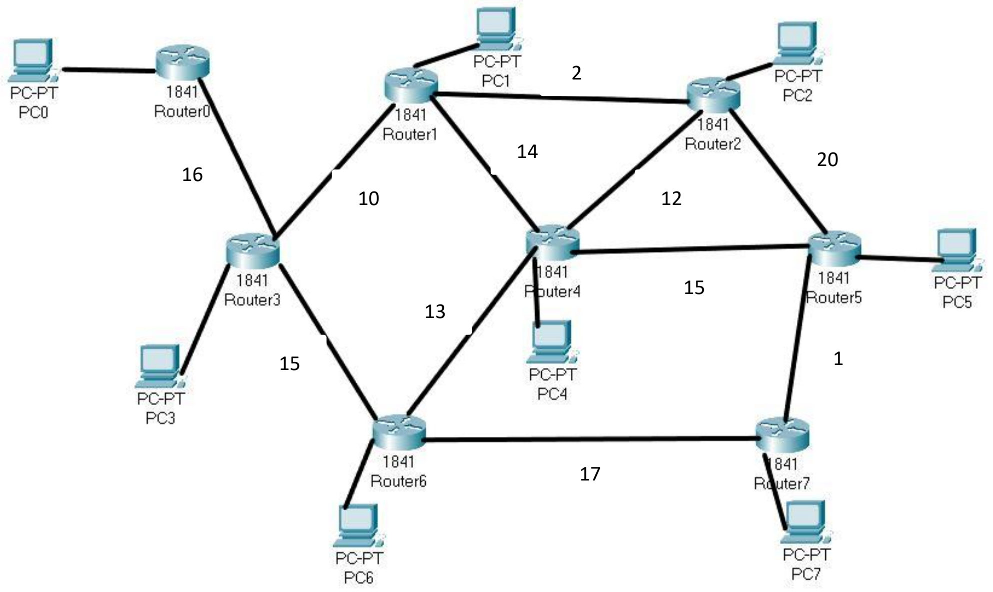
Problem Statement

The proposed problem is that there is a hypothetical Data communication computer network system show in diagram below from the diagram we can see 7 computers connected to their individual router and in this computer network system we wanted to send a message from PC0 to PC7 by routing the packet from the source router that has PC0 to other routers connected to it and arrive at the destination PC7. As we can see from the diagram there are multiple route from our source to its destination so the problem is to find the shortest path from PC0 to PC7 by considering the link cost between each router this link cost can represent the relative cost off communication between routers.

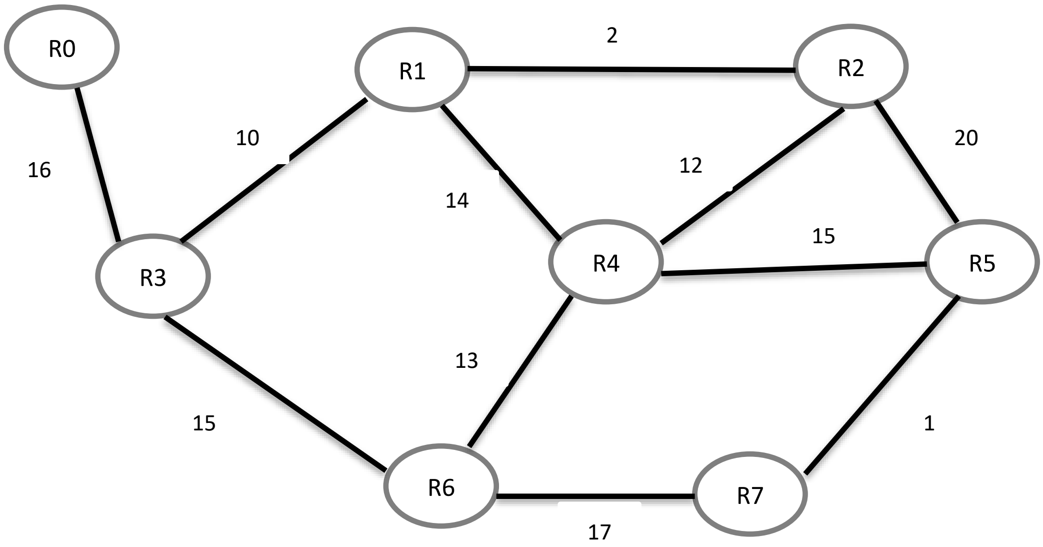


16, 10, 15…are Link cost

Proposed solution

Use Graph to represent the problem above by making the routers as the Vertex, the network connection between them as Edges and link cost as weight of the graph.

Step 1: Model your problem as graph



Assuming there is no cost between the routers and their individual PC (computer) the problem can be model as a graph as shown above and there is a bidirectional connection between the vertexes.

Step 2: Try to identify a well known graph problem that is similar to the problem you are solving

A well known graph problem that is similar to the problem above is Weighted Short path problems that are used to find the shortest path between source and destination or target by taking in the cost of the edges.

