Communication Network Design Lab – Task Submission (Task 1*)

Full name	Person code
Mohammadreza Zamani	10869960
Karim Al Wazze	11041528

1. **Report for task submission to CND LAB**

A. Describe the main objective of the task

The main objective of Task 1 is to carry out and analyze two different network traffic routing methods (splittable & unsplittable) flow formulations in order to comprehend their impact on the network performance.

1-<u>Implementing them on the flow formulations:</u>

- **Splittable Flow allows the traffic to be spread across multiple paths and optimize the capacity usage while Unsplittable flow limits the demand to single path which will increase the capacities on individual links.**

2-<u>Capacity Analysis:</u>

a. **Run simulations for both Splittable and Unsplittable with capacities C = {10, 15, 17, 20, 30, 40, 50} to observe how they use the resources of the network.**

3-<u>Performance Comparison:</u>

Compare the results to know where the splittable flow achieves better load flexibility and balance, on the other side unsplittable flow may require higher link capacities due to routing constraints.

B. Describe the main changes in the code, and report the code modifications (it can also be a screenshot)

Add Library

```
import java.util.HashSet; // Add library
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import java.util.Set; // Add library
```

Set LB and UB for X_lkc

```
// Set the IsInteger to true, Lower_Band = 0, Upper_Band = 1, to make it binary according to the formulation
op.addDecisionVariable("x_lkc" , true , new int [] {D , E } , 0 , 1);
```

Define V_c from Net2lan

```
// Define V_c from Net2Plan implemented function getVectorDemandOfferedTraffic
op.setInputParameter("v_c", netPlan.getVectorDemandOfferedTraffic() , "row");
```

Change objective function

```
// Change the objective function according to the Unsplittable equations
op.setObjectiveFunction("minimize" , "sum (v_c * x_lkc)");
```

Change the Constraint

```
    // Change the constraint. V_c to 1 and -V_c to -1
    op.addConstraint ("sum(x_lkc(c,OutgoingLinks)) - sum(x_lkc(c,IncomingLinks)) == 1");
else if (n == d.getEgressNode())
    op.addConstraint ("sum(x_lkc(c,OutgoingLinks)) - sum(x_lkc(c,IncomingLinks)) == -1");
else
    op.addConstraint ("sum(x_lkc(c,OutgoingLinks)) - sum(x_lkc(c,IncomingLinks)) == 0");
```

Change the capacity constraint

```
// Change the capacity constraint by adding V_c to the Sigma
op.addConstraint ("sum(v_c * x_lkc(all, lk)) <= linkCapacity");
```

C. Report the main results. For example, in a table as the one shown below

Link Capacity [Gbps]	Splittable	Unsplittable
10	NA (An optimal solution was not found)	No Result
15	NA	No Result
17	200	No Result
20	196	233
30	196	196
40	196	196
50	196	196

D. Comment the main takeaways from your results. What is the main message you get from the numerical results? Is it what you expected? If yes, why, if not, why? Motivate your answer

The results showed that splittable flows allow the use of the network capacity to be more efficient by distributing the traffic on multiple paths, this helps in avoiding bottlenecks in addition to that it allows the network to carry more traffic.

In contrast the Unsplittable flow limit the demand to a single path which create bottlenecks earlier because the traffic cannot spread out, which requires a much higher capacities on specific links to be able to handle the same demand

The main takeaway of this results is the importance of routing flexibility because the Unsplittable flow even though it is more simple it may still require additional capacity on certain links due to their rigid routing constraint, while splittable flows use balancing loads to optimize the capacity use.

It was expected because the splittable flows is more flexible in adapting to network conditions and spread traffic which result in reducing the likelihood of congestions. By design, Unsplittable flows restrict the traffic into one path which will result in making certain links more prone to reach the capacity limit quicker.