R-i+1)$.

The occurrence of a particular trajectory takes a series of state transitions. Its probability combines the probabilities of each horizontal transition in Fig. 2 and the transition to state F. Specifically, the probability p_{x_i} for trajectory x_i is given in Eq. 1.

$$p_{x_i} = \left[\prod_{j=0}^{i-1} \frac{(R-j)}{(R-j+1)} \right] \frac{1}{R-i+1} \tag{1}$$

Similarly, the delay along a trajectory x_i has to consider a series of times spending at each state waiting for an encounter to happen so to transit to the next state until encountering W. The latter leads to state F. Taking the aggregation factors at each state into consideration, for transitioning between horizontal states, the factor at state $N+j$ is $1/(N+j)(R-j)$; for transitioning to state F from $N+j$, the factor is $1/(N+j)$. The expected delay along trajectory x_i is the summation of the delays at each horizontal states, denoted as D_{x_i}. Recall that the expected pair-wise inter-meeting time is $1/\lambda$, D_{x_i} is given by Eq. 2.

$$E[D_{x_i}] = \left[\sum_{j=0}^{i-1} \frac{1}{(N+j)(R-j)\lambda} \right] + \frac{1}{(N+i)\lambda} \tag{2}$$

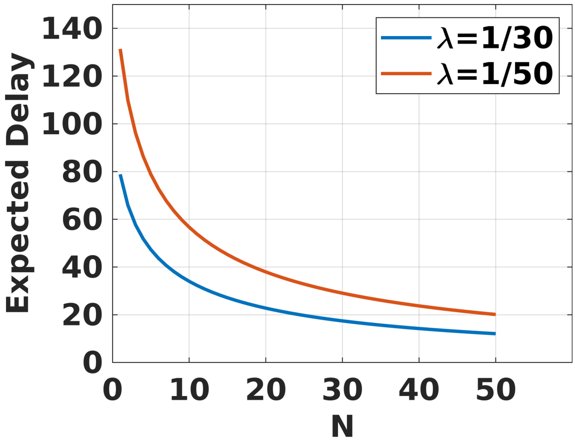

Fig. 3. Worker solicitation delay vs number of AVs with total vehicles = 100

Take trajectory $X = x_0$ as an example, the probability of N CAVs encountering W so transiting to state F, p_{x_0}, is given as $N/(RN + N) = 1/(1 + R)$, where the average delay transitioning from state N to the absorbing state F is $\frac{1}{N\lambda}$ due to the aggregation of N CAVs encountering W. Further, take trajectory $X = x_1$ as another example. Trajectory x_1 transits from state N to state $N+1$, then F. The existence of state $N+1$ is based on the event that one of the AVs at state N has passed the interest to one of the non-worker CVs. Thus, the probability is $RN/(RN + N) = R/(R+1)$. At state $N + 1$, there are $N + 1$ holding vehicles, and $R - 1$ non-worker CVs plus the worker W without the interest. Thus, the probability that one of the $N + 1$ vehicles passes the interest to W is $(N + 1)/(R - 1)(N + 1) + (N + 1) = 1/R$. Combining the two probabilities, we have the probability of trajectory x_1 be $[R/(R + 1)] * (1/R) = 1/(R + 1)$. The delay of trajectory x_1 has to count the delays at states N and $N + 1$, which is given as $\frac{1}{N\lambda} + \frac{1}{(N+1)\lambda}$.

Now, with Eqs. 1 and 2, the expectation of the solicitation delay over all the trajectories is given by Eq. 3.

$$E[D] = \sum_{i=0}^{R} E[D_{x_i}]p_{x_i} \tag{3}$$

Based on Eq. 3, as seen from Fig. 3, the expectation of the solicitation delay at a particular value of *lambda* decreases with the increase in number of nodes having the initial solicitation, N. Also, with the increasing value of N, the rate of change of average delay decreases, meaning with the optimized numbers of the initial solicitors present in the system, an optimistic value of delay can be realized.

