Assignment 3 Tips

CPSC 441 - TUTORIAL 9

WINTER 2020

Steps

1. Initialize topology
2. Initialize event list
3. Set routing algorithm
4. Main loop
    1. Call starts
        1. Find best route for call
            1. If no route is available call is dropped
            2. Increase load on all links in path
    2. Call ends
        1. Decrease load on all links in path
5. Print results

If these are both set completely at the start, it should be possible to rerun the experiment with all routing algorithms without having to reinitialize topology and events

Initialize Topology

topology.dat

```
A  B  10  5
A  C  15  2
B  C  20  6
B  D  30  8
C  D   8  5
```

	P1	P2	Delay	Max Circ.	Current Circ.
0	'A'	'B'	10	5	5
1	'A'	'C'	15	2	2
2	'B'	'C'	20	6	6
…	…	…	…	…	…

```
FILE *file; //variable for file dsecriptor
file = fopen("filename.txt","r"); //"r" for reading
int i = 0;
while (fscanf(file, "%c %c %d %d\n", p1[i], p2[i], delay[i], maxCircuits[i]) == 4){
  ◦ currentCircuits[i] = maxCircuits[i]
  ◦ i++;
}
fclose(file);
```

Initialize Event List

callworkload.dat

```
 0.123456   A   D   12.527453
 7.249811   B   C   48.129653
 8.975344   B   D   6.124743
10.915432   A   C   106.724339
15.817634   B   C   37.634569
```

	Type	Time	Src	Dest
0	'S'	0.123456	'A'	'D'
1	'S'	7.249811	'B'	'C'
2	'S'	8.975344	'B'	'D'
…	…	…	…	…

```
FILE *file; //variable for file dsecriptor
file = fopen("filename.txt","r"); //"r" for reading
int i = 0;
char src, dest;
float time, duration;
while (fscanf(file, "%f %c %c %f\n", time, src, dest, duration) == 4){
  addCallStartEvent(time, src, dest);
  addCallEndEvent(time + duration, src, dest);
}
fclose(file);
```

Main Loop

```
for each event in eventList
        if event.type == 'S' //call start
                totalCalls++;
                 event.route = getBestRoute(event.origin, event.dest, routingAlgorithm);
                if (route != NULL){
                        increaseLoadOnRoute(event.route);
                        totalCallsSucceeded++;
                        totalPropDelay += getPropDelay(event.route);
                        totalHops += getHops(event.route);
                else
                        totalCallsDropped++;
                end if-else
        end if
        if event.type == 'E' //call end
                decreaseLoadOnRoute(event.route);
        end if
end for
```

Dijkstra's Algorithm

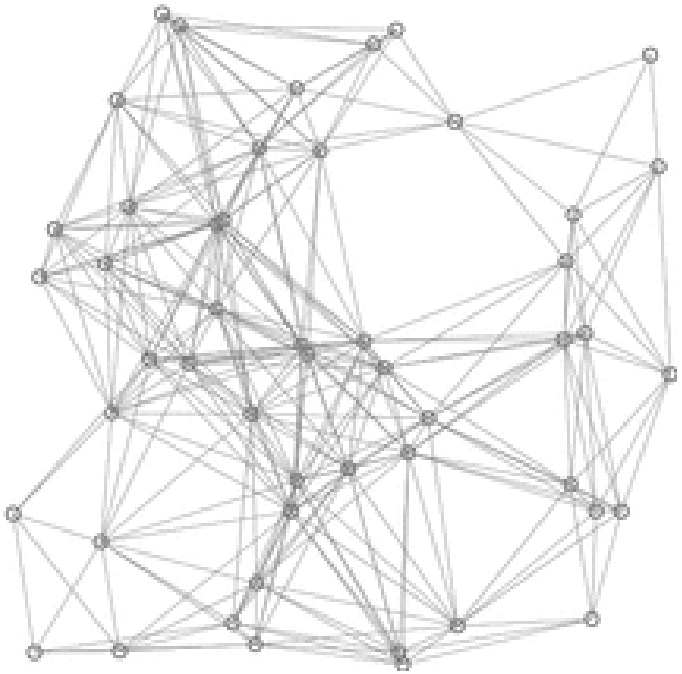
Returns shortest path spanning tree from source node

