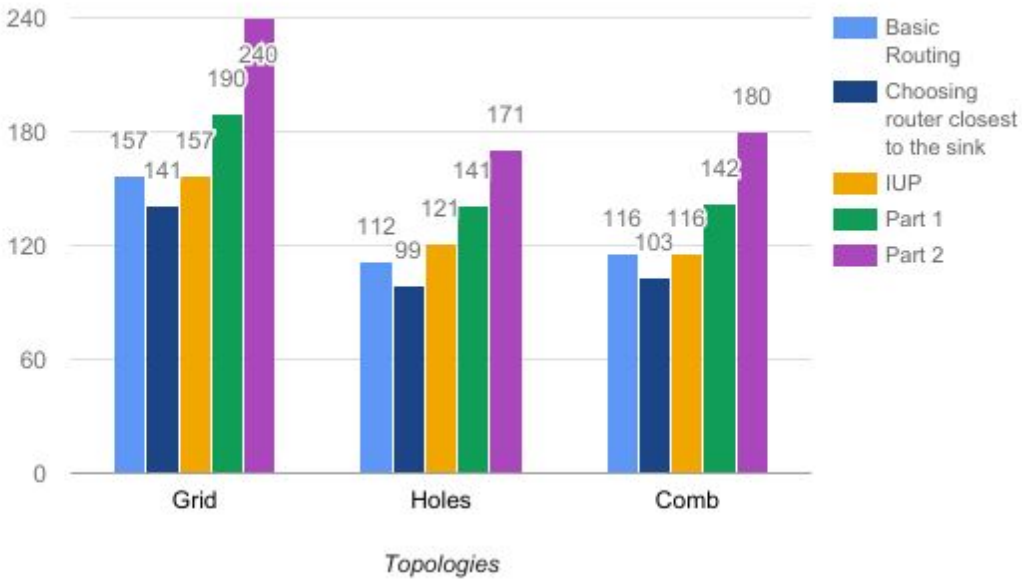
Report on Lab 2

- Kruthika Suresh Ved & Romi Zaragatzky (Group 6)

1. Results

	Basic Routing	Part 1	Part 2
Grid	157	190	240
Holes	112	141	178
Comb	116	142	180



2. Part 1 - Basic Routing

Idea behind the algorithm

In the basic routing algorithm, as soon as a node received an announcement from a node that was closer to the sink, it just chose this node as its router. The algorithm ignored all announcements received after this and the router wasn't recalculated until the next round.
Our first improvement of the basic routing algorithm was to update the router every time the node receives a new announcement, such that it provides the node a shorter distance to the sink than through the current router.

The second improvement is that the battery lifetime of a candidate for being router is taken into consideration. The battery lifetime is sent in the announcement message and when a node receives an announcement, it checks if the battery level of the candidate is enough to choose it as the router.

Failed Improvements

1) Using Independent Update Procedure
   We first used the IUP given in A Responsive Distributed Routing Algorithm for Computer Networks, but then we realized that our skeleton code sends periodic announcements, while in the paper announcement by normal node is sent when it updates its router to Sink. We changed our code accordingly, but then the improvement was not significant enough.

2) Upon announcement receive, update the router if router was not assigned or if distance of mess->from to the sink is less than the current router's distance to sink. It did not consider the distance from node to the router.

2. Part 2 - Clustered Routing

Idea behind the algorithm

The improvement in this assignment consists of several steps. In order to get the clustering to work correctly, we had to add a third round to the existing algorithm.
In the **first round**, each node becomes a cluster head with the probability 50% (using a provided random() method). Then, the cluster heads send out announcements to everyone. When a cluster head receives an announcement, it checks if the sender is a better router than the current one (exactly as in part 1) and updates the router if necessary. Every node has a global variable that says who is this node's cluster head. In the case of a cluster head, this global variable is set to it's own id. When a normal node receives an announcement, it checks if the sender is a better cluster head than its current cluster head (same checking candidate to be a router). If that is the case, it makes the sender its new cluster head.
In the **second round**, the normal nodes send their messages to their respective cluster heads. The cluster heads use a count variable that is incremented each time a new content message is received from a normal node. In this way, they aggregate the content sent by the nodes in their cluster.
In the **third round**, the cluster heads send their respective counts to the sink. If a cluster head receives a content message from another cluster head (which uses this one as a router), it just passes the message on to its own router.
The sink is also initiated to be a cluster head.

Failed Improvements

1) We first considered the router going to nearest cluster head in more than one hop. For this, we added 4 rounds which consisted of :
    I. All cluster heads send announcement. Non cluster heads (Routers) receive announcements from cluster heads, they update their router and clusterhead info to the nearest cluster head. Cluster heads updates their route to the Sink using algo given in Part 1.
    II. Routers send announcements. The cluster heads ignore them but the other routers update the cluster heads if the total distance to the sender's cluster head is less than distance to the currently assigned cluster head.
    III. Router sends content to respective cluster heads. Cluster head receives the content message and calculates the total count
    IV. Clusterhead send the total count to the Sink.

    The problem lay in Step 2. Since we in this case used four rounds to make the sink collect one round of content messages, it increased the time of the total execution and therefore simulation stopped even before the battery turned 0.

2) Improve by changing the probability of choosing the cluster heads. One major problem in our design is that we do not handle cluster head crashes. Therefore, by decreasing the probability of choosing the cluster head meant more nodes were dependent on each cluster head. Therefore, upon cluster head crash, much fewer messages were collected by the sink.
    On the other hand, increasing the probability only added more routing and more messages, defeating the whole purpose of clustering. Therefore, we have kept the probability to be 50%.