

算法题技术报告

目录

- 算法题技术报告..... 1
- 一、 题目分析..... 2
- 二、 设计思路..... 2
 - 2.1 输出文件的生成..... 2
 - 2.2 算法的选择..... 3
 - 2.3 算法的设计..... 3
 - 2.3.1 生成邻接矩阵..... 3
 - 2.3.2 路径寻找..... 4
 - 2.3.3 路由的输出..... 5
- 三、 控制器的 APP 实现..... 6
 - 3.1 远程登录 SDN 控制器..... 6
 - 3.2 建立拓扑结构..... 6
 - 3.3 导入算法..... 7
 - 3.4、建立拓扑结构..... 8
 - 3.5、算法的运行..... 9
- 四、 实验数据..... 10

一、题目分析

根据题目的要求，需要完成以下工作。

- 1、由网络拓扑结构图和业务需求生成规定格式的 input.txt 作为算法的输入文件；
- 2、编写路由算法，为业务寻找路由并输出要求格式的输出文件（output.txt），该路由算法支持多条业务流同时发起路由请求；
- 3、把编写的路由算法作为一个实际 SDN 平台的 APP(Application,即应用程序)，加入到其控制器的，实现 SDN 网络的路由功能，并得出实验结果。

二、设计思路

根据对题目的分析，具体采用以下设计思路。

2.1 输出文件的生成

输出文件（命名为 input.txt）中包含了整个 SDN 网络的拓扑结构以及业务的路径需求。

其中，input.txt（如图 1 所示）的第 2 行至第 15 行，前两列分别为各个链路的两端节点的地址，最后一列表示节点之间的带宽值。以第 17 行的分号为分割线，之后的第 19 行和第 20 行中，前两列分别为源节点和目标节点的地址，最后一列表示业务需求带宽值。

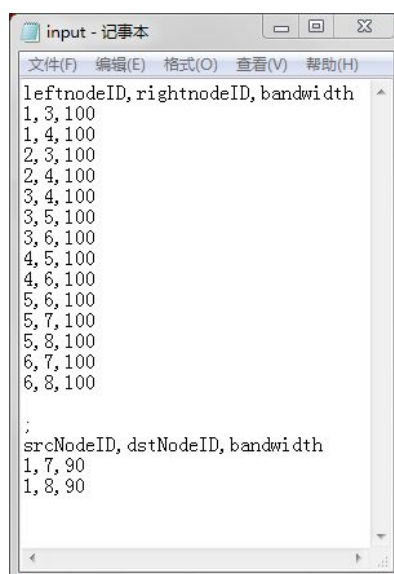


图 1 输入文件 input 的格式

注意事项：在华为平台下，input.txt 的格式是需要严格按照上边的格式的。需要注意的是，在倒数第二行（即“1,8,90”一行）之后，还有一个换行符，否则输入错误。

2.2 算法的选择

对于图 $G=(V, E)$ ，其中 V 为节点的集合， E 为路劲的集合可采用以下表格 1 中的几种算法来实现最优路径的查找。

算法	Dijkstra 算法	Bellman-Ford 算法	SPFA 算法	DAG 图算法
优点	时间复杂度为 $O((V+E) \lg V)$	边的权值可以为正数，也可以为负数	最好的情况下，时间复杂度为 $O(E)$ 。在负边权图上可以取代 Bellman-ford 算法	时间复杂度为 $O(V+E)$
缺点	边的权值只能为正数	时间复杂度为 $O(VE)$	最坏的情况下，退化为 Bellman-Ford 算法。	只对有向无回路图计算单源最优路径

表格 1 几种算法的比较

由对题目的分析可知我们所建立的拓扑图为双向图，且所有的权值均为正值，经过对以上几种算法的对比，最终决定采用 Dijkstra 算法来完成题目的要求。

然而，简单地使用 Dijkstra 算法并不能实现多个业务同时的路径请求。因此，可以把多个业务的路径请求单独分开，每个业务的路径请求采用 Dijkstra 算法找出最优路径后进行拓扑结构数据的更新，再处理下一个业务的路径请求。

2.3 算法的设计

拟采用 Dijkstra 最短路径的方法寻找路由，寻找满足业务需求的需要消耗带宽资源最少的路径。使用该方法需要先建立网络拓扑的图论模型，本设计用邻接矩阵表示网络拓扑图。

