

实验名称: Lab6: 网络层路由协议模拟与观察 Part1

实验人: 谢志康

学号: 22307110187

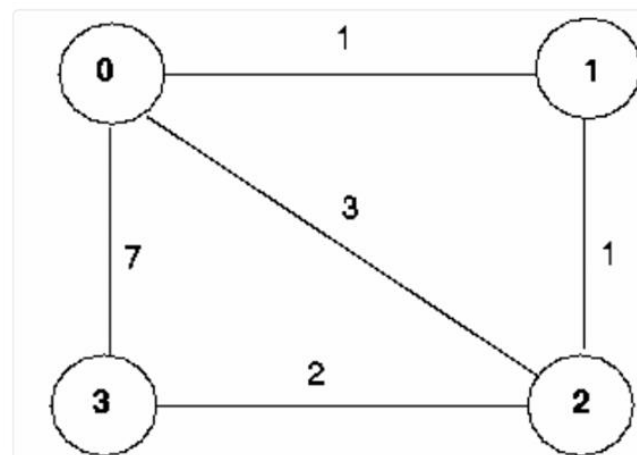
时间: 24.11.29 – 24.11.30 (24.12.9 修改)

最初还是有点环境问题, 解决了很久。具体说来就是前段时间重装了一下 vscode, gcc 版本更到最新了, 到 14 点几版本, 然后 prog3.c 文件中很多诸如: implicit declaration of function 'prindt3'; 或是 main 函数没有类型等等, 全都是 error 而不是 warning, 导致完全没法跑。咨询助教许久, 将 **gcc 降级为 9.2.0 或者 11.4.0** 即可:

```
>> gcc --version
```

```
<< gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
```

代码部分简要解释: 网络环境仿真, prog3.c 中 #define LINKCHANGES 设置为 0, 然后实际跑起来还有个错误, 有个地方 exit 未传入参数, 翻看了实验文档的链接, 原实验的代码就是这样, 学生还挂了 issue, 感觉就是单纯写错了, 没啥问题, 改成 **exit(0);**即可。其余都不要动 (prog3.c) 中也警告了, 不需要改任何地方。



网络拓扑与链路成本

四个 node, 逻辑一样, 简单讲讲 node0.c, 其余三个文件基本上复制粘贴即可。

按照 todo 的指引做即可。Rtinit0() 初始化: 初始化节点 0 中的距离表, 以反映出到达节点 1、2 和 3 的直接开销分别为 1、3 和 7。也就是第 0 行 0 到 j (j from 0 to 3) 设为 0, 1, 3, 7, 表示最初直接相连距离。其余行设为 -1。其次向其直接连接的邻居 (在本情况中为节点 1、2 和 3) 发送它到所有其他网络节点的最低开销路径的开销信息, 创建一个路由选择包, 用于发送最低开销信息给邻居节点, source 设为自身 (0), 目的为可以直接相连、即距离不为 -1 的节点设为 dest, 调用 tolayer2 函数直接发送包即可。Rtupdate0() 函数, 跟新最短距离信息, 实质上是 bellman-ford 算法的思路, 核心代码就一段:

```
if (dt0.costs[j][0] + rcvdpkt->mincost[i] < dt0.costs[i][j] || dt0.costs[i][j] == -1) {  
    dt0.costs[i][j] = dt0.costs[j][0] + rcvdpkt->mincost[i];
```

0 从 j 到 i 的距离大于 0 直接到 j 的距离加上 j 到 i 的 mincost, 则进行更新, 就是 bellman-ford 的思路。并且若做出修改, 要 tolayer2 发送包通知相邻节点, 几乎与 init 中那一段逻辑一样。

Trace 为 2, 完整输出结果如下 (四个高亮表示四个节点的最终的距离表):

root@kurumi:/mnt/c/Users/12980/Desktop/ 谢志康 -22307110187-lab6/ 谢志康 -  
22307110187-lab6/lab6\_part1# ./prog

