Vehicular Networks [C2X]

Part 2: Car-to-X Networking

Broadcast, Geocast, Routing
Routing

- Classical approaches to routing
  - Distance Vector Routing
    - Nodes keep vector of known destinations, store distance and next hop
    - Ex: DSDV
  - Link State Routing
    - Nodes keep track of all links in network
    - Pro: fast and guaranteed convergence
    - Con: high overhead
    - Ex: OLSR
Classical approaches to routing (II)

- Reactive (on demand) routing
  - Routes established when needed
  - Routing messages only exchanged if (or while) user data is exchanged
  - Unused routes expire
  - Ex: AODV, DYMO

- Proactive (table driven) routing
  - Routes are established and maintained continuously
  - No route setup delay when data needs to be sent
  - High overhead
  - Ex: OLSR, DSDV
Classical approaches to routing (III)

- **Hop-by-Hop Routing**
  - Each packet contains destination address
  - During routing, each hop chooses best next hop
  - Ex: AODV

- **Source Routing**
  - Each packet contains complete route to destination
  - During routing, nodes rely on this information
  - Ex: DSR
Routing

Georouting

- Primary metrics: position / distance to destination
- Requires node positions to be known (at least for the destination)
- Two operation modes (typ.):
  - Greedy mode: choose next hop according to max progress
  - Recovery mode: escape dead ends (local maxima)
- Must ensure that message never gets lost
Routing

- **Georouting: CBF**
  - „Contention Based Forwarding“
  - Reduction (or complete avoidance) of duplicates

- **Outline**
  - Given: position of message destination, position of last hop
  - Do not forward message immediately, but wait for time T
  - Choose wait time T according to suitability of node
  - Do not forward message if another forward was overheard

- **Problem**
  - Potential forwarders must be able to overhear each others’ transmissions

Georouting: CBF

Potential forwarders are contained in Reuleaux triangle (1) (use estimated communication range for thickness of triangle)

Waiting time is $T = 1 - P$

(z: destination, f: last hop forwarder)

If last hop overhears no node forwarding the message, message is re-sent for nodes in (2), then (3)

$$P(f, z, n) = \max \left\{ 0, \frac{\text{dist}(f, z) - \text{dist}(n, z)}{r_{\text{radio}}} \right\}$$
Routing

- Reflection on classical routing approaches
  - Q: Can (classical) routing work in VANETs?
  - A: Only in some cases.
  - Commonly need multicast communication, low load, low delay
  - Additional challenges and opportunities:
    network partitioning, dynamic topology, complex mobility, ...

Flooding

- Flooding (Multi-Hop Broadcast)
  - Simplest protocol: „Smart Flooding“:
    - Problem: Broadcast Storm
      - Superfluous re-broadcasts overload channel
Flooding

- Consequences of a broadcast storm
  - Interference → impact on other systems
  - Collision → impact on other users
  - Contention → impact on other applications

![Graphs showing delay and packet loss ratio versus node density.](image)
Flooding

- Solving the broadcast storm problem

- Classical approaches
  - Lightweight solutions (e.g., probabilistic flooding)
  - Exchange of neighbor information, cost/benefit estimations
  - Topology creation and maintenance (Cluster, Cord, Tree, ...)

- Drawbacks
  - Blind guessing (or scenario dependent parameterization)
  - Additional control message overhead
  - Continuous maintenance of topology
Flooding

- VANET specific solution: Broadcast Suppression
  - Needs no neighbor information
  - Needs no control messages
  - Maximizes distance per hop
  - Minimizes packet loss

- Approach
  - Node receives message, estimates distance to sender
  - Selectively suppresses re-broadcast of message
  - Alternatives
    - weighted p-persistence
    - slotted 1-persistence
    - slotted p-persistence

Flooding

Broadcast Suppression

- Estimate distance to sender as $0 \leq \rho_{ij} \leq 1$

- GPS based
  \[ Q_{ij} = \begin{cases} 
  0 & \text{if } D_{ij} < 0 \\
  \frac{D_{ij}}{R} & \text{if } 0 \leq D_{ij} < R \text{ (approx. transmission radius)} \\
  1 & \text{otherwise}
  \end{cases} \]

