Multicast Operation of AODV
Outline

• Routing Protocols for Ad hoc Networks
  ➢ Multicast operation of AODV
Multicast challenges in MANET

• Multicast routing in the Internet
  ➢ Shortest Path tree algorithms
  ➢ Minimum Cost tree algorithms
  ➢ Constrained tree algorithms

• In MANET, need to optimize several resources simultaneously
  ➢ Nodes have limited power (battery-power)
  ➢ Wireless bandwidth is scarce
  ➢ No centralized access point or existing infrastructure to keep track of node mobility
  ➢ Potentially unpredictable and rapidly changing topology
Multicast in OADV

• AODV is capable of unicast, broadcast, and multicast
  ➢ Discover unicast and multicast routes on-demand, using a broadcast route discovery mechanism
  ➢ Route information obtained when searching for a multicast route can also increase unicast routing knowledge, and vice versa

• Broadcast data delivery
  ➢ Use source IP address and identification fields of IP header as unique packet identifier
  ➢ Processing and propagation of a data packet multiple times is prevented
### Routing Tables

**Route Table** *(next hop for routes to other nodes in network)*
- Destination IP address
- Hope count to destination
- Lifetime

**Multicast Route Table** *(multicast groups of which the node is a router)*
- Multicast group IP address
- Multicast group leader IP address
- Hop count to multicast group leader
- Lifetime

**Request Table** *(for optimization purposes)*
- Multicast group IP address
- Requesting node IP address (group leader)
Unicast Operation 1/2

• RREQ

  ➢ J_Flag, R_Flag, Broadcast_ID, Source.Addr, Source_Seq#, Dest.Addr, Dest_Seq#, Hop_Cnt

  ✓ J_Flag: Join flag  R_Flag: Repair flag

• RREP

  ➢ R_Flag, U_Flag, Dest.Addr, Dest_Seq#, Hop_Cnt, Lifetime

  ✓ R_Flag: Repair Flag  U_Flag: Update Flag

  ✓ Dest_Seq#: responding node’s record of the destination’s sequence number
Unicast Operation 2/2

(a) RREQ

(b) RREP
Multicast operation

• Use RREQ/RREP as previously discussed for unicast routing
  ➢ One new message called *Multicast Activation* (MACT)

• As nodes join the multicast group
  ➢ A multicast tree composed of group members and nodes connecting group members is created

• A multicast group leader (typically first node that requests membership to the group) maintains multicast group sequence number

• Multicast group members agree to be routers in the multicast tree
Route Request Message Generation 1/3

• Send a RREQ when
  ➢ Wish to join a multicast group (set J_flag in RREQ)
  ➢ Has data to send to a multicast group and no route exists
  ➢ Set Dest_Addr of RREQ to multicast group IP address
  ➢ Destination seq# set to last known seq# for that group

• If source node is aware of multicast group leader (through request table) and has valid route, unicast (include an extension field including IP address of multicast group leader), else broadcast

• A router for the group may respond to a join RREQ

• If not a join RREQ, any node with a fresh enough route to multicast group may respond
Route Request Message Generation 2/3

• If no RREP before timeout, broadcast another RREQ with an incremented $Broadcast_ID$

• Continue broadcasting RREQ up to $rreq\_retries$. If reach maximum number of times, node becomes multicast group leader and init group sequence number to 1

• If original RREQ is unicast, and RREP not received, all further RREQs are broadcast (do not include group leader extension field)

• Nodes receiving a join RREQ
  
  ➢ Check request table for entry
  
  ➢ If no entry, insert entry for multicast group address and requesting node IP address
Route Request Message Generation

(a) RREQ Message Propagation
Reverse Route Establishment

• Propagation of *non-join* RREQs is similar to unicast routing case

• For a *join* RREQ
  
  ➢ Maintain a corresponding routing entry in multicast route table (in addition to Route table)
  
  ➢ *Enabled flag* for entry set to FALSE (will be set to TRUE later, if route selected to be added to multicast tree)
Route Reply Message Generation

• If receive a join RREQ, reply (generate a RREP) if
  ➢ A router for multicast group with a recorded sequence number
    >= sequence number in RREQ
  ➢ Group leader

• Unicast RREP back to node indicated by Source_Addr in RREQ
  ➢ RREP contains last known sequence number for group and IP address of
    group leader
  ➢ A special extension field (Mgroup_Hop). Initialized to zero and
    incremented every time packet is forwarded (gives an indication to source
    about number of hops between itself and nearest member of multicast tree)
  ➢ Extension field (Group_Leader.Addr) containing group leader IP address
Route Reply Message Generation

