Mobile Netw Appl (2015) 20:773–780
DOI 10.1007/s11036-014-0553-4

QoS-Aware Data Dissemination for Dense Urban Regions in Vehicular Ad Hoc Networks

Amit Dua · Neeraj Kumar · Seema Bawa

Published online: 23 December 2014

Abstract With an evolution of 5G and related technologies, there is a need of high data rate transfer between different communication objects such as vehicles, sensor nodes, or mesh clients connected to the Internet to maintain Quality of Service (QoS) at various levels. As vehicles in Vehicular Ad Hoc Networks (VANETs) are mainly used for safety related applications, so there is a need of an efficient QoS-aware data dissemination for various applications in different regions especially in dense urban regions where congestion is one of the major issues. Hence, to address this issue, in this paper, we propose a new mechanism to maintain QoS for data dissemination among the different vehicles in VANETs. An intelligent forwarding mechanism is used by newly defined metric which assigns weights to the different routing paths from source to destination. Separate algorithms are designed for route construction, and maintenance in the proposed scheme. The performance of the proposed scheme is studied by extensive simulations with respect to various metrics such as delay incurred, percentage of active links, and load on the network. The results obtained show that the proposed scheme is better than the other state-of-the-art existing schemes in the literature.

Keywords VANET · 5G · Congestion control · QoS · Dense urban regions

A. Dua · N. Kumar (✉) · S. Bawa
Department of Computer Science and Engineering,
Thapar University, Patiala Punjab, India
e-mail: neeraj.kumar@thapar.edu

A. Dua
e-mail: mail.amitdua@gmail.com

S. Bawa
e-mail: seema@thapar.edu

1 Introduction

Vehicular ad hoc networks (VANETs) have emerged as a new class of efficient data dissemination from past few years. In VANETs, vehicles act as an intelligent machine for data collection and processing from various other devices connected to the Internet by various short and long range transmissions using IEEE 802.11 a/b/g, and 802.11p/WAVE standards [1, 2]. For communication am-ongst among vehicles and through using RSU, IEEE 802.11p/WAVE standard is formed. In the Wireless Access in Vehicular Environment (WAVE), the data exchange takes place in 5.9 GHz licensed band. For short range and medium range communication, Dedicated Short Range Communication (DSRC) Standards and protocols are used. In 1999, 75 MHz spectrum was allocated for Intelligent Transportation System (ITS) so that users can access various services even with higher velocity. For connecting any device like laptop, Personal Digital Assistants (PDA), and desktops using these standards with Internet, the device must be uniquely identifiable. This unique identification and their virtual representation is termed as Internet of Things (IoT) which is one of the core technologies in 5G networks [4]. Connecting vehicles to 5G is no exception, although its communication requires complex infrastructure and synchronization between multiple partners so that they can access various resources from the Internet seamlessly. The mutual exchange of information between cars for safe driving is one of the applications of IoT in 5G networks [5].

With an advancement of technology and availability of higher frequency bands, there has been an evolution of 4G technology from 3G in which 1Gbit/s speed is offered for short distance communication. In contrast to previous evolutions, there is a less scope of improvement in terms of frequency bands, and channel bandwidth in 5G. With an increase in number of devices connected to internet, there can be extremely dense networks with high data rates required along with other constraints associated in these dense regions. So, there is a requirement of an efficient solution to overcome this problem. 5G aims to improve not only peak bit rates but aiming to solve scalability problem, battery consumption problem, coverage, latency, cost of deploying infrastructure and versatility problems also.

Although many state-of-the-art protocols are already available for routing in VANETs, but these solutions are not adequate to provide QoS to various applications in these networks. The existing solutions in literature can be classified into opportunistic or geographic routing protocols [7–14].

Intelligent Transport System (ITS) integrates advanced electronics, communications, wireless sensor technology, and information for the sake of safety of the passengers [6, 9]. ITS can be broadly classified into two categories as: Road to Vehicle communication (V2I/RVC) and Inter-Vehicle Communication (V2V/IVC).

