

实验预习报告

专业班级 计科 1804 班 实验地点 SY309 实验日期 Nov.14th.2020

学生姓名 王子龙 学号 18281218 同组者姓名 周天宸 田震 武斯全 万奕晨

实验项目名称 Routing Algorithms Programming

一、实验目的和要求

理解路由算法协议，并通过编程实现不同路由算法，对不同的路由算法能有进一步的理解和掌握。

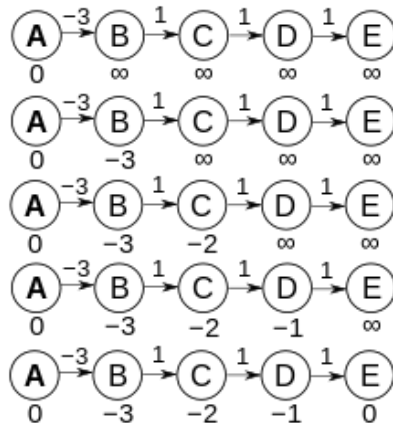
二、实验原理

动态路由

若某个设置好的路径无法使用时，现存的节点必须决定另一个发送资料到目的地的路径。他们通常使用以下两种形式的路由协议来达成：距离向量算法与连线状态算法。所有路由算法几乎都可以分类到这两种算法中。

距离向量算法

距离向量算法使用 Bellman-Ford 算法。对于每一条网络上节点间的路径，算法指定一个“成本”给它们。节点会选择一条总成本（经过路径的所有成本总和）最低的路径，用来把资料从节点甲送到节点乙。



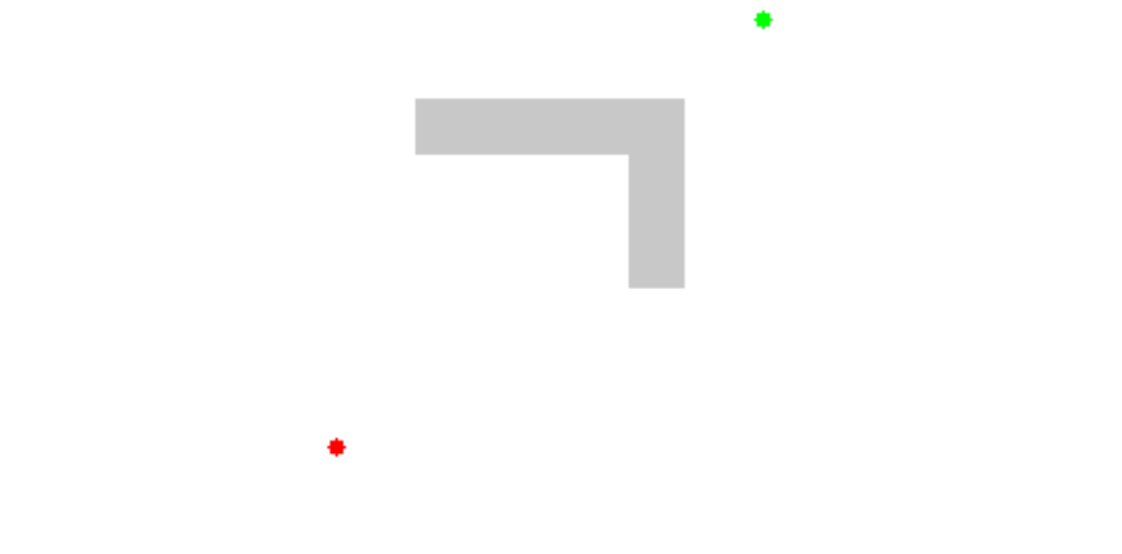
此算法非常的简单。当某节点初次引导时，将只知道它的邻居节点（直接连接到该节点的节点）与到该节点的成本。（这些信息、目的地列表、每个目的地的总成本，以及到某个目的地所必须经过的“下一个节点”，构成路由表，或称距离表。）每个节点定时地将目前所知，到各个目的地的成本的信息，送给每个邻居节点。邻居节点则检查这些信息，并跟目前所知的信息做比较；如果到某个目的地的成本比目前所知的低，则将收到的信息加入自己的路由表。经过一段时间后，网络上的所有节点将会了解到所有目的地的最佳“下一个节点”与最低的总成本。

