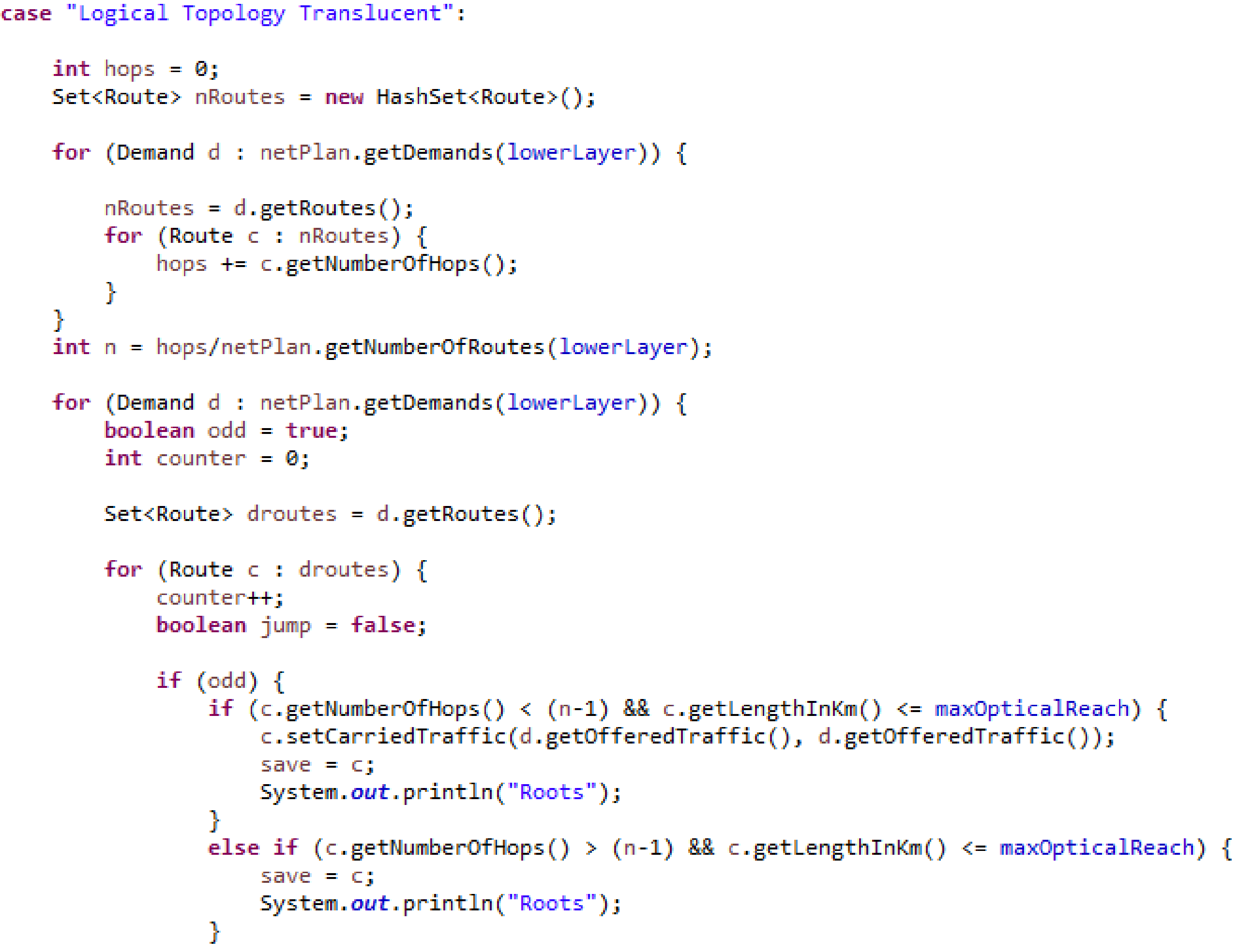

```
case "Logical Topology Translucent":

    int hops = 0;
    Set<Route> nRoutes = new HashSet<Route>();

    for (Demand d : netPlan.getDemands(lowerLayer)) {

        nRoutes = d.getRoutes();
        for (Route c : nRoutes) {
            hops += c.getNumberOfHops();
        }
    }
    int n = hops/netPlan.getNumberOfRoutes(lowerLayer);

    for (Demand d : netPlan.getDemands(lowerLayer)) {
        boolean odd = true;
        int counter = 0;

        Set<Route> droutes = d.getRoutes();

        for (Route c : droutes) {
            counter++;
            boolean jump = false;

            if (odd) {
                if (c.getNumberOfHops() < (n-1) && c.getLengthInKm() <= maxOpticalReach) {
                    c.setCarriedTraffic(d.getOfferedTraffic(), d.getOfferedTraffic());
                    save = c;
                    System.out.println("Roots");
                }
                else if (c.getNumberOfHops() > (n-1) && c.getLengthInKm() <= maxOpticalReach) {
                    save = c;
                    System.out.println("Roots");
                }
```