Keeping focus on all the above points, in this paper, we propose a novel QoS-aware routing algorithm which is quickest to route the packets to the destination from any source. Four parameters are taken for Intelligent Forwarding (IF) which along with reward penalty function is used to assign weight to a link. The calculation to route is computed using Dijksatra algorithm [16] which is described in detail in the coming sections. The protocol works successfully for both sparse and dense regions in VANETs, but its performance is found better for dense urban regions.. The main objective of this paper is to achieve high packet delivery ratio (PDR) in shortest time with minimum overhead. The protocol overcomes broadcast storm, and gray zone problems which exist in most of the earlier solutions. Also, the proposed scheme recovers quickly in case of link failures to avoid fault tolerance issue.

In this paper, we have used a Hybrid architecture in which road side units (RSUs) are installed along the road and vehicles are equipped with Application Unit (AU) for communication amongst themselves and RSUs. VANET is a distributed, self-organized communication network with moving vehicles and fixed RSU.

For communication amongst vehicles, VANET architecture comprises of two main components namely AU, and On board Unit (OBU) which are installed within vehicles. An important component for vehicle to infrastructure communication is RSUs, which may be installed statically along the roads at specific distances. These RSUs may be connected to gateways which in turn connected to the Internet. For vehicle to infrastructure communication, fixed RSUs act like a router and are service providers to the AUs and OBUs installed in vehicles. AUs and OBUs may act as consumers and use wireless standards like IEEE 802.11p and 802.11 a/b/g for communication [15]. AU on one vehicle communicate with AU of other vehicle or to RSU through OBU. These AUs might be integrated with OBU to form a single unit or may be a separate device placed in vehicles. The communication from RSU can be either through Dedicated Short Range Communication (DSRC) or using IEEE 802.11 a/b/g standard as shown in Fig. 1.

The roads considered in the proposed scheme are bidirectional and each direction has 2 lanes. There are N vehicles in the network following Freeway mobility model. With the movement of vehicles, the network topology changes continuously. To minimize the effects of location changes, Mobile Internet Protocol Version 6 (MIPV6) is used.

Fig. 1 Network Model used

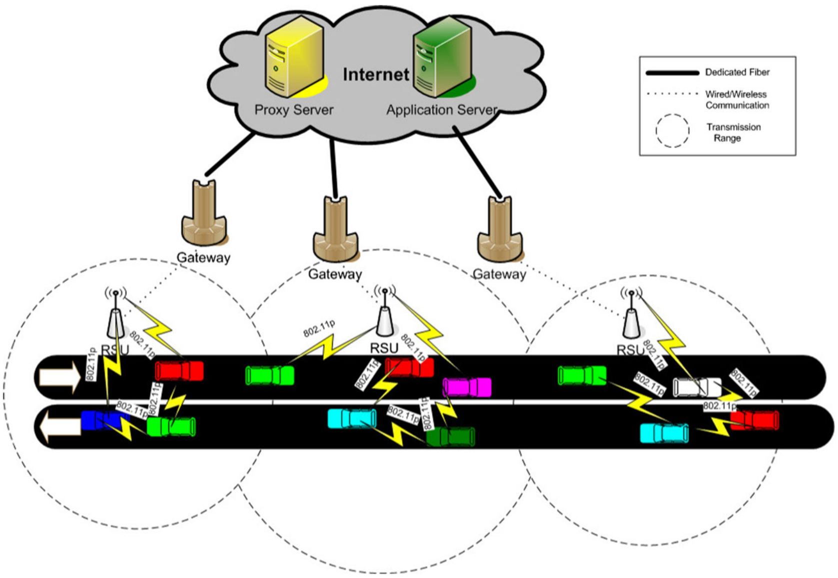

Table 1 Symbols and their meaning

Symbols	Meaning
$RF_{i,j}$	Reward Function of link between i & j
$RP_{i,j}$	Reward penalty value of link between i & j
ζ	Threshold value for RF
ϕ	Threshold time
S,D	Source and Destination
LV	Leaving Vehicle
$\mathrm{IF_{i,j}}$	Intelligent forwarding of link
$Weight_{i,j}$	Calculated weight metric of link
$distance[]$	Cumulative weight from source to destination
$parent[]$	Array used to calculate the best path

IEEE 802.11a supports 8 data rates with a maximum of 54 Mbps and minimum of 6 Mbps. It supports 12 non interfering channels while IEEE 802.11b has 3 non interfering channels with four data rates varying from 1 Mbps to 11 Mbps. Each vehicle has necessary infrastructure to use multiple interfaces from these standards.

