E-NC :PSO based Enforced Network Coding in Vehicular Networks

Pratap Kumar Sahu, Abdelhakim Hafid
DIRO, University of Montreal,
Montreal (QC), Canada
{sahupk, ahafid}@iro.umontreal.ca

Jagruti Sahoo
CIISE, Concordia University,
Montreal (QC), Canada,
jagrutiss@gmail.com

Soumaya Cherkaoui
Dept. of ECE, University of Sherbrooke,
Sherbrooke (QC), Canada
Soumaya.Cherkaoui@USherbrooke.ca

Abstract **In wireless networks, spectral efficiency is a vital issue in the wake of ever increasing bandwidth usage. To improve throughput, in wired and wireless networks, network coding is one of the promising solutions which has been used for a quite some time. On getting coding opportunity in a hop, coding gain is around 100% if physical layer networking, unlike linear network coding which offers maximum coding gain upto 33%. However, PNC has symbol-level synchronization, carrier-frequency synchronization, and carrier-phase synchronization. The throughput elevate with number of nodes involved in physical layer network coding. In the literature, there is no proposal which increases the involvement of number nodes to elevate overall throughput of the network. In this paper, we are forcing multiple opposite direction routes to share common road segments without compromising delay and reliability requirements. We designed a new particle swarm optimization (PSO) mechanism which offers maximum throughput gain for the entire network. Through network coding such voluntary congestion is solved. Thus, higher packet delivery ratio can be achieved while keeping lower end to end delay which is verified by graphs.**

Keywords— Congestion Control; Network Coding; Unicast Routing; Spectral Efficiency

I. INTRODUCTION

Network coding [1] has been studied extensively in wireless networks to improve throughput of unicast routing. Traditionally, network coding is applied to packets that belong to different unicast flows. The network coding which involves packets is referred to as digital network coding (DNC) or packet level network coding. The intermediate relay nodes perform XOR operation on multiple input packets and produce multiple output coded packets. Network Coding can be applied in wireless sensor networks (WSN), wireless mesh networks (WMN) and vehicular networks in different ways. In WSN, the resources such as battery power, memory and computational capabilities are the major constraints. Thus, general purpose network coding approaches may not be suitable for WSNs as in most schemes, nodes need to store overheard packets [1][2][3] to enhance throughput. In wireless mesh networks, the static nature of routers is exploited to design network coding schemes. MIXIT[5] is a layered architecture for bulk transfer over static mesh networks. The network coding is performed at the granularity of symbols.

Unicast routing [4] [6][7][8][17]is widely used as a data delivery model in vehicular networks for both safety and infotainment applications. Generally, the content in vehicular networks needs to be delivered to a destination few hops away from the source. . In a city, multiple sources may communicate at the same time. This may lead to considerable overhead to the entire network. The congestion of entire network matters when emergency messages have to wait to avail bandwidth. Apart from congestion, unicast routing in vehicular networks has to deal with network fragmentation. Due to network fragmentation (i.e. longer store and carry period) end-to-end delay increases in a routing path. If connectivity becomes the parameter to decide on a routing path to avoid fragmentation, increased number of routing paths may decide to pass through road segments with higher connectivity. In the race to become part of connected paths, many of the routing paths may share same road segments. As a consequence, severe contention results in the shared road segments. Packets are dropped due to increased collisions. Thus, delivery is delayed at the destinations which eventually lead to a decrease in throughput. This is the ideal scenario for leveraging network coding to improve throughput in vehicular networks.

In the traditional method, delay computation for all routing paths is not accurate as delay for each routing path is computed independent from others. The real delay depends upon the interactions of routing paths. Thus it is necessary to compute the delay of every routing path in consideration with other routing paths. In this paper we apply linear network coding to improve the throughput of unicast routing in vehicular networks. We have proposed a new particle swarm optimization method to enforce networking which increases throughput and reduces overall delay. Initially, routing paths are fetched based on Dijkstra's algorithm (i.e. minimum cost path). The cost of each path is chosen to be end-to-end delay between source destination pair. Paths are grouped based on their distance to their nearest neighbors. After that, our PSO algorithm is executed to find all possible routing paths with higher throughput and minimum delay. Fig 1 shows a traditional vehicular environment of a busy street which is ideal for network coding. The traditional routing protocols [4][6][7][8] run similar to Fig 2. In Fig 3, routes are forced to share one of the busiest road segments. Thus, coding opportunity is generated and this reduces total network load of the entire network.