2.3.1 生成邻接矩阵

邻接矩阵如图 2 所示。其中， i 为左节点的地址， j 为右节点的地址， a_{ij} 为节点 i 到节点 j 之间路径的权值，即带宽。

$$A_{IJ} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,j-1} & a_{1,j} \\ a_{2,1} & a_{2,2} & \dots & a_{2,j-1} & a_{2,j} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{i-1,1} & a_{i-1,2} & \dots & a_{i-1,j-1} & a_{i-1,j} \\ a_{i,1} & a_{i,2} & \dots & a_{i,j-1} & a_{i,j} \end{bmatrix}$$

图 2 网络拓扑的邻接矩阵

2.3.2 路径寻找

Dijkstra 算法是已知网络的拓扑结构和各链路的权值，寻找从源节点到网络中其他各节点的路径权值之和最小的路径。本题目需要修改 Dijkstra 算法使其满足多路径的带宽需求。

多路径带宽路由算法思想

初始化拓扑图，在输入文件里读入拓扑图的节点和路径权值，建立拓扑图。继续从输入文档里读入源节点和目标节点以及消耗值，采用 Dijkstra 算法找出源节点和目标节点的最短路径。刷新拓扑结构中路径的权值，再次读入新的路径查找请求。重复以上操作便可实现多路径的查找需求。

多路径带宽算法执行步骤

设 maxnum 为图 $G=(V,E)$ 中的顶点数， $\text{dist}[\text{maxnum}]$ 存放从源点到每个终点当前最短路径的长度， $\text{mprev}[\text{maxnum}]$ 存放相应路径。

(1) 获取源节点（假设为节点 v ），初始化与节点 v 相邻的节点， $\text{mprev}[i]$ 记录下第 i 个节点与节点 v 之间的路径关系，节点 v 标记为被访问。

(2) 比较节点 v 到所有未被访问过的节点，找出与节点 v 距离最短的节点（假设为节点 u ），节点 u 标记为被访问。

(3) 对于与 u 相邻的且未被访问过的节点（假设为节点 j ），比较源节点 v 到节点 j 的距离（ $\text{dist}[j]$ ）与节点 v 经过节点 u 再到节点 j 的距离（ $\text{dist}[u] + c[u][j]$ ）之间的大小，取 $\text{dist}[j]$ 和 $\text{dist}[u] + c[u][j]$ 中的最小值作为源节点 v 到节点 j 的最短距离， $\text{mprev}[j]$ 记录下节点 j 与节点 v 之间的路径关系。

(4) 重复步骤 (2) 和 (3)，便可找出源节点 v 到拓扑图 G 中其他所有节点最短距离的路径。

(5) 将拓扑图中路径的权值减去数据包的消耗值得到路径的新权值，重新确定新的源节点，重复以上步骤 (1) - (4) 的内容即可实现多个业务的路径请求。

2.3.3 路由的输出

在多个业务同时路径请求的情况下，按顺序一个一个地寻找最优路径并通过文件输出到 `output.txt`。在 `output.txt`（如图 3 所示）中每一行代表一个业务的路径请求的最优路径所经过的所有节点。

