



# **An Efficient Geographic Multicast Protocol for Mobile Ad Hoc Networks**

Proceedings of the 2006 International Symposium on a World of  
Wireless, Mobile and Multimedia Networks (WoWMoM'06)

Su-Yi-Chen



# outline

- **Introduction**
- **Related work**
- **Efficient geographic multicast protocol**
- **Simulation and Performance**
- **Conclusions**



# Introduction

- **tree-based protocols**
  - maintaining tree structure in these conventional protocols is very difficult
  - the tree connection is easy to be broken and the transmission is not reliable
- **mesh-based protocols**
  - providing redundant paths between the source and destination pairs at the cost of higher forwarding overhead
  - these conventional multicast protocols generally do not have good scalability due to the overhead



# Introduction

- EGMP: efficient geographic multicast protocol
- EGMP uses a hierarchical structure to achieve scalability.
- The network terrain is divided into geographical non-overlapping square zones, and a leader is elected in each zone to take charge of the local group membership management.



## Related work

- three components including group membership management, creation and maintenance of a tree- or mesh-based multicast structure and multicast packet forwarding.
- Adopting nongeographic mechanisms to implement the three components prohibits these protocols from scaling to large network size and large group size over the dynamic topology.



## Related work

- [3], [7] and [19] need to put the information of all the group members into the packet header, which is only applicable for the small group case.
- [21] made an effort to improve the scalability of geographic multicast protocol with group size, but it requires periodic local-range and network-range membership flooding.



# Notations and definitions

- *pos*: A mobile node's position coordinates  $(x, y)$ .
- *zone*
- *rzone*: The length of a side of the zone square
- *zone ID*: The identification of a zone
- *zone center*
- *zLD*: Zone leader
- *tZone*: The zones on the multicast tree
- *core zone*: The core zone is the root of the multicast tree
- *zone depth*: For each multicast session, a zone's depth reflects its distance to core zone.
- *zNode*: a node located in the same zone as the node being mentioned

# Zone structure building

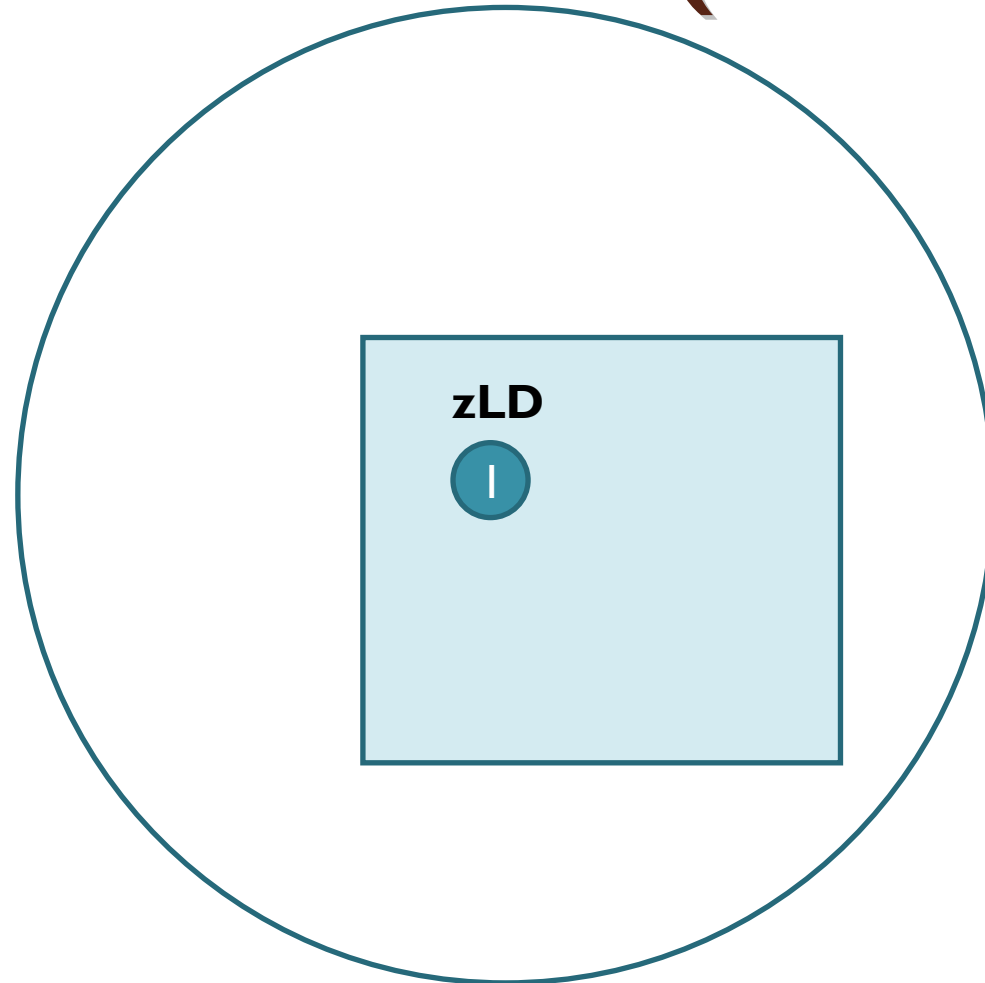
- In the underneath geographic unicast routing protocols, every node periodically broadcasts a BEACON message to distribute its position.
- On receiving a beacon from a neighbor, a node puts the node ID, *pos* and *flag* contained in the message into its neighbor table.

- $f_a = \left[ \frac{x-x_0}{r_{zone}} \right]$  be  
 $f_b = \left[ \frac{y-y_0}{r_{zone}} \right]$  nei  
 u... by t

