Delay-bounded Routing in Vehicular Ad-hoc Networks

MobiHoc 2008

Presented by L. W. Lee
2008/11/13
Outline

- Introduction
- System Model
- Proposed Algorithms
- Performance Evaluation
- Conclusions
Introduction

- Two categories of VANET-based applications
  - Those that require **broadcasting** of information from one vehicle to many nearby vehicles, e.g. for collision avoidance
  - Those that require the propagation of information **hop-by-hop** to a single destination point or area, e.g. Sending an emergency message from an accident side to the closest roadside unit that is connected to a fixed network
    - Motivating example: ambient traffic sensor application
    - Accident, road fault, traffic congestion
    - Different priority and different user-defined delay threshold
Introduction

- Network resources will be shared by applications that provide internet access to passengers, provide the driver with safety information and so on.
- This matter is further aggravated if we take into account inter-system interference.
- The goal is to design algorithms that try to optimize bandwidth utilization, by being frugal in wireless packet.
  - Leverage knowledge of traffic information (density, speed).
System Assumptions

- We assume location-aware vehicle that obtain their geographical position from a GPS
- Vehicles also have access to a digital map of the area
- We assume that the map is preloaded with historical traffic statistics about the street network (average speed, average vehicle density per road segment)
- The street map is abstracted as a directed graph $G(V,E)$
- Message containing the event description and generation time $t_g$ and a time-to-live value $\lambda$

Figure 1: Our Model. Shaded circles indicate the communication range.
System Objective

- Considering a densely populated urban area where typically the wireless medium is shared by a large number of vehicles, running a variety of application competing the network resources
  - It is crucial to be frugal in the use of the wireless channel
  - We aim to minimize bandwidth utilization by minimizing the number of transmitted messages
- When traffic density is low, the vehicular network often becomes disconnected
  - Carry-and-forward
  - Assume that vehicles have very large buffers to store messages
- Our algorithms intend to minimize the number of transmissions while forwarding a message to an access point within the message-specific delay threshold
Forwarding strategies

- **Multihop Forwarding**
  - Aggressive forwarding messages to vehicles that are better positioned
  - Traffic needs to be dense enough so that better positioned vehicles exist within communication range

- **Data Muling**
  - Buffering messages in local memory and carrying them at the vehicle’s speed
  - It is a feasible option as long as the current vehicle is traveling on the path selected by the routing algorithm

- The novelty of the proposed algorithms lies in their careful alternation between the Multihop Forwarding and Data Muling to achieve a good tradeoff between delay and communication cost
Delay-bounded Greedy Forwarding (D-Greedy)

- D-Greedy assumes that the best path to an access point is the shortest one.
- Each vehicle maintains a neighbor list by periodically broadcast beacons:
  - Vehicle identifier (id)
  - Length of the shortest path (distToAP)
- D-Greedy assumes that the remaining message delay budget can be uniformly distributed among the edges that compose the shortest path to the AP.
- Each edge on the path is allocated delay budget that is proportional to its length.
\[ \text{Del} = \text{TTL} \times \frac{\text{distToInt}}{\text{distToAP}} \]

\[ \text{Del}_{DM} = \frac{\text{distToInt}}{u} \]

\( u \): average speed during a k-second

\( \text{distToInt} \): the remaining length until the next intersection

\[ \text{Del}_{DM} \leq \text{Del} \quad \text{: Data Muling strategy} \]

Otherwise \quad \text{: Multihop Forwarding strategy}

Figure 2: Node a will choose to forward the message to node c, the closest node to the AP among those in range.
Delay-Bounded Minimum Cost Forwarding (D-MinCost)

- To annotate each edge with two metrics: 1) **cost (C)**, representing the number of message transmissions along the edge, and 2) **delay (Del)**, denoting the time required to forward a message along the edge.

- We convert the original directed graph $G(V,E)$ that represents the street map to a new $G'(V,E')$, which contains the same set of vertices, and twice as many edges.

![Diagram](image)

*Figure 4: Replacing edge $(d,a)$ in $G$ by two sibling edges, one per strategy.*
D-MinCost

- **Data Muling strategy:** \[ Del_{DM} = \frac{\ell}{u}, \quad C_{DM} = 1 \]

- **Multihop Forwarding strategy:** \[ C_{MH} = \frac{\ell}{R}, \quad Del_{MH} = C_{MH} \times q \]
  
  q: the time required for the node to check its neighbor list and identify the best next hop

- D-MinCost utilizes the Delay Scaling Algorithm to efficiently compute delay-constrained least cost path from the vehicle’s location to all access points
  
  - The access point that can be reached with the least cost.
  