Rest of the paper is organized as follows. Section 2 gives the detailed explanation of the proposed scheme. The performance evaluation is discussed in Section 3. Finally, conclusions and future work are described in Section 4.

List of various symbols used in the paper along with their meaning are given in Table 1.

2 Proposed scheme

Based upon the above issues and challenges, this section illustrates the proposed solution to the problems discussed as above. Every node is assumed to have GPS installed to know its position. Though GPS is not accurate, the nodes exchange beacon messages in order to know their speed, direction in which they are moving and exact location. The node keeps data in distributed hash table (DHT) and $< key, value >$ pair is stored for fast searching. The repository is maintained and shared in peer-to-peer manner amongst other vehicles. The proposed solution consists of various phases such as calculation of reward-penalty, intelligent forwarding, weight calculation and construction and maintenance phases which are described in detail in coming section.

2.1 Reward-penalty (RP) function

We have used the concept of reward-penalty for efficient transmission from source to destination which is explained as follows. When a new node enters in to the network, the link connecting to other node is having RP value of 0.5. Vehicles exchange beacon messages while moving to know the neighbors, their location and velocity. For the vehicles within the transmission range of other vehicles, the time taken for successful delivery of hello packet determines whether vehicle gets a reward or a penalty using following function:

$$RewardFunction_{i,j} = \frac{Distance_{i,j}}{LinkSpeed_{i,j}} \tag{1}$$

If the Reward Function (RF) value,i.e., $RF_{i,j} \leq \zeta$, then there is a reward of 10 % to current $RP_{i,j}$. If $RF_v > \zeta$, then there is a penalty of 10 % to current RP_v. The next $RF_{i,j}$ is calculated after ϕ time interval and if $RF_{i,j} \leq \zeta$, there is a reward else a penalty of 10 % . As the vehicles move links will be broken and after some time RP_v for that link becomes very low. Following condition is tested for the same:

$$0 \leq RP_{i,j} \leq 1 \tag{2}$$

If the link persists with good speed and less distance, then $RP_{i,j}$ goes higher. According to Eq. 2, the $RP_{i,j}$ must be less than 1. So the $RP_{i,j}$ approaching to 0 is considered as very poor link and the link whose $RP_{i,j}$ tends to 1 is considered reliable and trusted link for data transmission.

2.2 Intelligent forwarding

There are four factors on which the next link is selected. These are defined as follows.

Link stability: The link must have maximum bandwidth.

Distance: The maximum distance towards the destination.

Speed: The vehicle having maximum speed towards destination is given priority.

Density: The vehicle moving in high density area is given preference over vehicle moving in less dense area. Based upon the above four parameters, we have defined the following:

$$IF_{i,j} = IF_{Max}\left(\frac{\alpha}{LinkStability} + \frac{\beta}{Distance} + \frac{\gamma}{Speed} + \frac{\delta}{Density}\right) \tag{3}$$

The value of *IF* is stored in an array data structure.

2.3 Weight calculation

The routing is done according to weight computed for each link. Each link has some RP_v as computed above and has a IF as computed in Eq. 3. The weight for each link is directly proportional to IF and inversely proportional to RP_v. The calculation of weight for link between vertex i and vertex j is as follows:

$$Weight_{i,j} = \frac{IF_{i,j}}{RP_{i,j}} \tag{4}$$

Using the value of the variable in Eq. 4, packets are sent to their final destination.

2.4 Construction phase

Every vehicle calculates the weight of link according to Eq. 4. These links are exchanged by vehicles along with hello messages. When source vehicle (S) decides to transmit a message to a particular destination (D), it has the complete knowledge of the position and weights of links between vehicles. It uses the modified Dijkstra's algorithm [16] to compute the shortest path based upon weights calculated by the $< Reward$, IF $>$ value.