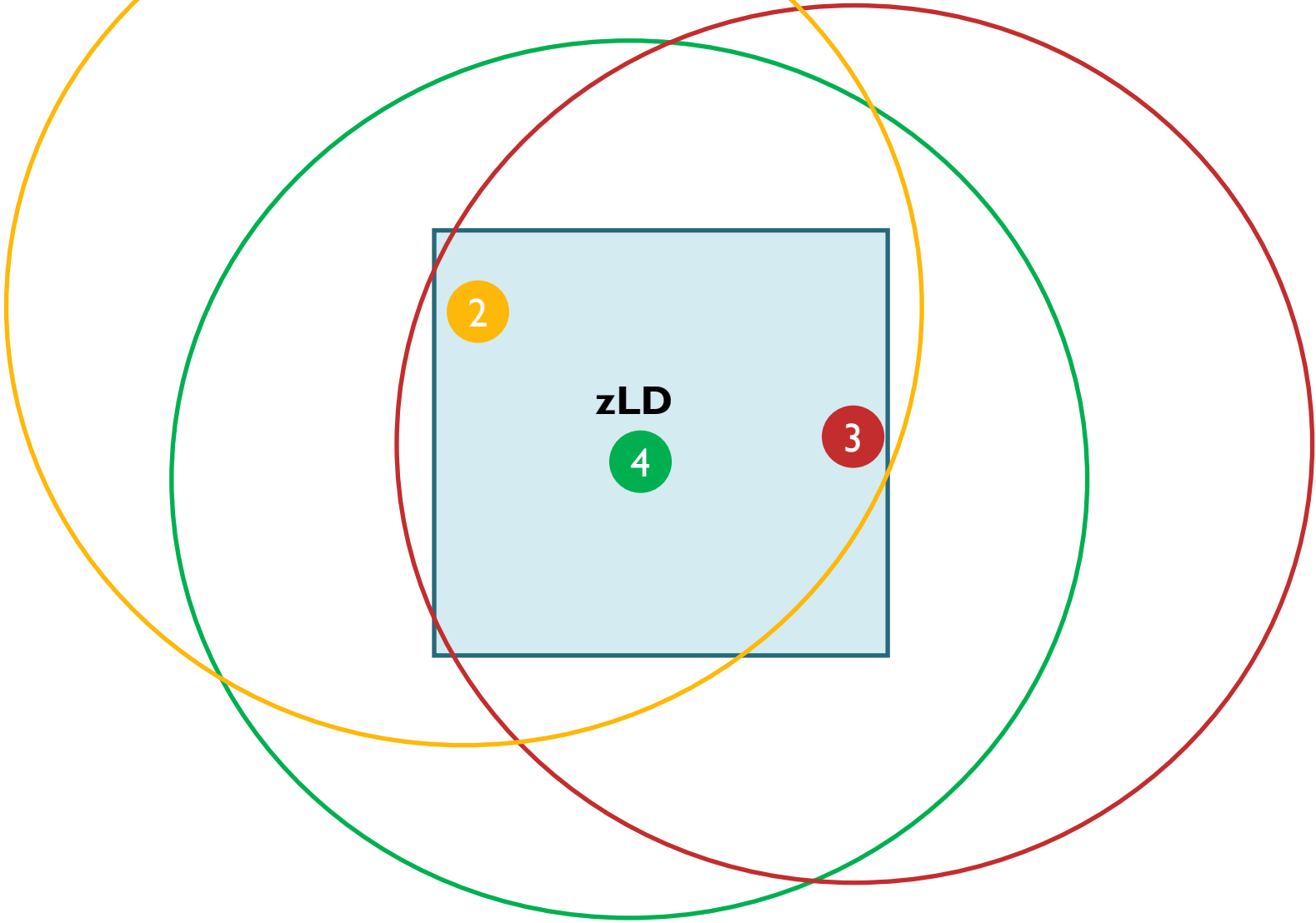
node ID	position	flag	zone ID
16	$(x_{16}, y_{16})$	1	(1, 1)
7	$(x_7, y_7)$	1	(0, 1)
1	$(x_1, y_1)$	0	(0, 2)
13	$(x_{13}, y_{13})$	1	(1, 2)



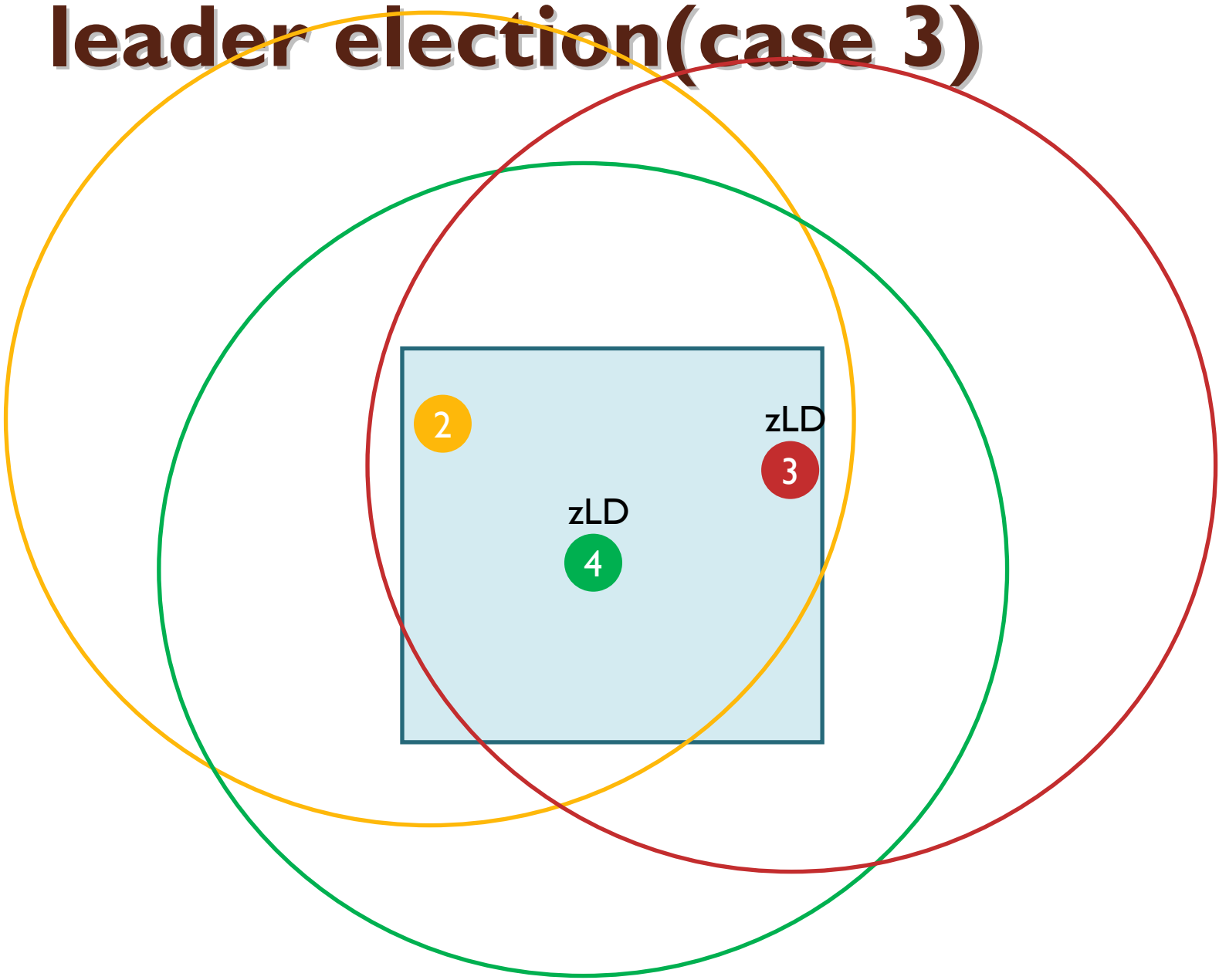
# leader election(case 1)



