A Position-based Routing Module for Simulation of VANETs in NS-3

K. Katsaros\textsuperscript{1}, M. Dianati\textsuperscript{1}, K. Roscher\textsuperscript{2}

\textsuperscript{1}Centre for Communication System Research
University of Surrey, United Kingdom

\textsuperscript{2}Fraunhofer ESK
Munich, Germany

Workshop on NS-3, 2012
Outline

1. Introduction
   - VANETs Vs. MANETs
   - Routing Protocols

2. CLWPR Protocol
   - Introduction
   - Neighbor Discovery
   - Repositories
   - Forwarding
   - Enhancements

3. Evaluation
   - Using CLWPR
   - Performance Evaluation
Differences between VANETs and MANETs

- Lack of strict energy constraints
- High but predictable mobility, constraint by the road topology
- Relatively short lived communication links
- Unique characteristics of communication channel
Routing protocols for VANETs

- NS-3 has currently ONLY routing protocols for MANETs (OLSR, AODV, DSDV)
- Position-based routing protocols perform better in VANET environments\(^1\)
- Enhance performance of the protocol with the use of mobility and cross-layer information

Routing protocols for VANETs

- NS-3 has currently ONLY routing protocols for MANETs (OLSR, AODV, DSDV)
- Position-based routing protocols perform better in VANET environments\(^1\)
- Enhance performance of the protocol with the use of mobility and cross-layer information

Routing protocols for VANETs

- NS-3 has currently ONLY routing protocols for MANETs (OLSR, AODV, DSDV)
- Position-based routing protocols perform better in VANET environments
- Enhance performance of the protocol with the use of mobility and cross-layer information

---

Position-based routing protocols

- Select the next hop among the neighbors using a metric based on the position of that node.

- Problem faced when current node lays closer to destination than any other neighbor; known as *local maximum* problem.

- Use additional information, such as map information, to minimize the effect of local maximum. Use a recovery strategy to cope with this problem (e.g., perimeter forwarding, carry-n-forward)

Katsaros et al. CLWPR module for NS-3
Position-based routing protocols

- Select the next hop among the neighbors using a metric based on the position of that node.
- Problem faced when current node lays closer to destination than any other neighbor; known as *local maximum* problem.

![Diagram showing a network with nodes and a void area]

- Use additional information, such as map information, to minimize the effect of local maximum. Use a recovery strategy to cope with this problem (e.g., perimeter forwarding, carry-n-forward).
Position-based routing protocols

- Select the next hop among the neighbors using a metric based on the position of that node.
- Problem faced when current node lays closer to destination than any other neighbor; known as *local maximum* problem.
- Use additional information, such as map information, to minimize the effect of local maximum. Use a recovery strategy to cope with this problem (e.g., perimeter forwarding, carry-n-forward)
Cross-Layer, Weighted, Position-based Routing Module

- A module for realizing position-based routing in VANETs
- Enhances routing decisions with navigation information
- Exploits cross-layer information to make more effective routing decisions
- Supports *carry-n-forward* mechanism
Cross-Layer, Weighted, Position-based Routing Module

- A module for realizing position-based routing in VANETs
- Enhances routing decisions with navigation information
- Exploits cross-layer information to make more effective routing decisions
- Supports carry-n-forward mechanism
Cross-Layer, Weighted, Position-based Routing Module

- A module for realizing position-based routing in VANETs
- Enhances routing decisions with navigation information
- Exploits cross-layer information to make more effective routing decisions
- Supports carry-n-forward mechanism
Cross-Layer, Weighted, Position-based Routing Module

- A module for realizing position-based routing in VANETs
- Enhances routing decisions with navigation information
- Exploits cross-layer information to make more effective routing decisions
- Supports carry-$n$-forward mechanism
Neighbor Discovery Mechanism

Is based on 1-hop periodic “HELLO” messages carrying mobility, link and utilization information

```
+----------+----------+----------+
| 0 bit    | 15 bit   | 31 bit   |
+----------+----------+----------+
|          |          |          |
|***********|***********|***********|
|          |          |          |
|          |          |          |
| ORIGINATOR POSITION (X,Y) |          |          |
|          |          |          |
| ORIGINATOR VELOCITY (X,Y) |          |          |
|          |          |          |
| ORIGINATOR HEADING |          |          |
| Orig.RoadID | Htime   | Utilization |
| Utilization | MAC INFO |          |
| MAC INFO   | Carry-n-Forward |          |
+----------+----------+----------+
```
Repositories

**Neighbor Set** is the list of directly accessible (1-hop) nodes. Includes mobility, link and utilization related information of the node.

**Position Association Set** is the list of destination nodes. Includes mobility related information of destination node.
Repositories

**Neighbor Set** is the list of directly accessible (1-hop) nodes. Includes mobility, link and utilization related information of the node.

**Position Association Set** is the list of destination nodes. Includes mobility related information of destination node.
Forwarding Mechanism

- Forwarding metric in CLWPR is the result of joint weighting function:

\[
\text{Weight} = f_1 \text{Distance} + f_2 \text{NormAngle} + \\
 f_3 \text{NormRoad} + f_4 \text{Utilization} + \\
 f_5 \text{MAC}_\text{info} + f_6 \text{CnF}_\text{info} + \\
 f_7 \text{SNIR}_\text{info}
\]

- The node with the least weight will be selected as next hop.
- If current node has least weight (*local maximum* problem), then the packet is cached.
- Currently \( f_i \) parameters are fixed, no optimizations.
Forwarding Mechanism

- Forwarding metric in CLWPR is the result of joint weighting function:

  \[
  \text{Weight} = f_1 \text{Distance} + f_2 \text{NormAngle} +
  f_3 \text{NormRoad} + f_4 \text{Utilization} +
  f_5 \text{MAC}_{\text{info}} + f_6 \text{CnF}_{\text{info}} +
  f_7 \text{SNIR}_{\text{info}}
  \]

- The node with the least weight will be selected as next hop.
- If current node has least weight (local maximum problem), then the packet is cached.
- Currently, \( f_i \) parameters are fixed, no optimizations.