To get shortest path from node *a* to node *b*

- Run Dijkstra's algorithm with *a* as source
- (Can stop when spanning tree includes *b)*
- Trace back from *b* to *a* to get shortest path

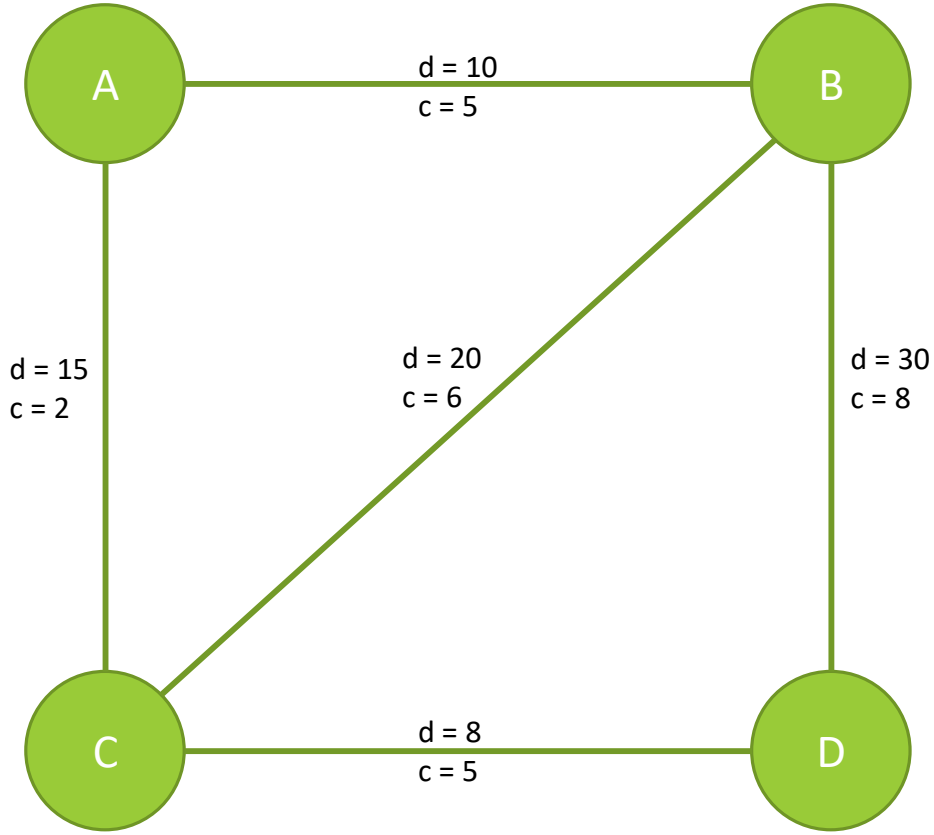
Dijkstra's Algorithm

```
1  function Dijkstra(Graph, source):
2
3      create vertex set Q
4
5      for each vertex v in Graph:
6          dist[v] ← INFINITY
7          prev[v] ← UNDEFINED
8          add v to Q
9
10     dist[source] ← 0
11
12     while Q is not empty:
13         u ← vertex in Q with min dist[u]
14
15         remove u from Q
16
17         for each neighbor v of u: // only v that are still in Q
18         alt ← dist[u] + length(u, v)
19         if alt < dist[v]:
20             dist[v] ← alt
21             prev[v] ← u
22
23     return dist[], prev[]
```



Example: Propagation Delay

Node	Distance	Previous
A	0	
B	10	A
C	15	A
D	23	C



Shortest Path

```
1 S ← empty sequence
2 u ← target
3 if prev[u] is defined or u = source: // Do
  something only if the vertex is reachable
4   while u is defined:
5     insert u at the beginning of S // Push
6     u ← prev[u] // Traverse back to
      source
```

Node	Distance	Previous
A	0	N/A
B	10	A
C	15	A
D	23	C

References

https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

https://pages.cpsc.ucalgary.ca/~carey/CPSC441/assignment3.html

Skeleton code for Assignment 3:

- https://pages.cpsc.ucalgary.ca/~carey/CPSC441/a3/skeleton.c

Youtube video for explaining Dijkstra's:

- https://www.youtube.com/watch?v=pVfj6mxhdMw&t=276s