3.2 Update Collection Delay

The need for updating a model usually only stays valid for a certain amount of time. Thus, knowledge about how many workers may help the training within a time frame is important. This analysis focuses on the number of holding workers with regards to time. The analysis is based on the assumption that the time is slotted. The duration of the time slot is long enough for establishing a connection and completing the model transmissions. At the same time, the actual time for the training process is not taken into the consideration. We assume that every worker that has a model will create an update.

For a time slot t, the reasons that a worker CV doesn't have an update can either be due to the expiration of the received model at time $t-1$, or due to the fact that the worker didn't have a model at time $t-1$ and it doesn't receive a model at time t from any CAV either.

In the analysis, let $p(j,t)$ be the probability that a worker j has an update at time t. And let τ be the probability that the update expires due to aging of the model. Further, suppose $q(j,t)$ be the probability that j didn't receive a model for training at t; and $n(i,j,t)$ be the probability that worker j receives a model from CAV i at time t. Given there are a total of N CAVs, $q(j,t)$ can be expressed in Eq. 4. As such, the problem in question can be described by Eq. 5.

$$q(j,t) = \prod_{k=1}^{N} (1 - n(k,j,t)) \tag{4}$$

$$1 - p(j,t) = \tau p(j,t-1) + (1 - p(j,t-1))\, q(j,t) \tag{5}$$

Solve Eqs. 4 and 5 and assume t be large enough such that the products approach zero, the limits of $p(j,t)$ $q(j,t)$ and $n(k,j,t)$ become $p(j)$, $q(j)$ and $n(i,j)$ respectively. Thus, we obtain the limit $p(j)$ in Eq. 6:

$$p(j) = \frac{1 - q(j)}{1 - q(j) + \tau} = \frac{1 - \prod_{l=1}^{N}(1 - n(l,k))}{1 - \prod_{l=1}^{N}(1 - n(l,k)) + \tau} \tag{6}$$

The task of Update Collection completes when the CAV encounter any CV that has completed its training. Here, we derive the latency needed for collecting the first update by a CAV. Given the collection process starts after the model distribution, we use the limiting probability that a worker has an update $p(j)$ from 6 to continue our analysis.

To analyze when the first update can be collected by a CAV, we start with capturing the mobility history that a CAV may encounter several non-participating worker CVs before encountering the one with an update, we use an encounter progression matrix. For a CAV y, let F_y be its encounter progression matrix. An element (i,j) in the matrix is the probability $f(i,j)$ that describes the progression of y having encountered CV i, then connecting to CV j. Given there are total M worker CVs, F_y is an $M \times M$ matrix as shown below 7, where $f(k,k) = 1, k = 1, 2, ..., M$.

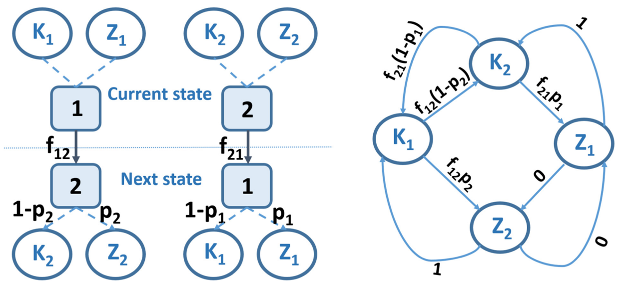

Fig. 4. An example state transition of two nodes for the update collection and the respective Markov Chain

$$F_y = \begin{bmatrix} f(1,1) & f(1,2) & ... & f(1,M) \\ f(2,1) & f(2,2) & ... & f(2,M) \\ & & ... & \\ f(M,1) & f(M,2) & ... & f(M,M) \end{bmatrix} \tag{7}$$

The update collection process can be described similar to worker solicitation Sect. 3.1. A CAV y may encounter multiple CVs without being able to collect any updates until the time of meeting the CV i. Now, during the encounter progression, if the newly connected CV j has an update, we say the progression yielded an update collection for the CAV.

