

COMPUTER COMMUNICATION HW2 REPORT

STUDENT NAME : HÜSEYİN ERDOĞAN
STUDENT NO : 040100054

CRN : 12337

BLG433E HW2 REPORT

1 Question 1

1.1 How do you create random network?

First i created vector that stores is of all edges. Than i choose a id for a edge from this vector and removed it from vector. If chosen edge's current_degree is 0 or chosen edge equals to current edge or current edge and chosen edge are conected, than i chose a random id from created vector. Than i controlled that current edge and chosen edge have current_degree that is higher than 0. If both have current_degree that is higher than 0, than i connect current edge and chosen edge. And i repeated this for all edges in graph.

Code for random network:

```
for(int i=0 ; i<edges.size() ; i++){
    vector<int> t_vector2;
    vector<int> t_vector = create_random_vector();
    while( t_vector.size() > 0 ){
        int t = rand()%t_vector.size() + 0;
        int t_index = t_vector[t];
        t_vector2 = delete_element_in_vector(t_index,t_vector);
        t_vector = t_vector2;
        while( edges[t_index].current_degree == 0 || t_index == i ||
            is_edges_connected(i,t_index) == true ){
            if(t_vector.size() != 0){
                t = rand()%t_vector.size() + 0;
                t_index = t_vector[t];
                t_vector2 =
                    delete_element_in_vector(t_index,t_vector);

                t_vector = t_vector2;
            }
            else{
                break;
            }
        }
        if( edges[i].current_degree > 0 && edges[t_index].current_degree >
            0 && i != t_index){
            insert_edge(i,0,t_index,0);
        }
    }
}
```

1.2 How do you implement flooding?

For packets i create packet class

```
class packet{
    int packet_id; //for storing id of sender
    int hop_count; //for storing hop count of packet
};
```

For edges i create edge class

```
class edge{
    int id; //for storing id of edge
    int real_degree; //for storing degree of edge that is constant
    int current_degree; //for storing degree of edge that can change when another
    edge is connected
    bool is_visited; // for storing if this edge is visited
    vector<packet> packets; // for storing packets that this edge received
```

BLG433E HW2 REPORT

```
vector<edge*> neighbor_vector; // for storing neighbors of this edge
};
For graph i create graph class
class graph{
    vector<edge> edges; //for storing all edges of graph
    int num_edges; //for storing number of edges in graph
};
```

First i traverse graph with using bfs search algorithm. When i enqueue a edge from queue, i decrease hop count of packet and send packets to edge's neighbors. And if hop count of any packet is equal to 0 then i quit the loop.