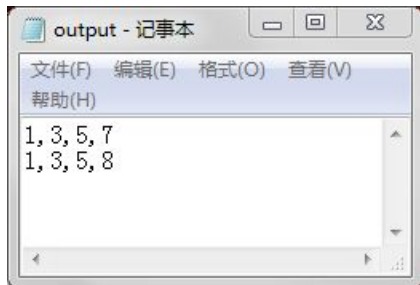


图 3 算法输出文件的格式

根据以上的设计思路，本报告的路由算法的系统设计图可以用图 4 的流程图来总结。算法的详细源代码见附录 A 及详细的代码分析见附录 B。

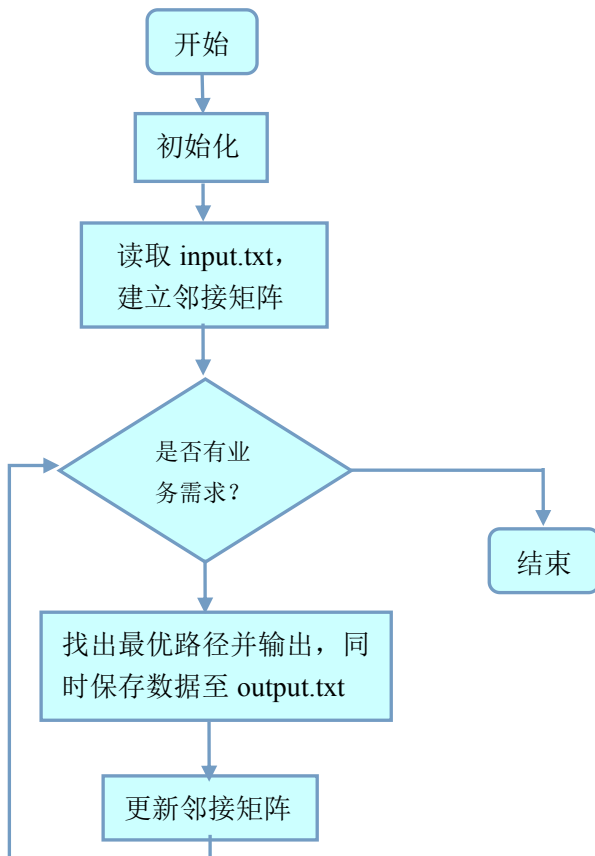


图 4 算法流程图

三、控制器的 APP 实现

我们采用华为的 SDN 平台实现本报告的路由算法。路由算法的 APP 安装如下。

3.1 远程登录 SDN 控制器

首先预约测试平台，获得密码。在本机上点击“开始”，输入“远程桌面连接”，连接到计算机“hw.sdn.scut.edu.cn:9000”上，用户名为“sdn”，点击“连接”，在弹出的窗口中输入密码。进入服务器的桌面。

注意事项：登陆远程桌面时要更改用户名时需点击“选项”下拉菜单(如图 5 所示)。



图 5 远程登录界面

3.2 建立拓扑结构

新建拓扑结构：进入测试平台,在下拉菜单下的“General”中，点击“topo 图库管理”->“新建 topo 图”，根据弹出的窗口填写对应的信息。（如图 6 所示）添加之后要选择建立的拓扑图。