- RSS based
  \[ Q_{ij} = \begin{cases} 
  0 & \text{if } RSS_x < RSS_{min} \\
  \frac{RSS_{max} - RSS_x}{RSS_{max} - RSS_{min}} & \text{if } RSS_{min} \leq RSS_x < RSS_{max} \\
  1 & \text{otherwise}
  \end{cases} \]
Broadcast Suppression

- Weighted p-persistence
  - Probabilistic flooding with variable $p_{ij}$ for re-broadcast
  - Thus, higher probability for larger distance per hop
**Flooding**

- **Broadcast Suppression**
  - **Weighted p-persistence**
    - Wait \( \text{WAIT\_TIME} \) (e.g., 2 ms)
    - Choose \( p = \min(p_{ij}) = \min(p_{ij}) \) of all received packets (probability for re-broadcast of packet)
    - Ensure that at least one neighbor has re-broadcast packet
Flooding

- Broadcast Suppression
  - Slotted 1-persistence
    - Suppression based on waiting and overhearing
    - Divide length of road into slots
    - More distant slots send sooner
    - Closer slots send later (or if more distant slots did not re-broadcast)
    - Thus, higher probability to transmit over longer distance

\[
p_{ij}^{t=0} \quad t=\tau \quad t=2\tau \quad t=3\tau
\]
Flooding

Broadcast Suppression

- Slotted 1-persistence
  - Divide “communication range” into $N_s$ slots of length $\tau$
  - Nodes wait before re-broadcast, waiting time $T_{ij} = \tau \times \lceil N_s (1 - \rho_{ij}) \rceil$
  - Duplicate elimination takes care of suppression of broadcasts
Flooding

- Broadcast Suppression
  - Slotted $p$-persistence
    - Cf. slotted 1-persistence
    - Fixed forwarding probability $p$ (instead of 1)
Flooding

- Broadcast Suppression
  - Slotted $p$-persistence
    - Wait for $T_{ij}$ (instead of fixed WAIT_TIME)
    - Use probability $p$ (instead of 1)
    - Ensure that at least one neighbor has re-broadcast the packet by waiting for $\delta' > \max(T_{ij})$
Flooding

- Broadcast Suppression
  - Solves Broadcast Storm Problem
  - Maximizes distance per hop
  - Minimizes packet loss

![Graph 1](image)

![Graph 2](image)
Flooding

- Broadcast Suppression
  - But: Much higher per-message delay
Flooding

- Remaining problems
  - Temporary network fragmentation

- Undirected message dissemination
Flooding + X

**DV-CAST**

- Idea: detect current scenario, switch between protocols
- Check for fragmented network
  - Network connected $\rightarrow$ perform broadcast suppression
  - Network fragmented $\rightarrow$ perform Store-Carry-Forward

Flooding + X

- **DV-CAST: Mechanism**
  - Nodes periodically send *Hello* beacons containing position, speed
  - Nodes maintain 3 neighbor tables
    - Same direction, ahead
    - Same direction, driving behind
    - Opposite direction
  - Messages contain source position and Region of Interest (ROI)

- For each message received, evaluate 3 Flags:
  - **Destination Flag (DFlg):**
    - Vehicle in ROI, approaching source
  - **Message Direction Connectivity (MDC):**
    - ∃ neighbor driving in same direction, further away from source
  - **Opposite Direction Connectivity (ODC):**
    - ∃ neighbor driving in opposite direction
Flooding + X

- DV-CAST

**Algorithm:**

- **IDLE**
  - Packet Arrival
  - MDC = ?
    - 1: Broadcast Suppression
    - 0: ODC = ?
      - 1: Rebroadcast
      - 0: WAIT I
        - Pkt Timer Expires
          - Hello Pkt arrives from ODN
        - ODC = 1 or MDC = 1
          - DFImg = ?
            - 1: WAIT II
            - 0: Rebroadcast
              - Pkt Arrival <HOP = N+2>
              - or Pkt Timer Expires
## Flooding + X

