Optical_Network_addSecondaryDemands.java

```java
/* Copyright (c) 2013-2014 Pablo Pavon-Marino, Jose-Luis Izquierdo-Zaragoza.

import com.net2plan.interfaces.networkDesign.IAlgorithm;
/**
 * @author Adolfo Oliveira
 * @version 1.0, June 2015
 */
public class Optical_Network_addSecondaryDemands implements IAlgorithm
{
    @Override
    public String executeAlgorithm(NetPlan netPlan, Map<String, String> algorithmParameters, Map<String, String> net2planPa
    {
        /* Initialize some variables */
        int N = netPlan.getNumberOfNodes();
        int E = netPlan.getNumberOfLinks();
        int D = netPlan.getNumberOfDemands();
        if (N == 0 || E == 0 || D == 0) throw new Net2PlanException("This algorithm requires a topology and a demand set");
        Set<Long> demandIds = netPlan.getDemandIds();
        int [] demands = new int[D];
        int i = 0;
        int lastDemand = 0;
        for(long demandId : demandIds)
        {
            demands[i] = (int) demandId;
            lastDemand = (int) demandId;
            i++;
        }
        double offeredTrafficInErlangs = Double.parseDouble(algorithmParameters.get("TrafficInErlangs"));
        i=0;
        for(int d=lastDemand ; d<((N*(N-1))+lastDemand) ; d++)
        {
            long a_d = netPlan.getDemandIngressNode(demands[i]);
            long b_d = netPlan.getDemandEgressNode(demands[i]);
            netPlan.addDemand(a_d, b_d, offeredTrafficInErlangs, null);
            i++;
        }
        return "Ok!";
    }
    @Override
    public List<Triple<String, String, String>> getParameters()
    {
        List<Triple<String, String, String>> parameters = new ArrayList<Triple<String, String, String>>();
        parameters.add(Triple.of("TrafficInErlangs", "2.5", "Traffic in Erlangs for each demand"));
```