1.3 How does your implementation work?

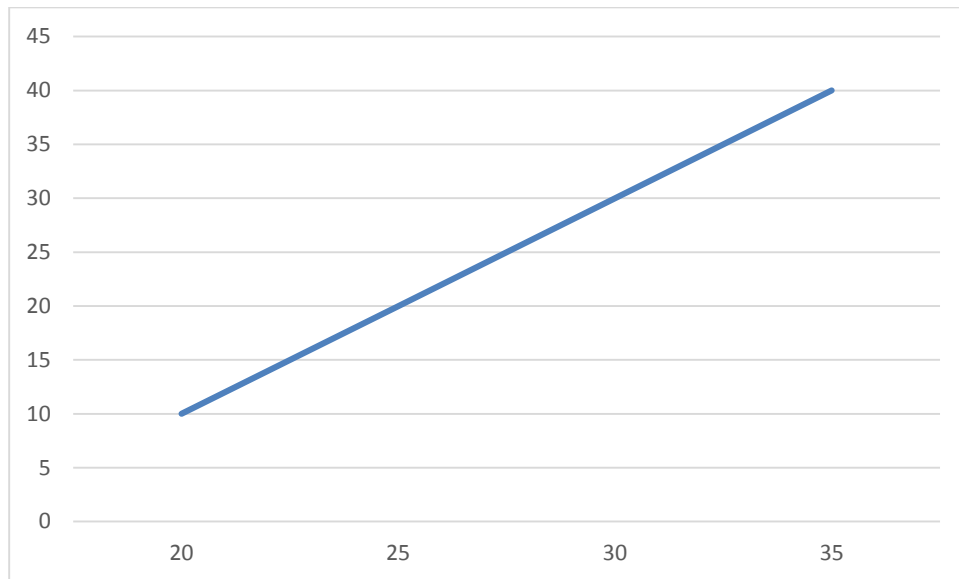
Explanenation of code for Flooding

```
//if queue is not empty, enter the while loop
while( t_queue.empty() == false ){
    //pop first element that is edge id from queue
    int t_enqueue = t_queue.front();
    t_queue.pop();
    //if popped edge is not visited
    if(edges[t_enqueue].is_visited == false){
        //set edge visited
        edges[t_enqueue].set_is_visited(true);
        //modifiy created packet for this edge
        t_packet.set_packet_id(t_enqueue);
        //visit all neighbors
        for(int i=0 ; i<edges[t_enqueue].neighbor_vector.size() ; i++){
            //send packet to neighbor
            edges[t_enqueue].neighbor_vector[i]
            ->packets.push_back(t_packet);
            //if hop count is zero, then exit from for loop
            if(t_packet.hop_count == 0){ break; }

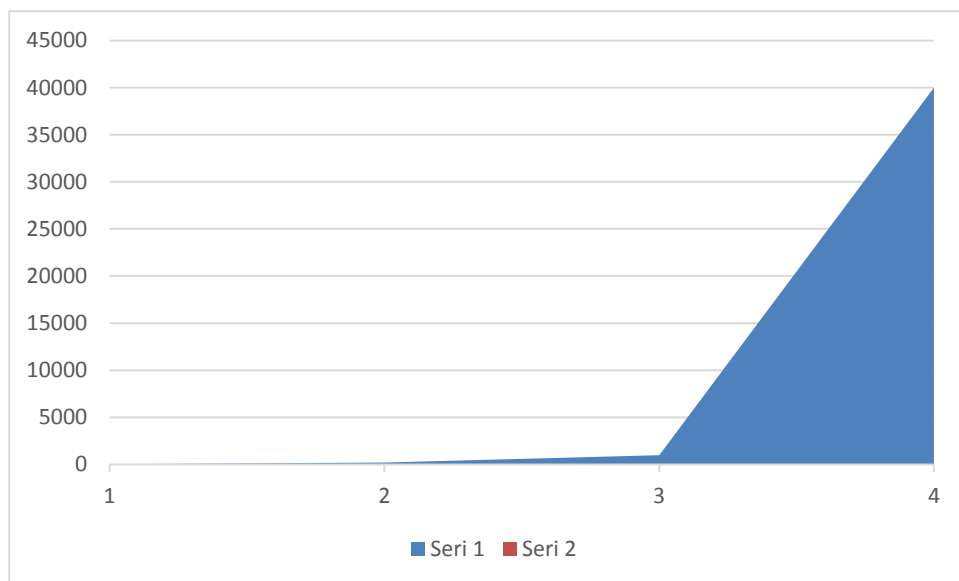
            //push this edge to queue
            if(is_in_vector(t_vector1,edges[t_enqueue].neighbor_vector
            [i]->id) == false ){
                t_queue.push(
                edges[t_enqueue].neighbor_vector[i]->id );
                t_vector1.push_back(
                edges[t_enqueue].neighbor_vector[i]->id );
            }
        }
        //decrease hop count of this packet
        t_packet.decrease_hop_count();
    }
    //if hop count is zero, then exit from while loop
    if(t_packet.hop_count == 0){ break; }
}
```

2 Question 2

BLG433E HW2 REPORT



3 Question 3



4 Question 4

As seen in the charts, if we have a graph that has a small number of edges, then flooding can be used. But when the number of edges increased, flooding is harmful for our network. Also, hop count has to be selected carefully. Because if we select hop count too low, then our packet may not go to destination.