The rest of the paper is organized as follows. A brief discussion of related work is presented in Section 2. Section 3 describes the proposed mechanism. Section 4 presents performance evaluation. Finally, Section 5 concludes the paper.

978-1-4673-9292-1/16/$31.00 ©2016 IEEE

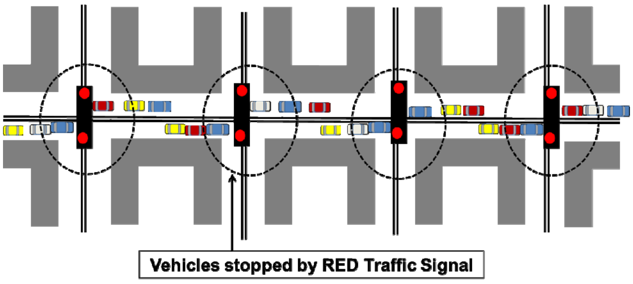

Fig 1. Vehicular Scenario in busy streets

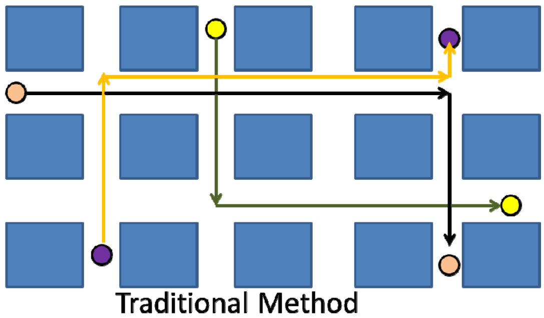

Fig 2. Routing in Traditional Methods

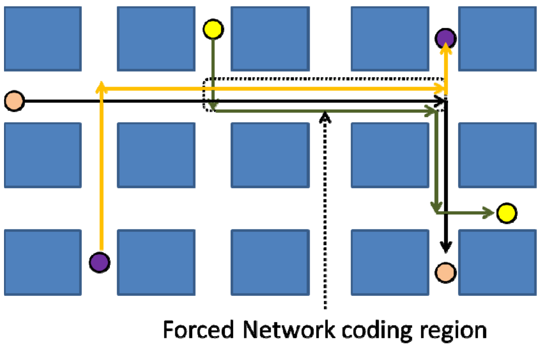

Fig 3. Routing through Forced Network Coding

II. Related Works and Motivation

Network coding (i.e. DNC) was first proposed for wired networks (e.g. butterfly network scenario) in [1]. The authors did show that multicast capacity is achieved by allowing the routers to perform network coding operation on different forwarded packets. The network coding operation is basically a bit -wise XOR operation which can also be viewed as linear coding over the finite field Gaulois Field GF (2^n). Later, in [9], Li et al. show that, for multicast traffic, it is possible to achieve the maximum capacity bounds using linear codes while coding and decoding can be performed in polynomial time. In[10], a distributed random linear network coding is proposed where routers pick random coefficients from a finite field without knowing network topology a -priori. The authors proved that with a sufficiently large field size, nodes can decode the XOR coded packets with high probability. The practical use of network coding for wireless networks has been investigated thoroughly in the literature [11][12]. The network coding we discussed so far falls under the category of DNC.

On busy roads, some of the vehicles always wait for red signal to be green to make a go in a round robin fashion (see Fig. 1). These busy roads allow for huge amount of data to be forwarded as these road segments share route s with multiple routing flows. Vehicular networks make use of IEEE 802.11p which supports multiple channels namely control channel and service channels. Using control channel message exchanges, one of the vehicles can be selected as the position bearer of an intersection. Again, using control message exchanges, a new vehicle can be chosen as the position bearer once the old position bearer makes a move. There is always a node to take care of routing packets in an intersection as a static node though virtual. A multi -hop network coding would be ideal for such situations[2][11]. Thus, we create scenarios where multiple routes share parts of their routes with each other. It is quite beneficial in terms of throughput for routing messages when data and acknowledgements share the common path in different directions. Here, messages from opposite directions exploit coding opportunities which cannot be addressed in existing methods while it is possible through our proposed solution.