当某个节点断线时，每个将它当作某条路径的“下一个节点”的节点会将该路由信息舍弃，再创建新的路由表信息。接着，他们将这些信息告诉所有相邻的节点，再找出到所有可抵达的目的地之新路径。

连线状态算法

在连线状态算法中，每个节点拥有网络的图谱（一个图）。每个节点将自己可以连接到的其他节点信息发送到网络上所有的节点，而其他节点接着各自将这个信息加入到图谱中。每个路由器即可根据这个图谱来决定从自己到其它节点的最佳路径。

完成这个动作的算法——Dijkstra 算法——创建另一种数据结构——树。节点产生的树将自己视为根节点，且最后这棵树将会包含了网络中所有其他的节点。一开始，此树只有根节点（节点自己）。接着在树中已有的节点的邻居节点且不存在树中的节点集合中，选取一个成本最低的节点加入此树，直到所有节点都存入树中为止。



这棵树即用来创建路由表、提供最佳的“下一个节点”等，让节点能跟网络中其它节点通信。

路由算法的比较

在小型网络中，距离向量路由协议十分简单且有效率，且只需要一些管理。然而，它们的规模性不好，且收敛性质也十分差，因此促进了较复杂但规模性较好的连线状态路由协议的开发，以使用在较大型的网络。距离向量路由协议也有无限计数问题（见[1] 页面存档备份，存于互联网档案馆（英文））。

连线状态路由协议的主要优点是在限制的时间内，对于连线改变（例如断线）的反应较快。而且连线状态路由协议在网络上所发送的数据包也比距离向量路由协议的数据包小。距离向量路由协议必须发送一个节点的整个路由表，但连线状态路由协议的数据包只需要传输该节点的邻居节点信息即可。因此，这些数据包小到不会占用可观的网络资源。连线状态路由协议的主要缺点则是比距离向量路由协议需要较多的存储空间与较强的计算能力。

三、实验方法与步骤

Bellman-Ford 算法实现的伪代码：

```
procedure BellmanFord(list vertices, list edges, vertex
```

```

source)

// 读入边和节点的列表并对 distance 和 predecessor 写入最短路径

// 初始化图
for each vertex v in vertices:
    if v is source then distance[v] := 0
    else distance[v] := infinity
    predecessor[v] := null

// 对每一条边重复操作
for i from 1 to size(vertices)-1:
    for each edge (u, v) with weight w in edges:
        if distance[u] + w < distance[v]:
            distance[v] := distance[u] + w
            predecessor[v] := u

// 检查是否有负权重的回路
for each edge (u, v) with weight w in edges:
    if distance[u] + w < distance[v]:
        error "图包含负权重的回路"

```

Dijkstra 算法伪代码:

```

1 procedure Dijkstra(G: 边全为正权的图)
2   {G带有顶点  $a = v_0, v_1, v_2 \dots$  和若干边  $w(v_i, v_j)$ }
3   for  $i := 1$  to  $n$ 
4      $D(v_i) := \infty$ 
5    $D(a) := 0$ 
6    $S := \emptyset$ 
7   while  $z \notin S$ 
8     begin
9        $u :=$  不属于  $S$  的  $D(u)$  最小的一个顶点
10       $S := S \cup \{u\}$ 
11      for 所有不属于  $S$  的顶点  $v$ 
12        if  $D(u) + w(u, v) < D(v)$  then  $D(v) := D(u) + w(u, v)$ 
13    end{ $D(z) =$  从  $a$  到  $z$  的最短路长度}

```