Katsaros et al.  CLWPR module for NS-3
Forwarding Mechanism

- Forwarding metric in CLWPR is the result of joint weighting function:

\[
\text{Weight} = f_1 \text{Distance} + f_2 \text{NormAngle} + \\
\quad f_3 \text{NormRoad} + f_4 \text{Utilization} + \\
\quad f_5 \text{MAC}_{\text{info}} + f_6 \text{CnF}_{\text{info}} + \\
\quad f_7 \text{SNIR}_{\text{info}}
\]

- The node with the least weight will be selected as next hop
- If current node has least weight (local maximum problem), then the packet is cached
- Currently $f_i$ parameters are fixed, no optimizations

Katsaros et al. CLWPR module for NS-3
Forwarding Mechanism

- Forwarding metric in CLWPR is the result of joint weighting function:

\[
\text{Weight} = f_1 \text{Distance} + f_2 \text{NormAngle} + \\
    f_3 \text{NormRoad} + f_4 \text{Utilization} + \\
    f_5 \text{MAC}_{\text{info}} + f_6 \text{CnF}_{\text{info}} + \\
    f_7 \text{SNIR}_{\text{info}}
\]

- The node with the least weight will be selected as next hop
- If current node has least weight (local maximum problem), then the packet is cached
- Currently \( f_i \) parameters are fixed, no optimizations
How packets flow through the Protocol Stack in NS-3

Application
  ↓ Socket::send()
  ↓ [Tcp/Udp]::SocketImpl
  ↓ ::RouteOutput()
  ↓ [Ipv4L3]::RoutingProtocol
  ↓ ::Send()
  ↓ [Tcp/Udp]::L4Protocol
  ↓ ::Send()
  ↓ Ipv4L3::Protocol
  ↓ ::Send()
  ↓ Arp::Arp4Interface
  ↓ ::Lookup()
  ↓ Arp::L3Protocol
  ↓ ::Send()
  ↓ NetDevice

Application
  ↓ ::m_rxCallback -> ForwardUp()
  ↓ [Tcp/Udp]::SocketImpl
  ↓ ::ForwardUp()
  ↓ [Tcp/Udp]::L4Protocol
  ↓ ::Receive()
  ↓ Ipv4L3::Protocol
  ↓ ::LocalDeliver()
  ↓ Ipv4::RoutingProtocol
  ↓ Protocol Handlers
  ↓ NetDevice

Katsaros et al. CLWPR module for NS-3
RouteOutput

- RouteOutput
- Query to Location Service
  - Lookup Routing Table for min weight
- Local Maximum
  - YES: Carry-n-Forward
  - NO: Create route for the selected node
    - Add DeferredRouteOutput Tag
    - Create Route for Local Delivery
- Return route
RouteInput

- `RouteInput`
  - `DeferredRouteOutputTag` (YES) → `Create QueueEntry and queue packet`
  - `Return`
  - `Local Delivery` (YES) → `LocalDeliverCallback`
- `Lookup Routing Table for min weight`
  - `Local Maximun` (YES) → `Carry-n-Forward`
  - `NO` → `Create route for the selected node`
  - `Add DeferredRouteOutputTag`
  - `UnicastForwardCallback`
  - `Create Route for Local Delivery`
  - `Return`
- `Query to Location Service`
- `Return`
Other Enhancements Used in CLWPR Module

- `ns3::GridMap` is a class that provides information related to navigation, such as road id and `courvemetric` distance.
- Use of packet Tags to get SINR values from PHY for “HELLO” messages.
- Use the `DeferredRouteTag` as per AODV implementation to cache packets in `local maximum`.
How to use CLWPR

- CLWPR comes with a helper class in order to be installed to the nodes
- Multiple attributes for configuration and optimization
- Current Limitations
  - GridMap is the only “navigation system"
  - Only IPv4 compatible
  - Works with single interface
Comparison with AODV, OLSR and DSDV

PDR Vs. Node Speed

- AODV
- OLSR
- DSDV
- CLWPR

Average Packet Delivery Ratio (%) vs. Average Node Speed (m/s)

Katsaros et al.
CLWPR module for NS-3
Comparison with AODV, OLSR and DSDV

E2ED Vs. Node Speed

- AODV
- OLSR
- DSDV
- CLWPR

Average End-to-End Delay (sec) vs. Average Node Speed (m/s)
Comparison with AODV, OLSR and DSDV

![Graph showing Elapsed Time Vs. Node Speed for AODV, OLSR, DSDV, and CLWPR](image-url)
Summary

- We have introduced a novel cross-layer, position-based routing protocol for VANETs.
- The performance evaluation results suggest it can provide higher PDR than the protocols already implemented in NS-3, without compromising end-to-end delay.

Outlook

- Optimization of $f_i$ parameters.
- Extend navigation class to real scenarios
- Extent support for IPv6 and multiple interfaces
The work was done within the joint research project DRIVE-C2X, which is funded by the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement nº 270410.

www.drive-c2x.eu

- The source code is available for review at http://codereview.appspot.com/5343044