Unlike multicast routing where entire network is considered for evaluation, in traditional unicast routing, a route usually does not consider other routes for computation. In reality, all the computation seems to be unreal if a route does not involve other road segments (i.e. edges) or intersections (i.e. vertices) which they may be sharing with other routes. Though ours' is a single source and single destination approach, we consider entire network for the network load and use particle swarm optimization to find best possible route selections.

In vehicular networks, multicast routing optimization primarily involves bio-inspired metaheuristics. A Bee-life Algorithm (BLA) [13] is proposed to solve a multi-constraint, multi-objective optimization, where the objectives include cost, jitter, delay, and bandwidth, while the constraints include maximum allowed delay, maximum allowed jitter, and maximum requested bandwidth. The authors use bee colony algorithms to solve the QoS multicast routing problem. The limitation of this work is that the weighted-sum approach is used to solve the problem as a single-objective optimization problem, and hence, lack pareto-optimal solutions. Also, the link cost simply denotes the road distance, which is not important in vehicular networks. This is because a shortest path in terms of road distance is not necessarily the least hop-count path due to high dynamics in vehicular networks. Another drawback is that the paper [13] does not discuss how a network node computes the several objective functions and constraints. Multicast with Ant Colony Optimization for VANETs (MAV-AODV) [14] uses ant-colony optimization to improve reliability of the MAODV scheme. Basically, the mobility information of vehicles is incorporated in the optimization algorithm to find a stable multicast tree for VANETs. It is important to note that both BLA and MAV-AODV find multicast trees where the links denote V2V communication links i.e. wireless link between two vehicles. In contrast, we consider the road segment as a link of the multicast tree. Such a consideration ensures connectivity up to a certain number of hops as opposed to only one hop in the case of BLA and MAV-AODV.

III. SYSTEM MODEL

A. Problem Formulation

Given M source-destination pairs, we have M routing paths. The objective is to find an optimal grouping of the M routing paths that minimizes the average end-to-end delay. Delay is minimized when routing paths are forced to share road segments with each other and the packets are transmitted using network coding. The higher the share, the higher is the possibility of applying network coding and the lower is the end-to-end delay. Hence, our objective is to find an optimal grouping of the routing paths that minimizes the delay.

Objective (f): The end-to-end delay (D_i) of any source-destination pair i given a particular grouping of M pairs.

B. Particle Swarm Optimization (PSO) based Heuristic

PSO is a metaheuristic based on evolutionary computing to solve optimization problems. The problem we address is a discrete combinatorial optimization problem. PSO was originally developed for continuous domain. Although, many discrete variants of PSO have been proposed in the literature, the design of PSO is still specific to the problem domain. In this paper, we design a PSO algorithm to solve the optimal grouping problem. More specifically, we propose new particle encoding and velocity operator. Moreover, we propose a grouping heuristic to find an initial grouping of routing paths. PSO starts with initializing a swarm of particles, where each particle denotes a candidate grouping of the routing paths. We apply the grouping heuristic to obtain the particles. Then, during each iteration, the particles are evaluated and compared to select local best and global best position. Local best position refers to the best position a particle has so far. Global best position is the best position in the swarm. Velocities of the particles are updated using their local best and global best positions. The updated velocity operator is then used to update the position. PSO continues till a maximum number of iterations is reached. The proposed PSO algorithm is described in detail in the following sub-sections.

1) Particle Encoding: Solution representation is critical to the efficiency of the search process. In PSO, each particle i has two vectors, position vector X_i and velocity vector V_i. The position vector X_i is encoded as an array of positive integers representing vertices:

$$X_i = \begin{bmatrix} x_{i11} & x_{i21} & \cdots & & x_{iN1} \\ x_{i12} & x_{i22} & \cdots & \cdots & x_{iN2} \\ \cdots & \cdots & \cdots & \cdots & \\ x_{i1M} & x_{i2M} & \cdots & \cdots & x_{iNM} \end{bmatrix}$$