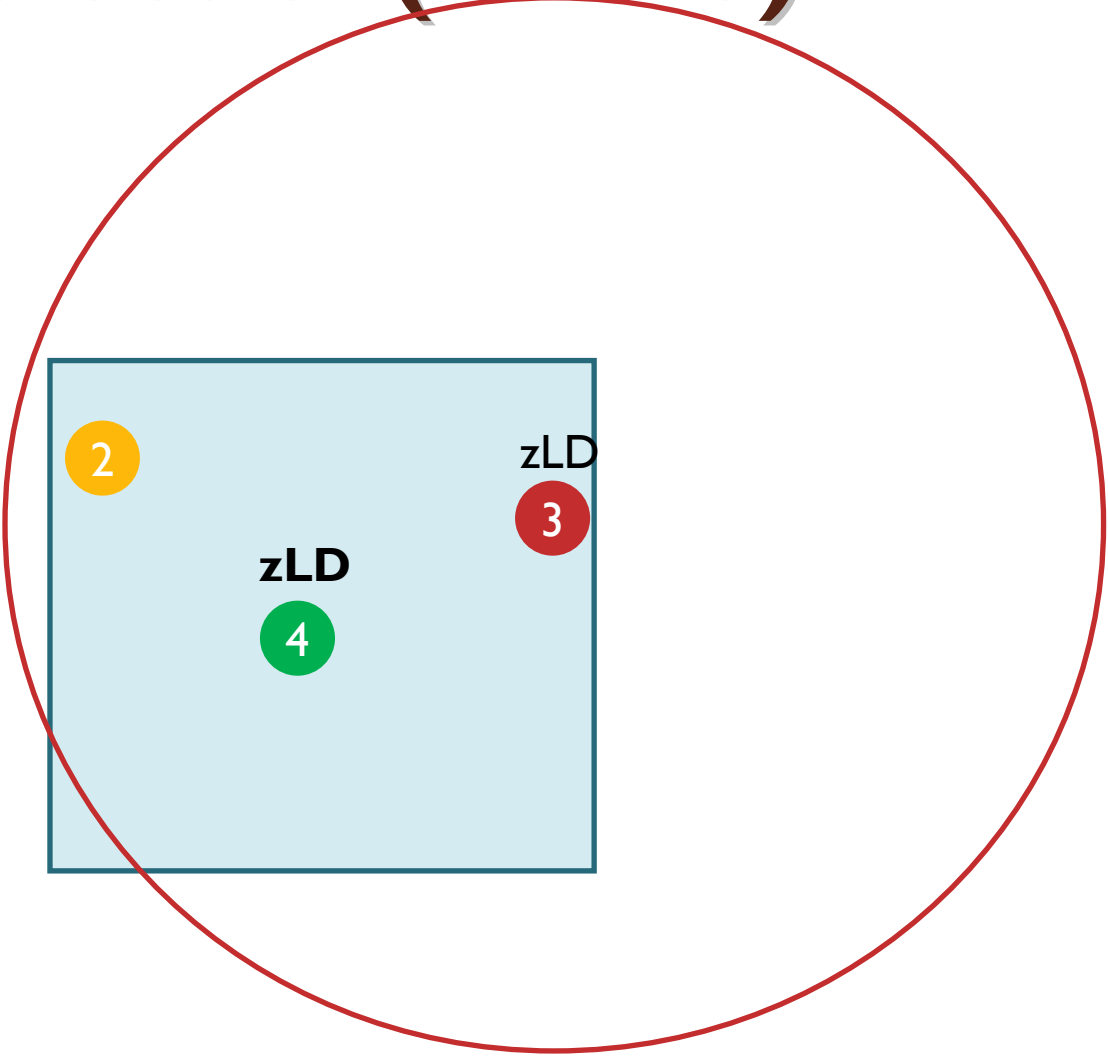
# leader election(case 2)



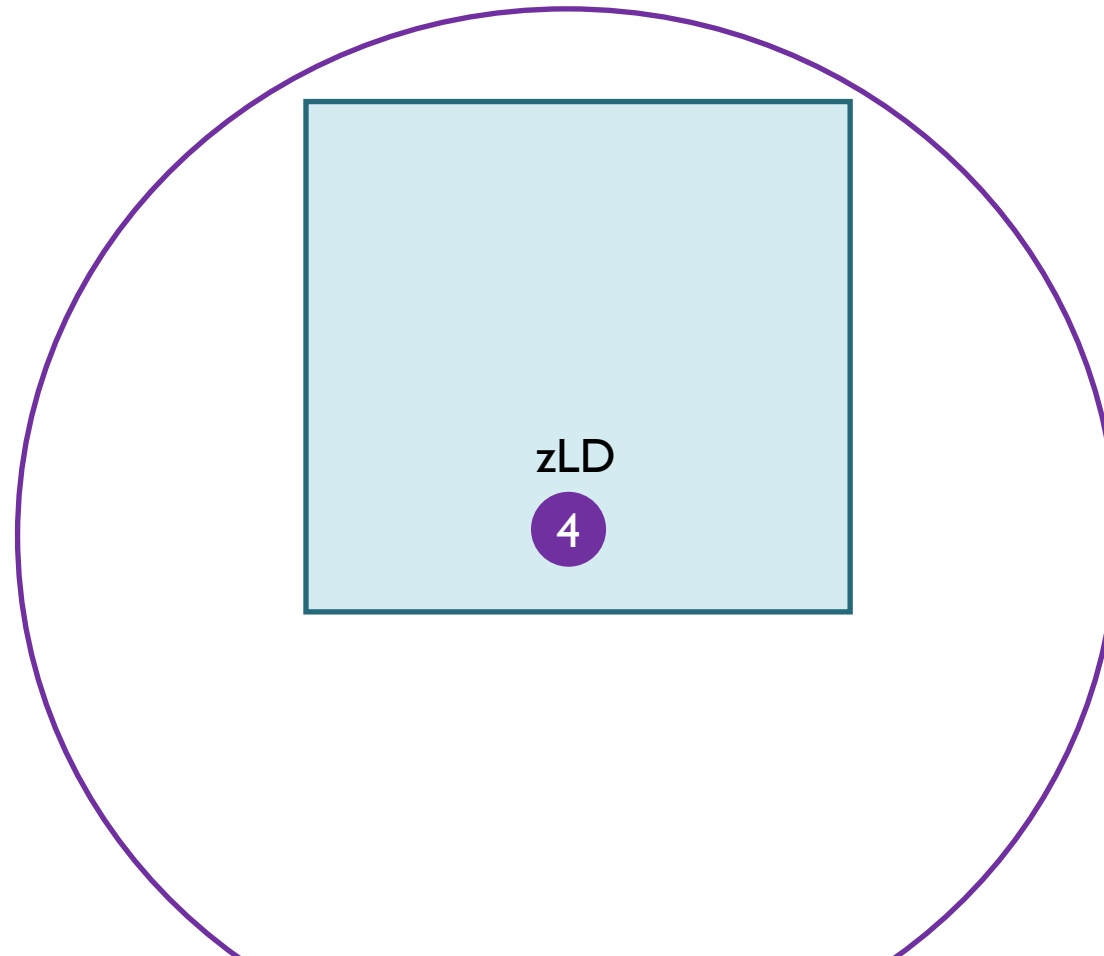
# leader election(case 3)