As described in Algorithm 1, first the distance of S is assigned zero as it is the origination vehicle, rest all links are assigned very high value. All the links are added to a priority Queue Q and covered nodes are included into array u. The priority queue is preemptive in nature as with changing network the queue also changes. For each adjacent vertex of v, the vehicle having minimum weight is assigned u. The combined weight of its parent in v and the weight of vertex is assigned X. If this X is less than the distance of u, then distance of u is assigned to X and parent of u is assigned to v. This process continues till destination is not reached. Once destination is reached, then the distance vector and parent vector are returned from which the path from source and destination is computed.

Algorithm 1 Shortest path from Source to Destination

Input: List of Vertices(V),Source (S), Destination(D),Weight of Connecting Edges

Output: Shortest path from S to D

```
 1: distance[$S$] = 0
 2: for ($i = 0toV - 1$) do
 3:   if ($v_i \neq S$) then
 4:     $distance[i] = \infty$
 5:     $parent[i] = Null$
 6:   end if
 7:   Create a priority Queue Q for all vertex according distance[ ]
 8: end for
 9: v = Extract minimum from Q
10: while ($Q \neq \phi \&\& v \neq D$) do
11:   Create a Queue ($Q_2$) of adjacent vertex of v
12:   while ($Q_2 \neq \phi$) do
13:     u = Extract a vertex from $Q_2$
14:     X= distance[v] + weight(u,v)
15:     if ($X < distance(u)$) then
16:       $distance[u] = X$
17:       parent[u] = v
18:     end if
19:   end while
20:   v = Extract minimum from Q
21: end while
22: return distance[ ] and parent[ ]
```

2.4.1 Time complexity

The first phase (lines number 1–8), i.e., updating the distance and parent for all the vertices take $O(V)$ time. Line number 9 takes $O(log(V))$ time. While loop from line 10 to 20 runs for all the vertices, i.e., V times. Line number 11 takes O(1) time and it is being in while loop runs for V times. While loop from line number 12-19 again runs for V iterations. Within it is decrease key operation which takes $O(logV)$ time. Extract minimum operation in line number 20 takes $O(log(V))$ time, which is repeated V times as it is in while loop.

So, the total complexity of the algorithm 1 is as follows:

$$\begin{aligned} Complexity &= O(V) + O(log(V)) \\ &\quad + V\left(O(log(V)) + O(Vlog(V))\right) \qquad (5) \\ &= O(V + log(V) + Vlog(V) + V^2 logV) \qquad (6) \end{aligned}$$

as $E = V \times (V - 1)$

$E = O(V^2)$. O(V) is very small as compared to $O(V^2)$ hence neglecting it and putting the value in Eq. 6

$$\begin{aligned} Complexity &= O\left(V + (V + E)\,(logV)\right) \qquad (7) \\ &= O\left((V + E)(logV)\right) \qquad (8) \end{aligned}$$

Hence, the total complexity is $O((V + E)logV)$

2.4.2 Space complexity

Initially, there is a requirement to store data structures namely $weight$. The space complexity for $weight$ data structure is $O(V^2)$.

The $distance$ data structure has two components in each entity namely link to parent and the value itself. The $distance$ data structure is for all the vertices hence, its space complexity is $O(V)$. The extra space consumed by $parent$ array is also of $O(V)$

Priority queue is explored for all adjacent vertices taking $O(V)$ space.The while loops from line 10 to line number 21 and from line number 12 to 19, use just comparing the values which takes fixed space.

The space complexity is $O(V) + O(V) + O(V^2)$, hence total space complexity is $O(V^2)$.