where, N_k denotes the number of vertices in the routing path for source-destination pair k. Each row represents the routing path for a source-destination pair. The position component x_{ijk} denotes the jth vertex along the shortest path for source-destination pair k. To encode velocity for each position component x_{ijk}, we consider the distance (a hop-metric) between the j^{th} vertex of routing path k and the vertex on the path p closest to path k where path k and the path p belongs to same group. Thus, velocity vector V_i is encoded similar to X_i as:

$$V_i = \begin{bmatrix} v_{i11} & v_{i21} & \cdots & & v_{iN1} \\ v_{i12} & v_{i22} & \cdots & \cdots & v_{iN2} \\ \cdots & \cdots & \cdots & \cdots & \\ v_{i1M} & v_{i2M} & \cdots & \cdots & v_{iNM} \end{bmatrix}$$

More specifically, the velocity v_{ijk} denotes a hop metric for the vertex represented by position x_{ijk}. The hop-metric calculated as follows.

Algorithm 1
Find Average Distance Between two Routing paths

1. *List1* and *List2* are list of nodes of two routing paths
2. *List1*={s1,x11, x12, …d1} and *List2*={ s2,x21, x22, …d2}. Here, s1, s2 are source nodes and d1,d2 are destination nodes respectively. x11, x12, x1, and x22 etc. are intermediate intersection positions.
3. For each member LS_i of *List1*
   a. Find a member LS_j of *List2* which is closest
   b. Add a member (LS_i, LS_j) in Tuple [].
4. For each member LS_j of *List2*
   a. Find a member LS_i of *List1* which is closest
   b. Add a member (LS_i, LS_i) in Tuple [].
5. TS=Sizeof (Tuple[])
6. For each member *m=1* to TS
   a. Sum=Distance (*m1,* m2)./* where *m1* and *m2* are two nodes of two different paths.*/
7. Avg_Dist=Sum/TS

2) Swarm Initialization Initially, PSO selects a random swarm of particles. The position vector is initialized randomly. Afterwards, for each particle, a heuristic shown in Algorithm 1 is used to group the routing paths. Then the velocity of a each particle component (i.e. a vertex on a routing path) is intialized to the distance between itself and the nearest vertex on another routing path in the same group. Algorithm 1 is also used to find the average distance between two routing paths.

3) Velocity Update: Because of discrete problem space, the velocity update equations of continuous PSO cannot be used. In continuous PSO, the velocity of a particle is updated by the following equation:

$$v_{ijk}^{new} = wv_{ijk} + c_1 r_1 (l_{ijk} - x_{ijk}) + c_2 r_2 (g_{ijk} - x_{ijk}) \quad (1)$$

Where w is the inertia weight linearly decreasing from 0.9 to 0.4 during running time, c1 and c2 are acceleration coefficients set as 2.0, r1 and r2 are random values in the range of [0, 1]. l_{ijk} and g_{ijk} are local best position and global best position respectively. For our discrete problem, the first velocity term remain unchanged as it represents a hop-metric value. However, the second and third velocity terms indicate invalid values i.e. difference between two vertex IDs will mean nothing. Thus, new approach is needed to find these terms. In our approach, the difference $l_{ijk} - x_{ijk}$ denotes the number of hops of the shortest path between l_{ijk} and x_{ijk}. Similarly, the difference $g_{ijk} - x_{ijk}$ denotes the number of hops of the shortest path between g_{ijk} and x_{ijk}.

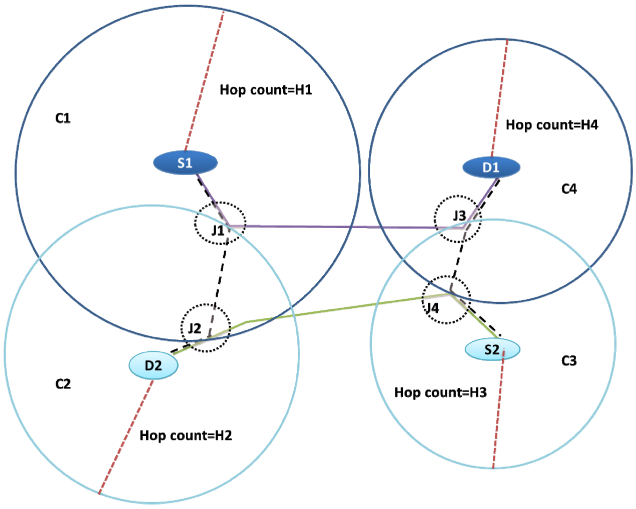

Fig 4. Applying Position Update

4) Position Update Once, the velocity of a particle is updated, the position is updated using the updated velocity. For continuous PSO, the velocity operator is usually the addition operator which is applied as:

$$x_{ijk}^{new} = x_{ijk} + v_{ijk}^{new} \tag{2}$$

In our case, the operator "+" indicates a heuristic which applies the hop-metric value to the vertex x_{ijk} to obtain a new vertex x_{ijk}^{new}. The operator is designed as follows.