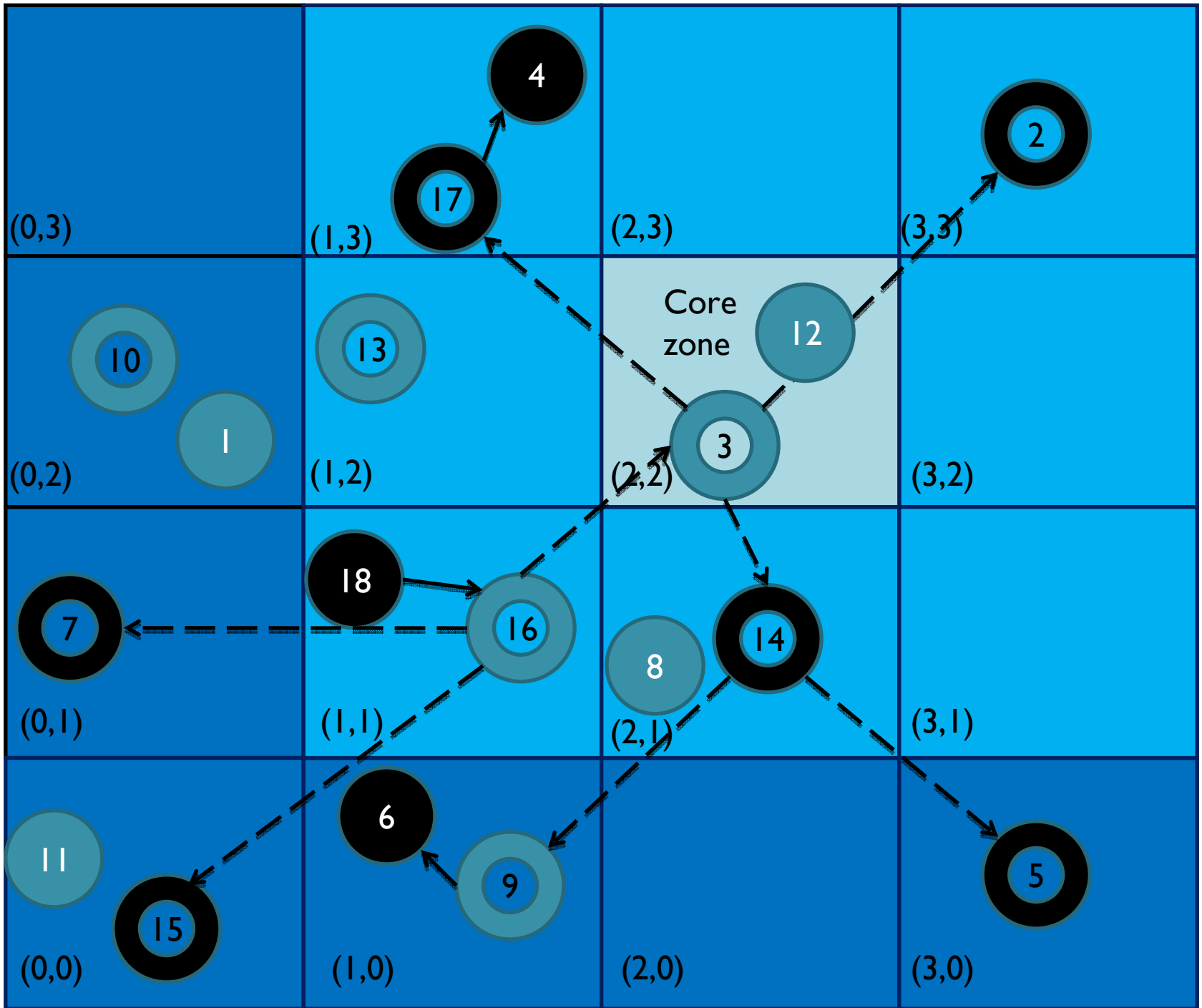


# leader election(case 3)



# leader election(case 4)





# Multicast session initiation and termination

- When S wants to start a multicast session G, it will announce the existence of G by flooding a message *NEW\_SESSION(G, zoneIDs)* into the whole network.
- When a zLD receives the *NEW\_SESSION* message, it will record the group ID and core-zone ID into its multicast table
- S floods a mess G.

group ID	$G$
core-zone ID	$(2, 2)$
upstream zone ID	$(2, 2)$
downstream zone list	$(0, 1), (0, 0)$
downstream node list	18



# Multicast session initiation and termination


- Every member node will keep a membership table with each entry as  $(G, core\_zID, isAcked)$

