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

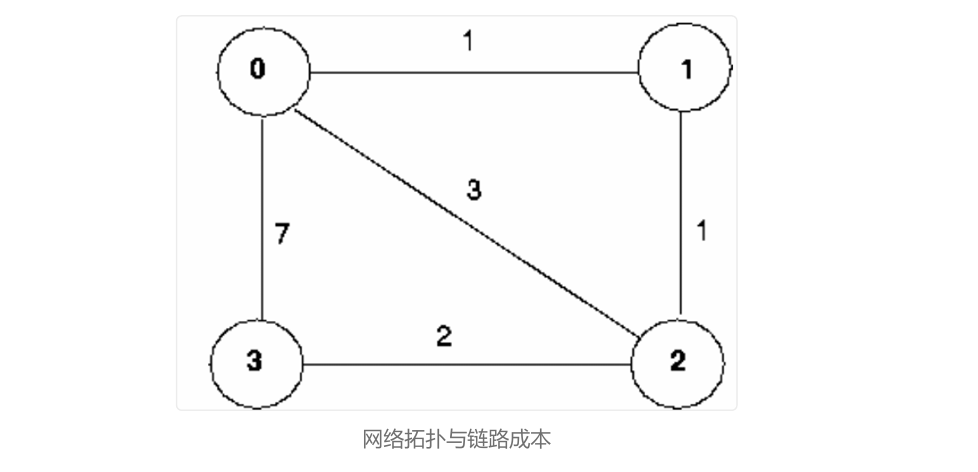
时间：24.11.29 – 24.11.30 （24.12.9修改）

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 ‘printdt3’; 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小错误，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。



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。