### DV-CAST

#### Decision matrix:

<table>
<thead>
<tr>
<th>MDC</th>
<th>ODC</th>
<th>DFbg</th>
<th>Derived Scenario</th>
<th>Actions Taken by DV-CAST Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>×</td>
<td>1</td>
<td>Well Connected</td>
<td>Broadcast Suppression</td>
</tr>
<tr>
<td>1</td>
<td>×</td>
<td>0</td>
<td>Well Connected</td>
<td>Help relay the packet by doing broadcast suppression</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Sparsely Connected</td>
<td>Rebroadcast and assume that the ODN will help relay or rebroadcast</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Sparsely Connected</td>
<td>Rebroadcast and help carry &amp; forward the packet to the first new neighbor in the opposite direction or in the message direction encountered</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>×</td>
<td>Totally Disconnected</td>
<td>Wait and forward the packet to the first neighbor in the opposite direction or in the message direction encountered.</td>
</tr>
</tbody>
</table>
Flooding + X

- **DV-CAST**
  - Simulation results:

![Graphs showing simulation results for DV-CAST compared to Broadcast.]
Intermediate Summary

- Remaining problems
  - Temporary network fragmentation
  - Undirected message dissemination
Geocast

TO-GO

„Topology-Assisted Geo-Opportunistic Routing“

Nodes periodically send *Hello* beacons; Contents:

- Number of neighbors
- Bloom filter of neighbor IDs
- IDs of neighbors furthest down the road/roads

Thus, nodes know about all 2-hop neighbors

Bloom Filter

**Idea:**

- Bloom filter is a bit field $X$
- Hash functions $h_1$ to $h_k$ map input data $x$ to one bit (each) in $X$
- Insertion of $x$: Set $X[h_i(x)] \leftarrow 1 \quad \forall i \in [1..k]$
- Test for $x \in X$: Check $X[h_i(x)] = 1 \quad \forall i \in [1..k]$

**Probabilistic test for “$x \in X$”**

- Possible results: no / maybe ($\Rightarrow$ chance of false positives)
- Allows for very compact representation of $X$

TO-GO

Step 1: Find best next hop (Target Node, T)
- Find N: Furthest neighbor towards destination
- Find J: Furthest neighbor towards destination, currently on junction
- Find N_J: Furthest neighbor towards destination, as seen by J
- if N, N_J are on the same road (and running in greedy mode), pick N
  else, pick J
Geocast

- TO-GO

  - Step 2: Find Forwarding Set (FS)
    - Nodes in the FS will compete for relaying of the message
    - Only one node in FS should relay
      thus, all nodes in FS must hear each other
    - Finding optimal solution is *NP complete*
    - TO-GO uses approximation:
      - Bloom filter entries indicate who can hear whom
      - Given the target node T,
        find its neighbor M with the maximum number of neighbors
      - Include all those neighbors in FS, which
        - can hear M, and
        - are heard by M, and
        - are heard by all current members of FS
Geocast

TO-GO

Step 3: Multicast message to all nodes in FS

- Nodes in the FS compete for relaying of the message
- Ensure maximum progress within FS
- Delay re-broadcast by $t$
- Suppress re-broadcast if another nodes forwards within $t$

$t = \tau \times \frac{d_T}{d_{\text{max}}}$

with:

- $\tau$: Maximum delay per hop
- $d_T$: Distance to Target Node
- $d_{\text{max}}$: Distance from last hop to Target Node
Intermediate Summary

- Remaining problems
  - Temporary network fragmentation
  - Undirected message dissemination
Scalability

Do the presented approaches scale?

Analytical evaluation [1]:
- Capacity of wireless channel is limited
- Amount of information transported across any (arbitrary) border must be upper-bounded

\[ \sum = \xi_r \]

Scalability

Solution?
- Define maximum dissemination range of any information
- Reduce update frequency with increasing distance
- Aggregate information as distance increases

Pre-condition for scalability of dissemination approach?
- Used bandwidth reduces as distance to source increases
- Upper bound: $1 / d^2$
Main Takeaways

- Classic information dissemination
  - Distance vs. link-state
  - Reactive vs. proactive
  - Hop-by-hop vs. source routing
  - Geo-routing (CBF)

- Examples of VANET-centric information dissemination
  - Flooding (Weighted/Slotted 1/p-Persistence)
  - Fragmentation (DV-Cast)
  - Directedness (To-Go)

- Scalability