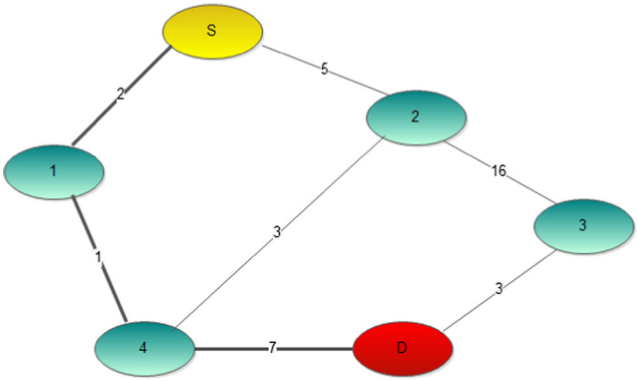

Fig. 2 Shortest path using for Algorithm 1

2.4.3 An example

Consider a scenario shown in Fig. 2, where source S (marked in yellow) has to transmit message to destination D (marked in red). The weight metrics calculated according to Eq. 4 are marked on each edge. The step by step calculation according to algorithm 1 is depicted in Fig. 3. Each node has been marked with two parameters as: its distance from source and its parent. Initially all vehicles, except source, are having distance ∞ and parent as $Null$. In second step, source distance is minimum with $Null$ as its parent. The distance of other vehicles from reached vehicle is calculated. In the third step, vehicle 1 is having minimum distance whose parent is S. Similarly in forth step, vehicle 4 with parent 1 and in fifth step vehicle 2 with parent as source are selected. In last step, Destination vehicle, D, is reached with distance 10 and vehicle 4 as parent. Now by calculating parent $D \rightarrow 4 \rightarrow 1 \rightarrow S$, the route is constructed.

2.5 Maintenance phase

There are two conditions that have have been considered in this phase:

- When an intermediate vehicle in the shortest path leaves the network or because of any reason one link breaks, or
- A better path than the existing path is available

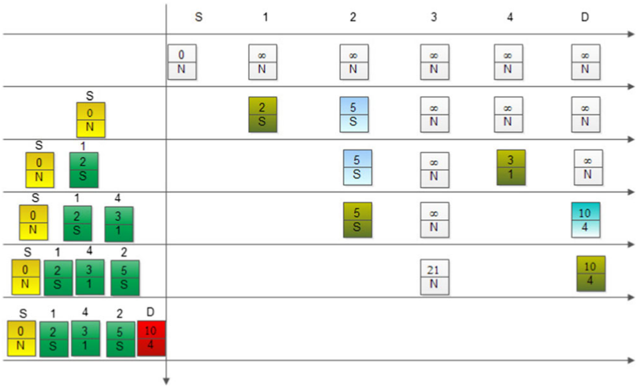

Fig. 3 Calculation of shortest path for graph used in Fig. 2

In first case, when an intermediate vehicle leaves the network or link breaks, then the path from the source to the vehicle previous to the leaving vehicle (LV) is the shortest. We have calculated the shortest path from LV to destination using Algorithm 1. It may also be possible that an alternate path from source to destination is shorter. So which ever of the two paths is shorter the packets are sent through that path. The sequence of actions for the same is explained in Algorithm 2.

Algorithm 2 Shortest path after link breakage

```
Input:Shortest path from source till node previous to (LV)
Output: Shortest path from source to destination
Calculate the shortest path from node previous to LV to destination using algorithm 1
($distance_1[S, D] = distance[S, LV] + distance[LV, D]$)
recalculate $distance_2[S, D]$ from Source using algorithm 1
if ($distance_1[S, D] < distance_2[S, D]$) then
    Use $diatance[] = distance_1[S, D]$
else
    Use $distnce[] = distance_2[S, D]$
end if
return distance[ ]
```

In second case, when there is a better path available, the source checks the optimal path after every ψ time and follows the best resultant path explained in Algorithm 3.