5) Objective Function Evaluation

In order to determine the fitness of a solution (i.e. particle), we compute the objective f1 i.e. end-to-end delay. We determine the end-to-end delay of each routing path. Basically, we compute the average delay in one hop. We then compute average end-to-end delay as the sum of delays in each hop of the routing path.

The average end-to-end delay, *D*, is expressed as:

$$D = h * D_h \tag{3}$$

where D_h denotes the average delay in one hop and *h* is the hop-count. Average delay in one hop comprises of three delay components [16]: average transmission delay (denoted by D_{tx}), average queuing delay (denoted by D_q) and average contention delay (denoted by D_c).

$$D_h = D_{tx} + D_q + D_c \tag{4}$$

Average transmission delay D_{tx} includes delay due to back-off, delay when packet is transmitted successfully and delay when retransmission occurs. D_{tx} is given as follows:

$$D_{tx} = t_{bk} + t_s + u * t_{col} \tag{5}$$

where t_{bk}, t_s, u and t_{col} denote back-off delay, delay of transmitting *B* bytes packet, number of unsuccessful transmission and collision duration respectively. tbk is average number of back-off slots multiplied by duration of single slot. We assume t_{col} is same as the transmission duration t_s. u is expressed as:

$$u = \sum_{k=1}^{r} k * p^k * (1-p) + (r+1) * p^{(r+1)} \tag{6}$$

where p is the collision probability, k is the random variable representing the number of attempts performed for the correct transmission of a given frame and r is the maximum retry limit which depends on the frame size and is specified by the IEEE 802.11 standard. Next, we determine the sum D_q + D_c. We model a vehicle as a discrete time M/M/1/K queue [16]. The packet arrival and service rate follow exponential distribution with parameter λ and μ respectively. The size of the queue is limited by the value K; when the queue contains *K* packets, every new packet is discarded. The queue is a classical FIFO (First in First Out). We assume no use of RTS and CTS messages; however, the analysis can be easily extended for the cases where RTS/CTS are present [19].

When $\mu > \lambda$, the service rate of the node is higher than the arrival rate; thus, there will be no accumulation in the queue leading to null queueing and contention delay. When, $\mu < \lambda$, the sum $D_q + D_c$ is bounded by the following maximum delay:

$$D_{\max} = \lim_{\rho \to \infty} (D_q + D_c) \approx \frac{K}{\mu} \tag{7}$$

Algorithm 2 (+ Operator)
1. List (L) the intersection nodes starting from node with Position x_{ijk} with a radius of v_{ijk} . 2. For each element(l_i)of the List L   a. Use Algorithm1 to find the distance (d_i) of its nearest neighbor node which is part of a routing path (must be member of same group) 3. End For 4. Sort all l_i in ascending order of d_i . 5. Choose randomly a l_i from list from top 5 members of the list. 6. Assign l_i to be new x_{ijk} 7. Update x_{ijk}

The service rate of the node is obtained as follows: We know the arrival rate to each node. If B denotes the total bandwidth, then a node can use B/3 since it has to contend with nodes on its either direction. If λ is the arrival rate at a node, then the service rate μ is obtained as B/3- λ. However, when a relay node performs network coding, it sends one coded packet instead of two separate packets. As a result, the service rate gets doubled i.e. μ= 2(B/3- λ).