  - The exact min-cost path to that access point.
  
  - The strategy that should be followed at each edge of the path in order to adhere to the message’s remaining delay budget.
**DSA (Delay Scaling Algorithm)**

*Inputs:* Graph $G$ with link costs $l_{ij}$ and delays $d_{ij}$ and a delay threshold $T$.

*Outputs:* Tables $L(v, t)$ and $P(v, t)$, $1 \leq v \leq n$, $0 \leq t \leq T$. The entry $L(v, t)$ is the cost of the cheapest path from 1 to $v$ whose delay is no more than $t$. The entry $P(v, t)$ encodes the cheapest path from 1 to $v$, whose delay is no more than $t$.

1. Initialize $L(1, t) = 0$, $t = 0, \ldots, T$
2. Initialize $L(j, 0) = \infty$, $j = 2, \ldots, n$
3. Compute $L(j, t) = \min\{ L(j, t-1), \min_{k, i} \{ l_{kj} + l_{ki} \} \}$

where $j = 2, \ldots, N$, $t = 1, \ldots, T$

Fig. 2. Subroutine DAD which iterates on integer delays.

$$L(3,15) = \min\{ L(3,14) , \min\{ L(2,9) + (l_{23}=10) \} \} = 16$$

$$L(2,9) = \min\{ L(2,8) , \min\{ L(1,8) + (l_{12}=2) \} \} = 6$$

$$L(S,-1) + (l_{12}=2)$$

$$\infty$$
Delay-Bounded Minimum Cost Forwarding (D-MinCost)

- When a message $p$ is generated at the node, the algorithm applies the DSA heuristic on the extended graph $G'$ for message $p$ with delay budget $TTL$.
- From the path returned by DSA($I, TTL$), D-MinCost selects the minimum cost path that leads to an access point and encodes it in the message header.
- The message path will be recomputed at the next intersection by its carrier only if it is not feasible to follow the suggested edge and its associated strategy.
  - In this case, the edge is removed from graph $G'$ and the DSA is reinvoked on the resulting graph in order to compute an alternative min-cost path.
Performance Evaluation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Iterations</td>
<td>30</td>
</tr>
<tr>
<td>Iteration Duration</td>
<td>1800 sec</td>
</tr>
<tr>
<td>Beacon Period</td>
<td>5 sec</td>
</tr>
<tr>
<td>Number of Vehicles</td>
<td>[200-1000]</td>
</tr>
<tr>
<td>Delay Threshold ((\lambda))</td>
<td>[300-1800] sec</td>
</tr>
<tr>
<td>Number of Messages Generated</td>
<td>100</td>
</tr>
<tr>
<td>Message Generation Interval</td>
<td>First 50 sec</td>
</tr>
<tr>
<td>Message Size</td>
<td>1500 KBytes</td>
</tr>
<tr>
<td>Communication Range</td>
<td>250m</td>
</tr>
<tr>
<td>Bitrate</td>
<td>500 Kbps</td>
</tr>
</tbody>
</table>

Table 1: Simulation Parameters

Figure 5: The section of the map of Zurich used in our experiments. Circles represent access points.
Delivery Ratio

Figure 7: Message Delivery Ratio varying the number of cars ($\lambda=1200$sec)

Figure 8: Message Delivery Ratio varying $\lambda$ (number of cars=900)
Transmitted Bytes

Figure 9: The total number of bytes sent for different car densities ($\lambda=1200$ sec)

Figure 10: The total number of bytes sent for different values of $\lambda$ (number of cars=900)
Message Delay

Figure 11: Average delivery delay for different values of $\lambda$ (number of cars=900)

Figure 12: CDF of message delivery delay (900 cars, $\lambda=1500$)
Effect of $\lambda$

- D-Greedy and D-MinCost gracefully alternate between the Multihop Forwarding and Data Muling strategies aiming to exhaust the message delay threshold and minimize the communication cost, effectively trading allowable delay for bandwidth.

**Figure 13:** Strategy chosen for low $\lambda = 600s$

**Figure 14:** Strategy chosen for high $\lambda = 1800s$
Conclusions

- Our algorithm leverage traffic statistics to reach forwarding strategy decisions that minimize communication cost and at the same time adhere to a per-packet application defined-delay threshold.
- The cost savings of D-Greedy and D-MinCost are derived from carefully alternation between the Multihop Forwarding and Data Muling strategies.