To analyze the encounter progression for y, let's create a Markov Chain where the state space includes every possible state for all the CVs in our analysis. Regarding every possible state for a CV, it can either have the update or not, meaning these two states completely define the possible states for a CV. As such let's create two different sets from every possible states in our analysis. The first set $K = \{k_1, k_2, ..., k_m\}$ contains the states for not having update of every CV, and the second set $Z = \{Z_1, Z_2, ..., Z_m\}$ contains the states for having update of every CV.

With the above definition of every possible states in terms of two sets, let's first define the transition matrix R for the CAV y based on the entries in the matrix. An entry (i, j) in R is the transition probability r_{ij} for transiting from encounter of the CV i to the encounter of the CV j, where the transition probability r_{ij} is expressed in terms of the encounter progression probability $f(i, j)$ and the probability of the CV j for having an update, $p(j)$, as given by Eq. 8. Figure 4 presents an example scenario involving two nodes for the transition. For the two nodes scenario, the CAV y could have been initially in connection with either of the two nodes. Similarly, these two nodes could have been in either of their own two states (having updates or not having updates). Now, if the CAV y moves such that there is the change in the connectivity, it either moves from the connectivity of node 1 to node 2 or from node 2 to node 1. These new nodes could have been in either of their states with the respective probability of the states. Based on this, we construct the Markov Chain. One thing that needs to be noticed in Fig. 4 is that if the node transitions from a state of already having update to a new state without an update, we have the transition probability to be 1 since the CAV already has the required update. Similarly, we have the probability of transition from state of already having update to the new state of already having update, the transition probability is 0, as we do not allow such transitions in our scenario (of first update collection by a CAV).

$$r(i,j) = \begin{cases} f(i,j)(1-p(j)), & \text{if } State(i) \in K, State(j) \in K \\ f(i,j)p(j), & \text{if } State(i) \in K, State(j) \in Z \\ 1, & \text{if } State(i) \in Z, State(j) \in K \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Based on the definition of the entries, we then construct the entire matrix $\mathbf{R}$. For this, we create a separate $M \times M$ diagonal matrix, $\mathbf{B}$, where each diagonal element is the probability, $p(j)$ for each CV. For example, the diagonal element in third row and third column has the probability of having update for the CV 3. Further, we define $\overline{\mathbf{B}} = I - \mathbf{B}$, where I is the identity matrix. Therefore, using Eqs. 7 and 8, and matrix $\mathbf{B}$, the transition matrix $\mathbf{R}$ is expressed by Eq. 9, where each term is $M \times M$ matrix.

$$R = \begin{bmatrix} \mathbf{F_y}\overline{\mathbf{B}} & \mathbf{F_y}\mathbf{B} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{9}$$

Thus, based on Eq. 9, we can conclude that the transitions follow the characteristics of a terminating Markov Chain, for which, the upper triangular part $\mathbf{F_y}\overline{\mathbf{B}}$ entirely characterizes the transition matrix. Matrices $\mathbf{0}$ and $\mathbf{1}$ are $M \times M$ matrices with all elements being 0 or 1 respectively. Based on the terminating Markov Chain of the transitions, we next derive the probability of a CAV collecting a new update at a time instance t, $w(i,t)$.

For such, if $B(i,0)$ is the initial connectivity of CAV i, based on the phase type distribution, the probability that an update is collected by an CAV i at time t for the first time, $w(i,t)$ is given by Eq. 10.

$$w(i,t) = B(i,0)\left(\prod_{k=1}^{t-1} F_i\overline{\mathbf{B}}\right)(F_i\mathbf{B}) \tag{10}$$

Thus, as seen from Eq. 10, the latency of the first update collection decreases with the increase in the probability of the CVs having model for training.

4 Performance Evaluation

In this section, we use simulations to compare the performance of NDN based system with the IP based system. The purpose of these comparisons is to see if the name-based routing and the inherent caching of NDN network brings advantage over IP network in terms of wireless channel usage, delay in model dispatch, and the resiliency over mobility. Further, the evaluations will compare more extended NDN based scenarios including pure fog environments and hybrid edge-fog environments. The evaluation will assess whether the former brings further improvements to the latter for the NDN based system.