Algorithm 3 Updation of shortest path

```
Input: Existing path $distance_{ex}[S, D]$
Output: Shortest path from Source and Destination
Calculate shortest path $distance_{new}[S, D]$from algorithm 1 after every $\psi$ time
if ($distance_{new}[S, D] < distance_{ex}$) then
    $distance[] = distance_{new}$
else
    $distance[] = distance_{ex}$
end if
return distance [ ]
```

3 Performance evaluation

3.1 Simulation settings

To test the performance of the proposed scheme, extensive simulations are done using NS2.35 [17] and sumo [19]. The area used is roads of Patiala city of India. The map of 6 Km × 4 Km is taken from OpenStreetMap [18] and SUMO [19] is used to generate vehicle movements. Each road is bidirectional and all major roads are having two lanes in each direction. The vehicle moves with a velocity ranging from 10 Km/hr to 100 Km/hr with a data rate of 2 Mbps. The number of vehicles in a cluster varies from 100 to 500 which takes care of both sparse and dense scenarios. There are 623 intersections and 1534 number of road segments. Each simulation runs for a total of 900 seconds. The protocol is compared with DDOR [8] and SOBP [10] because both of these protocols have considered similar conditions as used in the proposed scheme. Following are the routing metrics against which the protocol is compared and evaluated:

- **Delay incurred:** This is the amount of time a packet takes from generation at source to reception at destination.
- **Percentage number of active links:** This parameter indicates the percentage of total links to links active during transmission.
- **Load on network:** This parameter indicates the average load of all the links in the network.

3.2 Results and discussions

Figures 4 and 5 illustrate the variation of delay incurred and % number of active links which is measured with changing speed of vehicles and number of vehicles in a cluster. Figure 6 show the comparison of SOBP [10] and DDOR [8] with the proposed protocol.

3.2.1 Delay incurred

With an increase in speed of vehicles, delay incurred increases mainly because of frequent breakage of links and changing topology. As density increases, the delay incurred decreases showing the importance of availability of routes with better link quality. The size of cluster is 20-30 vehicles which vary dynamically. The formation of cluster, election of cluster head, maintenance and communication is performed according to ALCA [3]. The delay incurred is as low as only 3 seconds when number of vehicles in a cluster is 400 and speed of vehicles is 10 Km/hr which goes as high as 24 seconds when the network is sparse and speed of vehicles is 100 Km/hr. The variation of delay incurred is depicted in Fig. 4.

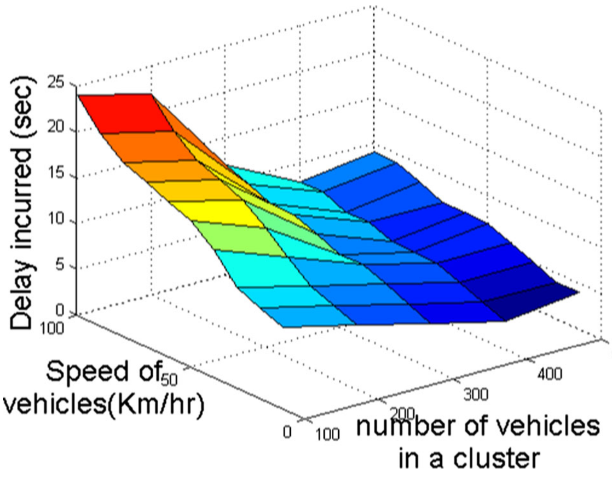

Fig. 4 Delay incurred

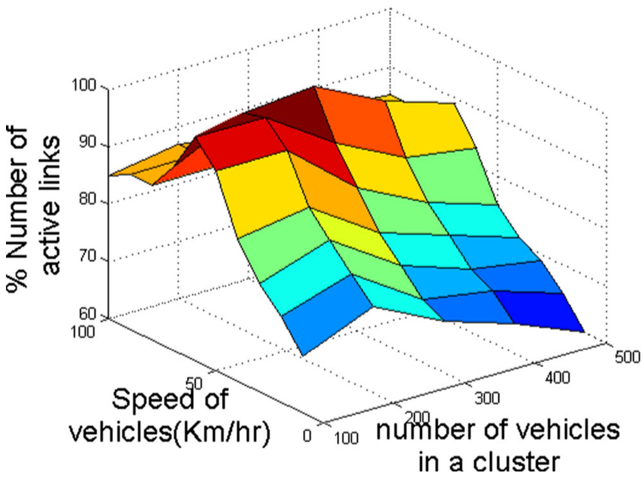

Fig. 5 Percentage of active links

3.2.2 % number of active links

As illustrated in Fig. 5, the percentage of active links is low with very less number of vehicles in cluster because of less links as most of the vehicles using store-carry-forward mechanism. As the speed and number of vehicles increase, the percentage of active links also increases which attains a saturation 99 % at 70 Km/hr speed and 300 vehicles after which it again decreases because of increase in turbulence in the network. Percentage number of active links decreases to 85 %, when there are 500 vehicles in cluster and speed is 100 Km/hr.

