Homework 1

Routing with adaptive link cost assignment

網工所 0656519 胡憲文

1. Overview

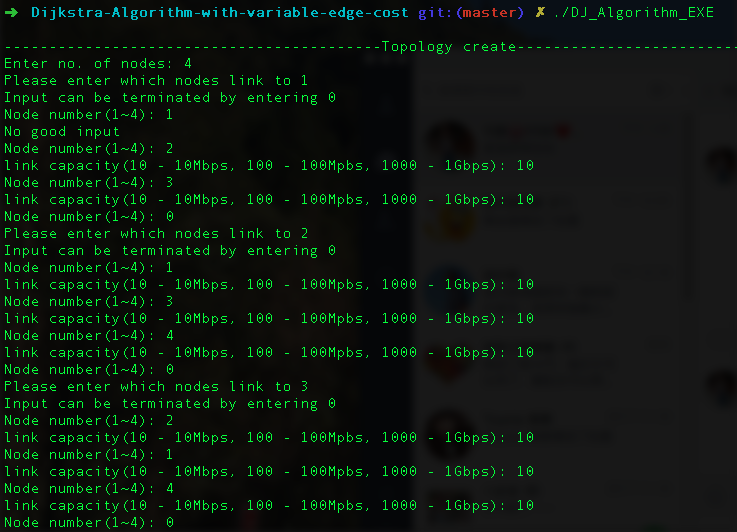
這是一個基於Dijkstra 路由演算法開發的簡易模擬程式。主要的功能有：

1. Topology creation: 可以依照使用者需求產生topology，支援的輸入有：
   * 1. Neighbors
     2. Link capacities: 也就是link的速度，此參數會被轉化成為Link cost
2. Route calculation: 可以依照使用者需求建立最佳路徑（基於Dijkstra method），同時會記錄之前輸入的路由資訊，當每一次有新的路由被輸入進程式，程式會重新計算路由，並顯示給使用者。
3. Satisfaction index: 計算有多少比例的Route 有符合原先的需求。

本文件會逐一說明以上所提到的相關細節

1. Topology creation

Topology creation 的介面為：



程式內部儲存拓墣的方式則是採用ajacent matrix。在圖中可以看到程式會要求使用者輸入link capacity並且以Mbps為單位。當使用者輸入拓墣後，會將這些link capacity 轉化成為link cost，轉換的方式如下：

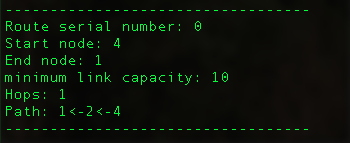
1. Route calculation

在前面有提到，本程式是採用Dijkstra 作為路由演算法，演算法細節不再次贅述，讀者可以參閱網路上或是書本資料。

本程式的路由輸入介面如下：



使用者可以輸入起始結點與目標節點，並且輸入希望的目標速度（unit: Mbps）。輸入完成後，程式為自動運算最佳路徑並且顯示出運算結果：

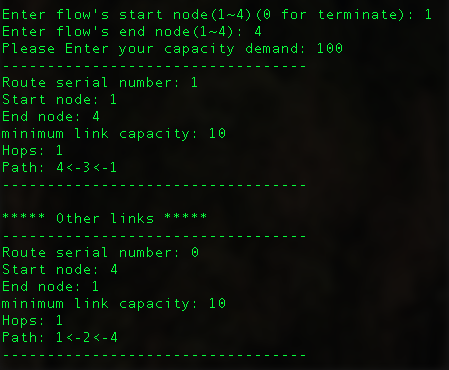


由上到下為：

* + 1. Route serial number: 第幾號routing information
    2. Start node: 起始節點
    3. End node: 目標結束節點
    4. Minimum link capacity: 路徑上的最小link capacity
    5. Hops: 共經過幾個節點（頭尾不算）
    6. Path: 路徑

每輸入一次新的路徑，程式會自動根據先前路徑所產生的link loading來重新計算最新的可能最佳路徑，而link loading 計算公式如下：

則重新計算的結果如下：

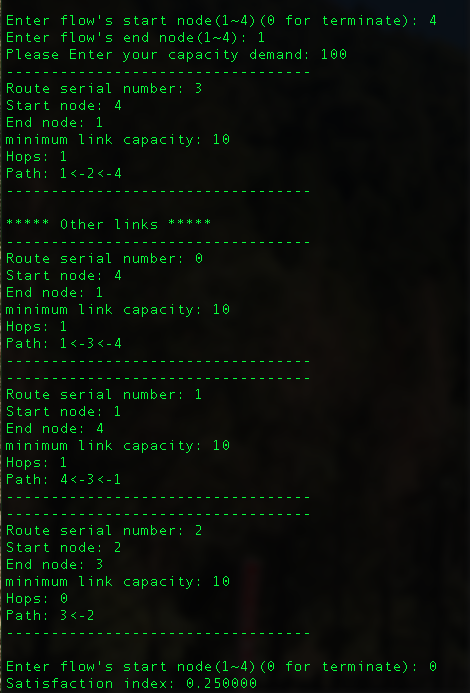


重新計算後的路徑

1. Satisfaction Index

整個程式運算結束後，會進行總檢討，計算出satisfaction index，數值是0至1，倘若找出的路徑有符合起初要求的link capacity，則算是符合需求，反之則否。

計算後的結果如下圖：



1. Program build and execution

可以在github 上下載到本程式，URL: <https://github.com/tomyhu1995/Dijkstra-Algorithm-with-variable-edge-cost>，詳細的build與運行方式在README.md有詳細的解說，程式中也有完整的導引。

1. Reference

<http://www.netadmin.com.tw/article_content.aspx?sn=1304170001&jump=3>

<http://blog.xuite.net/tolarku/blog/37809537-%5BCCNA%5D+Cisco+Router+%E7%9A%84+OSPF+%E8%B7%AF%E5%BE%91%E8%A8%AD%E5%AE%9A+%28Open+Shortest+Path+First%29>

<http://www.geeksforgeeks.org/greedy-algorithms-set-6-dijkstras-shortest-path-algorithm/>

<http://www.codewithc.com/dijkstras-algorithm-in-c/>