Therefore, the evaluations involves five different setups, namely, IP-CS1, IP-CS4, NDN-CS1, NDN-CS4, and NDN-AV0.4. IP-CS1 and IP-CS4 are cases of IP-based edge-fog hybrid environment. The two setups of NDN-CS1 and NDN-CS4 are for NDN-based edge-fog hybrid environment. In these settings, CS1 indicates that only one edge server is present in the simulation area, whereas CS4 indicates that four edge servers are in the simulation area, each locates at one of the four midpoints of the four subareas. The last setup NDN-AV0.4 is for the pure fog environment, the proposed system. The CAV presence ratio, which is the ratio of number of CAVs to the total cars in the simulation, is set to 0.4.

In this evaluation, communications mimic the update collection task, where CVs send interests to announce new updates. The CAVs will respond in the pure fog scenario presented in this paper, while the edge servers will respond for the update in the hybrid edge-fog environments.

4.1 Simulation Configuration and Evaluation Metrics

A one square kilometer map of urban San Francisco as shown in Fig. 5 is imported to the traffic simulator, Sumo [19] to generate real-world traces of vehicular mobility in the urban setup. The number of cars in the simulation is varied between 20 and 70 for different rounds of simulations. The mobility traces thus generated are then fed to ndnSIM simulator modified for use in VANETs [12] for the rounds of the network performance measures. The simulation time for all these rounds is set to 100 s so as to minimize the cars reaching the end of the road segments of the map within the simulation time. Similarly, the transmission range of every node is limited to 34 m and thus we use multi-hop communications.

In each round of the simulation, a random subset of cars is selected as the set of CVs that participate in actual training process. To analyze the presence of CAVs for the update collection from the CVs in the proposed system, another random subset of the cars is selected as the set of the CAVs. Every other cars are NDN-enabled for NDN based cases and thus can forward the Interest/Data packets. Similarly, the cars in the IP-based simulation are IP-enabled and can route the IP-packets towards the destination.

The following evaluation metrics are used to evaluate the various setups discussed above.

- **Satisfaction Ratio:** The measure of the satisfaction ratio defines the resiliency of a system to the mobility within the network. For NDN-based systems, we define Satisfaction ratio as the overall number of satisfied Interests per total Interests created during the simulation. Similarly, for IP-based systems, it is the ratio of overall number of packets received to the number of request packets created by the client vehicles.
- **Average Delay:** It is a measure of how quickly a request brings back the data packet. It is defined as the ratio of the sum of all the delays for the

