



路由算法实验报告

路由选择协议算法

2021 年 6 月 11 日

学院：智能与计算学部

班级：网络安全一班

姓名：谢远峰

学号：3019244283

一、算法描述

1 算法原理

距离向量算法是迭代的、异步的和分布式的。

1. 分布式: 每个节点从其直接连接的一个或多个邻居接收信息, 执行计算然后将结果分发回其邻居。
2. 迭代: 过程一直持续到没有更多的信息可以在邻居之间交换
3. 异步: 不要求其所有节点彼此都在限定步骤中操作。

“距离矢量路由算法”的基本思想如下: 每个路由器维护一个距离矢量(通常是以延时是作变量的)表, 然后通过相邻路由器之间的距离矢量通告进行距离矢量表的更新。每个距离矢量表项包括两部分: 到达目的节点的最佳输出线路, 和到达目的节点所需时间或距离, 通信子网中的其它每个路由器在表中占据一个表项, 并作为该表项的索引。每隔一段时间, 路由器会向所有邻居结点发送它到每个目的结点的距离表, 同时它也接收每个邻居结点发来的距离表。这样以此类推, 经过一段时间后便可将网络中各路由器所获得的距离矢量信息在各路由器上统一起来, 这样各路由器只需要查看这个距离矢量表就可以为不同来源分组找到一条最佳的路由。

2 算法过程

让 $d_x(y)$ 是从节点 x 到节点 y 的最小成本路径的成本。最低成本与 Bellman-Ford 方程相关

$$d_x(y) = \min_v \{c(x, v) + d_v(y)\}$$

其中 \min_v 是对所有 x 邻居采用的方程。从 x 到 v 后, 如果我们考虑从 v 到 y 的最小成本路径, 路径成本将为 $c(x, v) + d_v(y)$ 。从 x 到 y 的最小成本是 $c(x, v) + d_v(y)$ 对所有邻居的最小值。

使用距离矢量路由算法, 节点 x 包含以下路由信息:

- 对于每个邻居 v , 成本 $c(x, v)$ 是从 x 到直接连接的邻居 v 的路径成本。
- 距离向量 x , 即 $D_x = [D_x(y) : y \in N]$, 包含它到所有目的地 y 的成本, 在 N 中。
- 每个邻居的距离向量, 即对于 x 的每个邻居 v , $D_v = [D_v(y) : y \in N]$ 。

距离向量路由是一种异步算法, 其中节点 x 将其距离向量的副本发送给其所有邻居。当节点 x 接收到来自其相邻向量之一的新距离向量时, 它保存 v 的距离向量并使用 Bellman-Ford 方程更新自己的距离向量。方程如下:

$$d_x(y) = \min_v c(x, v) + d_v(y) \quad \text{for each node } y \text{ in } N$$

节点 x 使用上式更新自身距离向量表, 并将更新后的表发送给所有邻居, 以便他们更新自己的距离向量。

二、算法实现

1 实现规划

- 初始化距离向量表
发送包结点将到直接邻居的距离赋值到表中, 非直接邻居将距离设置为最大值
- 结点的最小距离选择和更新
获取发送结路由点到终点路由的最小距离数组并存储
- 初始化发送路由待发送的包
确认包的发送路由, 终点路由, 装载路由间的最小距离
- 发送三个包(不含发送自身路由的包)
发送包至链路端, 并进行信息打印
- 路由接收包后的处理
路由获取包后, 进行发送 ID 及终点识别, 根据邻居距离信息对自身的距离向量表进行更新, 并重新发送至其它结点

2 实现细节

- 初始化
发送路由到其它路由的直接距离数组、时间记录、距离矢量表、存储最小距离的数组、初始化包
- 最小距离查找
将每组的距离通过比较获取最小距离值
- 发送距离矢量信息
定义发送路由，终点路由，并将最小距离数组进行赋值，将包信息发送至所有直接邻居
- 发送包信息核查
将初始化距离表和更新后的距离表进行比较，如果有改变，将更新后距离表进行广播
- 路由初始化
将直接邻居距离加载进距离矢量表中，未知距离进行无穷大距离赋值
- 路由更新
获取发送路由 ID，终点路由，更新间接邻居的距离
- 打印最小距离
- 部分距离更新逆向获取原间接邻居距离，再与更新后的距离进行加和，重新获取最新的距离矢量

