GrooveNet

Hybrid Simulator for Vehicular Networks

Rahul Mangharam, Daniel Weller, Prof. Raj Rajkumar
Carnegie Mellon University, Pittsburgh, U.S.A.

Priyantha Mudalige & Fan Bai
General Motors Research Center, Warren, U.S.A.
General Motors – Carnegie Mellon Collaborative Research Laboratory

• 5 GM Research Vehicles
• Multiple Inter-Disciplinary Projects
  – V2V Network Protocols
  – Channel Modeling and Antenna Design
  – Navigation & User Interfaces
  – Autonomous Vehicles, AI and Computer Vision
  – Embedded Systems and bus design
What are Vehicle-to-Vehicle Networks?

• **Vehicle-to-Vehicle Wireless Broadcast Protocols for:**
  – On-road Safety
  – Traffic Congestion Probing

• **The Key Questions:**
  – Market Penetration to make v2v useful?
  – Performance in urban, rural and highway?
  – What mix of mobile gateways, infrastructure and on-board wireless?
  – Which multi-hop protocols work best and under what conditions?
Experimental Multi-hop Vehicular Network Test-bed

- 5.9 GHz DSRC Dedicated Short Range Communications Between vehicles
- Mobile Nodes in Pittsburgh, PA
- GPS Differential GPS reference station beacons
- EVDO Cellular Network connects mobile gateway
- Internet
- Remote Monitoring of Experiment in Detroit, Michigan

- Vehicle-to-Vehicle Multi-hop
- Vehicle-to-Mobile Gateway
- Vehicle-to-Infrastructure
• Driven 5 vehicles over 400 miles – Urban, Rural and Highway Roads
• Over 625,000 link measurements
On-road Vehicular Networking Platform
Why do we need Hybrid Simulation?

Key Benefits:
• Use same protocol implementation for Simulation and Prototype

• Observe effects of Network Scaling and Traffic Density

• Remote Monitoring of On-road Experiments

• Evaluate correctness of Physical, Link Layer and Vehicle Interaction Models
GM-CMU Collaborative Research Laboratory

Real Vehicles in vicinity

Simulated Vehicles

Real Vehicles

Network Connections
with Real Vehicles

Real Network Connectivity
GrooveNet Hybrid Simulator Design

- Map Database (ASCII)
- GrooveNet Topology Graph (binary)
- Simulator Test File
- On-board Diagnosis OBD-II Event Trigger
- DGPS Positioning
- Log Files
- Network Visualization

GrooveNet Simulator Core

- Event Queue
- Mobility Model
- Trip Model
- Car Model
- Network Model
- Other Models

Vehicle-to-Vehicle Network Interfaces
- DSRC
- 802.11
- 1xRTT
- EVDO
Modular Architecture

- Models
  - Vehicle Trip
  - Vehicle Mobility
  - V2V & V2I Network
  - Behavioral Traffic
  - Adaptive Broadcast
  - Geographic Routing
  - Multi-ch. Messaging

- Map Database
  - Network Visualizer
  - Test Generator
  - Event & Message Log

- Vehicle Operations Director
  - GrooveSim Simulator
  - Emulator

- Network Adaptation Layer
  - Virtual Vehicles
  - Real Vehicles

- Linux Network Stack
  - DSRC
  - 1xRTT
  - EVDO
  - 802.11
Modular Architecture (2)

- **Infrastructure Node Model**
  - Car Model
    - GPS Mode
    - Simulator Mode
    - Network Mode
    - Random Waypoint
  - Mobility Model
    - Fixed Mobility
    - Street Speed
    - Uniform Speed
    - Car Following
  - PHY Model
    - Simple PHY
    - Collision PHY
    - Multi-Channel PHY

- **Traffic Light Model**
  - Trip Model
    - Random Walk
    - Djikstra
    - Sightseeing

- **Visualizer**
  - Map Visual
  - Car List Visual

- **Comm. & Link Model**
  - Adaptive Re-broadcast
  - Groove Re-broadcast
Click to Add Model
1,000 Vehicles in Chicago, IL suburb

Routed with Minimum Cost Routing
Vehicles migrate to roads with higher speed limits
Why do we need GrooveNet?
Uniform Urban Distribution

GrooveNet Mobility - Chicago
Rural Area: Rnd Waypoint Vs GrooveNet
Topology-Mobility Models
Message Propagation Rate

![Graph showing message propagation rate over time with different scenarios: GrooveNet-Chicago, GrooveNet-Pittsburgh, Random Waypoint-Rural, GrooveNet-Rural, RWP-Urban, Chicago, Pittsburgh, Rural, RWP-Rural. The x-axis represents time in seconds, ranging from 0 to 100, and the y-axis represents the fraction of vehicles reached, ranging from 0 to 1.](image-url)
GeoRoute: Broadcast Scenarios