3.2.3 Load on network

Load on the network increases with increase in the average velocity and number of vehicles in cluster. But the load is considerably lower in the proposed scheme as it uses link stability, velocity, density and distance while calculating the weight of a link for transmitting the message. The load is 1425 Kbps for SOBP and 1140 Kbps for DDOR when average velocity is 100 Km/hr which is much higher than 760 Kbps of proposed protocol at similar conditions. The load of proposed protocol is 950 Kbps when there are 500 vehicles in cluster but load is 1710 Kbps and 1330 Kbps for SOBP

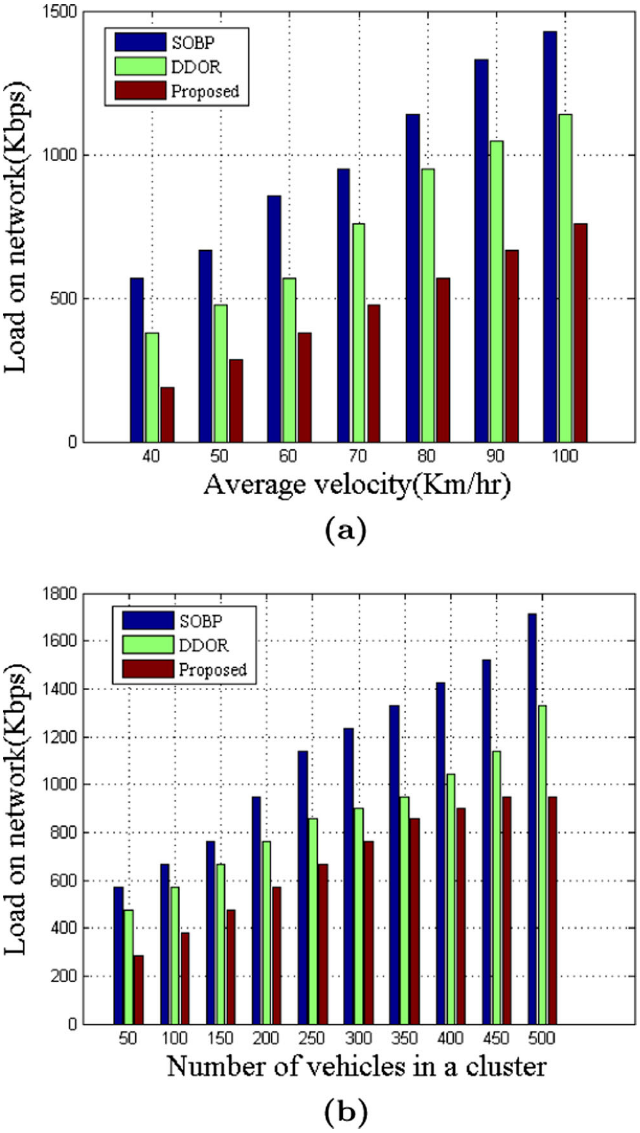

Fig. 6 Comparison of proposed protocol with SOBP and DDOR against (**a**) Load on the network with varying velocity (**b**) Load on the network with increasing number of vehicles in a cluster

and DDOR respectively in same conditions. The variations of load on the network are illustrated in Fig. 6a, b.

4 Conclusion and futute work

With an evolution of 3G, 4G technologies, users are still demanding high data rates for most of the applications they are using especially, when they are going from one place to another. In this regard, it is a challenging task to maintain the QoS for various applications in vehicular adhoc networks (VANETs). To address this issue, in this paper, we propose a new QoS-aware data dissemination for dense urban regions in VANETs. As the safety messages are the most important messages in VANETs, so to choose the best destination from a source, a new intelligent forwarding mechanism is proposed. Based upon this newly defined metric, new algorithms for route construction, and maintenance are designed in the proposed scheme. Finally, performance evaluation of the proposed scheme is done using extensive simulations with respect to various metrics. The results obtained show that the proposed scheme is effective in maintaining QoS for various applications in the upcoming 5G technology in VANETs.