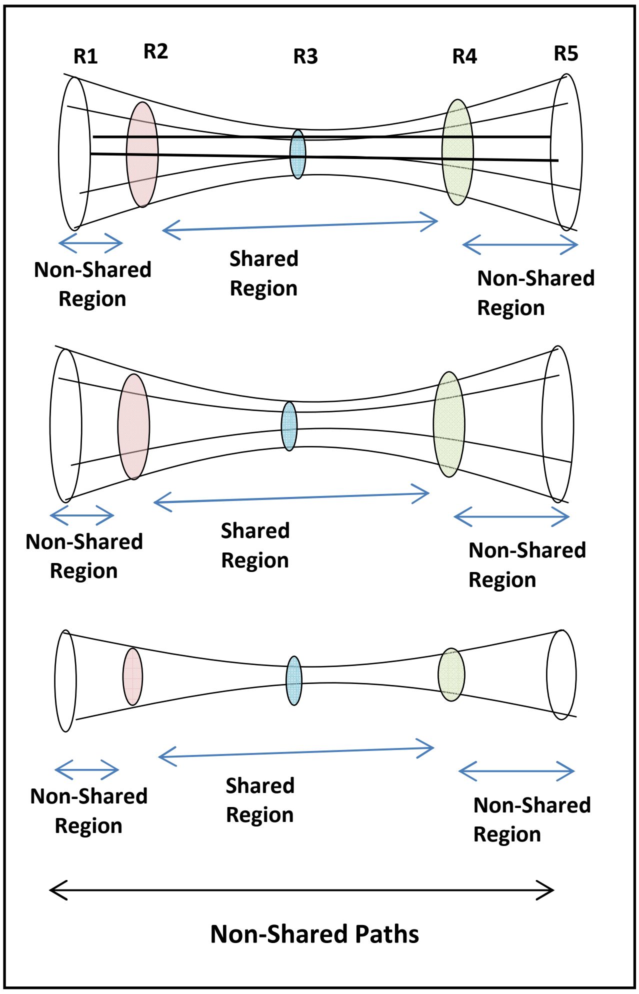

Fig. 5. Routing paths in shared and non-shared regions.

C. Throughput Analysis

In this section, we analyze upper bound on average per-node throughput for the E-NC and W/o NC (Without network coding) scheme.

Per-node throughput of a node is defined as the data rate at which each node can send or receive data. The wireless bandwidth is denoted as *W* bits/sec. To find throughput capacity, we consider the structure obtained by executing PSO. The structure is shown in Fig 5. where all routing paths are divided into shared regions (R2, R3 and R4) and non-shared regions (R1 and R5).

To find upper bound on per-node throughput, we consider the sparsity cut approach presented in [15]. For a random network, where nodes are placed on a Euclidean space and transmissions occur between neighboring nodes, a narrow class of cuts can be considered that are induced by a line segment (or a plane in the 3*D* case) that cuts the network into two regions. The *cut length* Γ is defined as the length of the cut line segment. The cut lines that we consider have zero width such that no nodes lie on them. The two subregions divided by the cut are denoted by Γ1 and Γ2. A *sparsity cut* [17] for a random network is defined as a cut induced by the line segment with the minimum length that separates the region into two equal areas/subregions. The *cut capacity* is defined as (ΛΓ1,2 ,ΛΓ2,1) where ΛΓ1,2 equals the transmission bandwidth, *W*, times the maximum possible number of simultaneous transmissions (broadcast or non-broadcast) across the cut from Γ1 to Γ2; and ΛΓ2,1 equals the same quantity from Γ2 to Γ1. This cut capacity constrains the information rate that the nodes from one side of the cut as a whole can deliver to the nodes at the other side as a whole.

The cut capacity is bounded by any upper bound on the maximum number of simultaneous transmissions across the cut. It is easy to see that all of the direct receivers of all transmissions across a cut Γ in one direction lie in a shaded rectangle region [15] with area $l_\Gamma \times$ R. It is shown that under the protocol model, disks of radius ΔR/2 centered at each receiver must be disjoint. We rely on the protocol model [15] to model the transmission behavior of a node.

IV. Performance Evaluation

A. Experiment Set up

In this section, we evaluate the performance of the proposed PSO based enforced network coding scheme. We consider two grid-shaped networks; a small-scale network of 10x10 intersections and a large-scale network of 50 x 50 intersections where the length of each road segment is 200m. We randomly generate source-destination pairs in the network in a way that the source and destinations reside near the edge of the network. Of course, some source and destinations also reside in the middle of the network.