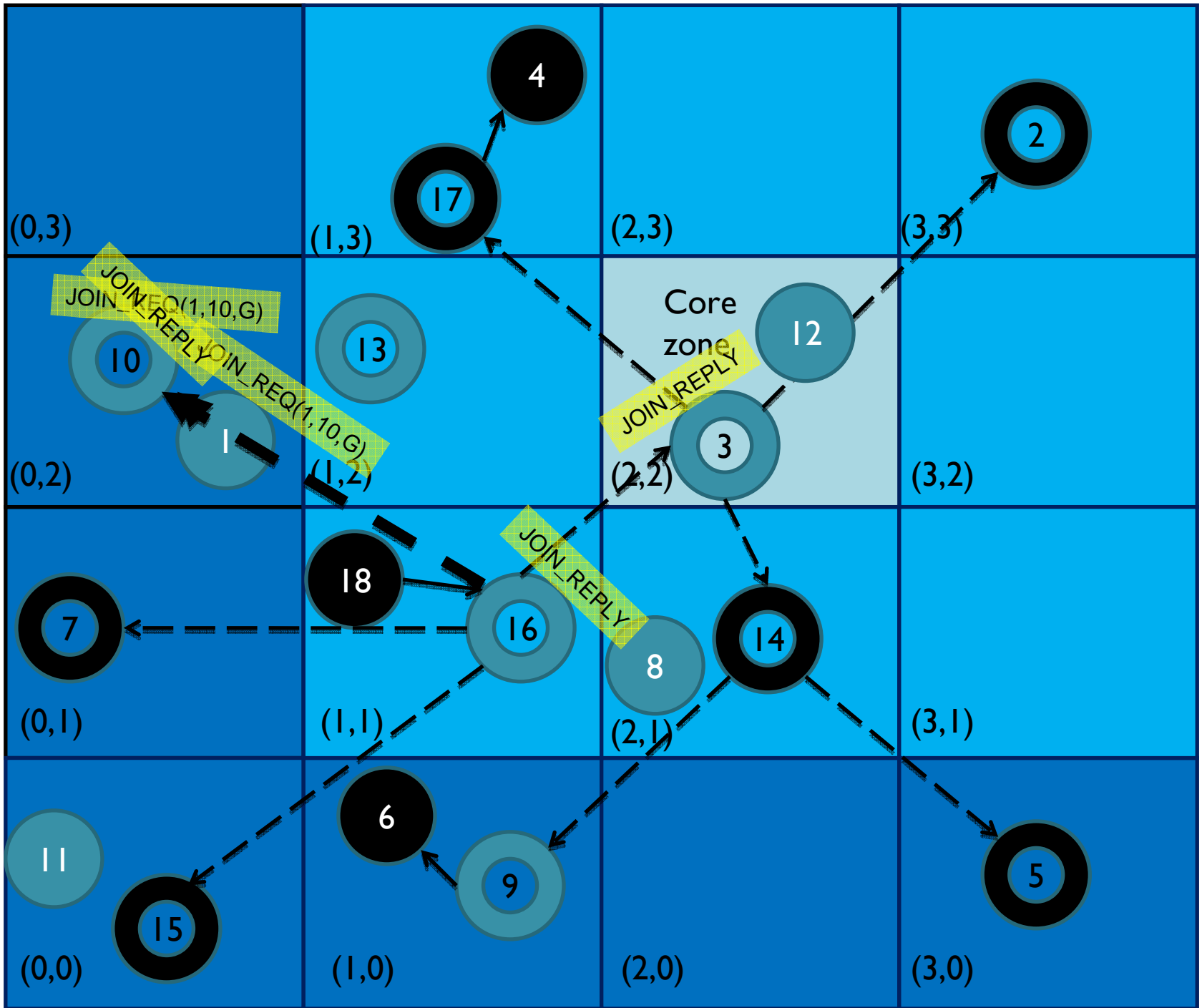




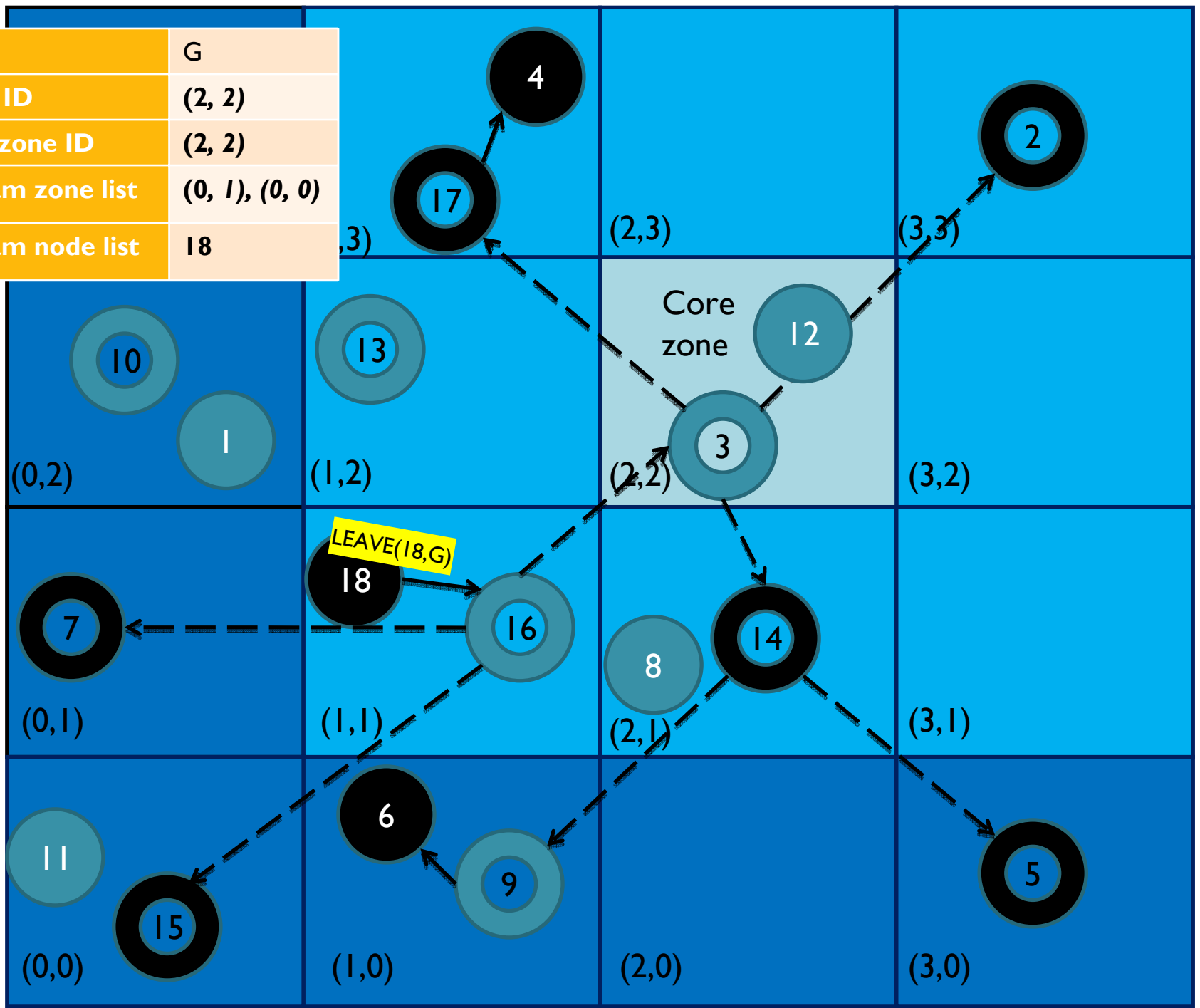
# **Multicast group joining and leaving**

---

- 
- If new nodes or zones are added to the downstream list, the leader will check the core-zone ID and the upstream zone ID and take corresponding action.
  - If it doesn't know the core zone, it starts an *expanded ring search*.
  - If its upstream zone ID is unset, the leader will represent its zone to send a JOIN\_REQ message towards the core zone; otherwise, the leader will send back a JOIN\_REPLY to the source of JOIN\_REQ