注意事项：新建拓扑结构时选择的方案为“方案一”。

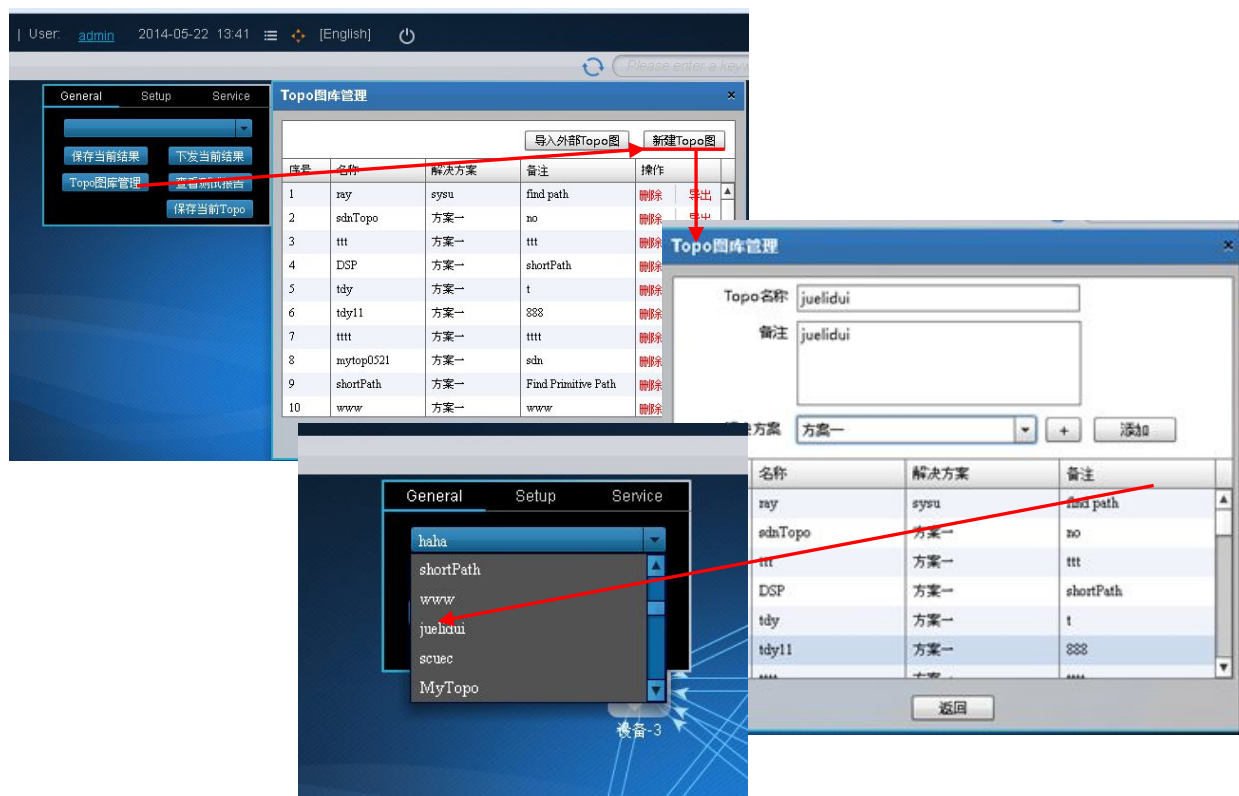


图 6 新建拓扑图界面

3.3 导入算法

在下拉菜单的“Service”中，点击“注册新算法，”窗口里填上对应的信息。注册成功后点击“变更”，选择注册的算法。（如图 7 所示）

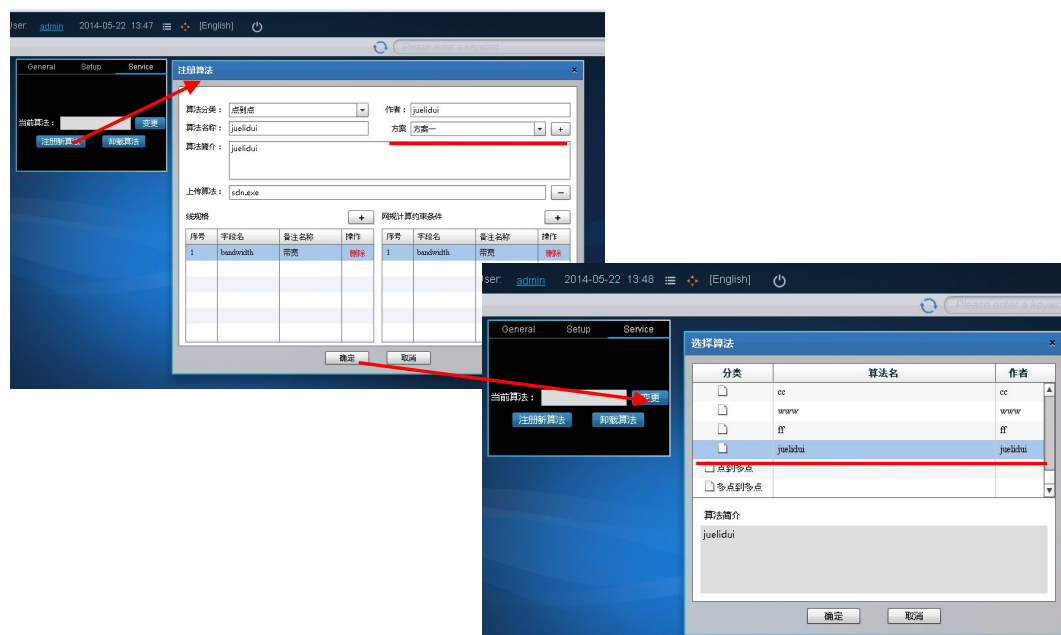


图 7 注册新算法界面

注意事项:

①注册算法选择的方案与新建拓扑时一样，这里为“方案一”。

②将（文件名）.exe 文件从主机传到服务器时，可以直接复制过去，同时“input.txt”文件要复制到服务器里。点击“注册”，显示“注册成功”即可。

③在变更算法时应先选择建立的拓扑结构，否则没有算法可选。

④华为提供的服务器装过 VS2010，而 VS2010 与 VS2013 版本有差别，所以用 VS2013 生成的 APP 可能会导致最后不能运行。此时可以改用 VS2010 生成 APP 再导入平台上测试。