Table 1. PSO Parameters

Parameter	Value
Acceleration Co-efficient, C1	2
Acceleration Co-efficient, C2	2
Inertia Weight, *w*	0.5
Population Size	50
Number of Iterations	50

Table 2. Routing Parameters

Parameter	Value
Data rate	6 Mbps
Propagation Model	Two Ray Ground
PHY/MAC Protocol	IEEE 802.11p
Transmission Range	300 m
Number of Source-Destination Pairs	5, 8,11,14, 17,20

B. Performance Metrics

1) *Convergence:* It shows the effectiveness of PSO in converging to global optimum.

2) *End-to-End Delay(s):* It is defined as the delay incurred at the destination in receiving packets from the source.

3) *Throughput (Mbps):* It is defined as the capacity of a node to transmit and receive packets.

C. Results and Discussions

Fig. 6 shows the convergence performance of PSO for the small-scale network. We also compute the global optimum for the network to show the effectiveness of the proposed scheme. We observe that the objective function value decreases very fast as the iterations pass by. At iteration 40, the objective function value converges to the global optimum.

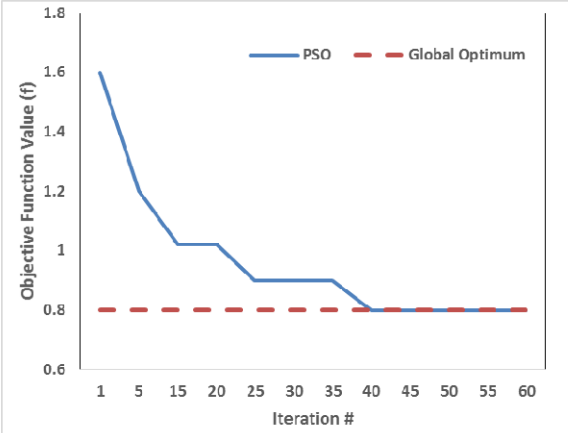

Fig. 6. Convergence of PSO

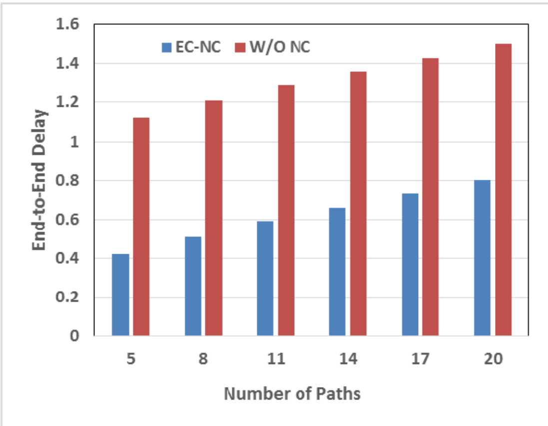

Fig. 7. End-to-End Delay (Large-scale Network)

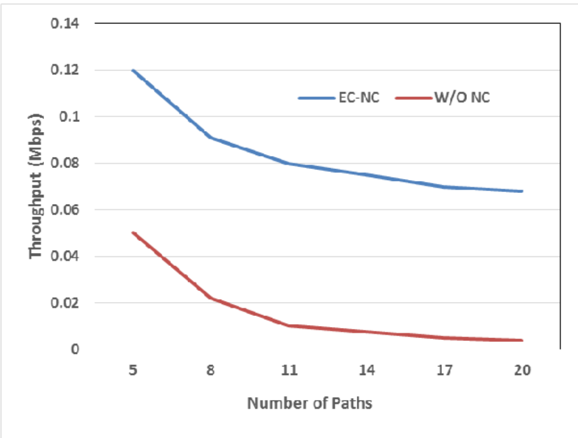

Fig. 8. Throughput (large-Scale Network)

Fig. 7 shows the end-to-end delay for two schemes: proposed E-NC scheme and the W/o NC scheme i.e. the scheme without network coding for large-scale network. We observe that as the number of source-destination pair increases, the end-to-end delay increases because of back off and queuing delays. However, E-NC outperforms W/o NC scheme by having a slow increases.

Fig. 8 shows the throughput for the two schemes for large-scale network. We observe that E-NC scheme achieves better throughput than W/o NC scheme. Although both schemes show decrease in throughput with more source-destination pairs, EC-NC is less sensitive over without NC.