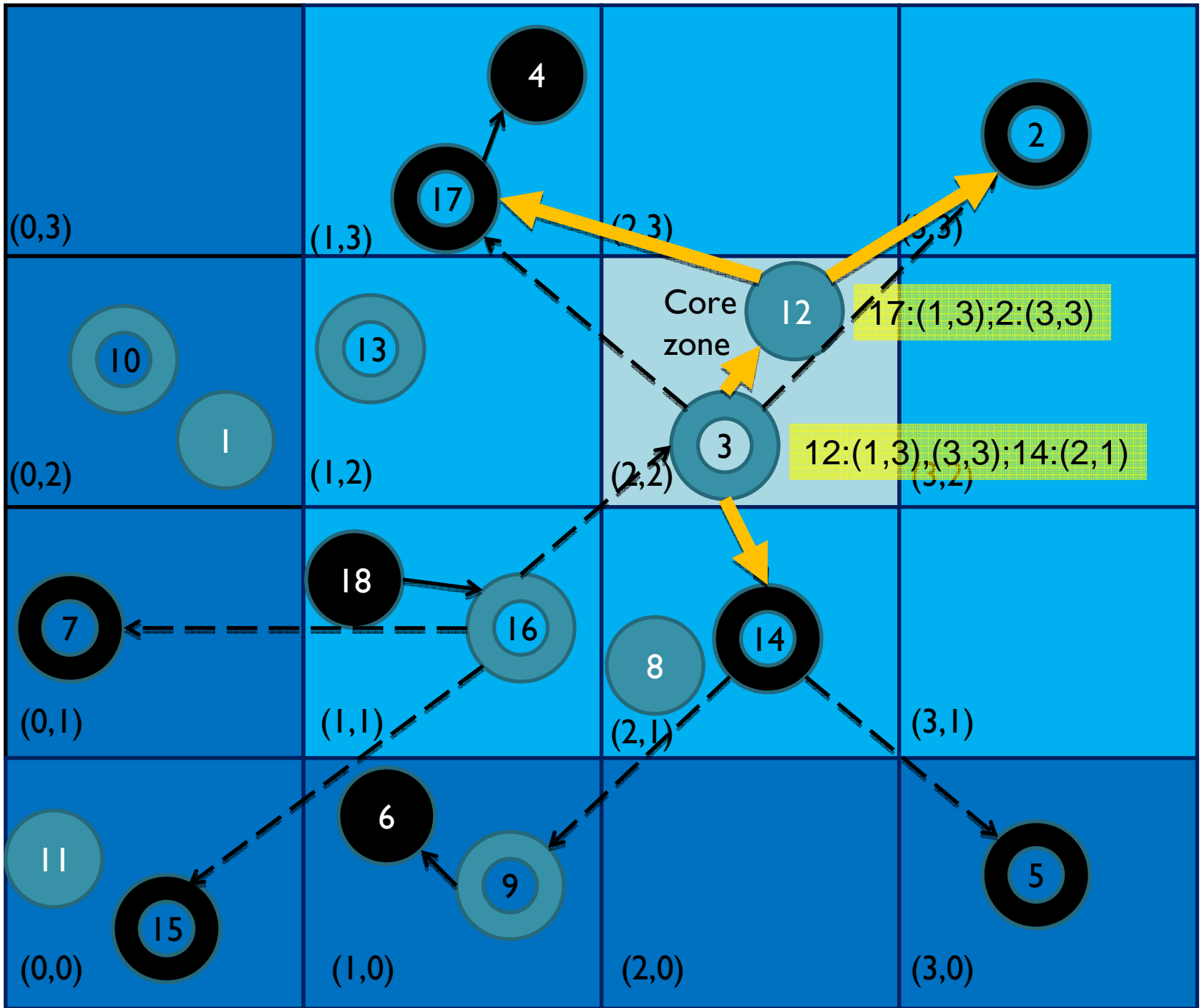
Group ID	G
core-zone ID	(2, 2)
upstream zone ID	(2, 2)
downstream zone list	(0, 1), (0, 0)
downstream node list	18





# **Multicast packet delivery**

---





## Moving between zones

- During this joining process, to reduce the packet loss, its BEACON message will also be unicast to its old zone to update its position and the old zLD will continue forwarding the multicast packet to M.
- When the rejoining process finishes, M sends a LEAVE message to its old zLD.



# Moving between zones

- To look for the new leader, the old zLD compares the positions of the zNodes in its neighbor table and selects the one closest to the zone center as the new zLD. It then sends its multicast table to the new zLD, which will announce its leadership role immediately through a BEACON.
- During the transition, the old zLD may still receive packets destined to the leader. It will forward all these packets to the new zLD when the process is completed



