1. **Group members**

Group members: Ziqi Liu 9364-4917

Enyang Wang 6494-1124

1. **How to run the code**

Steps: 1. Enter root folder (project2).

2. using commend: **mix escript.build**

3. type commend as: **escript project2 *num\_node topology algorithm***

1. **What is working**

Determine the convergence of gossip and push\_sum through different topologies(full, line, rand2D, 3Dtorus, honeycomb and randhoneycomb) based on actors written in Elixir.

1. **Largest network**

We test the gossip network from 1000 to 3000 as interval of 500 and push\_sum network from 100 to 500 as interval of 100.

1. **Time measuring**

Using :*timer.tc* in erlang to measure the time of our project. Measure the time from the first node begin spreading to the algorithm converge. Ignore the time of building graphs (finding neighbors).

1. **Result**
2. **Observation**

**Gossip**:

**Full**: time cost to converge increase linearly when number of nodes increased.

**Line**: same as full, but slowest topology when running gossip.

**Rand2D**: faster than two topologies above. Time cost to converge is not influenced much by the number of nodes. However, need more time to build the graph (find neighbors).

**3Dtorus**: time cost increase slightly when number of nodes increased. Faster than full, line and honeycomb.

**Honeycomb**: time cost increase linearly when number of nodes increased. Faster than line topology and slower than full topology.

**Randhoneycomb**: fastest topology. Time cost is not influenced much by number of nodes

**Push\_sum:**

**Full**: time cost to converge increase linearly when number of nodes increased.

**Line**: time cost to converge increase linearly when number of nodes increased. Slower than full.

**Rand2D**: time cost to converge increase rapidly when number of nodes change from 100 to 200, then increase slightly when number of nodes increase after 200.

**3Dtorus**: Time cost to converge increase to peak at 400 nodes, then drop down.

**Honeycomb**: faster than line topology before 300 nodes, then become the slowest topology to converge.

**Randhoneycomb:** fastest topology to converge. Time cost change slightly when number of nodes increase.