V. CONCLUSION

In this paper, we have explored the benefits of enforcing network coding to the routing paths in vehicular networks. To optimally generate routing paths we have proposed a new particle swarm optimization algorithm. Our numerical results show that, E-NC protocol ensures higher throughput and lower end to end delay. In a small scale network, we have shown that, PSO algorithm converges to the global optimum after 42 iterations. In a large scale network, the performance of E-NC is evident in terms of throughput and end-to-end delay.

REFERENCES

[1] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, “Network Information Flow,” IEEE Transactions on Information Theory, vol. 46, no. 4, pp. 1204-1216, July 2000.

[2] S. Zhang, S. Liew, and P. Lam, “Hot topic: physical -layer network coding,” in Proceedings of ACM MobiCom, Sept. 2006, pp. 358 –365.

[3] T. Ho, M. M´edard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” IEEE Trans.Inform. Theory, 52:10 (2006), pp. 4413 –4430.

[4] M. Jerbi, S. M. Senouci, T. Rasheed, and Y. Ghamri-Doudane, “Towards efficient geographic routing in urban vehicular networks,” IEEE Trans. Veh. Technol., vol. 58, no. 9, pp. 5048–5059, Nov. 2009

[5] S. Katti and D. Katabi, “MIXIT: The Network Meets the Wireless Channel,” in Proc. of the Sixth ACM Workshop on Hot Topics in Networks (HotNets-VI), 2007

[6] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, “Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions,” IEEE Communications Surveys & Tutorials, vol. 13, no. 4, pp. 584–616, 4th Quart., 2011.

[7] P.K. Sahu, E.H. Wu, J. Sahoo, M. Gerla, “BAHG: back-bone-assisted hop greedy routing for VANET’s city environments”, IEEE Trans. Intell. Transp. Syst., 14 (1) (2013), pp. 199–213

[8] J. Bernsern and D. Manivannan, “Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification,” Pervasive Mob. Comput., vol. 5, no. 1, pp. 1–18, Feb. 2009.

[9] S. R. Li, R. W. Yeung, and N. Cai, “Linear network coding,” IEEE Transactions on Information Theory, vol. 49, no.2, pp.371 -381, Feb. 2003.

[10] T. Ho, M. Medard, R. Koetter, D.R. Karger, M. Effros, J. Shi; B. Leong, "A Random Linear Network Coding Approach to Multicast,", IEEE Transactions on Information Theory, vol.52, no.10, pp. 4413 - 4430, Oct. 2006 .

[11] S. Katti, S. Gollakota, and D. Katabi,, “Embracing Wireless Interference: Analog Network Coding,” in Proceedings of. ACM SIGCOMM, Oct. 200 7, pp 397 -408 .

[12] .S. Liew, S. Zhang, L. Lu, “Physical -layer network coding: Tutorial, survey, and beyond,” Elsevier Physical Communication, vol. 6, pp. 4 - 42, March 2013.

[13] S. Bitam, A. Mellouk, “Bee life-based multi constraints multicast routing optimization for vehicular ad hoc networks,” Journal of Network and Computer Applications, Volume 36, Issue 3, pages 981-991, May 2013

[14] Souza, A.B.; Celestino, J.; Xavier, F.A.; Oliveira, F.D.; Patel, A.; Latifi, M., "Stable multicast trees based on Ant Colony optimization for vehicular Ad Hoc networks," Information Networking (ICOIN), 2013 International Conference on, vol., no., pp. 101, 106, 28-30 Jan. 2013.

[15] J. Liu, D. Goeckel, D. Towsley, "Bounds on the Gain of Network Coding and Broadcasting in Wireless Networks," IEEE INFOCOM 2007, pp. 724-732, 6-12 May 2007.

[16] C. Sarr, I.G. Lassous, “Estimating Average End-to-End Delays in IEEE 802.11 Multihop Wireless Networks”, Technical Report, INRIA-00166017, July 2007.

[17] A. Togou, A. Hafid, L. Khoukhi, “A Novel CDS-based Routing Protocol for Vehicular ad hoc Networks in Urban Environments,” IEEE Globecom, 2015