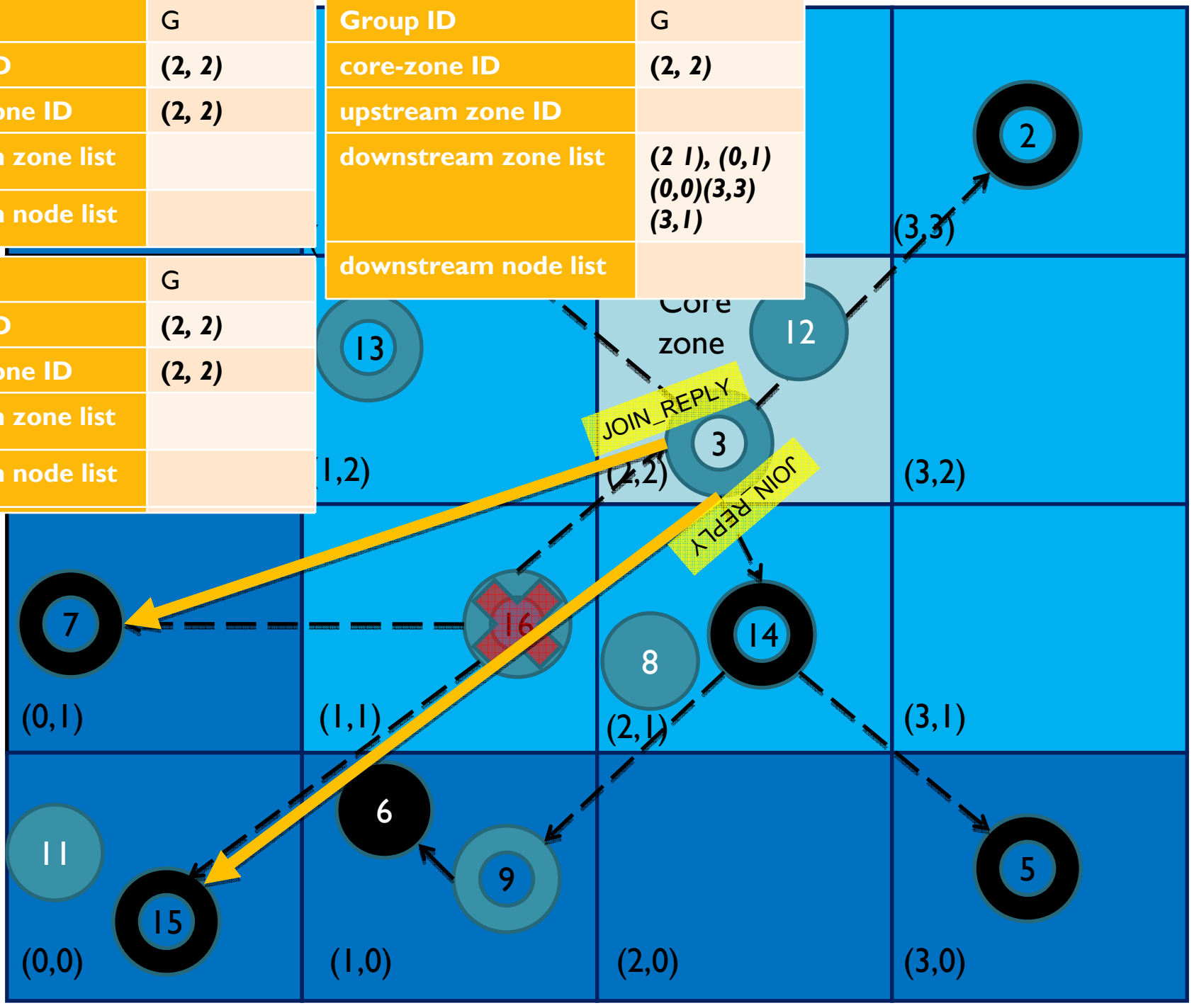



# **Dealing with empty zones**

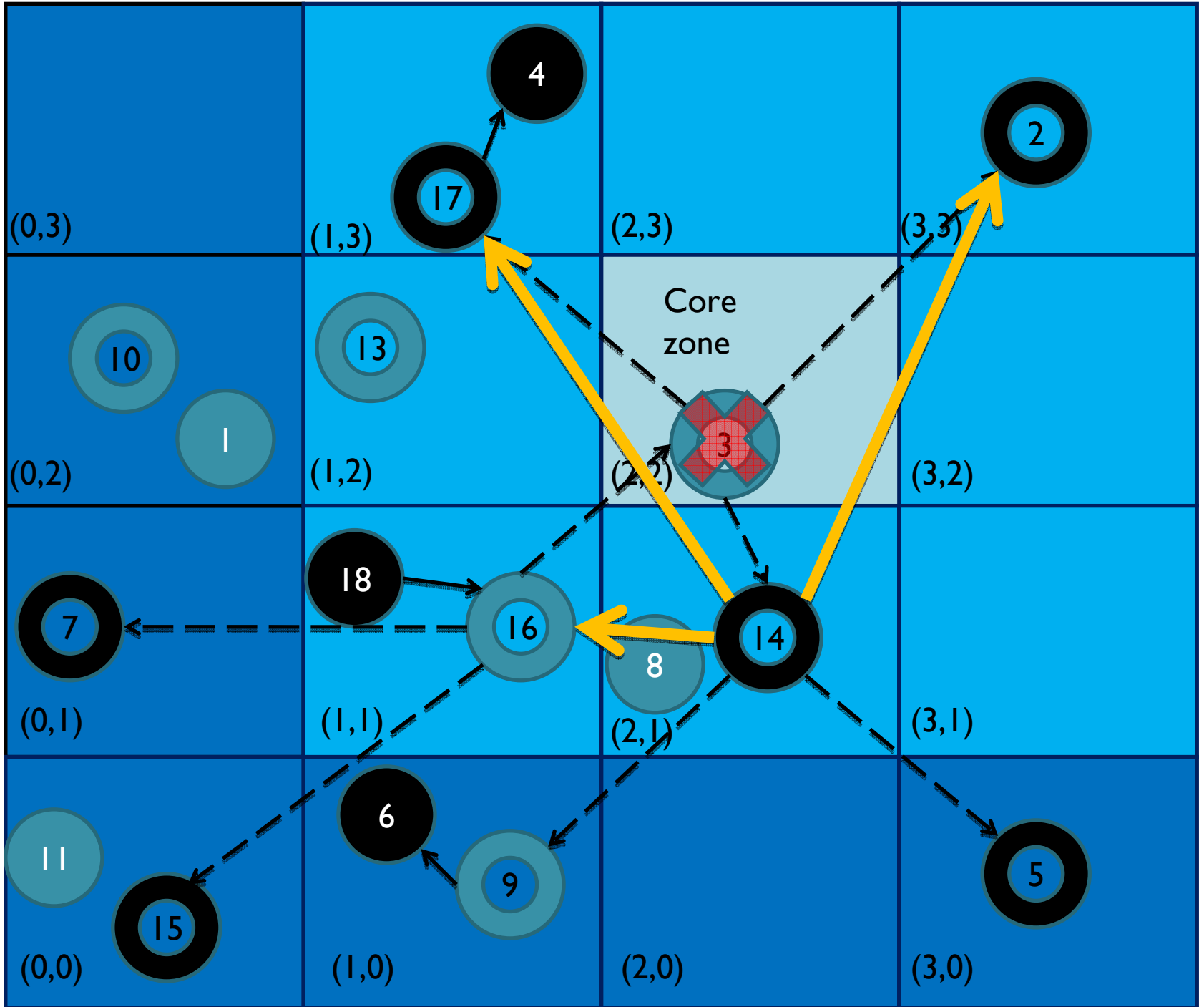
Group ID	G
core-zone ID	(2, 2)
upstream zone ID	(2, 2)
downstream zone list	
downstream node list	

Group ID	G
core-zone ID	(2, 2)
upstream zone ID	
downstream zone list	(2, 1), (0, 1) (0, 0)(3, 3) (3, 1)
downstream node list	

Group ID	G
core-zone ID	(2, 2)
upstream zone ID	(2, 2)
downstream zone list	
downstream node list	



- 
- the zLD will check its connected neighboring zones and choose the one closest to the core zone as the new core zone.
  - The zLD then forwards its multicast table to the new core zone, and floods a **NEW\_CORE** message throughout the network to announce the change.





# Tree branch maintenance and optimization

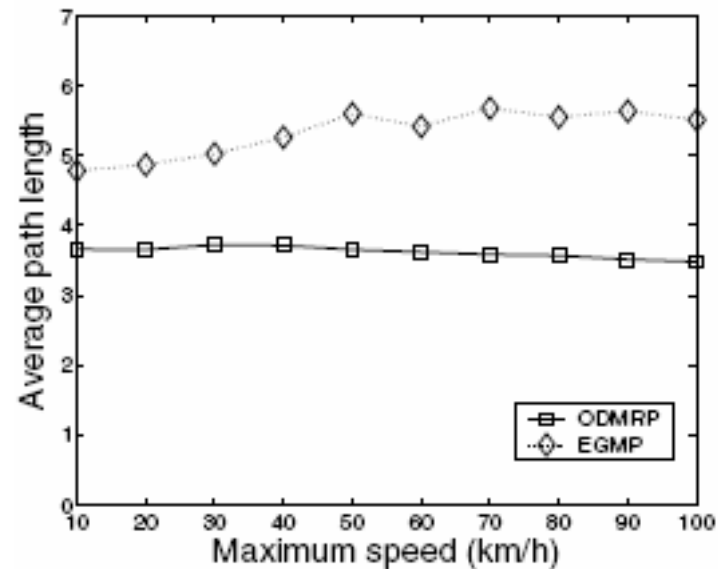
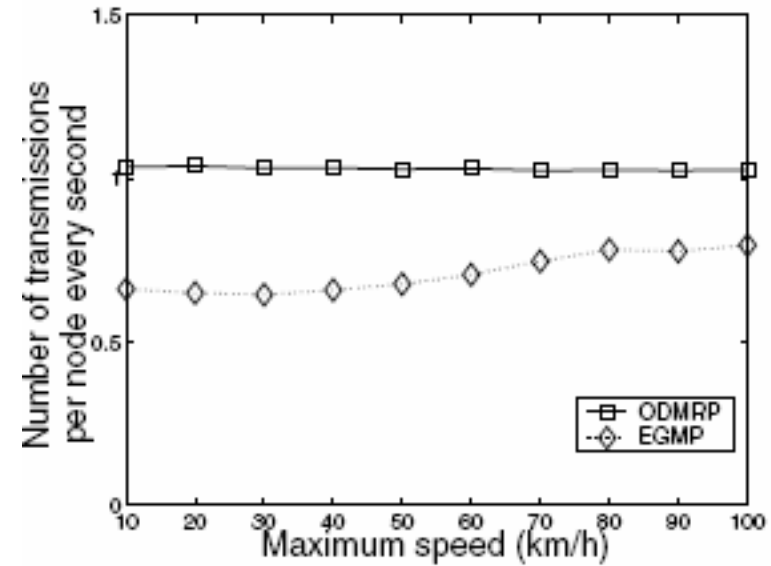
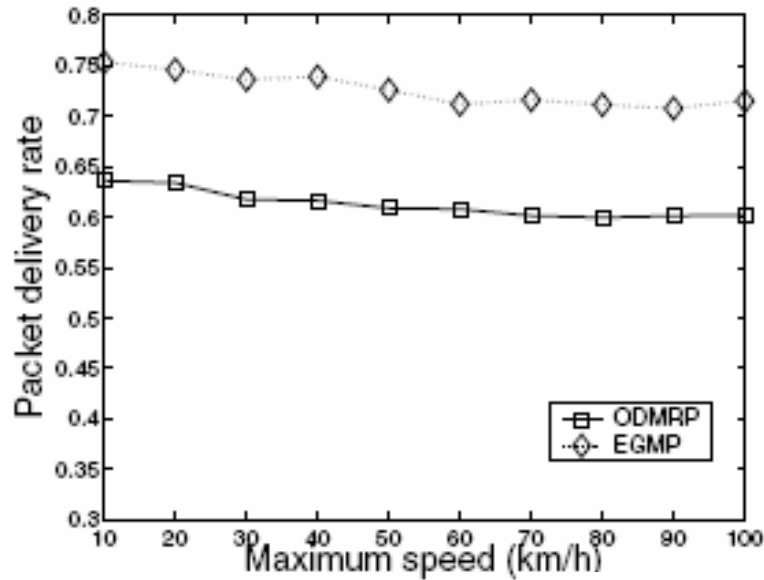
- if there are no multicast packets for delivering for a period of  $Intval_{active}$ , the zLD of a tZone will send an *ACTIVE* message to its downstream nodes and zones.
- When a tZone doesn't receive any packets longer than a period of  $2 * Intval_{active}$ , it assumes that it loses the connection to the multicast tree and restarts a joining process.
- If the old upstream zone is detected as still connected later and the joining process to the new upstream zone is finished, the one closer to the core zone will be kept, while the other one will be removed by sending a *LEAVE* message.
- If the two upstream zones have the same distances to the core zone, one of them is randomly selected.