Fig. 5. Simulation map

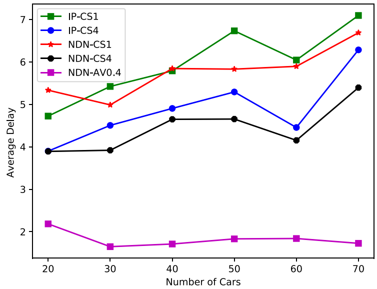

Fig. 6. Average delay in seconds

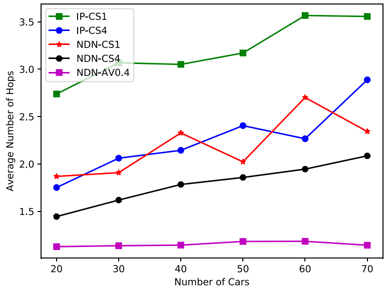

Fig. 7. Average number of hops

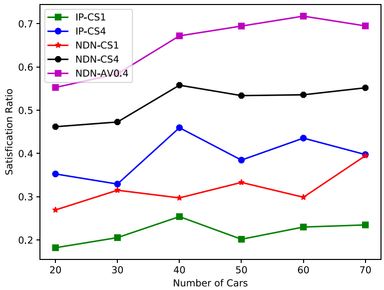

Fig. 8. Satisfaction ratio

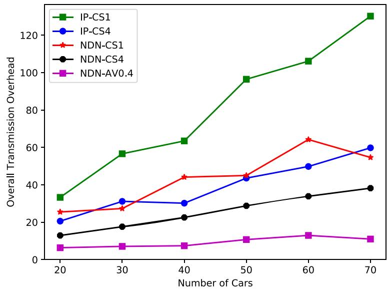

Fig. 9. Overall transmission overhead

satisfied interests to the count of such interests for NDN-based systems. And equivalently, for IP-based systems, it is the ratio of the sum of all the delays for quenched requests to the number of such requests.

– **Average Hop Counts:** Average Hop counts is the average number of hops travelled by the successful requests for the data. It is defined as the ratio of sum of number of hops and the count of satisfied interests for NDN-based

systems. Equivalently, for IP-based systems, it is the ratio of sum of number of hops travelled by the successful requests to the count of such request packets.
- **Overall Transmission Overhead:** This measure defines the exploitation of the wireless channel per completion of successful data transfer. It is the ratio of total number of packets (including the retransmissions) created throughout the simulation time and the total number of the satisfied interests for NDN-based systems. Equivalently, it is the ratio of sum of number of all the request packets created by the clients to the total number of satisfied requests.

4.2 Evaluation Results

In this section, we present the discussion on the results of simulation. As seen from Fig. 6, with single Edge Server setup, the average delay for the satisfied requests for both the IP-based Edge-Fog Hybrid system and the NDN based systems remains similar probably due to a few requests being satisfied, and the inherent caching of NDN not being able to provide substantial improvement in the case of the delay. However, the delay measure shows better performance for the hybrid NDN network when compared to IP based system with the scenario involving 4 edge servers. This is due to the fact that with 4 edge servers, more requests get satisfied, and the inherent caching provided by the NDN network helps in early satisfaction of later requests. While the difference between general NDN and IP are not very drastic, the utilization of the CAVs for the model dispatch to the CVs in pure fog environment brings drastic improvement in the delay measures, due to their mobility-assisted closeness to the CVs.

While the average delay of the satisfied requests between NDN and IP systems are comparable for single Edge Server setup, the average number of hops travelled by such requests differ more drastically. As seen from Fig. 7, for various number of cars in the simulation, the average number of hops is almost always lower by 1 for NDN system in comparision to the IP system. This shows that some of the few satisfied requests for the single edge server setup were quenched from the nearby cache provided by the NDN network. Similar differences can be seen for the four server setup as well. The higher number of edge servers provide nearby data source for both IP system and NDN system for the four server setup, but the presence of caching in NDN further improves the hop counts. On the other hand, the presence of CAVs for the model dispatch in the proposed system has substantial improvement in the hop count as well. This is due to the reason that the CAVs become more reliable close-by data dispatch sources and thus the average number of hops always becomes close to 1.

Figures 8 and 9 present the comparison between NDN and IP for Satisfaction Ratio, and Overall Transmission Overhead, respectively. The Satisfaction Ratio for both NDN and IP systems improve with the presence of more sources in 4 Edge Server setups. Similar to the previous metrics, the NDN-based systems have comparatively better Satisfaction Ratios than IP-based systems. Also, the proposed system has the best Satisfaction Ratio measure among all the compared setups. In terms of transmission overhead, NDN-based systems have lower transmission overhead compared to the IP-based systems as seen from Fig. 9.

Among the five compared setups, the proposed system has the best performance in terms of overhead (lowest value) with almost constant overhead for various number of cars in the simulation.

In a nutshell, based on the evaluation results, we see that the hybrid edge-fog based NDN system has better performance compared to the hybrid edge-fog based IP system in terms of every evaluation metrics discussed above. Similarly, the proposed pure fog-based system involving CAVs has the best performance among all the three setups in all the performance metrics.