Enter TRACE:2

MAIN: rcv event, t=0.947, at 3 src: 0, dest: 3, contents: 0 1 3 7

via  
D3	0 2
0| 7 -1  
dest 1| 8 -1  
2| 10 -1

MAIN: rcv event, t=0.992, at 0 src: 1, dest: 0, contents: 1 0 1 -1

via  
D0	1 2 3
1| 1 -1 -1  
dest 2| 2 -1 -1  
3| -1 -1 -1

MAIN: rcv event, t=1.209, at 3 src: 2, dest: 3, contents: 3 1 0 2

via  
D3	0 2
0| 7 5  
dest 1| 8 3  
2| 10 2

MAIN: rcv event, t=1.276, at 3 src: 0, dest: 3, contents: 0 1 2 7

via  
D3	0 2
0| 7 5  
dest 1| 8 3  
2| 9 2

MAIN: rcv event, t=1.642, at 2 src: 0, dest: 2, contents: 0 1 3 7

via  
D2	0 1 3
0| 3 -1 -1  
dest 1| 4 -1 -1  
3| 10 -1 -1

MAIN: rcv event, t=1.871, at 1 src: 0, dest: 1, contents: 0 1 3 7

via  
D1	0 2
0| 1 -1  
dest 2| 4 -1

```

3|    8    -1
MAIN: rcv event, t=2.166, at 2 src: 1, dest: 2, contents:  1  0  1 -1
                    via
D2 |    0    1    3
----|-----
0|    3    2   -1
dest 1|  4    1   -1
3|   10   -1   -1
MAIN: rcv event, t=2.407, at 0 src: 2, dest: 0, contents:  3  1  0  2
                    via
D0 |    1    2    3
----|-----
1|    1    4   -1
dest 2|  2    3   -1
3|   -1    5   -1
MAIN: rcv event, t=2.421, at 2 src: 3, dest: 2, contents:  7 -1  2  0
                    via
D2 |    0    1    3
----|-----
0|    3    2    9
dest 1|  4    1   -1
3|   10   -1    2
MAIN: rcv event, t=2.811, at 1 src: 2, dest: 1, contents:  3  1  0  2
                    via
D1 |    0    2
----|-----
0|    1    4
dest 2|  4    1
3|    8    3
MAIN: rcv event, t=3.293, at 2 src: 3, dest: 2, contents:  7  8  2  0
                    via
D2 |    0    1    3
----|-----
0|    3    2    9
dest 1|  4    1   10
3|   10   -1    2
MAIN: rcv event, t=3.602, at 3 src: 2, dest: 3, contents:  2  1  0  2
                    via
D3 |    0    2
----|-----
0|    7    4
dest 1|  8    3
2|    9    2
MAIN: rcv event, t=4.063, at 2 src: 0, dest: 2, contents:  0  1  2  7

```

```

          via
D2 |    0    1    3
----|-----
0|    3    2    9
dest 1|    4    1   10
3|   10   -1    2
MAIN: rcv event, t=4.104, at 0 src: 3, dest: 0, contents:  7  -1   2   0

```

```

          via
D0 |    1    2    3
----|-----
1|    1    4   -1
dest 2|    2    3    9
3|   -1    5    7
MAIN: rcv event, t=4.169, at 2 src: 3, dest: 2, contents:  5   3   2   0

```

```

          via
D2 |    0    1    3
----|-----
0|    3    2    7
dest 1|    4    1    5
3|   10   -1    2
MAIN: rcv event, t=4.330, at 0 src: 3, dest: 0, contents:  7   8   2   0

```

```

          via
D0 |    1    2    3
----|-----
1|    1    4   15
dest 2|    2    3    9
3|   -1    5    7
MAIN: rcv event, t=4.643, at 1 src: 0, dest: 1, contents:  0   1   2   7

```

```

          via
D1 |    0    2
----|-----
0|    1    4
dest 2|    3    1
3|    8    3
MAIN: rcv event, t=5.213, at 0 src: 3, dest: 0, contents:  5   3   2   0

```

```

          via
D0 |    1    2    3
----|-----
1|    1    4   10
dest 2|    2    3    9
3|   -1    5    7
MAIN: rcv event, t=5.384, at 3 src: 0, dest: 3, contents:  0   1   2   5

