Controlling Channel Congestion using CAM Message Generation Rate

John B. Kenney: Toyota InfoTechnology Center, USA
Daniel Jiang: Mercedes-Benz R&D North America, USA
Gaurav Bansal: Toyota InfoTechnology Center, USA
Tessa Tielert: Karlsruhe Institute of Technology, Germany

ETSI ITS Workshop
February 6, 2013

jkenney@us.toyota-itc.com
http://us.toyota-itc.com/
Goals & Assumptions

Goals for this talk

• Persuade that under-utilization of channel is to be avoided
• Present our adaptive message rate control solution

Assumptions:

• When there is channel congestion, the main cause will be frequent CAM messages.
• Other message types (e.g. DENM, SAM, SPaT, MAP, …) are either also controlled, are rare, are given differentiated EDCA priority, or are sent on another channel
IEEE 802.11 MAC collisions

PER and CBR corresponding to max. throughput

Throughput maximized when CBR in 60-70% range

Test Parameters
- 30 radios
- 6 Mbps
- 544 µsec
- AIFSN = 6
- CWmin = 7

- An Adaptive DSRC Message Transmission Rate Control Algorithm, Weinfeld, Kenney, Bansal, ITS World Congress, October 2011
Why not be conservative?

- Sender can’t always tell what is important to receiver
- Sender doesn’t know what has been received by whom
- RF channel varies widely and is unpredictable
- Therefore, asking sender to “only send when necessary” is apt to lead to failures

- More progressive sending is safe because we control it
Does it make a difference? Intersection use case

Compare CAM rates: 2 Hz, 6 Hz, 10 Hz

Velocity 20 m/s

Colors represent different OBEs

Suburban Closed Intersection (Sunnyvale, CA)

Urban Closed Intersection (San Jose, CA)

Suburban ¾ Open Intersection (San Jose, CA)

Configuration:
Tx distance 50 meters
18 dBm power
10 MHz channel
QPSK Modulation

*Comparing Communication Performance of DSRC OBEs from Multiple Suppliers, Kenney, Barve, Rai, and Kanai, ITS World Congress 2012*
Success related to CAM rate

- Define application success as:
  
  *receive at least one CAM within a given time window*

- Examine for two time windows:
  - 3 to 5 seconds before impact (60 to 100 meters)
  - 2 to 4 seconds before impact (40 to 80 meters)

- Compare performance for 2 Hz, 6 Hz, and 10 Hz

<table>
<thead>
<tr>
<th>P{Application Success}</th>
<th>Urban Closed</th>
<th>Suburban 3/4</th>
<th>Suburban Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance to intersection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 Hz</td>
<td>6 Hz</td>
<td>10Hz</td>
</tr>
<tr>
<td>60 - 100 m</td>
<td>54%</td>
<td>90%</td>
<td>98%</td>
</tr>
<tr>
<td></td>
<td>7%</td>
<td>19%</td>
<td>30%</td>
</tr>
<tr>
<td></td>
<td>2%</td>
<td>7%</td>
<td>12%</td>
</tr>
<tr>
<td>40 - 80 m</td>
<td>93%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>34%</td>
<td>71%</td>
<td>87%</td>
</tr>
<tr>
<td></td>
<td>12%</td>
<td>32%</td>
<td>47%</td>
</tr>
</tbody>
</table>

CAM rate strongly correlated to application success in harsh fading environment
Why Message Rate?

- Lots of control knobs
- Can be used in combination or alone
- Can be responsive to different stimuli

Reasons why we emphasize message rate:
- Predictable impact independent of topology
- Maintain connectivity at distances of interest
- Fine grained control
- Large dynamic range (no obvious minimum)
3 functional steps:

1) **When sending CAM:** include locally measured CBR and largest CBR reported by a 1-hop neighbor

   CBR\textsubscript{LOC,k} \hspace{0.5cm} CBR\textsubscript{1,k} \hspace{0.5cm} Payload

2) **When receiving CAM:** accumulate maximum 1-hop and 2-hop CBR

3) Every measurement interval (e.g. 200 msec):
   - Measure local CBR
   - Compute max CBR observed within 2 hops: CBR(t)
   - Update local message rate \( r(t) \) according to
     \[
     e(t) = CBR\textsubscript{Thresh} - CBR(T) \\
     r(t+1) = (1-\alpha)r(t) + \beta e(t)
     \]

*Note: CBR\textsubscript{Thresh} is chosen to achieve desired PER/Throughput trade off*
LIMERIC Convergence

Convergence:
- Provable conditions
- Fair
- Exact

LIMERIC, Total Number of Nodes = 180

Message Rate (msg/sec) vs. Number of Iterations

LIMERIC: A Linear Message Rate Control Algorithm for Vehicular DSRC Systems, Kenney, Bansal, Rohrs, ACM VANET, September 2011
PULSAR Global Fairness

Without Information Sharing

With PULSAR Information Sharing

- Long road with many vehicles
- PULSAR approaches Max-Min Fairness

Design Methodology and Evaluation of Rate Adaptation Based Congestion Control for Vehicle Safety Communications, Tielert, Jiang, Chen, Delgrossi, Hartenstein, IEEE VNC November 2011
Improvements

• Provide differentiated transmission opportunities based on vehicle dynamics
  – Example: acceleration, yaw rate
  – Remain responsive to channel load

• Integrate power control based on desired range, e.g. function of speed
  – Relatively coarse power control may be sufficient
Conclusions

• High throughput corresponds to 60-70% CBR
• Under-utilizing channel can harm applications
• Message rate is preferred control for channel load
• LIMERIC+PULSAR algorithm has been implemented and tested:
  – Provides provable, fair convergence
  – Information sharing promotes global fairness
• Improvements to include vehicle dynamics and transmit power control
Thank You

John Kenney
jkenney@us.toyota-ite.com
http://us.toyota-itc.com/
Backup Slides
Message Rate Adaptation:

At regular intervals t (e.g. 200 msec):
1. Measure local channel busy ratio $\text{CBR}_{\text{LOC}}$
2. Compute maximum neighborhood $\text{CBR}(t) = \max\{\text{CBR}_{\text{LOC}}, \text{CBR}_1, \text{CBR}_2\}$
3. Compute adaptation error: $e(t) = \text{CBR}_{\text{Thresh}} - \text{CBR}(t)$
4. Compute new message rate: $r(t+1) = (1-\alpha)r(t) + \beta e(t)$

Selected to achieve desired PER/Throughput trade off

Message Generation at node k:

1. Include in header local and 1-hop maximum CBR:
   - $\text{CBR}_{\text{LOC},k}$
   - $\text{CBR}_1,k$
   - Payload
2. When generating message at time t, schedule next at $t + 1/r(t)$

Message Reception from node j:

1. Extract $\text{CBR}_{\text{LOC},j}$. Compile 1-hop max $\text{CBR}_1 = \max\{\text{CBR}_1, \text{CBR}_{\text{LOC},j}\}$
2. Extract $\text{CBR}_{1,j}$. Compile 2-hop max $\text{CBR}_2 = \max\{\text{CBR}_2, \text{CBR}_{1,j}\}$