Problem Statement:

Modification of Epidemic Routing Algorithm to delete the delivered packets from other nodes

Solution:

The solution is, every data would have three extra fields,

For source node:

1. no of intermediate nodes which have the data for whom the source node was the immediate source,
2. no of intermediate nodes which have the data for whom the source node wasnt the immediate source,
3. no of intermediate nodes which have the “**has the data reached its destination data**”
4. data_reached_destination

For intermediate nodes:

1. no of other nodes to whom this node have sent the data packet,
2. no of other nodes to whom this node have notified that the data is received by the destination,
3. data_reached_destination

For the destination node:

1. data_reached_destination

Basic concept:

Each intermediate node has a count of how many new node have it transferred the data too.. and as soon as it gets the value of data_reached_destination=true, its work is to notify that no of nodes whom it has transferred the data to. For example, if it has sent the data to 5 nodes, it is bound to notify 5 nodes(any node) that the data has reached the destination.

As soon as any node gets its “data_reached_destination” value equal to true, it looses its capacity to send data to any node which doesnt have the data. It can send data to only those nodes, which have already received the data from any other node or previously from this node.

After getting the value of “ data_reached_destination” equal to true, the only objective of the node becomes to make

(the no. of other nodes to whom this node have sent the data packet) == (the no. of other nodes to whom this node have notified that the data is received by the destination)

As soon as the values becomes equal, the intermediate node deletes the data package and retains the uid so that any other node trying to send the same data would not be accepted.

Now comes the role of the source node.

Whenever the source node send a data to any intemediate node which doesnt already have the data, adds 1 to the value of **(no of intermediate nodes which have the data).**

And if the node already has the data, adds the *(no of other nodes to whom this node have sent the data packet) of the intermediate node* to ***(no of intermediate nodes which have the data)*** *for whom the source node wasnt the immediate source****.***
If (data_reached_destination)== false, ()=0,
else if(intermediateNode.data_reached_destination==true)adds *(no of other nodes to whom this node have notified that the data is received by the destination) of the intermediate node* to *(no of intermediate nodes which have the "****has the data reached its destination data****")*
*else adds 1 to (no of intermediate nodes which have the "****has the data reached its destination data****")*
And as soon as the value of (*data_reached_destination) becomes true,* the source node starts converting the intermediate nodes having the (*data_reached_destination) =false.*

As soon as the value of summation of first two fields becomes equal to third field, it starts removing the uids from all the infected intermidiate nodes. And finally all data packets get deleted.

Working:

There are some points to be noted,

1. Whenever an intemediate node gets a new data (at the time of insert into the data table), the value of 3 parameters will be (0,0,false)

There are two situations while a node sends a data to another node:

1. The receiving node already has the data:

```
if(sending_node .data_reached_destination== true)
    receiving_node.data_reached_destination=true

else if( receiving_node .data_reached_destination== true)
    sending_node .data_reached_destination= true
```

2. The receiving node doesnt have the data:

```
if(sending_node.data_reached_destination == false)
    sending_node.no of other nodes to whom this node have sent the data packet ++
else
    receiver rejects the data
```