```

```

          via
D3 |    0    2

```

```

-----|-----
    0|    7    4
dest 1|    8    3
    2|    9    2
MAIN: rcv event, t=5.820, at 1 src: 2, dest: 1, contents:  2  1  0  2
                via
    D1 |    0    2
-----|-----
    0|    1    3
dest 2|    3    1
    3|    8    3
MAIN: rcv event, t=6.042, at 2 src: 1, dest: 2, contents:  1  0  1  8
                via
    D2 |    0    1    3
-----|-----
    0|    3    2    7
dest 1|    4    1    5
    3|   10    9    2
MAIN: rcv event, t=6.071, at 0 src: 1, dest: 0, contents:  1  0  1  8
                via
    D0 |    1    2    3
-----|-----
    1|    1    4   10
dest 2|    2    3    9
    3|    9    5    7
MAIN: rcv event, t=6.532, at 1 src: 0, dest: 1, contents:  0  1  2  5
                via
    D1 |    0    2
-----|-----
    0|    1    3
dest 2|    3    1
    3|    6    3
MAIN: rcv event, t=7.021, at 0 src: 2, dest: 0, contents:  2  1  0  2
                via
    D0 |    1    2    3
-----|-----
    1|    1    4   10
dest 2|    2    3    9
    3|    9    5    7
MAIN: rcv event, t=7.160, at 2 src: 0, dest: 2, contents:  0  1  2  5
                via
    D2 |    0    1    3
-----|-----
    0|    3    2    7

```

dest 1| 4 1 5  
3| 8 9 2

MAIN: rcv event, t=7.405, at 0 src: 1, dest: 0, contents: 1 0 1 3

via

D0	1 2 3
1| 1 4 10  
dest 2| 2 3 9  
3| 4 5 7

MAIN: rcv event, t=7.579, at 3 src: 0, dest: 3, contents: 0 1 2 4

via

D3	0 2
0| 7 4  
dest 1| 8 3  
2| 9 2

MAIN: rcv event, t=7.941, at 1 src: 0, dest: 1, contents: 0 1 2 4

via

D1	0 2
0| 1 3  
dest 2| 3 1  
3| 5 3

MAIN: rcv event, t=8.086, at 0 src: 3, dest: 0, contents: 4 3 2 0

via

D0	1 2 3
1| 1 4 10  
dest 2| 2 3 9  
3| 4 5 7

MAIN: rcv event, t=8.639, at 2 src: 1, dest: 2, contents: 1 0 1 3

via

D2	0 1 3
0| 3 2 7  
dest 1| 4 1 5  
3| 8 4 2

MAIN: rcv event, t=8.943, at 2 src: 3, dest: 2, contents: 4 3 2 0

via

D2	0 1 3
0| 3 2 6  
dest 1| 4 1 5  
3| 8 4 2

MAIN: rcv event, t=9.960, at 2 src: 0, dest: 2, contents: 0 1 2 4

via

D2	0	1	3
---- -----			
0	3	2	6
dest 1	4	1	5
3	7	4	2

Simulator terminated at t=9.959651, no packets in medium

D0 的表, ij 是表示 0 经过 j 到 i 的最短距离, 比方 D0[1][3]是 10, 表示 0 经过 3 到 1 的最短距离是 10 (0321 路径)。所以只需看每一行的最小值即可。0 到 1 的最短距离是 1, 就是 0 直接到 1; 0 到 2 的最短距离是 2, 0 经过 1 到 2; 0 到 3 的最短距离是 4, 0 经过 1 到 3 (0123, 至于中间这个 2, 就要看 D1 表 1 到 3 的最短路径而得到, 下面解释)

D1 表, 1 经过 j 到 i 的最短距离, 1 到 0 的最短距离, 1, 1 直接到 0; 1 到 2 的距离是 1, 1 直接到 1; 1 到 3 的距离是 3, 1 经过 2 到 3。

D2 表, 2 到 0 的最短距离是 2, 2 到 1 的最短距离是 1, 2 到 3 的最短距离是 2。

D3 表, 3 到 0 的最短距离是 4, 3 到 1 的最短距离是 3, 3 到 2 的最短距离是 2。