Highway Driving
- Path with Intermediate points
- Static Source Routing

City Driving
- Radial Broadcast

Rural Driving
- Bounding Box
- Controlled Flooding
• Broadcast Safety Alerts to all vehicles in the vicinity
• Messages are valid in a specific geographic region
• Regions are determined by position, speed and direction
GrooveSim – On-Road Alerts (2)

![GrooveSim - Add Event Message](image)
GrooveSim – On-Road Alerts (3)

- Only vehicles in the relevant geographic region receive alerts
Performance: Message Delay

<table>
<thead>
<tr>
<th>Bounding Box Size</th>
<th># Active Vehicles</th>
<th>Message Delay (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>192</td>
</tr>
<tr>
<td>1</td>
<td>138</td>
<td>40.4</td>
</tr>
<tr>
<td>2</td>
<td>150</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>162</td>
<td>11</td>
</tr>
</tbody>
</table>
Performance: Message Lifetime

- Street Intersection Message Lifetime

- Graph showing the relationship between Bounding Circle Radius (m) and Message Lifetime (mins)

- Map displaying various street intersections and bounding circles of different radii.
How close are we to V2V?

Effect of Market Share on Message Diffusion

V2V Market Share

% of vehicles reached

Time (s)

V2V Market Share

100%

90%

80%

70%

60%

50%

40%

30%

20%

10%

0%

0

2

4

6

8

10

12

10%

5%

4%

3%
Simulator Scalability

The graph shows the relationship between the number of vehicles and the running time. As the number of vehicles increases, the running time also increases, indicating scalability issues. The curve suggests an exponential growth in running time with the number of vehicles.
Simulation Setup – 1000s of vehicles

![GrooveSim - Auto-Generate Nodes...](image)

- **Node Type**: SimModel
- **Parameter**: <Other Addresses>
- **Region Type**: Bounding Rectangle
- **Bounding box area**: 0.0723807 sq. km.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMM</td>
<td>Fixed</td>
<td>AdaptiveCommModel</td>
</tr>
<tr>
<td>DELAY</td>
<td>Fixed</td>
<td>0.2</td>
</tr>
<tr>
<td>DEPENDS</td>
<td>Fixed</td>
<td></td>
</tr>
<tr>
<td>DOLOG</td>
<td>Fixed</td>
<td>Yes</td>
</tr>
<tr>
<td>ID</td>
<td>Fixed</td>
<td>127.0.0.1</td>
</tr>
<tr>
<td>LINK</td>
<td>Fixed</td>
<td>SimpleLinkModel</td>
</tr>
<tr>
<td>MOBILITY</td>
<td>Fixed</td>
<td>CarFollowingModel</td>
</tr>
<tr>
<td>PHYS</td>
<td>Fixed</td>
<td>CollisionPhysModel</td>
</tr>
<tr>
<td>START</td>
<td>Fixed</td>
<td>0</td>
</tr>
<tr>
<td>TRACKSPEED</td>
<td>Fixed</td>
<td>No</td>
</tr>
<tr>
<td>TRIP</td>
<td>Fixed</td>
<td>SightseeingModel</td>
</tr>
</tbody>
</table>

**START (seconds)** -- The number of seconds from the beginning of the simulation until this vehicle starts moving.
Simulation Test Setup is Easy

---

**Model Name:** SimModel2

**Parameter** | **Value**
--- | ---
DOLOG | Yes
ID | 127.0.0.3
LINK | SimpleLinkModel2
MOBILITY | CarFollowingModel2
PHYS | CollisionPhysModel2
START | 0

**ID (IP address)** -- The IP address uniquely identifying this vehicle.
Running the Simulation test

Simulation Type
- Run Once
- Monte Carlo

Duration: 100

Simulation Time
- Real Time
- Fixed Time Increments: 10m

Emergency from 127.0.0.1: t=0->600, "Vehicle crash"

Log File Type
- GrooveSim Message Log: messages.txt
- Event Log 1: events.txt
- Vehicle Neighbor Log: neighbors.txt

Run Cancel
Easy to Add New Models

```cpp
int CarFollowingModel::Init()
{
    ....
}

int CarFollowingModel::PreRun()
{
    ....
}

int CarFollowingModel::ProcessEvent(SimEvent & event)
{
    ....
}

int CarFollowingModel::PostRun()
{
    ....
}

int CarFollowingModel::Cleanup()
{
    ....
}
```
What’s Next?

• Develop an Embedded in-vehicle platform

• General Motors to deploy 50 vehicles

• Congestion Probe and Stream Protocols

• GrooveNet is Free for academic use

Email: Rahul (rahul@cmu.edu)
“That's all Folks!”
Infrastructure Nodes