3.4、建立拓扑结构

(1)增加点：在空白处右击鼠标，点击“增加节点”，根据弹出的窗口填写对应的信息。

注意事项：节点类型为 CE8500。

(2)增加线：在空白处右击鼠标，点击“增加线”，根据弹出的窗口填写对应的信息。

注意事项：增加线时虽然第一次输入带宽后会一直显示带宽值，但每一条路径都要输入一次带宽，不然后输入的路径没有带宽。

(3)网规计算：在空白处右击鼠标，点击“网规计算”，根据弹出的窗口填写对应的信息。

(如图 8 所示)

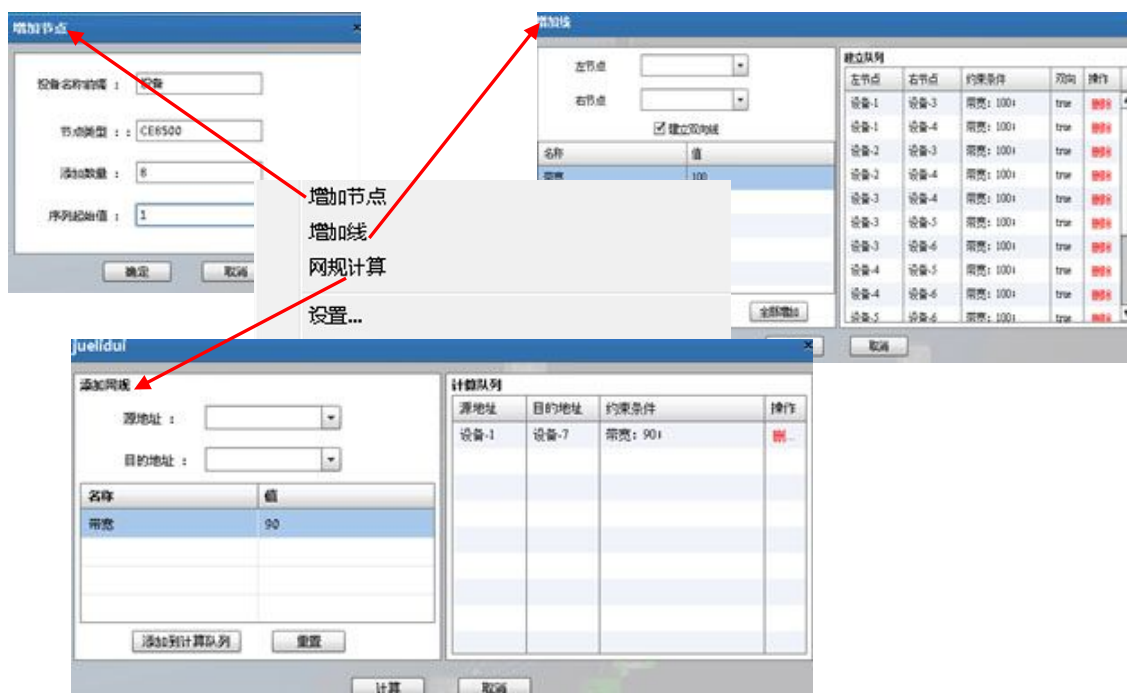


图 8 建立拓扑结构视图界面

3.5、算法的运行

以上操作后，最终点击“计算”，即可得到结果。(如图 9 所示)