In the future, we would like to explore security features of data dissemination in VANETs. Moreover, geographical protocols for data dissemination with respect to limited connectivity would also be explored.

Acknowledgements We would like to thank the anonymous reviewers for their constructive suggestions and comments which have greatly helped us to improve the content, quality, and presentation of this paper. The work presented in this paper is supported by the research grant from TCS, New Delhi (India).

References

1. Dua A, Kumar N, Bawa S (2014) A systematic review on routing protocols for vehicular ad hoc networks. Veh Commun 1(1):33–52
2. Kumar N, Chilamkurti N, Rodrigues JJPC (2014) Learning automata-based opportunistic data aggregation and forwarding scheme for alert generation in vehicular ad hoc networks. Comput. Commun. 59(1):22–32
3. Kumar N, Chilamkurti N, Park JH (2013) ALCA: agent learning-based clustering algorithm in vehicular ad hoc networks. Pers Ubiquit Comput 17(8):1683–1692
4. Spiekermann S, Pallas F (2006) Technology paternalism wider implications of ubiquitous computing. Poiesis Prax 4(1):6–18
5. Mattern F, Floerkemeier C (2010) From the internet of computers to the internet of things. Informatik-Spektrum 33(2):107–121
6. Luo J, Hubaux JP (2006) A survey of inter-vehicle communication Embedded Security in Cars 2004:111–122
7. Yang Q, Lim A, Li S, Fang J, Agrawal P (2010) ACAR: adaptive connectivity aware routingfor vehicular ad hoc networks in city scenarios. Mob Netw Appl 15(1):36–60
8. Sahu PK, Wu EHK, Sahoo J, Gerla M (2012) DDOR: Destination discovery oriented routing in highway/freeway VANETs. Telecommun Syst 50(4):267–284
9. Chen YS, Lin YW, Pan CY (2011) DIR: diagonal-intersection-based routing protocol for vehicular ad hoc networks. Telecommun Syst 46(4):299–316
10. Li G, Wang W, Yao X, Chen W (2013) SOBP: a sender-designated opportunistic broadcast protocol for VANET. Telecommun Syst 53(4):453–467
11. Li M, Zeng K, Lou W (2011) Opportunistic broadcast of event-driven warning messages in Vehicular Ad Hoc Networks with lossy links. Comput Netw 55(10):2443–2464
12. Wang T, Wang G (2010) TIBCRPH: trafc infrastructure based cluster routing protocol with handoff in VANET, In Proceedings of the 19th annual on wireless and optical communications conference (WOCC'10), vol 2010, pp 1–5

13. Lee KC, Lee U, Gerla M (2009) TO-GO: TOpology-assist geo-opportunistic routing in urban vehicular grids, In Proceedings of the 6th international conference on wireless on-demand network systems and services, WONS'09, vol 2009, pp 11–18
14. Shrestha RK, Sangman M, Ilyong C, Dongmin C (2010) Vertex-based multihop vehicle-to-infrastructure routing for vehicular ad hoc networks. In Proceedings of the 43rd Hawaii international conference on system sciences (HICSS'10), vol 2010, pp 1–7
15. Li F, Wang Y (2007) Routing in vehicular ad hoc networks: A survey. IEEE Veh Technol Mag 2(2):12–22
16. Cormen TH, Leiserson CE, Rivest RL, Stein C (2001) Introduction to Algorithms In 2nd ed. Cambridge, MA: MIT Press, Section 24.3:Dijkstra algorithm, vol 2001, pp 595–601
17. NS2.35. [Online]. Available: http://www.isi.edu/nsnam/ns/
18. OpenStreetMap. [Online].Available: http://www.openstreetmap.org/
19. SUMO Simulation of Urban Mobility. [Online]. Available: http://sumo.sourceforge.net/