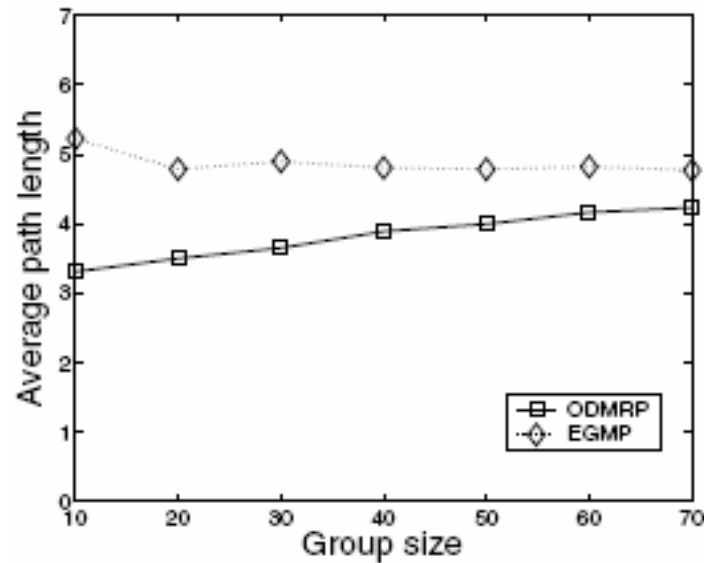
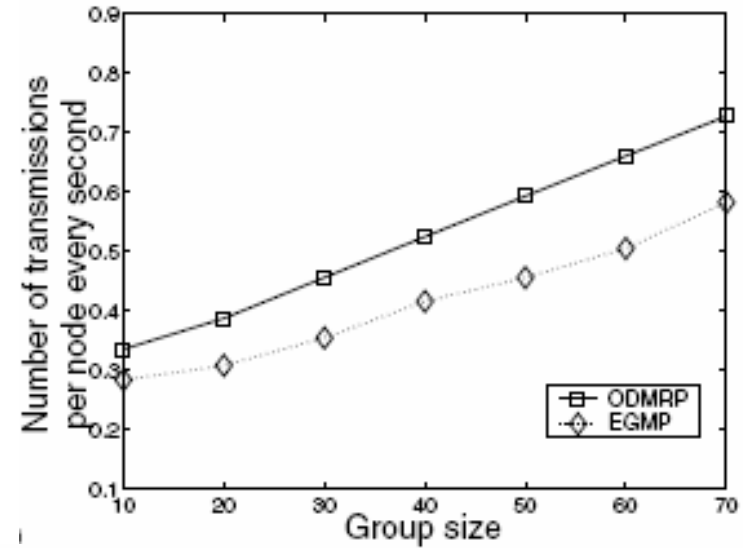
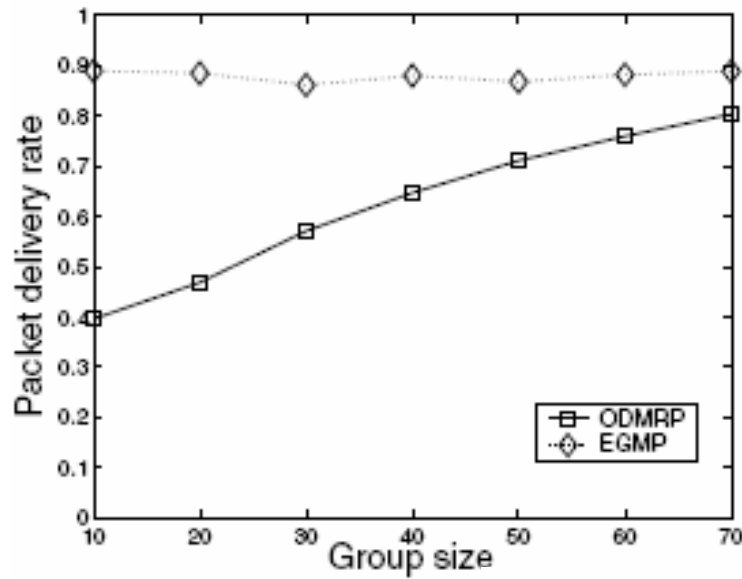
# Simulation environment

- distributed in the area of  $3000m * 1500m$
- The nodes move following the *random waypoint mobility model*
- The transmission range is 250m
- the zone size is 150m
- $(Intval_{min}, Intval_{max}) = (0.5s, 4s)$ ,
- $D_{beacon}$  and  $D_{border}$  are set as 50m and 5m
- *TimeoutZT* is 4.5s and *Intvalactive* is set as 2s.

# Performance(moving speed)



# Performance(group size)







## **future work**

- enhance our protocol without the help of core zone, to achieve more optimal routing and lower control overhead.