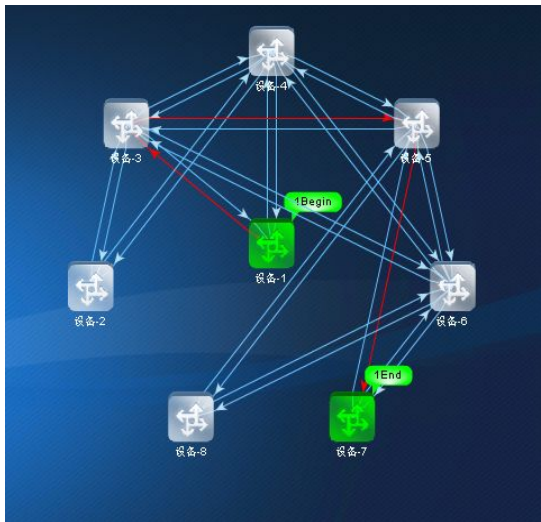


图 9 路径显示界面（1）

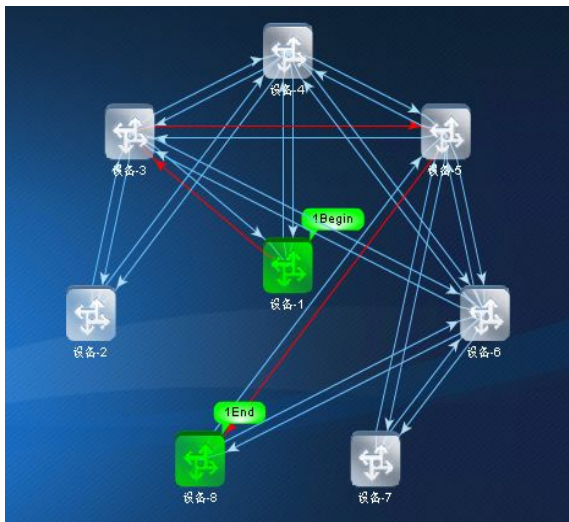


图 10 路径显示界面（2）

若图 8 中源节点为 1，目标节点为 8，则结果如图 10 所示。

四、实验数据

以下表格 2 是寻找源节点为 1，目标节点为 8 时的数据。可以看出总共有 8 个节点，28 条线（由于节点之间是双向导通的，所以有 14 对节点之间是有路径的）。运行结果是 1->3->5->8，与图 10 显示的路径一样，结果正确。若同时有多个业务发出求路请求，也能完成路径的查找工作（若源节点为 1，目标节点为 7 时，运行结果是 1->3->5->7）。而且每次寻找路径所用的时间仅需 0.004 秒，运算速度有很大的提高。

Topo节点数量：8个

设备ID	设备类型	设备名称
1	CE8500	设备-1
2	CE8500	设备-2
3	CE8500	设备-3
4	CE8500	设备-4
5	CE8500	设备-5
6	CE8500	设备-6
7	CE8500	设备-7
8	CE8500	设备-8

Link线数量：28条

左节点ID	右节点ID	开销	带宽	peerID	delay	左节点ID	右节点ID	开销	带宽	peerID	delay	左节点ID	右节点ID	开销	带宽	peerID	delay
设备-1	设备-3	空	100	空	空	设备-4	设备-3	空	100	空	空	设备-5	设备-6	空	100	空	空
设备-3	设备-1	空	100	空	空	设备-3	设备-5	空	100	空	空	设备-6	设备-5	空	100	空	空
设备-1	设备-4	空	100	空	空	设备-5	设备-3	空	100	空	空	设备-5	设备-7	空	100	空	空
设备-4	设备-1	空	100	空	空	设备-3	设备-6	空	100	空	空	设备-7	设备-5	空	100	空	空
设备-2	设备-3	空	100	空	空	设备-6	设备-3	空	100	空	空	设备-5	设备-8	空	100	空	空
设备-3	设备-2	空	100	空	空	设备-4	设备-5	空	100	空	空	设备-8	设备-5	空	100	空	空
设备-2	设备-4	空	100	空	空	设备-5	设备-4	空	100	空	空	设备-6	设备-7	空	100	空	空
设备-4	设备-2	空	100	空	空	设备-4	设备-6	空	100	空	空	设备-7	设备-6	空	100	空	空
设备-3	设备-4	空	100	空	空	设备-6	设备-4	空	100	空	空	设备-6	设备-8	空	100	空	空
左节点ID	右节点ID	开销	带宽	peerID	delay												
设备-8	设备-6	空	100	空	空												

运算线路：1条

始节点ID	末节点ID	带宽	delay	level
设备-1	设备-8	90	空	空

运算线路：1条

始节点ID	末节点ID	带宽	delay	level
设备-1	设备-8	90	空	空

运算时间：0.004秒

运算结果：

线路计算结果
设备-1 -->> 设备-3 -->> 设备-5 -->> 设备-8

表格 2 运行结果显示表