三、问题复现

1. 算法规划

Algorithm 1 Routing

```
1: Initialization: Min_Cost_Node, Node_Cost_Array, time, pkt_node
2: Min_Cost_Node = Distance of SourceID TO DestinationID
3: Node_Cost_Array = Two dimension Array of SourceID TO DestinationID VIA MidSide
   Node
4: pkt_node = Package sent to DestinationID
5: Min_Cost_Node = (For  $i \leftarrow 0$  to 3 in each row)  $\text{Min}(\text{Node\_Cost\_Array}[i])$ 
6: Send the Package to the direct neighbour
7: repeat
8:   Get the SourceID && DestinationID && Min_Cost_Node
9:   Update Distance Indirect neighbour VIA the Min_Cost_Node
10:  if Min_Cost_Node changed then
11:    Broadcast to neighbour nodes
12:  else
13:    Stay silence and Wait next Update
14:  end if
15:  if The Distance to DestinationID changed then
16:    Reverse acquisition of original indirect neighbor distance
17:    Update Distance Indirect neighbour VIA Rriginal Distance
18:  end if
19: until All Min_Cost_Node are not changed
Output: All Min_Cost_Node
```

2. 算法实现

对于输出表内容的理解，左侧为终点路由，上侧为到达终点路由的上一跳结点；间接邻居距离矢量的更新，利用发送路由到达上一跳结点的距离与获得的其它邻居距离矢量信息进行求和，更新自身的距离矢量。

四、结果展示

1 仿真过程

```
PS C:\Users\123456\Desktop\AssignmentCode\Routing\src> .\test.exe
Enter TRACE:1
rtinit0() is called at time t=: 0.000
      via
D0 | 1 2 3
----|-----
1 | 1 999 999
dest 2 | 999 3 999
3 | 999 999 7
At time t=0.000, node 0 sends packet to node 1 with: (0 1 3 7)
At time t=0.000, node 0 sends packet to node 2 with: (0 1 3 7)
At time t=0.000, node 0 sends packet to node 3 with: (0 1 3 7)
rtinit1() is called at time t=: 0.000
      via
D1 | 0 2
----|-----
0 | 1 999
dest 2 | 999 1
3 | 999 999
At time t=0.000, node 1 sends packet to node 0 with: (1 0 1 999)
At time t=0.000, node 1 sends packet to node 2 with: (1 0 1 999)
rtinit2() is called at time t=: 0.000
      via
D2 | 0 1 3
----|-----
0 | 3 999 999
dest 1 | 999 1 999
3 | 999 999 2
At time t=0.000, node 2 sends packet to node 0 with: (3 1 0 2)
At time t=0.000, node 2 sends packet to node 1 with: (3 1 0 2)
At time t=0.000, node 2 sends packet to node 3 with: (3 1 0 2)
rtinit3() is called at time t=: 0.000
      via
D3 | 0 2
----|-----
0 | 7 999
dest 1 | 999 999
2 | 999 2
At time t=0.000, node 3 sends packet to node 0 with: (7 999 2 0)
At time t=0.000, node 3 sends packet to node 2 with: (7 999 2 0)
```

图 4.1: 各节点的距离矢量表初始化

调用指令 `gcc prog3.c node0.c node1.c node2.c node3.c -o test.exe` 生成可执行文件并运行

2 仿真结果

```
Minimum cost didn't change. No new packets are sent
rtupdate2() is called at time t=: 20006.547 as node 3 sent a pkt with (4 3 2 0)
      via
D2 | 0 1 3
----|-----
0 | 3 2 6
dest 1 | 4 1 5
3 | 7 4 2

Minimum cost didn't change. No new packets are sent

Simulator terminated at t=20006.546875, no packets in medium
Minimum cost from 0 to other nodes are: 0 1 2 4
Minimum cost from 1 to other nodes are: 1 0 1 3
Minimum cost from 2 to other nodes are: 2 1 0 2
Minimum cost from 3 to other nodes are: 4 3 2 0
```

图 4.2: 更新迭代后各结点最小矢量数组

五、总结与分析

通过本次实验理解了 Routing 距离矢量表的更新过程，能够通过邻居结点的距离表信息更新自身路由表。网络主机、路由器通过路由选择算法形成路由表，以确定发送分组的传输路径，而路由选择协议是路由器用来完成路由表建立和路由信息更新的通信协议。