5 Conclusion

The paper introduces a NDN-based fog computing system built within the IoV designed for federated learning to be used in vehicular environments. The numerical analyses of the system show a promising prospects for fog computing in such environments. At the same time, the simulation based evaluations prove the usefulness of Named Data Networking over traditional IP networks for such systems.

References

1. Ma, Y., Wang, Z., Yang, H., Yang, L.: Artificial intelligence applications in the development of autonomous vehicles: a survey. IEEE/CAA J. Autom. Sin. **7**(2), 315–329 (2020)
2. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., et al.: Communication-efficient learning of deep networks from decentralized data (2016). arXiv preprint arXiv:1602.05629
3. Du, Z., Wu, C., Yoshinaga, T., Yau, K.-L.A., Ji, Y., Li, J.: Federated learning for vehicular internet of things: recent advances and open issues. IEEE Open J. Comput. Soc. **1**, 45–61 (2020)
4. Lomonaco, V., Maltoni, D.: Core50: a new dataset and benchmark for continuous object recognition (2017). arXiv preprint arXiv:1705.03550
5. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: a review. Neural Netw. **113**, 54–71 (2019)
6. Wu, W., He, L., Lin, W., Mao, R., Maple, C., Jarvis, S.A.: SAFA: a semi-asynchronous protocol for fast federated learning with low overhead. IEEE Trans. Comput. **70**, 655—668 (2020)
7. Zhou, Z., Liu, P., Feng, J., Zhang, Y., Mumtaz, S., Rodriguez, J.: Computation resource allocation and task assignment optimization in vehicular fog computing: a contract-matching approach. IEEE Trans. Veh. Technol. **68**(4), 3113–3125 (2019)
8. Zhu, C., Pastor, G., Xiao, Y., Li, Y., Ylae-Jaeaeski, A.: Fog following me: Latency and quality balanced task allocation in vehicular fog computing. In: 2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), pp. 1–9 (2018)
9. Peng, X., Ota, K., Dong, M.: Multiattribute-based double auction toward resource allocation in vehicular fog computing. IEEE Internet of Things J. **7**(4), 3094–3103 (2020)

10. Meisel, V.P.M., Zhang, L.: Ad hoc networking via named data. In: Fifth ACM International Workshop on Mobility in the Evolving Internet Architecture, ACM, Chicago, pp. 3–8. ACM (2010)
11. Grassi, G., Pesavento, D., Pau, G., Zhang, L., Fdida, S.: Navigo: interest forwarding by geolocations in vehicular named data networking.' In: IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1–10. IEEE (2015)
12. Chowdhury, M., Khan, J.A., Wang, L.: Leveraging content connectivity and location awareness for adaptive forwarding in NDN-based mobile ad hoc networks. In: Proceedings of the 7th ACM Conference on Information-Centric Networking, pp. 59–69 (2020)
13. Grassi, G., et al.: Vanet via named data networking. In: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2014, pp. 410–415. IEEE, Toronto (2014)
14. Meng Kuai, X.H., Yu, Q.: Delay-tolerant forwarding strategy for named data networking in vehicular environment. Int. J. Ad Hoc Ubiquit. Comput. **31**(1), 1 (2019)
15. Kuai, M., Hong, H.: Location-based deferred broadcast for ad-hoc named data networking. Fut. Internet **11**(6), 139 (2019)
16. Hamza-Cherif, A., Boussetta, K., Diaz, G., Lahfa, F.: Performance evaluation and comparative study of main VDTN routing protocols under small-and large-scale scenarios. Ad Hoc Netw. **81**, 122–142 (2018)
17. Shannigrahi, S., Fan, C., Partridge, C.: What's in a name? Naming big science data in named data networking. In: Proceedings of the 7th ACM Conference on Information-Centric Networking, pp. 12–23 (2020)
18. Groenevelt, R., Nain, P., Koole, G.: The message delay in mobile ad hoc networks. Perform. Eval. **62**(1–4), 210–228 (2005)
19. Lopez, P.A., et al.: Microscopic traffic simulation using sumo. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2575–2582. IEEE (2018)