PPT Contents

1. Transport Layer for VANET(from Current issues and future challenges):
   1. TCP: Reliable + in order
   2. UDP: un reliable and not in order
   3. TCP not suitable for VANET:
      1. Hand shake is time consuming
      2. retransmit vs send fresh message
      3. ACK cause traffic load and collision in MAC Layer
      4. cannot distinguish between failure and congestion
   4. UPD no longer faces congestion problem but has a new trouble: reliability
   5. solution: UPD + application protocol with FEC
      1. fountain
2. Application Layer:
   * 1. Safety application: alarm based & beacon based
        1. is single-hop beacon dissemination necessary?
        2. effect of distance on beacon reception rate: hidden terminal problem
           1. current solution : RTS/CTS but not suitable for broadcasting scheme
           2. main approach: control channel load: transmission range, interval, packet payload size

estimation based beacon dissemination( increase packet size instead of transmission interval) estimate beacon message and send in advance. every beacon message contains two categories of info: measured and estimated.

application layer scheduling: applications communicate with each other and disseminate their beacon in a specific order: SDMA

problems with SDMA:  
adaptive slot allocation

cluster maintenance

1. Routing/Forwarding/Dissemination
   1. Topology based:
      1. proactive/table drive： fisheye state routing[65]
      2. reactive/on-demand:
         1. route discovery: RREQ & RREP
         2. AODV:
            1. drawback: delay in route discovery
            2. route discovery packet lead to collision
            3. frequent route breakage
         3. DSR:
            1. stale cache info
            2. route maintenance mechanism does not repair a local link
            3. performance decreases with increasing mobility.
   2. Geography based： position of the destination should be available to the source node.
      * 1. DTN: forward continue in case of network fragmentation. store-carry-forward packet delivery at the cost of unbounded delay
        2. Non-DTN: The network is assumed to connect. reactive recovery process or drop package
        3. Beacon/non-Beacon
        4. Overlay & non-Overlay
   3. Broadcasting: target: suppress excessive rebroadcast message(storm)
      1. solutions: delay + probability(based on location and network density)
2. Challenges of routing protocols:
   * 1. connectivity and interference awareness
     2. interference comes from obstacles and channel contentions
   1. Table\_driven:
      1. a notion of end-to-end connectivity is not considered in **table-driven protocols**. (**connectivity**)
      2. update/exchange link state leads to high
      3. handle/predict route breakage leads to delay and waste of bandwidth
   2. Reactive :
      1. delay
      2. packet collision brought by broadcast storm
      3. no predict/handle network fragmentation
      4. Current solutions:
         1. VADD: beacon + DTN carry & forward strategy
         2. find neighbors through beacon message
         3. choose next hop by Location, Direction
         4. GPS, MAP info -> select path & model path delay
            1. downloading/updating map delay
            2. solution: path delay estimation or RSU dissemination
         5. performs well in light traffic condition
      5. GeOpps: DTN + opportunistically routing
         1. needs trajectory, may not suitable for security purpose
      6. GPSR: on-DTN, non-overlay, and beacon-based : greedy + rule-hand-rule

Multicast: (Muñoz A G. Multicast over Vehicle Ad Hoc Networks[J].)

* flooding: duplicate & disrupt
* tree: source based vs share-tree based: needs frequent rebuilt of trees(MAODV);

RREQ+ RREP + MACT

Tree link status managed by group leader (Hello)

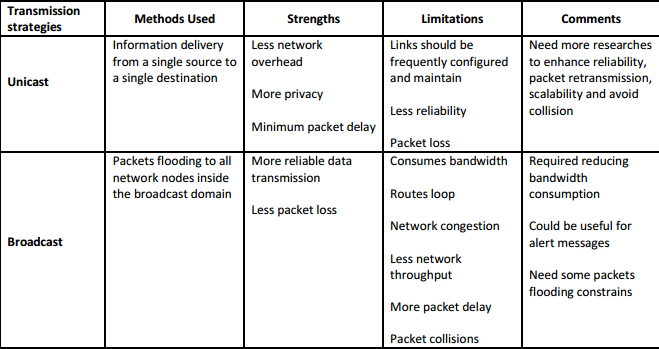
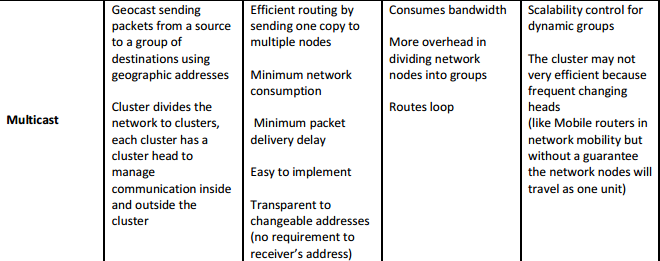
* mesh: alternative path. ODMRP

Broadcast JOIN QUERY + JOOIN REPLY

Mesh and tree based performance decreases with the increase of source numbers

* Overlay based: virtual network links among multicast group members
  + logical core to maintain multicast tree
  + JOIN\_REQ + JOIN\_ACK to establish unicast tunnel

Overview:

Thinks about Many Talks:

* + - 1. Pull & push
      2. Intermediate node processing to reduce redundancy
      3. Priority. Info requested by more nodes enjoys larger priority
  1. Main approach: two kinds of packets: Pull (request) and push (reply). Each packet contains the node ID of the source node and the forwarding node, temporal and spatial range the request is interested in, or that the reply package covers.

1. Request interested in popular ranges (ranges that many request packages specified ) will give be assigned with higher priority.
2. Remaining Problems:
   1. many-to-many can be simplified to one-to-many, no much difference.
   2. The main idea is based on geographical multicasting, which is, essentially, broadcast. That means this routing strategy is best-effort.
   3. What if there is no route reachable to the requested range? store-and-forward? the packet will time out soon. any recovery strategy?
   4. how to guarantee end-to-end QoS
   5. Flooding means both control and data packets are broadcasted. AODV only broadcasts control packet, and unicast data packet.
   6. ManyTalks is, in essence, an improved version of flooding.
   7. ROVER: reactive multicast protocol
      1. flooding ZRREQ to nodes in ZOF.
      2. Nodes in ZOR reply by ZRREP to its upper one-hop node, Nodes in ZOF but not in ZOR does not reply( unless it receive an reply from other nodes).
      3. Build a multicast-tree using ZRREQ & ZRREP. Each node stores next-hop information about the tree.
      4. Use VIN + Senquence Number to identify a route discovery process.

On-board devices: sensors, cameras, computing and communicating capabilities-> help driver make decision.

Broadcast: multi-hop versus single-hop

Multi-hop: flooding based.

Single-hop: carry-and-forward. rely on mobility to achieve connectivity. broadcast interval & info that needs to be broadcasted.

Metrics:

* Redundancy
* Reachability
* Failure Rate

Position based routing: GPSR: Greedy perimeter Stateless Routing