Step 3: Take advantage of existing graph algorithms to solve the problem you have identified

One of the algorithms that is used to find the shortest path of a weighted graph is Dijkstra algorithm. Table (1) implimentation

Vertex	Visited	cost	Previous
R0	T	0	0
R1	F	INF	0
R2	F	INF	0
R3	F	16	R0
R4	F	INF	0
R5	F	INF	0
R6	F	INF	0
R7	F	INF	0

(2)

Vertex	Visited	cost	Previous
R0	T	0	0
R1	F	26	R3
R2	F	INF	0
R3	T	16	R0
R4	F	INF	0
R5	F	INF	0
R6	F	31	R3
R7	F	INF	0

Vertex	Visited	cost	Previous
R0	T	0	0
R1	T	26	R3
R2	F	28	R1
R3	T	16	R0
R4	F	40	R1
R5	F	INF	0
R6	F	31	R3
R7	F	INF	0

(3)

Vertex	Visited	cost	Previous
R0	T	0	0
R1	T	26	R3
R2	T	28	R1
R3	T	16	R0
R4	F	40	R1
R5	F	48	R2
R6	F	31	R3
R7	F	INF	0

(4)

Vertex	Visited	cost	Previous
R0	T	0	0
R1	T	26	R3
R2	T	28	R1
R3	T	16	R0
R4	F	40	R1
R5	F	48	R2
R6	T	31	R3
R7	F	48	R6

(5)

Vertex	Visited	cost	Previous
R0	T	0	0
R1	T	26	R3
R2	T	28	R1
R3	T	16	R0
R4	T	40	R1
R5	F	48	R2
R6	T	31	R3
R7	F	48	R6

(6)

Vertex	Visited	cost	Previous
R0	T	0	0
R1	T	26	R3
R2	T	28	R1
R3	T	16	R0
R4	T	40	R1
R5	T	48	R2
R6	T	31	R3
R7	F	48	R6

(7)

Vertex	Visited	cost	Previous
R0	T	0	0
R1	T	26	R3
R2	T	28	R1
R3	T	16	R0
R4	T	40	R1
R5	T	48	R2
R6	T	31	R3
R7	T	48	R6

(8)

The above tables show the Dijkstra algorithm first we started by seating the cost to all routers infinity and the visited value to false then start from the source Router0 and change the cost of the adjacent router (R3) to 16 its previous is R0(Table (1)), since it's the only adjacent vertex we move on an choose the smallest cost out of the unvisited nodes in this case its R3 we then set R3 to visited True (T) change the adjacent routers cost to (R1→26, previous R3)(R6→31, previous R3) (Table(2))we repeat the process by choosing the smallest one and updating cost and path if we find small cost until all vertex have been visited.

Using the algorithm we can see that the shortest path from PC0 to every other PC our goal was to solve the shortest path from PC0 to PC7 and from the algorithm implemented using the tables above we can see that the shortest path using the minimum link cost is through Router 0(R0)➔Router 3(R3)➔ Router 6 (R6) ➔ Router 7 (R7)

Code implementation

Four classes were used to implement the Dijkstra algorithm

First class contains Edges with it it has the cost, the source and destination vertex

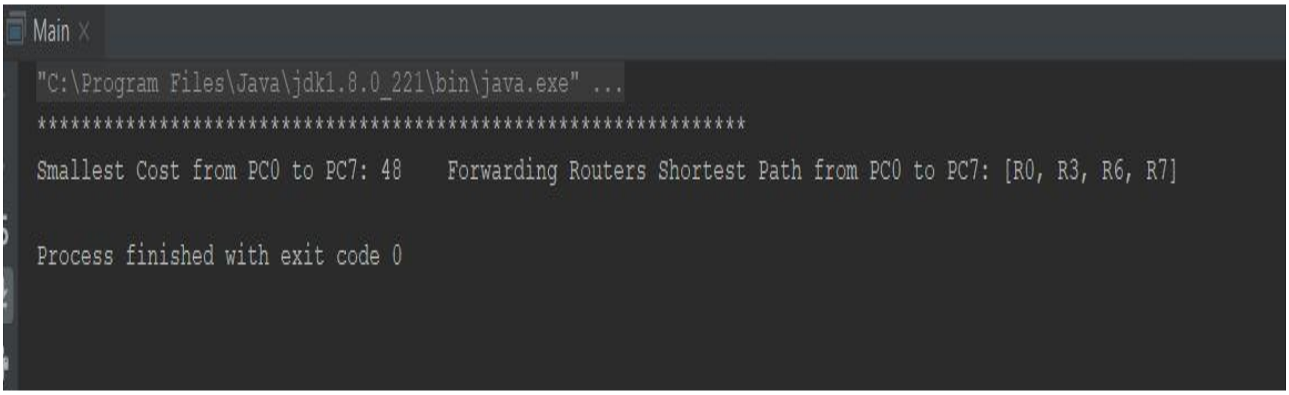
Second class contain vertex in this class we have a string for the name of the vertex eg.R0, R1… , used an Array List of edges for the adjacent vertexes, distance variable set to infinity to represent the cost, Boolean visited variable to see if a vertex is visited or not. A vertex method to insert vertex and an add edge method to add the edge with the cost this vertex class implements comparable so it is can be used in priority queue for comparing cost of vertex

Third class contains the Dijkstra algorithm where there is a method that takes in the source vertex and use a priority queue to find every short distance starting from the source the other method in this class uses the information from the above method were the cost of every vertex from the source is known so it gives us the path from source with minimum cost

Forth class is the main class was we imported the vertex, their edge and cost

The code is included in the file submitted for the project. In the code I used a simple for loop to retrieve the precious vertex to get to the routing of the routers (vertex).

The output of code implementation using Dijkstra algorithm



```
"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" ...
****************************************************************
Smallest Cost from PC0 to PC7: 48    Forwarding Routers Shortest Path from PC0 to PC7: [R0, R3, R6, R7]


Process finished with exit code 0
```

In conclusion using Dijkstra algorithm we can solve the problem of shortest path from a certain source vertex to target vertex other problem that can be solved using this algorithm could be finding the shortest distance from home to school by making vertex intersection rods and edges the rods connecting them.