• Nodes along path back to source node
  ➢ Add a route table and multicast route table entry for node from which RREP was received (forward path creation)
  ➢ Increment $Hop\_Cnt$ and $Mgroup\_Hops$

• If a node receives a unicast RREQ with its address as group leader, and it is not
  ➢ Ignore request (do not propagate further)
  ➢ Source node will send another RREQ without including the group leader field
Route Reply Message Generation

(b) RREPs sent back to source
Group Hello Messages

• Group leader responsible for maintaining group sequence number and disseminating it to group members

• Periodically broadcast a group hello message (unsolicited RREP with TTL > network diameter)

  ➢ Multicast group IP address and corresponding sequence numbers

• Sequence number incremented for each group Hello

• $Hop\_Cnt$ of group Hello init to zero, and incremented by each node receiving it

• Nodes use group Hello information to update request table
Multicast tree maintenance

• 3 main categories
  ➢ Selecting and activating the link to be added to the tree when a new node joins the group
  ➢ Pruning the tree when node leaves
  ➢ Repairing a broken link (re-establishing branches when a link fails and reconnecting the tree after a network partition)
Multicast Route Activation 1/2

• When a RREQ is broadcasted, multiple RREPs can be received (potential additions to multicast tree)

• Source node waits for a *timeout* value and meanwhile maintains the received route with the greatest sequence number and shortest number of hops to nearest member
  
  ➢ At end of timeout, enable next hop in multicast route table
  ➢ Unicast a Multicast Activation (MACT) to selected next hop
    
    📦 P_Flag (Prune), GL_Flag (Group Leader), Source_Addr, Source_Seq#, Dest_Addr
  
  ➢ If next hop, a member of multicast tree, does not propagate
  ➢ If not member, repeat steps above and send MACT to its next hop
  ➢ The process repeats until MACT reaches a member of the multicast group

• Nodes forward data packets along activated routes in multicast table
Multicast Route Activation 2/2

Branch added to multicast tree
Pruning

• If node not a leaf, revoke member status, but continue to serve as a router for the tree

• If leaf, unicast MACT message and set P_Flag (Prune Flag)

• A leaf node has only one next hop for the multicast group (father node)

• Father node removes information for sender node from multicast table

  ➢ If it had been made a leaf, similarly prune itself from tree (if not a member of the multicast group)

• Tree branch pruning terminates when either a multicast group member or a non-leaf node is reached
Link Breakage 1/2

• Nodes promiscuously record reception of any neighbor’s transmission
  ➢ Link breakage is detected if no packets received from neighbor within a specific time interval

• Node downstream of the break is responsible for repairing broken link
  ➢ Node further from multicast group leader

• Initiate repair by broadcasting a RREQ with $J_{Flag}$ set, and $M_{group\_Hop}$ set to distance of node from group leader
  ➢ Nodes which are at least as close to group leader (hop count to group leader smaller than indicated by $M_{group\_Hop}$), and have a fresh enough route may reply to RREQ
  ➢ If no RREP is received after a number of retries, assume network partitioned
**Link Breakage 2/2**

- If partitioned, need to choose a new group leader

- If node initiating tree rebuilding is a group member, it becomes the new leader

- If not a group member, and has only one next hop for tree
  - Prune itself from tree by sending MACT with P_Flag set
  - Next node if a multicast group member, becomes the new group leader, if not the process repeats until a new group leader is reached

- If not a group member, and has more than one next hop
  - Cannot prune itself from tree
  - Selects first next hop and unicasts a MACT with GL_Flag set
  - Next hop should become new group leader (if a group member)
  - New leader broadcasts a group Hello across its connected part of the network (U_Flag set, for nodes to update request table)
Reconnecting Partitioned Trees 1/2

- Two group leaders after partition

- If partitions reconnect, a node receives a group Hello that contains information differing from what is currently known

- If already a member of multicast group and member of partition whose leader has lower IP address, init rebuilding the tree (ensure that one group leader attempts rebuild)
  
  ➢ Unicast a RREQ with R_Flag set to its group leader

  ➢ Group leader sends back a RREP (no other rebuild efforts allowed)

  ➢ After receiving the RREP, unicast a RREQ to other group leader (using the node from which it received the group Hello as next hop)
Reconnecting Partitioned Trees 2/2

• The other group leader notes the repair flag
  ➢ Take larger of its record of the group sequence number and the received sequence number and increment by one
  ➢ Send back a RREP back to source node

• As RREP travels back to source node, it grafts a branch on the tree

• This group leader becomes the leader of the reconnected tree

• Next time, the new group leader sends a group Hello setting the U_Flag (nodes that used to belong to other partition note the new leader and the merge is complete)